David Zhang · Dongmin Guo
Ke Yan

Breath Analysis for Medical Applications

Springer

Breath Analysis for Medical Applications

David Zhang · Dongmin Guo
Ke Yan

Breath Analysis for Medical Applications

Springer

David Zhang
Biometrics Research Centre
The Hong Kong Polytechnic University
Hong Kong
China

Ke Yan
National Institute of Health
Bethesda
USA

Dongmin Guo
Wake Forest University
Winston-Salem, NC
USA

ISBN 978-981-10-4321-5 ISBN 978-981-10-4322-2 (eBook)
DOI 10.1007/978-981-10-4322-2

Library of Congress Control Number: 2017939534

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

Some components in human breath have been proven to be associated with certain diseases and the concentration of these components is linked to disease status. Recently, breath signal diagnosis has attracted increasing research interests. Many kinds of breath signal acquisition systems and breath signal processing methods have been reported. However, there are still a lot of challenging works to be done, for example, how to acquire breath signal in a fast, accurate, and informative way, how to preprocess the breath signal to rule out the outliers and increase the quality of the signal, and how to extract efficient features and find proper classifiers for breath diagnosis.

This book focuses on these challenging issues. Novel breath signal acquisition systems based on multiple breath sensors were described first. In order to collect samples effectively, we developed a sample acquisition system with sensor fusion technology. To detect the drift of breath signals, we provided optimized preprocessing frameworks, such as using transfer samples and regression models. To represent breath signals completely, we discovered different types of breath signal features, such as spatial feature, frequency feature, deep learning feature, etc. Moreover, we also provided many effective algorithms for breath signal classification and recognition, such as curve-fitting models and sparse representation classification.

All of the technologies, algorithms, and medical application cases described in this book were applied in our research work and have proven to be effective in breath signal analysis. First, this book presents a comprehensive introduction on the useful techniques of breath signal acquisition methods using different kinds of chemical sensors, cooperated with the optimized selection and fusion acquisition scheme. Then, this book introduces the effective preprocessing approaches, such as drift removing and feature extraction methods. Moreover, the classification methods used as case studies are also provided. Finally, this book provides discussions and concluding remarks to indicate some promising directions on the studies and medical applications of computerized breath diagnosis. This book will benefit the researchers, professionals, graduate and postgraduate students working in the field of breath sample diagnosis, signal processing, pattern recognition,biometrics, etc. This book will also be very useful for interdisciplinary research.

Our team has been working on the breath analysis research on computational TCM diagnosis over 10 years. Under the grant support (Grant No. 61332011) from National Natural Science Foundation of China (NSFC) and Hong Kong Polytechnic University, we had started our studies on this topic. The authors would like to thank Dr. Zhaotian Zhang, Dr. Xiaoyun Xiong, and Dr. Ke Liu from NSFC for their consistent support to our research work.

Hong Kong, China
Winston-Salem, USA
Bethesda, USA
February 2017

David Zhang
Dongmin Guo
Ke Yan

Contents

Part II Breath Acquisition Systems

Part IV Feature Extraction and Classification

Part I
Background

Chapter 1
Introduction

Abstract In this introductory chapter which sets the scene for this book, the background which stimulates this research work is first provided. The motivation for the focus of the work is then explained, highlighting the importance of the breath analysis used in disease diagnosis, of the development of breath analysis device, and of the design of specific pattern recognition algorithm for breath analysis. This is followed by a statement of the objective of the research, a brief summary of the work, and a general outline of the overall structure of the present study.

Keywords Breath analysis · Electronic olfaction · Therapy monitoring · Chemical sensor · Disease identification

1.1 Background and Motivation

As early as 1970s, Jellum et al. (1973) had stated that, *"If one were able to identify and determine the concentration of all compounds inside the human body, including high molecular weight as well as low molecular weight substances, one would probably find that almost every known disease would result in characteristic changes of the biochemical composition of the cells and of the body fluids."* From the metabolism point of view, the "metabolic profile" is representative of the internal chemistry of human body. Therefore, its external expression, like breath, blood, urine, and skin excretions, can be exploited as the source of information to diagnose diseases or detect them at an early stage (D'Amico et al. 2008).

The exploitation strictly associates with finding the reliable ways of accessing it, developing the consequent range of suitable instruments used to perform the measurements, and designing the appropriate data analysis algorithm to discover pathological features from the "metabolic profile" (D'Amico et al. 2008). Because of these restrictions, at present, only the composition of human fluids such as blood and urine is analyzed and utilized as an effective approach for diagnostic tasks. Even though such approach is accurate, it is invasive, time consuming, and harmful to not only the subjects but also the personnel who collect the samples. Recently, great efforts have been done about seeking effective approaches to noninvasive and self-help diagnosis

D. Zhang et al., *Breath Analysis for Medical Applications*,
DOI 10.1007/978-981-10-4322-2_1

of disease. As a result, there are increasing concerns about the applications of breath analysis in medicine and clinical pathology both as a diagnostic tool and as a way to monitor the progress of therapies.

The present work includes three aspects: designing a specific device, collecting breath data, and analyzing these data. The following sections highlight the significance of the study and discuss the reasons why develop the current work.

1.1.1 Why Is Breath Analysis Used in Disease Diagnosis?

Breath analysis is the examination of breath for the presence of certain compounds to determine the presence of some diseases and conditions in the human body. Endogenous molecules in exhaled breath, such as acetone, nitric oxide, hydrogen, and ammonia, are produced by metabolic processes. They are separated from blood and enter into the alveolar air via the alveolar pulmonary membrane (D'Amico et al. 2008; Schubert et al. 2004; Miekisch et al. 2004). Variation in the concentration of these molecules can suggest various diseases or at least changes in metabolism (Amann et al. 2005). For instance, nitric oxide in breath can be measured as an indicator of asthma or other conditions characterized by airway inflammation (Deykin et al. 2002). Increased pentane and carbon disulfide have been observed in the breath of patients with schizophrenia (Phillips et al. 1993). Breath concentration of volatile organic compounds (VOCs) such as cyclododecatriene, benzoic acid, and benzene are much higher in lung cancer patients than in control groups (Phillips et al. 2007).

Acetone has been found to be more abundant in the breath of diabetics (Deng et al. 2004; Fleischer et al. 2002), and breath ammonia is significantly elevated in patients with renal diseases (Davies et al. 1997). These molecules are considered as biomarkers of the presence of diseases and clinical conditions. Much can be learnt from them about the overall state of an individual's metabolism or physical condition.

Breath analysis has many advantages compared with other traditional methods such as blood and urine tests (Van Berkel et al. 2008). First, breath analysis is a non-invasive method, and it causes the least harm to both the subjects and the personnel who collect the samples. Secondly, its result can be obtained immediately, and third, the sample collection is quite easy for a subject, since the only requirement to collect a breath sample is that the subject must be breathing. Therefore, increasing interest has been expressed about the applications of breath analysis in medicine and clinical pathology, both as a diagnostic tool and as a way to monitor the progress of therapies (Di Francesco et al. 2005; Dweik and Amann 2008).

1.1.2 Why Should Breath Analysis System Be Developed?

As aforementioned, the exploitation of "metabolic profile" strictly associates with the reliable ways of accessing it and the suitable instruments used to perform the measurements. It is obvious that breath analysis is a possible way to access the "metabolic profile", so finding an efficient apparatus to discover the pathological features is a significant and critical task for breath analysis.

Currently, the measurement of exhaled breath is usually performed by two common gas analysis apparatuses, gas chromatography (GC) (Phillips 1997), or electronic nose (e-nose) (Thaler and Hanson 2005). GC can separate and identify molecules that are responsible of typical odors occurring in specific diseases. It is very accurate for disease identification. But this kind of apparatus is expensive and not portable, its sampling and assaying processes are complicated and time consuming (about one hour for one sample), and its results require expert's interpretation (Amann et al. 2004). Therefore, it is hard to use such apparatus as a domestic or clinical tool.

A less expensive and more portable alternative is e-nose. Unlike GC that identifies molecules directly, e-nose works like human olfaction that functions as a nonseparative mechanism. It is cheaper and faster (requiring only 30 min for one sample) and is often used outside of medicine, in fields related to food, chemistry, fragrances, and environment (Rock et al. 2008). Recently, e-nose has gradually been used in medicine for the diagnosis of renal disease (Lin et al. 2001), diabetes (Yu et al. 2005), lung cancer (Blatt et al. 2007), and asthma (Dragonieri et al. 2007). While all of these methods work satisfactorily in breath analysis, the results could possibly be improved. That is because, commercial e-noses, for the sake of their marketing concerns, have to provide some versatility in applications, such as coffee, wine, and fragrances identification. The versatility, in contrast, limits their performance in disease detection since their sensor selection has to match broad applications.

As a result, it is necessary to develop a specific device for breath analysis. The device should have the advantage of both GC and e-nose: noninvasive, painless, high accuracy, low cost, and user-friendly. A possible solution is to select highly accurate chemical sensors that are specifically sensitive to compounds of human breath and the indicators of certain diseases. Each sensor in the sensor array should have a different sensitivity profile over a range of compounds expected in the target application, e.g., the detection of an unknown disease. Therefore, the sensor array provides distinct response patterns to different analytes.

1.1.3 Why Should Specific Algorithms Be Designed for Breath Analysis?

1.1.3.1 Sensor Selection

The breath analysis system that the book proposes includes multiple sensors. This kind of design offers system broad applications in medicine, but is problematic in practice. Since each sensor has a specific contribution in identifying a type of disease, not all sensors in the system are needed when we only want to detect one type of disease. For sensors that are not sensitive to the biomarkers of a given disease, they may only generate slightly different responses. These sensors would provide redundant information, which might interfere with the identification. However, it is difficult to decide which sensor is more useful for an unknown sample because some sensors are cross-sensitive to the biomarkers of the diseases. Therefore, a proper sensor selection approach should be involved in the system.

1.1.3.2 Feature Extraction

The shape of odor signal obtained by chemical sensor is different from other signals like speech, image, and electric current. Compared with other signals, the odor signal is smooth and weakly distinguishable and therefore has less distinct features to extract. Additionally, breath signals obtained by chemical sensors are often in such case: (1) these data are with high dimensionality since there are often tens of chemical sensors included in a sensor array; (2) the amount of signal samples is limited, due to the time cost of diseased breath sample collection. However, it is known that if the number of features extracted from the original samples is too large relative to the number of the training samples, the features will become less distinguishable and may lead to the degrading of the performance of the trained classifier. Accordingly, finding a compact feature representation is the critical problem we are facing when conducing odor signal recognition.

1.1.3.3 Classifier Design

Breath signals are with high dimensionality and the number of samples is limited, as aforesaid, which means the number of samples utilized to train the classifier is too small relative to the dimensionality of data in each sample. In this case, the traditional statistical pattern recognition methods may not work well, since the efficiency of these methods is highly dependent on the interrelationship between sample sizes, number of features, and structure of classifiers (Jain and Mao 2000).

Besides, in the medical applications of breath analysis, one of them is physical condition monitoring, which monitors the subjects' physical condition by grouping the samples into "well controlled", "somewhat controlled", "poorly controlled", and "not controlled" according to the status of disease control and development. It is clear that there is an order among these values. Therefore, common classification methods, which treat the samples as a set of unordered data, are not suitable for our application tasks.

Accordingly, specific pattern recognition algorithms should be designed for the breath signals in order to achieve different medical application tasks.

1.2 Relative Technologies

Based on the aforementioned statement, the present work focuses on designing breath analysis system and analyzing the breath data for medical application, such as disease diagnosis and physical condition monitoring. Specifically, the following research objective will be achieved:

- System design:
  - Develop a special system for breath analysis;
  - Define the standard breath gas collection and signal sampling procedure;
- Data collection:
  - Collect samples and build database for all kinds of diseased samples associated with the compounds of human breath, such as diabetes, lung cancer, and renal diseases.
  - Build database for patients with different level of diseases;
- Pattern recognition algorithm design:
  - Design a technique to choose an optimal configuration of sensors from a whole sensor set for a given medical application;
  - Establish signal preprocessing algorithms to correct drift in e-nose signals such as instrumental variation and time-varying drift;
  - Extract multi-type of features and design a multi-feature fusion technique to determine the optimal feature set for a given medical application;
  - Develop different classifiers for different medical applications;
- Applications:
  - Disease diagnosis: identify a subject as healthy or as with either diabetes, renal disease, airway inflammation, gastroenteritis, or lung cancer.
  - Condition monitoring: monitor the development of diabetes by measuring the breath acetone of patients;
  - Medical treatment evaluation: evaluate the medical treatment of patients with end-stage renal failure by measuring the concentration of ammonia in patient's breath.

1.3 Outline of the Work

The organization of this book is presented as follows.

Chapter 2 reviews the general methods used in breath analysis. The limitations and drawbacks of these methods are pointed out and some valuable conclusions from the current investigation are presented.

In Chap. 3, a specific breath analysis system is described. The system structure, user interface, and system performance are introduced first. Then, the data collection and preprocessing are presented. Finally, the databases formed by the samples collected by the designed system are described.

Chapter 4 proposes an automatical sensor selection method, which utilizes LDA technique to compute the weight of each sensor, and therefore to choose an optimal configuration of sensors from a whole set of available sensors for a given disease identification task.

In Chap. 5, four techniques will be developed to evaluate the sensors performance in a breath acquisition system. Proposed statistics aim at judging the importance, unique discriminating information and redundancy of each sensor based on the exhaustive search results of all the possible sensor arrays.

Chapter 6 describes two approaches, windowed piecewise direct standardization (WPDS) and standardization error-based model improvement (SEMI), to correct instrumental variation and make the prediction models of e-noses more transferable.

Chapter 7 introduces transfer-sample-based multitask learning (TMTL) to simultaneously address instrumental variation and time-varying drift.

Chapter 8 proposes an approach named drift correction autoencoder (DCAE) to deal with complex drift of e-noses.

In Chap. 9, maximum independence domain adaptation (MIDA) is presented for unsupervised drift correction.

Chapter 10 applies the classical support vector machine recursive feature elimination (SVM-RFE) algorithm to feature selection and improves it by incorporating a correlation bias reduction (CBR) strategy into the feature elimination procedure.

Chapter 11 proposes a Sparse Representation-based Classification (SRC) method for breath sample identification. The method expresses an input signal as the linear combination of a small number of the training signals, which are from the same category as the input signal. The selection of a proper set of training signals in representation, therefore, gives us useful cues for classification.

Chapter 12 introduces a breath analysis system to measure acetone in human breath, and therefore to evaluate the blood glucose levels of diabetics.

Chapter 13 investigates the potential of breath signals analysis as a way for blood glucose monitoring.

Chapter 14 proposes several targeted approaches to improve the accuracy of diabetes screening and blood glucose level (BGL) prediction.

In Chap. 15, a novel optimized medical e-nose system specially for disease diagnosis and blood glucose level (BGL) prediction is proposed.

Finally, the conclusions drawn from this study and the directions of future work are summarized in Chap. 16.

References

Amann A, Poupart G, Telser S, Ledochowski M, Schmid A, Mechtcheriakov S (2004) Applications of breath gas analysis in medicine. Int J Mass Spectrom 239(2–3):227–233

Amann A, Schmid A, Scholl-Burgi S, Telser S, Hinterhuber H (2005) Breath analysis for medical diagnosis and therapeutic monitoring. Spectrosc Eur 17(3):18–20

Blatt R, Bonarini A, Calabro E, Della Torre M, Matteucci M, Pastorino U (2007) Lung cancer identification by an electronic nose based on an array of MOS sensors. In: International joint conference on neural networks 2007, pp 1423–1428

D'Amico A, Di Natale C, Paolesse R, Macagnano A, Martinelli E, Pennazza G, Santonico M, Bernabei M, Roscioni C, Galluccio G et al (2008) Olfactory systems for medical applications. Sens Actuators B Chem 130(1):458–465

Davies S, Spanel P, Smith D (1997) Quantitative analysis of ammonia on the breath of patients in end-stage renal failure. Kidney Int 52(1):223–228

Deng C, Zhang J, Yu X, Zhang W, Zhang X (2004) Determination of acetone in human breath by gas chromatography-mass spectrometry and solid-phase microextraction with on-fiber derivatization. J Chromatogr B 810(2):269–275

Deykin A, Massaro A, Drazen J, Israel E (2002) Exhaled nitric oxide as a diagnostic test for asthma: online versus offline techniques and effect of flow rate. Am J Respir Crit Care Med 165(12):1597–1601

Di Francesco F, Fuoco R, Trivella M, Ceccarini A (2005) Breath analysis: trends in techniques and clinical applications. Microchem J 79(1–2):405–410

Dragonieri S, Schot R, Mertens B, Le Cessie S, Gauw S, Spanevello A, Resta O, Willard N, Vink T, Rabe K et al (2007) An electronic nose in the discrimination of patients with asthma and controls. J Allergy Clin Immunol 120(4):856–862

Dweik R, Amann A (2008) Exhaled breath analysis: the new frontier in medical testing. J Breath Res 2(030):301

Fleischer M, Simon E, Rumpel E, Ulmer H, Harbeck M, Wandel M, Fietzek C, Weimar U, Meixner H (2002) Detection of volatile compounds correlated to human diseases through breath analysis with chemical sensors. Sens Actuators B Chem 83(1–3):245–249

Jain A, Mao R (2000) Statistical pattern recognition: a review. IEEE Trans Pattern Anal Mach Intell 22(1):4–37

Jellum E, Stokke O, Eldjarn L (1973) Application of gas chromatography, mass spectrometry, and computer methods in clinical biochemistry. Anal Chem 45(7):1099–1106

Lin Y, Guo H, Chang Y, Kao M, Wang H, Hong R (2001) Application of the electronic nose for uremia diagnosis. Sens Actuators B Chem 76(1–3):177–180

Miekisch W, Schubert J, Noeldge-Schomburg G (2004) Diagnostic potential of breath analysis-focus on volatile organic compounds. Clin Chim Acta 347(1–2):25–39

Phillips M (1997) Method for the collection and assay of volatile organic compounds in breath. Anal Biochem 247(2):272–278

Phillips M, Sabas M, Greenberg J (1993) Increased pentane and carbon disulfide in the breath of patients with schizophrenia. J Clin Pathol 46(9):861–864

Phillips M, Altorki N, Austin J, Cameron R, Cataneo R, Greenberg J, Kloss R, Maxfield R, Munawar M, Pass H et al (2007) Prediction of lung cancer using volatile biomarkers in breath. Cancer Biomark 3(2):95–109

Rock F, Barsan N, Weimar U (2008) Electronic nose: current status and future trends. Chem Rev 108(2):705–725

Schubert J, Miekisch W, Geiger K, Nöldge-Schomburg G (2004) Breath analysis in critically ill patients: potential and limitations. Expert Rev Mol Diagn 4(5):619–629

Thaler E, Hanson C (2005) Medical applications of electronic nose technology. Expert Rev Med Devices 2(5):559–566

Van Berkel J, Dallinga J, Möller G, Godschalk R, Moonen E, Wouters E, Van Schooten F (2008) Development of accurate classification method based on the analysis of volatile organic compounds from human exhaled air. J Chromatogr B 861(1):101–107

Yu J, Byun H, So M, Huh J (2005) Analysis of diabetic patient's breath with conducting polymer sensor array. Sens Actuators B Chem 108(1–2):305–308

Chapter 2
Literature Review

Abstract This chapter discusses some of the key issues in breath analysis and reviews some previous research work in the areas which are particularly relevant to the present study. Following a brief introductory overview of the field, the chapter first presents the development of breath analysis. Traditional approaches like GC which have been used to analyze the compounds of breath and identify several diseases are then described. This is followed by a detailed introduction of current major approaches, e-noses, for breath analysis. The final section gives a short summary of the chapter.

Keywords Breath analysis · Electronic olfaction · Therapy monitoring · Chemical sensor · Disease identification

2.1 Introduction

Breath analysis is the examination of breath for the presence of certain compounds to determine the presence of some diseases and conditions in the human body. The breath is largely composed of oxygen, carbon dioxide, water vapor, nitric oxide, and numerous VOCs (Cao and Duan 2007). The type and quantity of the VOCs in the breath of any particular individual will vary but there is nonetheless a comparatively small common core of breath which is present in all humans (Phillips et al. 1999b). The molecules in an individual's breath may be exogenous or endogenous depending on their origin (Miekisch and Schubert 2006). Exogenous molecules are those that have been inhaled or ingested from the environment or other sources such as air or food and are hence of no diagnostic value (Risby and Solga 2006). Endogenous molecules are produced by metabolic processes. They pass from the blood through the alveolar pulmonary membrane and enter the alveolar air. As a result, the molecules in the breath have a direct relationship with their types, concentrations, volatilities, lipid solubility, and rates of diffusion when they circulate in the blood and cross the alveolar membrane (Sehnert et al. 2002). Table 2.1 summarizes the typical compositions from the breath of healthy persons and their concentrations (Phillips et al. 1999b, Risby and Solga 2006). Changes in the concentration of these molecules can suggest various diseases or at least changes in the metabolism.

D. Zhang et al., *Breath Analysis for Medical Applications*,
DOI 10.1007/978-981-10-4322-2_2

Table 2.1 Typical compositions from the breath of healthy persons and their concentrations

Concentration (v/v)	Molecules
Percentage	Oxygen, water, carbon dioxide
Parts-per-million	Acetone, carbon monoxide, methane, hydrogen, isoprene, benzenemethanol
Parts-per-billion	Formaldehyde, acetaldehyde, 1-pentane, ethane, ethylene, other hydrocarbons, nitric oxide, carbon disulfide, methanol, carbonyl sulfide, methanethiol, ammonia, methylamine, dimethyl sulfide, benzene, naphthalene, benzothiazole, ethane, acetic acid

Table 2.2 Physiological origins of some endogenous breath molecules

Breath molecules	Physiological origins
Acetaldehyde	Ethanol metabolism
Acetone	Decarboxylation of acetoacetate
Ammonia	Protein metabolism
Carbon disulfide	Gut bacteria
Carbon monoxide	Production catalyzed by heme oxygenase
Carbonyl sulfide	Gut bacteria
Ethane	Lipid peroxidation
Ethanol	Gut bacteria
Ethylene	Lipid peroxidation
Hydrocarbons	Lipid peroxidation/metabolism
Hydrogen	Gut bacteria
Isoprene	Cholesterol biosynthesis
Methane	Gut bacteria
Methanethiol	Methionine metabolism
Methanol	Metabolism of fruit
Methylamine	Protein metabolism
Nitric oxide	Production catalyzed by nitric oxide synthase
Pentane	Lipid peroxidation

By studying the components of the breath, much can be learnt about the overall state of an individual's metabolism or physical condition. Table 2.2 presents some physiological origins of endogenous breath molecules (Risby and Solga 2006). These molecules are considered as biomarkers of the presence of diseases and clinical conditions. For instance, nitric oxide in breath can be measured as an indicator of asthma or other conditions characterized by airway inflammation (Deykin et al. 2002). Breath isoprene is significantly lower in cystic fibrosis patients with acute respiratory exacerbation (McGrath et al. 2000). Increased pentane and carbon disulfide have been observed in the breath of patients with schizophrenia (Phillips et al. 1993). Acetone has been found to be more abundant in the breath of diabetics (Deng et al.

2004; Fleischer et al. 2002), and breath ammonia is significantly elevated in patients with renal diseases (Davies et al. 1997). By detecting these molecules in breath, one can identify the diseases in an early stage and monitor their development.

Breath analysis has many advantages compared with other traditional methods such as blood and urine tests, including the following major ones. First, breath analysis is a noninvasive method, and it causes the least harm to both the subjects and the personnel who collect the samples. Second, its result can be obtained immediately, and third, the only requirement to collect a breath sample is that the subject must be breathing (Van Berkel et al. 2008). Therefore, increasing interest has been expressed about the applications of breath analysis in medicine and clinical pathology, both as a diagnostic tool and as a way to monitor the progress of therapies (Di Francesco et al. 2005; Dweik and Amann 2008).

2.2 Development of Breath Analysis

The breath analysis for the purpose of diagnosis has a long history. The ancient Greek physicians already knew that human breath could provide clues to diagnosis (Phillips 1992). For example, doctors in ancient Greece knew the existence of sweet breath was a dangerous sign and modern clinicians know that exhaled air from patients with diabetic ketoacidosis smells sweet like rotting apples. Ancient Greek physicians also recognized musty and fishy odors indicated a problem with liver, a urine-like smell indicated failing kidneys, and a putrid stench indicated a lung abscess. Olfaction diagnosis is also one of the basic diagnostic methods of Chinese Traditional Medicine, which has a history of 5000 years. The ancient Chinese doctors stated that the aroma of human breath could indicate the condition of the human body (Zhufan 2000). They found that foul breath is due to pathogenic heat in the stomach or indigestion and sour breath indicates food accumulation in the stomach.

Modern breath analysis started in the 1970s when Pauling et al. (1971) pioneered the analytical assessment of breath components by the GC analysis of exhaled air and identified more than 200 compounds in human breath exhaled after passing the blood/air interface within the lungs. Some of these compounds were associated with different pathological conditions.

With the technical progress of various analytical methods such as GC and the sensor system during the past few decades, breath measurement by GC and e-nose have become two common approaches. GC is a chemical analysis instrument for separating chemicals in a complex sample. By coupling with a detector, like Mass Spectrometry (MS) or Flame Ionization Detection (FID), it can positively identify the actual presence of a particular substance in a given sample. Despite its excellent sensitivity, GC usually requires the preprocessing of breath samples and separation for addressing target analytes, which renders this method less suitable for analyzing samples in real time. Besides, GC is expensive and hard to move. It requires skilled operators and qualified expert's interpretation. Therefore, it is difficult to implement GC as an online screening and quick diagnosis tool. For these reasons, e-nose might provide an alternative means of breath analysis.

E-nose utilizes chemical sensors to obtain 'smell-prints' of various gaseous sources and distinguish them with the help of pattern recognition algorithms, providing discrimination of gas mixtures irrespective of the individual molecular components. Compared with GC, e-nose measurement is regarded as a nonspecific test which principally follows an empirical approach. Although largely qualitative or semi-quantitative in nature, such approach is ideal for rapid screening for infectious diseases because the results can be obtained in minutes, rather than the days taken by traditional techniques (Turner and Magan 2004).

In the following sections, the current literatures about breath analysis by using both GC and e-nose are reviewed in detail. The reviewed contents are categorized according to the type of the diseases.

2.3 Breath Analysis by GC

In virtue of GC or GC linked with Mass Spectrometry (GC/MS), researchers can find out which biomarkers indicate some diseases and explain the pathological mechanisms associated with these diseases. The list of diseases reported below, is related to a series of works found in literature. Each of diseases is associated with certain biomarkers, which can be detected by GC or GC/MS.

2.3.1 Lung Cancer

In the past two decades, a noteworthy body of research about breath analysis has been oriented toward the identification of some particular VOCs as markers of lung cancer, one reason may be that the lung has a close connection with breath.

As early as 1985, by using a specially developed breath collection technique and computer-assisted GC/MS, Gordon et al. (1985) identified 22 VOCs, such as hexane, methylpentane, and benzene derivatives, in the exhaled air of patients with lung cancer. The GC/MS profiles of 12 diseased samples and 17 controlled samples were analyzed to distinguish patients from controlled group with the accuracy of over 80%.

Three years later, in 1988, O'Neill et al. (1988) also analyzed the compounds of exhaled breath from both lung cancer patients and healthy subjects, in virtue of GC/MS, and classified the compounds into 16 chemical classes, and then sorted all compounds into these chemical classes and classified the compounds at the >75% and >90% occurrence levels. Both the occurrence-rate components were then evaluated as diagnostic markers in a discriminant function model.

About a decade later, in 1999, Phillips et al. (1999a) collected breath samples from 108 patients with an abnormal chest radiograph and analyzed them by GC. The investigation found that a combination of 22 breath VOCs, predominantly alkane, alkane derivatives, and benzene derivatives, could discriminate between patients with and without lung cancer with 100% sensitivity and 81.3% specificity.

Table 2.3 The definition of sensitivity and specificity

		Test outcome		Sensitivity	Specificity
		Positive	Negative		
Actual condition	Positive	*tp*	*fn*	$\frac{tp}{tp+fn}$	$\frac{tn}{tn+fp}$
	Negative	*fp*	*tn*		

It is necessary to introduce the definition of sensitivity and specificity. In medicine, the reliability of a diagnosis is measured in terms of sensitivity and specificity, with the outcome being either positive (unhealthy) or negative (healthy). In the classification, the number of genuine sick subjects is denoted *tp*; misidentified healthy subjects is *fp*; genuine healthy subjects is *tn*; the misdiagnosed sick subjects is denoted as *fn* (Blatt et al. 2007). Sensitivity and specificity are thus defined as in Table 2.3.

And then, in 2003, to evaluate VOCs in the breath as tumor markers in lung cancer, Phillips et al. investigated the breath compounds of 178 bronchoscopy patients and 41 healthy volunteers by using GC (Phillips et al. 2003a). In this study, the number of biomarkers of lung cancer was reduced to nine in comparison with the report issued in 1999 (Phillips et al. 1999a). The results showed that a predictive model employing the nine VOCs could identify the primary lung cancer with a sensitivity of 89.6% and a specificity of 82.9%.

In these studies, it turned out that some specific compounds occur in anomalous concentration in the breath of lung cancer patients.

2.3.2 *Lipid Peroxidation*

Alkanes (principally ethane and pentane) in the breath result from cellular injuries which cause an intracellular accumulation of oxygen-free radicals and accelerated peroxidation of polyunsaturated fatty acids (Van Gossum and Decuyper 1989). The peroxidation of lipids may result in membrane injury, with the dysfunction and death of the affected cells. From 1991, several research groups started to find the connection between the breath pentane and diseases related to lipid peroxidation.

Weitz et al. (1991) first measured pentane in the breath of 10 healthy control subjects and 20 consecutive patients with suspected acute myocardial infarction. The results showed the breath pentane concentration was higher in the acute myocardial infarction group than in the patient control and healthy control groups.

Then, by using a GC, Sobotka et al. (1993a) measured compounds in the breath of patients with chronic heart failure (CHF) and age matched controls in 1993, and found out that the patients with CHF excreted high concentrations of pentane.

In the same year, to determine the concentrations of pentane and other VOCs in the breath of patients with schizophrenia, Phillips et al. (1993) measured the exhaled breath in 25 patients with acute schizophrenic psychosis, 26 patients with psychiatric disorders other than schizophrenia, and 37 normal volunteers by GC/MS. The results demonstrated that the mean alveolar gradients of pentane and carbon disulfide were significantly higher in the patients with schizophrenia than in the control groups. As a result, schizophrenia could be detected by measuring the concentration of pentane and carbon disulfide in breath.

Next year, in 1994, Sobotka et al. (1993b) studied 37 consecutive outpatients with stable cardiac allograft function. Breath pentane levels were measured with GC. The investigation found out that breath pentane could be measured as a potential marker of acute cardiac allograft rejection.

In 1995, Phillips et al. (1995) first combined GC with a self-designed Breath Collecting Apparatus (BCA) to analyze the breath samples. The composition of the subject database was the same as Ref. Phillips et al. (1993). Pattern recognition models using 11 VOCs, such as 2-methylbatane, pentane, and dichloromethane, identified the patients with schizophrenia with a sensitivity of 80% and a specificity of 61.9%. The paper also indicated that the VOCs in breath were not significantly affected by drug therapy, age, sex, smoking, diet, or race.

In 1997, to determine if exhaled pentane levels were increased in acute asthma, Olopade et al. (1997) collected 12 acute asthma patients, 11 stable asthma patients, and 17 normal control subjects and analyzed them using a GC. The result showed exhaled pentane levels were similar in patients with stable asthma and in normal control subjects, while the levels were increased in patients with acute asthma.

In 2003, by using GC, Phillips et al. (2003c) analyzed breath VOCs in 30 patients with unstable angina confirmed by coronary angiography and in 38 age-matched healthy volunteers. They selected 8 VOCs, like pentane and hexane as biomarkers to construct a predictive model that correctly classified unstable angina patients with a sensitivity of 90% and a specificity of 73.7%.

2.3.3 Renal Diseases

This kind of disease is due to the inability of kidneys to filter blood substances, resulting in the accumulation of nitrogen-bearing waste products (urea), which are usually excreted in urine and blood. It then eventually causes ammoniacal breath of patients.

In 1997, Davies et al. (1997) used selected ion flow tube(SIFT) technique to quantify ammonia on the breath of 23 patients with end-stage renal failure. The study showed several compounds were present in patients' breath samples, including amine and alcohol, and in quantitative terms ammonia was by far the most significant abnormality. The study also monitored the reduction of breath ammonia during hemodialysis. Accordingly, ammonia can be regarded as a critical biomarker to detect renal failure and monitor the medical treatment of this disease.

2.3.4 Liver Diseases

Liver diseases were first investigated by Sehnert et al. (2002), based on abnormal concentrations of metabolic products in exhaled breath. Exhaled breath collected from 86 liver diseases patients and 109 healthy subjects were analyzed by GC. The experiments showed that subjects with chronic liver diseases could be differentiated from those with normal liver function by comparing the levels of breath carbonyl sulphide, carbon disulphide, and isoprene. These differences were confirmed and correlated by comparing the levels with standard clinical blood markers of liver diseases.

2.3.5 Breast Cancer

Breast cancer is accompanied by increased oxidative stress and induction of polymorphic cytochrome P-450 mixed oxidase enzymes (CYP) (Phillips et al. 2003b). Both processes affect the abundance of VOCs in the breath because oxidative stress causes lipid peroxidation of polyunsaturated fatty acids in membranes, producing alkanes and methylalkanes which are catabolized by CYP (Phillips et al. 2003b).

In 2003, Phillips, et al. (2003b) collected 201 breath samples from women with breast cancer and analyzed them by GC/MS in order to determine the volatile markers of breast cancer. Eight breath VOCs, like nonane and tridecane, 5-methylused were used to identify this disease. The breath test distinguished between women with breast cancer and healthy volunteers with a sensitivity of 94.1% and a specificity of 73.8% (Phillips et al. 2003b).

2.3.6 Diabetes

It has long been known that the blood of diabetics contains acetone. Diabetes occurs when the glucose produced by the body cannot enter the bloodstream to provide energy to cells. Glucose enters the cells of body with the help of insulin. If the body is not producing insulin (type 1 diabetes), or the body becomes less responsive to insulin (type 2 diabetes), glucose cannot get into the cells. As a result, the cells have to use fat as an energy source. In the process of metabolizing fat for energy, one of the by-products is ketones. When ketones are accumulated in the blood, it first causes ketosis, and then progresses to ketoacidosis, a form of metabolic acidosis (Laffel 1999). There are three ketone bodies—acetoacetate, acetone, and β-hydroxybutyrate in the blood. Among them, β-hydroxybutyrate is the predominant ketone present in severe diabetic ketoacidosis (Umpierrez et al. 1995).

As early as 1969, Tassopoulos et al. (1969) measured the breath acetone of 251 diabetics after overnight fasting, by using GC. At the same time, the authors also measured the patients' venous β-hydroxybutyrate and blood glucose values, and showed that the concentration of breath acetone has quite a high correlation with both venous β-hydroxybutyrate and blood glucose values.

The relationship between breath acetone and plasma acetone was confirmed by Sulway et al. (1970) in 1970, who tested the plasma and breath acetone of 27 diabetics and discovered that the concentration of breath acetone and plasma acetone was linearly correlated with some scatter at the higher concentration.

Additionally, Crofford et al. (1977) proved that the concentration of acetone in the head space of the sealed container containing whole blood was approximately equal to the alveolar air acetone concentration. And then, in 1982, Owen et al. (1982) studied acetone metabolism in nine diabetic patients in moderate to severe ketoacidosis and observed that there was a positive linear relationship between the breath acetone production rate and the plasma acetone concentration. In 2004, Deng et al. (2004) analyzed the breath of healthy persons and patients with diabetes by using GC/MS. The results proved that the increased concentration of acetone in diabetics' breath could be used as a marker for diagnosis of diabetes.

2.3.7 Pulmonary Tuberculosis

Pulmonary tuberculosis may alter the VOCs in breath because both mycobacteria and oxidative stress resulting from mycobacterial infection generate distinctive VOCs in human body (Phillips et al. 2007).

Phillips et al. (2007) studied the breath of patients with pulmonary tuberculosis to determine if the breath contains biomarkers of this kind of disease in 2007. 130 different VOCs were consistently detected. The most abundant were naphthalene, 1-methyl-, 3-heptanone, etc. These VOCs were assayed by GC/MS in the breath of 42 patients hospitalized for suspicion of pulmonary tuberculosis and 59 healthy controls. Pattern recognition methods distinguished the healthy controls from the hospitalized patients with 100% sensitivity and 100% specificity.

2.3.8 Summary

Table 2.4 summarizes the key breath compounds associated with different disease types analyzed by both GC and pathological mechanism. Even though the clinical application of GC might be hampered by the need for expensive analytical equipment, the degree of expertise required to operate such instruments, and the length of time required to obtain results (Turner and Magan 2004), GC plays a critical role in confirming these compounds associated with certain diseases. These compounds not only help explain the pathological mechanism of these diseases, but also are of benefit to selecting proper sensors when designing the specific breath analysis system.

Table 2.4 Summary of key breath compounds associated with different disease types

Breath compounds	Associated conditions
Acetone	Diabetes (Deng et al. 2004)
Carbonyl sulphide, carbon disulphide, isoprene	Liver diseases (Sehnert et al. 2002)
Naphthalene, 1-methyl-, 3-heptanone, methylcyclododecane, etc.	Pulmonary tuberculosis (Phillips et al. 2007)
Nonane, tridecane, 5-methyl, undecane, 3-methyl, etc.	Breast cancer (Phillips et al. 2003b)
Benzene,1,1-oxybis-, 1,1-biphenyl, 2,2-diethyl, furan, 2,5-dimethyl-, etc.	Lung cancer (Phillips et al. 2003a)
Ammonia	Renal disease (Davies et al. 1997)
Octane, 4-methyl, decane, 4-methyl, hexane, etc.	Unstable angina (Salazar 2003)
Propane, 2-methyl, octadecane, octane, 5-methyl, etc.	Heart transplant rejection (Phillips et al. 2004)
Pentane, carbon disulfide	Schizophrenia (Phillips et al. 1993)
Pentane	Acute myocardial infarction (Weitz et al. 1991)
Pentane	Acute asthma (Olopade et al. 1997)
Pentane	Rheumatoid arthritis (Humad et al. 1988)
Ethane	Active ulcerative colitis (Sedghi et al. 1994)
Nitric oxide	Asthmatic inflammation (Baraldi and Carraro 2006)
Nitric oxide, carbon monoxide	Bronchiectasis (Kharitonov et al. 1995), (Horvath et al. 1998)
Nitric oxide	COPD (Maziak et al. 1998)
Ethane, propane, pentane, etc.	Cystic fibrosis (Barker et al. 2006)

2.4 Breath Analysis by E-Nose

The idea of e-nose was inspired by the mechanisms of human olfaction. In general, basic elements of an e-nose system include an ‘odor’ sensor array, a data preprocessor, and a pattern recognition engine (Craven et al. 1996). Among them, the sensor array, like signal receptors, is the key part of e-nose. The application of sensor array on odor recognition was demonstrated firstly by Persaud and Dodd (1982). Currently, e-nose has undergone much development and been used to fulfill a large number of industrial needs, such as food, chemistry, fragrances, security, and environment (Rock et al. 2008). In addition to its contributions to analytical chemistry and biotechnology, artificial olfaction also has a significant impact on the field of medicine since the compounds listed in Table 2.4 may be detected by chemical sensors (Dickinson et al. 1998). Recently, the feasibility of using e-noses for monitoring the health of human and diagnosing diseases in an early stage has been demonstrated (Lin et al. 2001; Yu et al. 2005; Blatt et al. 2007; Dragonieri et al. 2007).

As early as 1997, Wang et al. (1997) designed an e-nose with one SnO_2 thin film sensor for diabetes diagnosis. The authors tested their device by using the breath samples collected from 18 patients and 14 healthy persons. The concentration of blood sugar of the subjects was used as reference. The results showed that the e-nose was able to diagnose diabetes with a sensitivity of 77.8% and a specificity of 35.7%.

In 2001, Lin et al. (2001) reported a study about the application of e-nose with six quartz crystal sensors to detect renal diseases. Discriminant Analysis (DA) was carried out to analyze the sensor signals. The clinical test result showed that the e-nose could discriminate the breath samples from 30 normal subjects, 83 uremia patients, and 61 chronic renal disease patients with a total correct classification of 86.78%.

In 2003, Yu et al. (2004) developed an e-nose with two SAW sensors for lung cancer detection. The breath samples of four patients with lung cancer and four normal subjects were collected by using Tedlar bags and then pre-concentrated by solid phase micro extraction (SPME) to increase the sensitivity. The e-nose was calibrated by 9 VOCs identified as the markers of lung cancer. An Artificial Neural Network (ANN) was used to recognize the lung cancer patients. The result showed that in four healthy samples, three of them were recognized correctly and one of them was recognized as suspected patient; in four patients, three of them were diagnosed correctly and one of them was diagnosed as suspected.

In 2003, Di Natale et al. (2003) used an e-nose composed by eight quartz microbalance (QMB) gas sensors to analyze the breath samples, which were collected from 60 individuals, 35 of them were affected by lung cancer, 18 individuals were measured as healthy, and 9 were measured after the surgical therapy. The application of a Partial Least Squares Discriminant Analysis (PLS-DA) found out that 100% lung cancer-affected patients were classified correctly, 94% healthy individuals were classified correctly, and 44% of post-surgery patients were classified correctly.

In 2005, Yu et al. (2005) developed a gas analyzing system using four conducting polymer sensors to analyze the breath samples from three diabetics and three normal people. The discrimination between patients and normal persons were interpreted by the PCA plus Euclidean distances with 100% sensitivity and 100% specificity.

In 2005, Machado et al. (2005) investigated exhaled breath of people by using a commercial e-nose, the Cyrano Sciences' Cyranose 320, comprising an array of 32 polymer carbon black composite sensors. PCA and Canonical Discriminant Analysis (CDA) sensor data were used to determine whether exhaled gases could discriminate between cancer and non-cancer. Support Vector Machine (SVM) analysis was used to create a cancer prediction model prospectively in a separate group of 76 individuals, 14 with cancer, and 62 without cancer. The results showed a sensitivity of 71.4% and a specificity of 91.9% of lung cancer detection.

In 2007, Dragonieri et al. also used Cyranose 320 to obtain the responses of exhaled air of patients with asthma and healthy controls. The responses were analyzed by LDA. Cross-validation values plus Mahalanobis distance were calculated for classification. The accuracy to classify the mild asthma and young controls is 100%, to classify severe asthma and old controls is 90%, to classify mild and severe asthma is 65%, and to classify two controlled groups is 50%.

In 2007, Blatt et al. (2007) reported their work about lung cancer detection by using an e-nose with 6 MOS sensors. They analyzed the breath of 101 persons, of which 58 as controls and 43 suffering from different types of lung cancer (primary and not) at different stages. Nonparametric LDA was used to extract the features of the sensors' responses. The features were classified by several supervised pattern classification techniques, based on different K-nearest neighbor (KNN) approaches, linear and quadratic discriminant classifiers, and on a feed forward ANN. The observed results showed an accuracy of 92.6%, a sensitivity of 95.3%, and a specificity of 90.5% for lung cancer diagnosis.

In 2009, Ogorodnik et al. (2008) analyzed VOCs from a breath sample of a patient with different lung diseases by using an e-nose with ten MOSFET sensors and four SnO_2 sensors. In total, 66 individuals—23 with asthma, 3 with chronic obstructive pulmonary disease (COPD), 12 with pneumonia, 13 with lung cancer, 4 in the past operation state (removed lung cancer), and 11 healthy volunteers were tested at two different times and ANN analysis was employed to classify the samples of cancer and other lung diseases. The results showed that the e-nose could identify lung cancer with 100% accuracy, identify healthy subjects with 100% accuracy, and identify asthma with 82.6% accuracy.

In 2009, using Cyranose 320, Dragonieri et al. (2009) analyzed the exhaled breath samples to discriminate patients with lung cancer from COPD patients and healthy controls. The breath samples were collected from 30 subjects, 10 patients with non-small cell lung cancer, 10 patients with COPD, and 10 healthy controls. The responses were analyzed by onboard statistical software. The method could distinguish non-small cell lung cancer from COPD and from normal people with 85% and 90% accuracy, respectively.

In 2010, Guo et al. (2010b) designed a breath analysis system, which includes 12 chemical sensors that are specially sensitive to the biomarkers and compositions in human breath. 108 healthy breath samples, 117 samples from diabetics, 110 samples from patients with renal diseases, and 110 samples from patients with airway inflammation were collected. PCA + KNN were used to evaluate the performance. The results showed that the system was not only able to diagnose these diseases with quite high accuracy, but in the case of renal failure was also helpful in evaluating the efficacy of hemodialysis (treatment for renal failure).

In 2010, by using the same system and the same diabetes breath samples, Guo et al. (2010c) proposed a method of monitoring the blood glucose levels of diabetics via measuring the concentration of breath acetone. A SVM classifier was used to evaluate the accuracy of classifying the samples into the groups with different blood glucose levels. The results indicated that the system was not only able to distinguish between breath samples from patients with diabetes and healthy subjects, but also to represent the fluctuation of blood glucose of diabetics. In the same year, Guo et al. (2010a) improved accuracy of diabetes condition monitoring by using a SRC method. Coupling with SRC, the system was able to classify these levels with a much better accuracy than the accuracy reported in Guo et al. 2010c.

In 2013, Saraolu et al. (2013) tried to develop an e-nose with 9 quartz crystal microbalance (QCM) sensors. The e-nose was used to measure the breath of 30 diabetes patients. Signals from 6 sensors were normalized then fed into a radial basis function neural network (RBFNN). The final average accuracy rate was 83.03 and 74.76% for HbA1c parameter predictions and glucose parameter predictions, respectively.

In 2014, an e-nose with 6 MOS sensors, 3 temperature modulated MOS sensors, a carbon dioxide sensor, and a temperature-humidity sensor was proposed by Yan et al. (2014). It was optimized for diabetes screening and blood glucose level prediction. Several optimization strategies, such as sensor selection, humidity and alveolar air ratio compensation, and inter-subject variance reduction, were implemented. The sensitivity and specificity of diabetes screening were 91.51% and 90.77%, respectively. The mean relative absolute error for BGL prediction was 21.7%. Experiments showed that the system was effective and that the strategies adopted in the system could improve its accuracy.

The same e-nose was further applied to collect breath samples from 5 kinds of patients, see Table 2.5. They have been proved to be related to certain breath biomarkers. The paper (Yan and Zhang 2016) proposed drift correction autoencoder (DCAE) to deal with instrumental variation and complex time-varying drift of e-noses. Experiments in the paper exhibited the potential of breath analysis systems as adjunct tools for disease screening.

To sum up, Table 2.5 concludes the current reports about the medical applications of e-noses.

From Table 2.5, we can see some limitations about the current researches: (1) Even though some works provided promising disease identification results, the sample number they used are not enough to provide a stronger statistical evidence to support the claim. (2) Most of the relevant systems have fewer sensors. We agree that it is not going to be very useful by simply adding more sensors. But it is necessary to provide a sufficiently redundant amount of sensors thus we can pick up the most sensitive ones in applications. Consequently, it therefore requires us to add more sensors in our system and collect enough typical samples for analysis.

2.5 Summary

This chapter reviewed some previous researches about breath analysis. General breath analysis approaches, like GC and e-nose, were introduced according to the type of the diseases analyzed. And some summaries were made about the disease biomarkers and current approaches. From these summaries, we can see that even though all of these methods work satisfactorily in breath analysis, the results could possibly be improved. The portable and low cost device is required to achieve a broad application in breath analysis.

Table 2.5 The application of e-noses in medicine

Diseases	Sensors	Database	Algorithm	Results
Diabetes (Wang et al. 1997)	1 SnO_2 thin film sensor	18 patients	Fuzzy clustering	Sensitivity: 78%
		14 healthy persons		Specificity: 36%
Renal diseases (Lin et al. 2001)	6 quartz crystal sensors	30 healthy persons	DA	CRI/CRF: 90.16%
		83 uremia		Uremia: 79.52%
		61 chronic renal disease		Healthy: 100%
Lung cancer (Yu et al. 2004)	2 SAW sensors +GC	4 lung cancer	ANN	3 lung cancer
		4 healthy persons		2 suspected
				3 healthy persons
Lung cancer (Di Natale et al. 2003)	8 QMB gas sensors	35 lung cancer	PLS-DA	Lung cancer: 94
		9 post-surgery		Post-surgery: 44
		17 healthy persons		healthy: 100%
Diabetes (Yu et al. 2005)	4 conducting polymer	3 diabetics	PCA + Euclidean distances	Sensitivity: 100%
		3 healthy persons		Specificity: 100%
Lung cancer (Machado et al. 2005)	32 carbon black and polymers sensors	14 lung cancer	SVM	Sensitivity: 71.4%
		62 healthy persons		Specificity: 91.9%
Asthma (Dragonieri et al. 2007)	32 carbon black and polymers sensors	10 mild asthma	PCA+Mahalanobis distances	Mild asthma and young controls: 100%
		10 severe asthma		Severe asthma and old controls: 90%
		10 younger controls		Mild and severe asthma: 65%
		10 older controls		Two controlled groups: 50%
Lung cancer (Blatt et al. 2007)	6 MOS sensors	43 lung cancer	Fuzzy-KNN	Sensitivity: 95.3%
		58 controlled patients		Specificity: 90.5%

(continued)

Table 2.5 (continued)

Diseases	Sensors	Database	Algorithm	Results
Lung cancer (Ogorodnik et al. 2008)	6 MOSFET sensors	23 asthma	ANN	Lung cancer: 100%
	4 MOS sensors	3 COPD		Healthy: 100%
		12 pneumonia		Others: 82.6%
		13 lung cancer		
		4 post surgery		
		11 healthy persons		
Lung cancer (Dragonieri et al. 2009)	32 carbon black and polymers sensors	10 lung cancer	PCA + Mahalanobis distances	Distinguish lung cancer from COPD: 85%
		10 COPD		From healthy: 90%
		10 healthy controls		
Diabetes renal diseases airway inflammation (Guo et al. 2010b)	12 MOS sensors	108 healthy	PCA+KNN	Diabetes: sensitivity: 87.67%
		117 diabetes		Specificity: 86.87%
		110 renal diseases		Renal diseases: sensitivity: 86.57%
		110 airway Inflammation		Specificity: 83.47%
				Airway inflammation: sensitivity: 70.20%
				Specificity: 75.07%
Diabetes (Guo et al. 2010a)	12 MOS sensors	90 diabetes:	PCA + SRC	Level 1: 50%
		4 level 1		Level 2: 83.67%
		49 level 2		Level 3: 60%
		20 level 3		Level 4: 76.47%
		17 level 4		
Diabetes (Saraoğlu et al. 2013)	9 QCM sensors	30 patients	RBFNN	HbA1c: 83.03%
				BG: 74.76%

(continued)

Table 2.5 (continued)

Diseases	Sensors	Database	Algorithm	Results
Blood glucose (BG) and HbA1c level for diabetics (Yan et al. 2014)	6 MOS sensors	295 healthy 279 diabetes	PCA + SVM	Diabetes: 82.16%
	3 temperature modulated MOS sensors			CKD: 84.27%
	1 carbon dioxide sensor			Cardiopathy: 89.94%
	1 temperature-humidity sensor			Lung cancer: 81.34%
				Breast cancer: 82.92%
Diabetes chronical kidney disease (CKD) cardiopathy lung cancer breast cancer (Yan and Zhang 2016)	6 MOS sensors	125 healthy	DCAE + logistic regression	Diabetes: 82.16%
	3 temperature modulated MOS sensors	431 diabetes		CKD: 84.27%
	1 carbon dioxide sensor	340 CKD		Cardiopathy: 89.94%
	1 temperature-humidity sensor	97 cardiopathy		Lung cancer: 81.34%
		156 lung cancer		Breast cancer: 82.92%
		215 breast cancer		

References

Baraldi E, Carraro S (2006) Exhaled NO and breath condensate. Paediatr Respir Rev 7:20–22

Barker M, Hengst M, Schmid J, Buers H, Mittermaier B, Klemp D, Koppmann R (2006) Volatile organic compounds in the exhaled breath of young patients with cystic fibrosis. Eur Respir J 27(5):929–936

Blatt R, Bonarini A, Calabro E, Della Torre M, Matteucci M, Pastorino U (2007) Lung cancer identification by an electronic nose based on an array of MOS sensors. In: International joint conference on neural networks 2007, pp 1423–1428

Cao W, Duan Y (2007) Current status of methods and techniques for breath analysis. Crit Rev Anal Chem 37(1):3–13

Craven M, Gardner J, Bartlett P (1996) Electronic noses-development and future prospects. Trends Anal Chem 15(9):486–493

Crofford O, Mallard R, Winton R, Rogers N, Jackson J, Keller U (1977) Acetone in breath and blood. Trans Am Clin Climatol Assoc 88:128–139

Davies S, Spanel P, Smith D (1997) Quantitative analysis of ammonia on the breath of patients in end-stage renal failure. Kidney Int 52(1):223–228

Deng C, Zhang J, Yu X, Zhang W, Zhang X (2004) Determination of acetone in human breath by gas chromatography-mass spectrometry and solid-phase microextraction with on-fiber derivatization. J Chromatogr B 810(2):269–275

Deykin A, Massaro A, Drazen J, Israel E (2002) Exhaled nitric oxide as a diagnostic test for asthma: online versus offline techniques and effect of flow rate. American J Respir Crit Care Med 165(12):1597–1601

Di Francesco F, Fuoco R, Trivella M, Ceccarini A (2005) Breath analysis: trends in techniques and clinical applications. Microchem J 79(1–2):405–410

Di Natale C, Macagnano A, Martinelli E, Paolesse R, D'Arcangelo G, Roscioni C, Finazzi-Agrò A, D'Amico A (2003) Lung cancer identification by the analysis of breath by means of an array of non-selective gas sensors. Biosens Bioelectron 18(10):1209–1218

Dickinson T, White J, Kauer J, Walt D (1998) Current trends in "artificial-nose" technology. Trends Biotechnol 16(6):250–258

Dragonieri S, Schot R, Mertens B, Le Cessie S, Gauw S, Spanevello A, Resta O, Willard N, Vink T, Rabe K et al (2007) An electronic nose in the discrimination of patients with asthma and controls. J Allergy Clin Immunol 120(4):856–862

Dragonieri S, Annema J, Schot R, van der Schee M, Spanevello A, Carratú P, Resta O, Rabe K, Sterk P (2009) An electronic nose in the discrimination of patients with non-small cell lung cancer and COPD. Lung Cancer 64(2):166–170

Dweik R, Amann A (2008) Exhaled breath analysis: the new frontier in medical testing. J Breath Res 2(030):301

Fleischer M, Simon E, Rumpel E, Ulmer H, Harbeck M, Wandel M, Fietzek C, Weimar U, Meixner H (2002) Detection of volatile compounds correlated to human diseases through breath analysis with chemical sensors. Sens Actuators B: Chem 83(1–3):245–249

Gordon S, Szidon J, Krotoszynski B, Gibbons R, O'Neill H (1985) Volatile organic compounds in exhaled air from patients with lung cancer. Clin Chem 31(8):1278–1282

Guo D, Zhang D, Li N (2010a) Monitor blood glucose levels via breath analysis system and sparse representation approach. In: Sensors, 2010 IEEE. IEEE, pp 1238–1241

Guo D, Zhang D, Li N, Zhang L, Yang J (2010b) A novel breath analysis system based on electronic olfaction. IEEE Trans Biomed Eng 57(11):2753–2763

Guo D, Zhang D, Li N, Zhang L, Yang J (2010c) Diabetes identification and classification by means of a breath analysis system. In: International conference on medical biometrics, pp 52–63

Horvath I, Loukides S, Wodehouse T, Kharitonov S, Cole P, Barnes P (1998) Increased levels of exhaled carbon monoxide in bronchiectasis: a new marker of oxidative stress. British Med J 53(10):867–870

Humad S, Zarling E, Clapper M, Skosey J (1988) Breath pentane excretion as a marker of disease activity in rheumatoid arthritis. Free Radiac Res 5(2):101–106

Kharitonov S, Wells A, O'connor B, Cole P, Hansell D, Logan-Sinclair R, Barnes P (1995) Elevated levels of exhaled nitric oxide in bronchiectasis. Am J Respir Crit Care Med 151(6):1889–1893

Laffel L (1999) Ketone bodies: a review of physiology, pathophysiology and application of monitoring to diabetes. Diabetes/Metab Res Rev 15(6):412–426

Lin Y, Guo H, Chang Y, Kao M, Wang H, Hong R (2001) Application of the electronic nose for uremia diagnosis. Sens Actuators B Chem 76(1–3):177–180

Machado R, Laskowski D, Deffenderfer O, Burch T, Zheng S, Mazzone P, Mekhail T, Jennings C, Stoller J, Pyle J et al (2005) Detection of lung cancer by sensor array analyses of exhaled breath. Am J Respir Crit Care Med 171(11):1286–1291

Maziak W, Loukides S, Culpitt S, Sullivan P, Kharitonov S, Barnes P (1998) Exhaled nitric oxide in chronic obstructive pulmonary disease. Am J Respir Crit Care Med 157(3):998–1002

McGrath L, Patrick R, Mallon P, Dowey L, Silke B, Norwood W, Elborn S (2000) Breath isoprene during acute respiratory exacerbation in cystic fibrosis. Eur Respir J 16(6):1065–1069

Miekisch W, Schubert J (2006) From highly sophisticated analytical techniques to life-saving diagnostics: technical developments in breath analysis. Trends Anal Chem 25:665–673

Ogorodnik V, Kleperis J, Taivans I, Jurka N, Bukovskis M (2008) Electronic nose for identification of lung diseases. Latv J Phys Techn Sci 45(5):60–67

Olopade C, Zakkar M, Swedler W, Rubinstein I (1997) Exhaled pentane levels in acute asthma. Chest 111(4):862–865

O'Neill H, Gordon S, O'Neill M, Gibbons R, Szidon J (1988) A computerized classification technique for screening for the presence of breath biomarkers in lung cancer. Clin Chem 34(8):1613

Owen O, Trapp V, Skutches C, Mozzoli M, Hoeldtke R, Boden G, Reichard G (1982) Acetone metabolism during diabetic ketoacidosis. Diabetes 31(3):242–248

Pauling L, Robinson A, Teranishi R, Cary P (1971) Quantitative analysis of urine vapor and breath by gas-liquid partition chromatography. Proc Natl Acad Sci 68(10):2374–2376

Persaud K, Dodd G (1982) Analysis of discrimination mechanisms in the mammalian olfactory system using a model nose. Nature 299:352–355

Phillips M (1992) Breath tests in medicine. Sci Am 267(1):74–79

Phillips M, Sabas M, Greenberg J (1993) Increased pentane and carbon disulfide in the breath of patients with schizophrenia. J Clin Pathol 46(9):861–864

Phillips M, Erickson G, Sabas M, Smith J, Greenberg J (1995) Volatile organic compounds in the breath of patients with schizophrenia. J Clin Pathol 48(5):466

Phillips M, Gleeson K, Hughes J, Greenberg J, Cataneo R, Baker L, McVay W (1999a) Volatile organic compounds in breath as markers of lung cancer: a cross-sectional study. Lancet 353(9168):1930–1933

Phillips M, Herrera J, Krishnan S, Zain M, Greenberg J, Cataneo R (1999b) Variation in volatile organic compounds in the breath of normal humans. J Chromatogr B Biomed Sci Appl 729(1–2):75–88

Phillips M, Cataneo R, Cummin A, Gagliardi A, Gleeson K, Greenberg J, Maxfield R, Rom W (2003a) Detection of Lung Cancer With Volatile Markers in the Breath. Chest 123(6):2115–2123

Phillips M, Cataneo R, Ditkoff B, Fisher P, Greenberg J, Gunawardena R, Kwon C, Rahbari-Oskoui F, Wong C (2003b) Volatile markers of breast cancer in the breath. Breast J 9(3):184–191

Phillips M, Cataneo R, Greenberg J, Grodman R, Salazar M (2003c) Breath markers of oxidative stress in patients with unstable angina. Heart disease (Hagerstown, Md) 5(2):95–99

Phillips M, Boehmer J, Cataneo R, Cheema T, Eisen H, Fallon J, Fisher P, Gass A, Greenberg J, Kobashigawa J et al (2004) Heart allograft rejection: detection with breath alkanes in low levels (the HARDBALL study). J Heart Lung Transplant 23(6):701–708

Phillips M, Cataneo R, Condos R, Ring Erickson G, Greenberg J, La Bombardi V, Munawar M, Tietje O (2007) Volatile biomarkers of pulmonary tuberculosis in the breath. Tuberculosis 87(1):44–52

Risby T, Solga S (2006) Current status of clinical breath analysis. Appl Phys B Lasers Opt 85(2):421–426

Rock F, Barsan N, Weimar U (2008) Electronic nose: current status and future trends. Chem Rev 108(2):705–725

Salazar M (2003) Breath markers of oxidative stress in patients with unstable angina. Heart Dis 5(2):95–99

Saraoğlu HM, Selvi AO, Ebeoğlu MA, Taşaltin C (2013) Electronic nose system based on quartz crystal microbalance sensor for blood glucose and hba1c levels from exhaled breath odor. IEEE Sens J 13(11):4229–4235

Sedghi S, Keshavarzian A, Klamut M, Eiznhamer D, Zarling E (1994) Elevated breath ethane levels in active ulcerative colitis: evidence for excessive lipid peroxidation. Am J Gastroenterol 89(12):2217–2221

Sehnert S, Jiang L, Burdick J, Risby T (2002) Breath biomarkers for detection of human liver diseases: preliminary study. Biomarkers 7(2):174–187

Sobotka P, Brottman M, Weitz Z, Birnbaum A, Skosey J, Zarling E (1993a) Elevated breath pentane in heart failure reduced by free radical scavenger. Free Radic Biol Med 14(6):643–647

Sobotka PA, Gupta DK, Lansky DM, Costanzo MR, Zarling EJ (1993b) Breath pentane is a marker of acute cardiac allograft rejection. J Heart Lung Transplant Off Publ Int Soc Heart Transplant 13(2):224–229

Sulway M, Malins J (1970) Acetone in diabetic ketoacidosis. Lancet 296(7676):736–740

Tassopoulos C, Barnett D, Russell Fraser T (1969) Breath-acetone and blood-sugar measurements in diabetes. Lancet 293(7609):1282–1286

Turner A, Magan N (2004) Electronic noses and disease diagnostics. Nat Rev Microbiol 2(2):161–166

Umpierrez G, Watts N, Phillips L (1995) Clinical utility of beta-hydroxybutyrate determined by reflectance meter in the management of diabetic ketoacidosis. Diabetes Care 18(1):137–138

Van Berkel J, Dallinga J, Möller G, Godschalk R, Moonen E, Wouters E, Van Schooten F (2008) Development of accurate classification method based on the analysis of volatile organic compounds from human exhaled air. J Chromatogr B 861(1):101–107

Van Gossum A, Decuyper J (1989) Breath alkanes as an index of lipid peroxidation. Eur Respir J 2(8):787

Wang P, Tan Y, Xie H, Shen F (1997) A novel method for diabetes diagnosis based on electronic nose. Biosens Bioelectron 12(9–10):1031–1036

Weitz Z, Birnbaum A, Sobotka P, Zarling E, Skosey J (1991) High breath pentane concentrations during acute myocardial infarction. Lancet 337(8747):933–935

Yan K, Zhang D (2016) Correcting instrumental variation and time-varying drift: a transfer learning approach with autoencoders. IEEE Trans Instrum Meas 65(9):2012–2022

Yan K, Zhang D, Wu D, Wei H, Lu G (2014) Design of a breath analysis system for diabetes screening and blood glucose level prediction. IEEE Trans Biomed Eng 61(11):2787–2795

Yu H, Xu L, Cao M, Chen X, Wang P, Jiao J, Wang Y (2004) Detection volatile organic compounds in breath as markers of lung cancer using a novel electronic nose. Proc IEEE Sens IEEE 2:1333–1337

Yu J, Byun H, So M, Huh J (2005) Analysis of diabetic patient's breath with conducting polymer sensor array. Sens Actuators B Chem 108(1–2):305–308

Zhufan X (2000) Selected terms in traditional chinese medicine and their interpretations. Chin J Integr Med 6(3):230–232

Part II
Breath Acquisition Systems

Chapter 3
A Novel Breath Acquisition System Design

Abstract Certain gases in the breath are known to be indicators of the presence of diseases and clinical conditions. These gases have been identified as biomarkers using equipments such as gas chromatography (GC) and electronic nose (e-nose). GC is very accurate but is expensive, time consuming, and non-portable. E-nose has the advantages of low-cost and easy operation, but is not particular for analyzing breath odor and hence has a limited application in diseases diagnosis. This chapter proposes a novel system that is special for breath analysis. We selected chemical sensors that are sensitive to the biomarkers and compositions in human breath, developed the system, and introduced the odor signal preprocessing and classification method. To evaluate the system performance, we captured breath samples from healthy persons and patients known to be afflicted with diabetes, renal disease, and airway inflammation respectively and conducted experiments on medical treatment evaluation and disease identification. The results show that the system is not only able to distinguish between breath samples from subjects suffering from various diseases or conditions (diabetes, renal disease, and airway inflammation) and breath samples from healthy subjects, but in the case of renal failure is also helpful in evaluating the efficacy of hemodialysis (treatment for renal failure).

Keywords Breath analysis · Electronic olfaction · Therapy monitoring · Chemical sensor · Disease identification

3.1 Introduction

In recent years, there were increasing concerns about the applications of breath analysis in medicine and clinical pathology both as a diagnostic tool and as a way to monitor the progress of therapies (Di Francesco et al. 2005; Dweik and Amann 2008). Comparing with other traditional methods such as blood and urine test, breath analysis is noninvasive, real-time, and least harmless to not only the subjects but also the personnel who collect the samples (Van Berkel et al. 2008). The measurement of breath air is usually performed by gas chromatography (GC) (Phillips 1997) or electronic nose (e-nose) (Thaler and Hanson 2005). GC is very accurate but is expensive

D. Zhang et al., *Breath Analysis for Medical Applications*,
DOI 10.1007/978-981-10-4322-2_3

and not portable, its sampling and assaying processes are complicated and time consuming (about one hour for one sample), and its results require expert interpretation (Amann et al. 2004). A less expensive and more portable alternative is e-nose. It is cheaper and faster (requiring only 30 min for one sample) and is often used outside of medicine, in fields related to food, chemistry, fragrances, security, and environment (Rock et al. 2008). Recently, e-nose has gradually been used in medicine for the diagnosis of renal disease (Lin et al. 2001), diabetes (Yu et al. 2005), lung cancer (Blatt et al. 2007), and asthma (Dragonieri et al. 2007). While all of these methods work satisfactorily, they can each identify only one particular disease. One reason for the limited applications of e-noses in breath analysis might be the design of commercial e-noses for broad applications rather than for breath analysis specifically. We thus propose a new specific breath analysis system in this chapter in order to extend the applications in medicine.

The system makes use of chemical sensors that are particularly sensitive to the biomarkers and compositions in human breath to trigger responses to a patient's breath sample. In contrast to the broad panel of nonspecific sensors used in commercial e-noses, the sensors of our system were specifically selected for their responses to known components of human breath. The sample is injected into the system using an auto-sampler at a fixed injection rate to guarantee all samples are sampled under the same criterion. The chemical sensors sense the sample and accordingly form a kind of "odorprint" that is typically associated with a given disease or condition. The "odorprint" is then sent to computer for signal processing and pattern recognition. We evaluated the system in two experiments. In the first, we classified subjects with renal failure before and after hemodialysis. In the second, we applied the system to distinguish between healthy subjects and subjects suffering from three types of diseases/conditions (diabetes, renal disease, and airway inflammation). The experimental results show that our system can fairly accurately measure whether hemodialysis has been effective and can identify the three conditions/diseases with quite a high level of accuracy.

The remainder of this chapter is organized as follows. Section 3.2 describes the composition of the human breath and the certain diseases that may be associated with certain gaseous compounds. Section 3.3 describes how a subject's breath is sampled, the setup of the sensor array, and how data is processed. Section 3.4 explains the experimental details. Section 5 gives the experimental results and discussion. Section 3.6 summarizes our work.

3.2 Breath Analysis

Human breath is largely composed of oxygen, carbon dioxide, water vapor, nitric oxide, and numerous volatile organize compounds (VOCs) (Cao and Duan 2007). The type and number of the VOCs in the breath of any particular individual will vary but there is nonetheless a comparatively small common core of breath which are present in all humans (Phillips et al. 1999). The molecules in an individual's

Table 3.1 Typical compositions from the endogenous breath of the healthy persons

Concentration (v/v)	Molecule
Percentage	Oxygen, water, carbon dioxide
Parts-per-million	Acetone, carbon monoxide, methane, hydrogen, isoprene, benzenemethanol
Parts-per-billion	Formaldehyde, acetaldehyde, 1-pentane, ethane, ethylene, other hydrocarbons, nitric oxide, carbon disulfide, methanol, carbonyl sulfide, methanethiol, ammonia, methylamine, dimethyl sulfide, benzene, naphthalene, benzothiazole, ethane, acetic aide

breath may be exogenous or endogenous (Miekisch and Schubert 2006). Exogenous molecules are those that have been inhaled or ingested from the environment or other sources such as air or food and hence no diagnostic value (Risby and Solga 2006). Endogenous molecules are produced by metabolic processes and partition from blood via the alveolar pulmonary membrane into the alveolar air. These endogenous molecules are present in breath relative to their types, concentrations, volatilities, lipid solubility, and rates of diffusion as they circulate in the blood and cross the alveolar membrane (Sehnert et al. 2002). Changes in the concentration of the molecules in VOCs could suggest various diseases or at least changes in the metabolism. Table 3.1 summarizes the typical compositions found in the endogenous breath of healthy persons (Phillips et al. 1999; Risby and Solga 2006).

Some molecules such as nitric oxide, isoprene, pentane, benzene, acetone, and ammonia may indicate specific pathologies (DAmico et al. 2007; Schubert et al. 2004; Miekisch et al. 2004). To take a few examples, nitric oxide can be measured as an indicator of asthma or other conditions characterized by airway inflammation (Deykin et al. 2002). Breath isoprene is significantly lower in patients with acute respiratory exacerbation of cystic fibrosis (McGrath et al. 2000). Increased pentane and carbon disulfide have been observed in the breath of patients with schizophrenia (Phillips et al. 1993). The concentration of VOCs such as cyclododecatriene, benzoic acid, and benzene are much higher in lung cancer patients than in control groups (Phillips et al. 2007a). Acetone has been found to be more abundant in the breath of diabetics (Deng et al. 2004). Ammonia is significantly elevated in patients with renal disease (Davies et al. 1997). Table 3.2 lists some breath compounds and the conditions that research has found to be associated with them. The compounds and conditions listed in Tables 3.1 and 3.2 were the focus of the work being described in this chapter.

3.3 Description of the System

The proposed system operates in three phases (Fig. 3.1), gas collection, sampling, and data analysis, with a subject first breathing into a Tedlar gas sampling bag. This

Table 3.2 Some breath compounds and associated conditions

Breath compounds	Associated conditions
Acetone	Diabetes (Deng et al. 2004)
Carbonyl sulfide, carbon disulfide, isoprene	Liver diseases (Sehnert et al. 2002)
Naphthalene,1-methyl-, 3-heptanone, methylcyclododecane, etc.	Pulmonary tuberculosis (Phillips et al. 2007b)
Nonane, tridecane, 5-methyl, undecane, 3-methyl, etc.	Breast cancer (Phillips et al. 2003b)
Benzene,1,1-oxybis-, 1,1-biphenyl,2,2 -diethyl, furan,2,5-dimethyl-, etc.	Lung cancer (Phillips et al. 2003a)
Ammonia	Renal disease (Davies et al. 1997)
Octane,4-methyl, decane, 4-methyl, hexane, etc.	Unstable angina (Salazar 2003)
Propane,2-methyl, octadecane, octane, 5-methyl, etc.	Heart transplant rejection (Phillips et al. 2004)
Pentane, carbon disulfide	Schizophrenia (Phillips et al. 1993)
Pentane	Acute myocardial infarction (Weitz et al. 1991)
Pentane	Acute asthma (Olopade et al. 1997)
Pentane	Rheumatoid arthritis (Humad et al. 1988)
Ethane	Active ulcerative colitis (Sedghi et al. 1994)
Nitric oxide	Asthmatic inflammation (Baraldi and Carraro 2006)
Nitric oxide, Carbon monoxide	Bronchiectasis (Kharitonov et al. 1995; Horvath et al. 1998)
Nitric oxide	COPD (Maziak et al. 1998)
Ethane, propane, pentane, etc.	Cystic fibrosis (Barker et al. 2006)

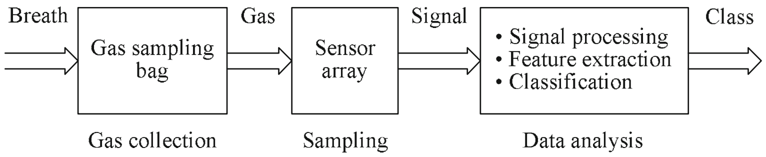


Fig. 3.1 The working flow defined in our system

gas is then injected into a chamber containing a sensor array where a measurement circuit measures the interaction between the breath and the array. The signals are then filtered and amplified and sent to computer for further analysis. Figure 3.2 shows our system (left) and its laptop interface.

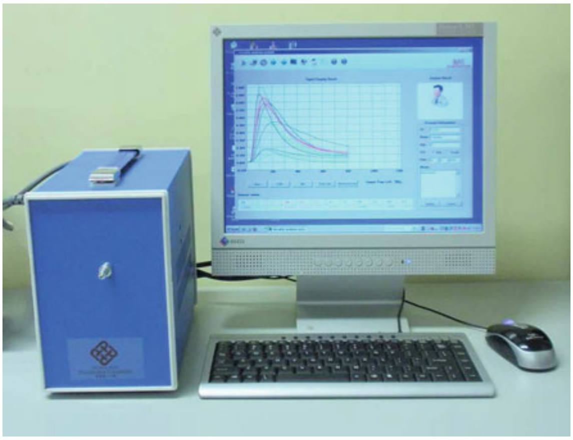


Fig. 3.2 Breath analysis system and the working interface

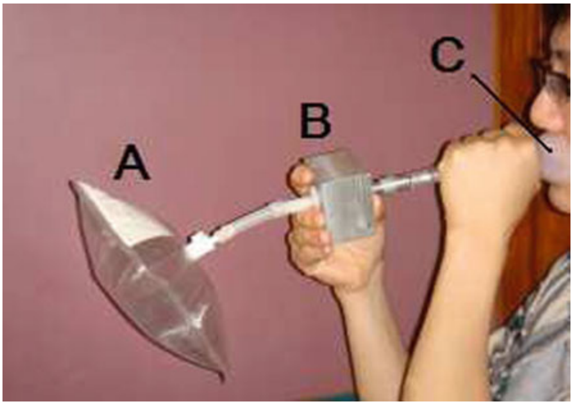


Fig. 3.3 Exhaled air is collected with a gas sampling bag

3.3.1 Breath Gas Collecting

Figure 3.3 shows how the subject's breath is collected using a 600 ml Tedlar gas sampling bag (A), an airtight box (B) filled with disposable hygroscopic material to absorb the water vapor from the breath, and, the last component, a disposable mouthpiece (C). The hygroscopic material is silica gel. It is stable and only reacts with several components, such as fluoride, strong bases, and oxidizers. None of them is involved in the breath components showed in Tables 3.1 and 3.2. Our previous experiments had shown there was no obvious effect on the disease identification by using silica gel as a hygroscopic material. In any case, the mouthpiece is equipped with an anti-siphon valve that prevents inhalation of the gel.

Subjects are required to give their breath sample in one of two different ways depending on whether the condition under consideration typically exhibits its biomarkers (compounds) in what are known as, dead-space air from the upper airway, or alveolar air from the lungs. Depending on the type of biomarkers and on the breath

Table 3.3 Detailed information with respect to collected diseases

Compounds	Breath sampling locations	Conditions
Acetone	Alveolar air	Diabetes
Ammonia	Dead-space air	Renal disease
Nitric oxide	Dead-space air	Airway inflammation

test tracks, dead-space air may be either a necessity or a contaminant (Cao and Duan 2007). Alveolar air is required where a condition is typified by biomarkers that are found when there is an exchange from circulating blood. In contrast, dead-space air is required when the biomarkers are released into the airways, and thus into the dead-space air. Table 3.3 lists some of the compounds, conditions, and breath sampling locations used in this work (Davies et al. 1997; Turner and Magan 2004).

Alveolar and dead-space airs are collected in different ways. Alveolar air is collected by having the subject take a deep breath before breathing into the bag. The first 150 ml of the collected breath is discarded because it may be contaminated (DAmico et al. 2007). This method would be applied to a subject with, for example diabetes. Dead-space air is collected with a pump that draws the breath from the subject's mouth into a sampling bag. The pump is shown as component (B) of the apparatus in Fig. 3.3. This method would be applied with subjects with, for example, airway inflammation and renal disease. There is no need for the subject to exhale in this process.

3.3.2 Signal Sampling

The second phase is signal sampling, which involves acquiring dynamic responses to the interactions between a breath sample and the sensing elements, chemical sensors which form a sensor array in the signal measurement module in the hardware framework. These sensors sense gas particles and generate measurable electronic signals. The signals are then filtered, amplified, and digitized, and then sent to the computer for feature extraction, pattern analysis and classification.

3.3.2.1 Chemical Sensors

The function and the performance of our system highly depend on the capabilities of the chemical sensors. In our system, each sensor in the array has a unique "odorprint" corresponding to the compounds listed in Tables 3.1 and 3.2. Most of the compounds are VOCs but some are inorganic compounds such as ammonia, nitric oxide, carbon dioxide, and hydrogen. Table 3.4 summarizes the main disease biomarkers and compositions in human breath and the type of sensor required. Table 3.5 lists the types

Table 3.4 Compounds detected and sensors required

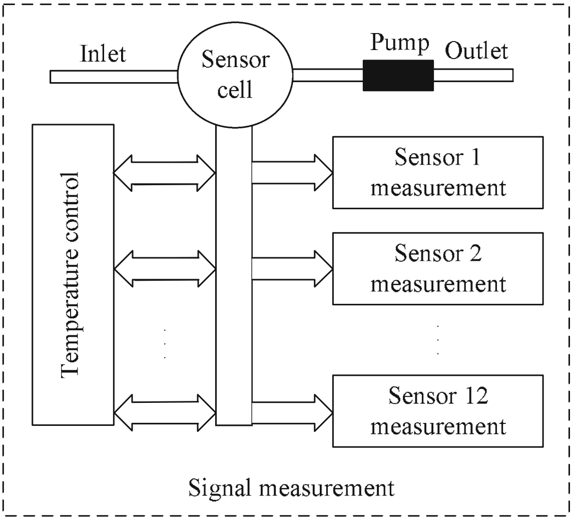
Main compounds in human breath	Requisite sensors
Acetone, isoprene, pentane, benzene etc.	VOC sensor
Ammonia	NH_3 sensor
Nitric oxide	NO sensor
Carbonyl sulfide, carbon disulfide	Sulfide sensor
Hydrogen	H_2 sensor
Carbon monoxide, carbon dioxide	CO and CO_2 sensor

Table 3.5 Type of sensors and corresponding sensitive gas

No.	Sensors	Gases	Sensitivities (ppm)
1	TGS2600	H_2, CO and VOCs	1–30
2	TGS2602	VOCs	1–30
3	TGS2611-C00	VOCs	500–10000
4	TGS2610-C00	VOCs	500–10000
5	TGS2610-D00	VOCs	500–10000
6	TGS2620	VOCs and CO	50–5000
7	TGS825	H_2S	5–100
8	TGS4161	CO_2	350–10000
9	TGS826	NH_3	30–300
10	TGS2201	NO and NO_2	0.1–10
11	TGS822	VOCs	50–5000
12	TGS821	H_2	10–1000

of sensors used in our system, the gases they are sensitive to and at what sensitivity. These sensors used in our work are metal oxide semiconductor gas sensors from FIGARO Engineering Inc. This kind of sensors is very sensitive, robust, and resistant to humidity and ageing (Turner and Magan 2004). Seven of the sensors are each sensitive to VOCs. One sensor detects only carbon dioxide. One sensor is sensitive to ammonia, which is associated with renal disease. One sensor is sensitive to nitric oxide which is associated with bronchiectasis, airway inflammation, and COPD. One sensor is sensitive to sulfides, what are associated with liver diseases. Finally, one sensor is sensitive to hydrogen. These sensors are able to sensitive to most of biomakers and compositions in human breath, therefore they have better responses than those commercial e-noses.

Fig. 3.4 Basic structure of sensor array used in our system



3.3.2.2 Signal Measurement

The framework of the system consists of three modules: signal measurement, signal conditioning, and signal acquisition. The signal measurement module contains a sensor array, temperature control circuit, and measurement circuit (Fig. 3.4). The temperature control circuit provides negative feedback voltage to the heater of sensors so as to guarantee that the sensors are at stable temperature. The measurement circuit is responsible for transforming odor signals into electronic signals.

The sensor array is composed of 12 sensors set in a 600 ml stainless steel chamber. Breath samples from subjects are collected with a 600 ml Tedlar gas sampling bag and then injected into the chamber through an auto-sampler at 120 ml/s. Since the capacity of the sampling bag is 600 ml, the total injection time for one sample is $t = 600/120 = 5s$. The resistances of the sensors change from R_0 to R_s when they are exposed to sampled gas. The output voltage is

$$V_{Out} = \frac{1}{2} V_{CC}(1 - \frac{R_s}{R_0}), \tag{3.1}$$

where V_{CC} is the transient voltage crossing the sensor and V_{Out} is the transient output voltages of the measurement circuits.

The signal measurement module measures these voltages and converts them into analog electrical signals. The analog signals are subsequently conditioned by signal filtering and amplifying. Finally, the signals are sampled at a 9 Hz sampling frequency and transmitted through a USB interface to a computer. This component is controlled by a 16-bit microprocessor. After data collection, a pump works at a rate

Table 3.6 Fundamental performance parameters of the proposed system

System parameters	Specifications
Working temperature	25 ± 10 °C
Carrier flow	10 ml/s
Chamber volume	600 ml
Sampling injection rate	120 ml/s
Sampling frequency	9 Hz
Sampling time	100 s

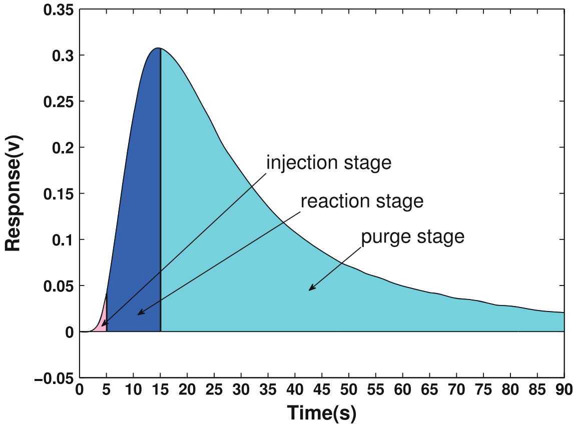


Fig. 3.5 A typical sensor response curve which undergoes three stages

of 10 ml/s to purge the chamber. Table 3.6 summarizes the fundamental performance parameters of the proposed system.

3.3.2.3 Sampling Procedure

The sampling procedure, program controlled by the system to ensure all samples are sampled under the same criterion, involves two sub-procedures, a purge cycle, and a sampling cycle. In the purge cycle, a pump pulls and purges the air over the sensor array, supplying background air to the array for the baseline measurement as well as refreshing it after sampling. In the sample cycle, the analyte is injected into the chamber. When the sensor array is exposed to the analyte, changes in resistances are measured and recorded. The action of the system is different in each time-slice (Fig. 3.5). The following explains this in detail.

1. −10–0 s (baseline stage): The chamber is purged and the sensor returns to a steady state. The baseline value is measured and recorded for data manipulation and normalization;

2. 1–5 s (injection stage): Sampled gas is injected into the chamber at an invariable rate. Particles of sampled gas inside the chamber accrue during injection, producing a changing of resistance of the sensor, and causing the amplitude of the signal to rise;
3. 6–10 s (reaction stage): Particles in the chamber continue to accumulate on the sensors but the accumulation rate is decreasing. The resistance of the sensor monotonically increases at a decreasing rate, as does the amplitude of the signals;
4. 15–90 s (purge stage): The chamber is purged again. The pump quickly draws out the remaining analyte, thereby shortening the sampling time as well as refreshing the air for the next use.

In our database, the characteristic curve of one sample is taken from the data for the period from 1 s to 90 s. Since the sampling frequency is 9 Hz, one sensor in one sample creates a $90 \times 9 = 810$-dimension feature vector.

3.3.3 Data Analysis

The system measures changes in voltage across each sensor and converts the raw signal into a digital value that can be applied to future analysis. This analysis involves three steps: signal preprocessing, feature extraction, and classification.

3.3.3.1 Signal Preprocessing

The purpose of signal preprocessing is to compensate for drift and eliminate irrelevant information so to improve the performance of the subsequent pattern recognition and classification. It involves baseline manipulation and normalization. Baseline manipulation is implemented for drift compensation, contrast enhancement, and scaling. Its basic idea is to subtract the baseline of each sensor from the sensor response. We assume that one data set has e samples, where $e = 1, \ldots, N_e$. Each sample consists of s sensor transients, where $s = 1, \ldots, N_s$. There are k dimensions per transient, where $k = 1, \ldots, N_k$. The dynamic response of one sample at time t_k is denoted as $R_{e,s}(t_k)$. There are b dimensions in baseline stage, where $b = 1, \ldots, N_b$. The baseline response of this sample is $B_{e,s}(t_b)$. The relative change for a particular sensor is defined as the preprocessed response

$$R^{B}{}_{e,s}(t_k) = R_{e,s}(t_k) - \frac{1}{N_b}\sum_{t_b=1}^{N_b} B_{e,s}(t_b), \quad \forall e, s, k, b. \tag{3.2}$$

Normalization is used to compensate for sample-to-sample variations caused by analyte concentration and pressure of oxygen (PO$_2$). $R^{B}{}_{e,s}(t_k)$ is the response of the sensor N_s to the N_e sample in data set, which has been processed by baseline manipulation. The normalized response is defined as

$$R^{BN}{}_{e,s}(t_k) = \frac{R^{B}{}_{e,s}(t_k)}{max(R^{B}{}_{e,s}(t_k))}, \quad \forall e, m. \tag{3.3}$$

3.3.3.2 Feature Extraction

The purpose of feature extraction is to find a low-dimensional mapping $f : x \in R^N \mapsto y \in R^M (M < N)$ that preserves most of information in the original feature vector x. In this chapter, we employed principal components analysis (PCA) to extract characteristic features of samples from m classes. We calculated the eigenvectors and eigenvalues of the training set and sort the eigenvectors, i.e., principal components of PCA, by descendant eigenvalues, then projected both test data and training data onto the PCA subspace spanned by selected principal components. The criteria for principal component selection is

$$r_\lambda = \sum_{k=1}^{s} \lambda_i \Big/ \sum_{k=1}^{n} \lambda_i, \tag{3.4}$$

where r_i is the eigenvalue, s is the number of selected principal components, and n is the total number of eigenvalues. Assume $r_\lambda > 99\%$ counts for enough variability in the dataset, $s = 10$ eigenvectors as features. We hence formed a s-dimensional training vector space and test vector space respectively for data classification.

3.3.3.3 Classification

K-nearest neighbor voting rule (KNN) was used as a classifier for the features that extracted by PCA. Basically, it classifies an unlabeled test sample by finding the K nearest neighbors in the training set using Euclidean distance and assigning the label of that class represented by a majority among the K neighbors (Gutierrez-Osuna 2002). There are many voting rule to decide which class the unlabeled sample belongs to. In our experiment, we used the following vote rule: assuming there are m classes and one sample has $K_1, K_2, \ldots, K_m$ -nearest neighbors for the m classes, where $\sum_{i=1}^{m} K_i = K$, the classification result is given by

$$c = \underset{i=1,\ldots,m}{\arg\max}\{\frac{K_i}{K}\}, \tag{3.5}$$

where c is the label of the predicted class. The training vectors were classified in advance into m classes, labeled as either healthy or diseased. The test vector was then predicted using Eq. 3.5.

3.4 Experiments

In the first experiment, we used our system to distinguish between pre- and post-treatment breath samples from 52 subjects with end-stage renal failure, a kind of condition associates with the accumulation of urinary waste products in the blood because the kidneys are not working effectively (Table 3.7). A standard treatment for the condition is hemodialysis to help patient remove more urea and creatinine from the blood. There is a reduction in the ammonia concentration in expired breath of patients as hemodialysis proceeds (Narasimhan et al. 2001). The results for these experiments are given in Sect. 3.4.1.

In the second set of experiments, we tested the ability of the system to distinguish between subjects assumed to be healthy on the basis of recent health check and subjects known to be afflicted with either diabetes, renal disease, or airway inflammation. Totally, We collected 108 healthy samples, 117 diabetes samples, 110 renal disease samples, and 110 airway inflammation samples using the gas collection and signal sampling procedures described in Sects. 3.3.1 and 3.3.2. Table 3.8 details the composition of the subject database. All patients were inpatient volunteers from the Harbin Hospital. Their conditions were confirmed and correlated by comparing their levels with standard clinical blood markers for the relevant diseases and conditions. In each case, these diseases and conditions are associated with characteristic molecules in the breath. Diabetes arises when the glucose produced by the body cannot enter cells and so cells have to use fat as an energy source. One of the by-products of metabolizing fat for energy is ketones. When ketones accumulate in the blood, there is ketoacidosis, which is characterized by the smell of acetone on the patient's breath (Holt et al. 2006). Renal disease arise from the inability of the kidneys to effectively filter the blood. This results in an accumulation of nitrogen-bearing waste products (urea), which accounts for the odor of ammonia on the breath of patients (Greenberg and Cheung 2005). As for airway inflammation, it has been shown that exhaled nitric oxide levels are higher when there is airway inflammation, especially asthmatic airway inflammation (Ashutosh 2000).

Table 3.7 Composition of the renal failure database

Type of subjects	Number	Male/Female	Age
Subjects with renal failure	52	33/19	34–70

Table 3.8 Composition of the subject database

Type of subjects	Number	Male/Female	Age
Healthy subjects	108	58/50	23–60
Subjects with diabetes	117	65/52	32–70
Subjects with renal disease	110	63/47	28–70
Subjects with airway inflammation	110	54/56	16–62

3.4.1 Evaluating Outcomes of Hemodialysis

Figure 3.6 shows the responses of the twelve different sensors (S1–S12) to the samples of renal failure patients over the 90 s sampling period. Figure 3.6a shows a typical response of one patient before hemodialysis and Fig. 3.6b shows a response of the same patient after hemodialysis. The curves represent the output of each sensor, S1–S12. These curves have been preprocessed by baseline manipulation and normalization according to Eqs. 3.2 and 3.3. As shown in Table 3.2, the dominant compound marking renal disease is ammonia. From these figures, it is very clear that before hemodialysis (Fig. 3.6a), the amplitude of the ninth sensor is very high, which indicates that the concentration of ammonia in the breath is quite large. However, after treatment (Fig. 3.6b), the amplitude of the ninth sensor clearly decreases, indicating the concentration of ammonia in the subject's breath has fallen.

Figure 3.7 presents the mean responses of the twelve sensors showing the response of each sensor to two kinds of samples. The error bar represents the standard deviation, showing the difference between the responses of all samples in one classes and their mean. The mean response is defined as

$$MeanR_{e,s} = \frac{1}{N_k} \sum_{t_k=1}^{N_k} R_{e,s}(t_k), \qquad \forall e, s, k. \tag{3.6}$$

The definitions of these variables are the same as Sect. 3.3.3.1. For each sensor, the left bar presents the class before hemodialysis and the right bar is the class after hemodialysis. After the treatment, the values of several responses clearly fall, especially the ninth sensor, which is sensitive to ammonia.

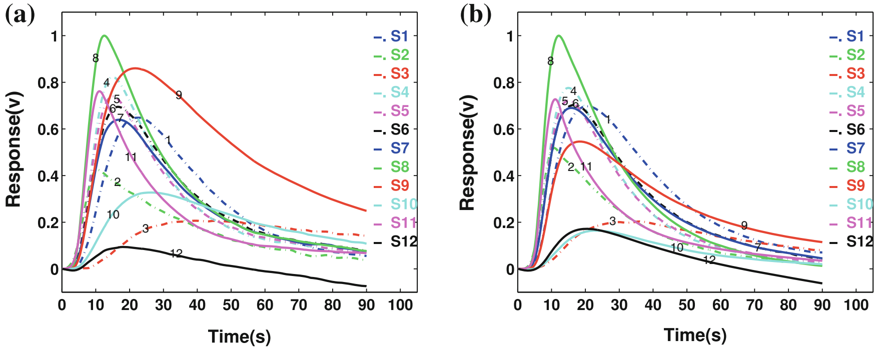


Fig. 3.6 Typical responses from the same patient: **a** before treatment, **b** after treatment. The horizontal axis stands for the sampling time (0–90 s) and the vertical axis shows the amplitude of the sensor output in volts. The number in each curve is the index of the sensor

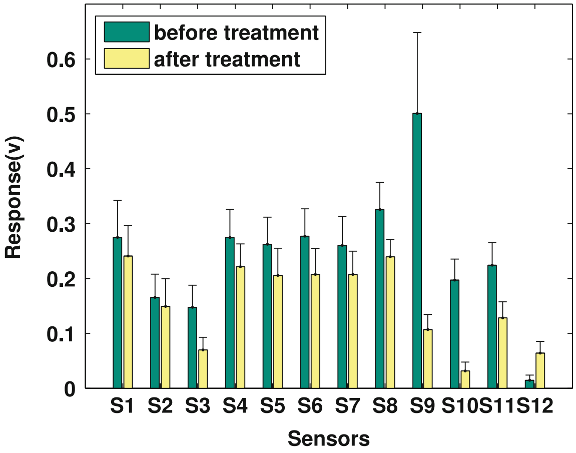


Fig. 3.7 Mean response of twelve sensors to two cases: before treatment and after treatment. The error bar represents the standard deviation (only the upper bar is drawn). The horizontal axis denotes the twelve sensors and the vertical axis is the mean value of each normalized response

3.4.2 Distinguishing Between Subject Breath Samples

Figure 3.8 shows the responses of twelve different sensors (S1–S12) to the four different air samples over the 90 s sampling period. Figure 3.8a is a typical response to a healthy sample. Figure 3.8b is to a diabetes sample. Figure 3.8c is to a renal disease sample. And Fig. 3.8d is to an airway inflammation sample. The curves represent the output of each sensor.

Figure 3.9 gives the mean responses of the twelve sensors in the four types of air samples as defined by Eq. 3.6. The definition of error bar is the same as that in Sect. 3.4.1. In each of the four categories it is possible to find the combinations of sensors which could unambiguously identify each of the four conditions. Thus, the strongest responses to healthy samples came from the sixth, seventh, and eighth sensors while the strongest response to diabetes came from the second, fourth, fifth, and twelfth sensors. It is worth mentioning that the twelfth sensor gave a very significant response, though it is not used for VOCs detection. In China, the special diet recommended for diabetics features large amounts of fermentable dietary fiber, which leads to colonic fermentation of indigestible carbohydrates (Brighenti et al. 2006). One product of colonic fermentation is hydrogen (Le Marchand et al. 2006), which is absorbed into the bloodstream and excreted through the breath. Therefore, the breath air of diabetics we have collected would include hydrogen. The strong response to the renal disease samples came from the first, third, ninth, and eleventh

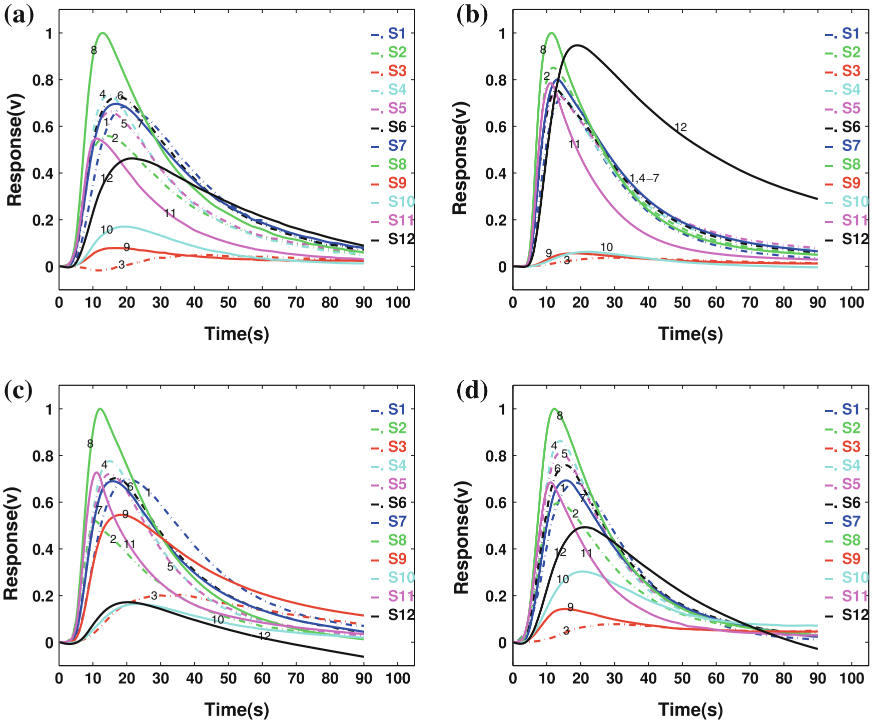
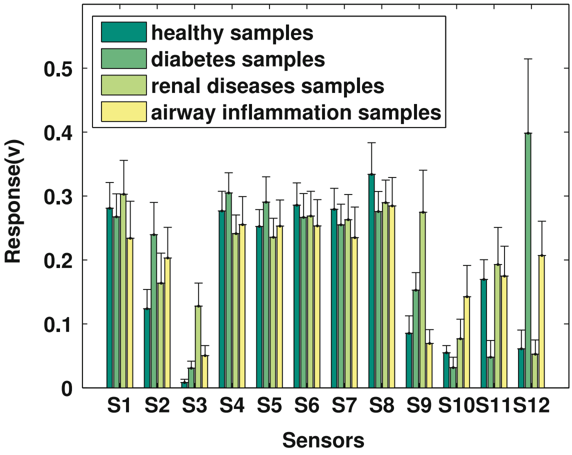


Fig. 3.8 Typical responses from four subject categories: **a** healthy subjects, **b** subjects with diabetes, **c** subjects with renal disease, and **d** subjects with airway inflammation. The horizontal axis stands for the sampling time (0–90 s) and the vertical axis denotes the amplitude of sensor output in volts. The number in each curve is the index of the sensor

Fig. 3.9 Mean response of twelve sensors to four classes: healthy, diabetes, renal disease, and airway inflammation. The error bar represents the standard deviation (only the upper bar is drawn). The horizontal axis denotes the twelve sensors and the vertical axis is the mean value of each normalized response



sensors, especially the ninth sensor, which is particularly sensitive to ammonia. The largest response to airway inflammation came from the tenth sensor, which is used to detect nitric oxide.

3.5 Results and Discussion

The outcomes of both sets of experiments were evaluated using PCA coupled with KNN, as introduced in Sects. 3.3.3.2 and 3.3.3.3.

3.5.1 Results Evaluating Outcomes of Hemodialysis

Figure 3.10 shows a two-dimensional PCA analysis of the responses with the first principal component (PC1) plotted against the second (PC2). $*$ stands for the samples before treatment and $+$ for the samples after treatment. The two dimensions explained 73.01% of the variation in the data, 53.4% for PC1 and 19.61% for PC2. The two classes are discriminative even though some samples overlap.

To measure the classification accuracy of system, we randomly selected a training set of 26 samples from each disease class of 52 samples. The remainder was used as the test set. PCA was used to extract characteristic features of the samples. We calculated the eigenvectors and eigenvalues of the training set and sorted the eigenvectors by descendant eigenvalues. We then used Eq. 3.4 and the condition $r_\lambda >$ 99% to select the first 12 eigenvectors as principle components. Next, we projected all samples onto the PCA subspace spanned by principal components. Then, KNN (K = 5) classifier defined by Eq. 3.5 predicted the class that a test sample belonged

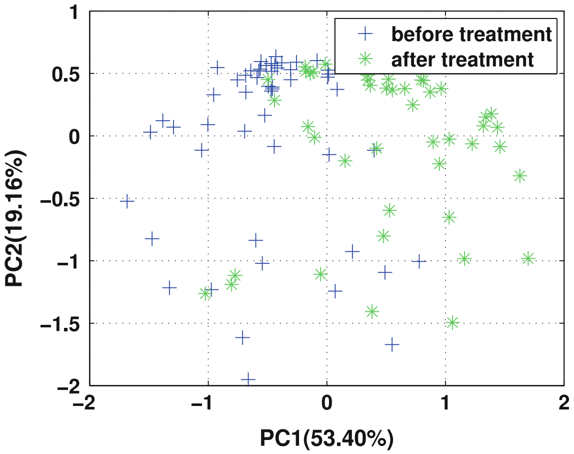


Fig. 3.10 PCA two-dimensional plot of the sensor signals corresponding to two classes: **a** renal failure samples before treatment ($*$), **b** renal failure samples after treatment ($+$)

to. We ran this procedure 50 times and computed the average classification rate over all 50 runs.

Table 3.9 shows the classification results. In the 26-sample pre-treatment test set, an average of 20.84 samples were classified correctly and 5.16 samples were classified incorrectly, an overall accuracy of 80.15%. In the 26-sample post-treatment test set, an average of 21.32 samples were classified correctly and 4.68 samples were classified incorrectly, an overall accuracy of 82%. Clearly, the proposed system would have some value in evaluating the efficacy of hemiodialysis, and may take the place in some cases of blood tests, given that it is simple, low-cost, and non-invasive.

3.5.2 Results Distinguishing Between Subject Breath Samples

The classifications of the four types of breath samples were evaluated with PCA coupled with KNN and the results were measured by sensitivity and specificity. The samples from patients with diabetes, renal disease, and airway inflammation and the healthy samples were formed three groups. One group contained healthy subjects and subjects with diabetes, the second contained healthy subjects and subjects with renal disease, and the third group contained healthy subjects and subjects with airway inflammation.

Figure 3.11 shows the PCA two-dimensional plot of the responses from the two classes with the first principal component (PC1) plotted against the second (PC2). The $*$ stands for the samples classified as being from patients and $+$ for healthy subjects. In the PCA plot of diabetes samples and healthy samples, the two dimensions explained 79.96% of the variation in the data, 65.29% for PC1 and 14.67% for PC2. In the PCA plot of renal disease samples and healthy samples, the two dimensions explained 72.45% of the variation in the data, 55.56% for PC1 and 16.89% for PC2. In the PCA plot of airway inflammation samples and healthy samples, the two dimensions explained 77.4% of the variation in the data, 52.49% for PC1 and 24.91% for PC2.

Table 3.9 Classification results of two classes: renal failure samples before treatment and after treatment

Actual group member	Number of samples		Predicted group member		Accuracy (%)
	Training set	Test set	Before treatment	After treatment	
Before treatment	26	26	20.84	5.16	80.15
After treatment	26	26	21.32	4.68	82

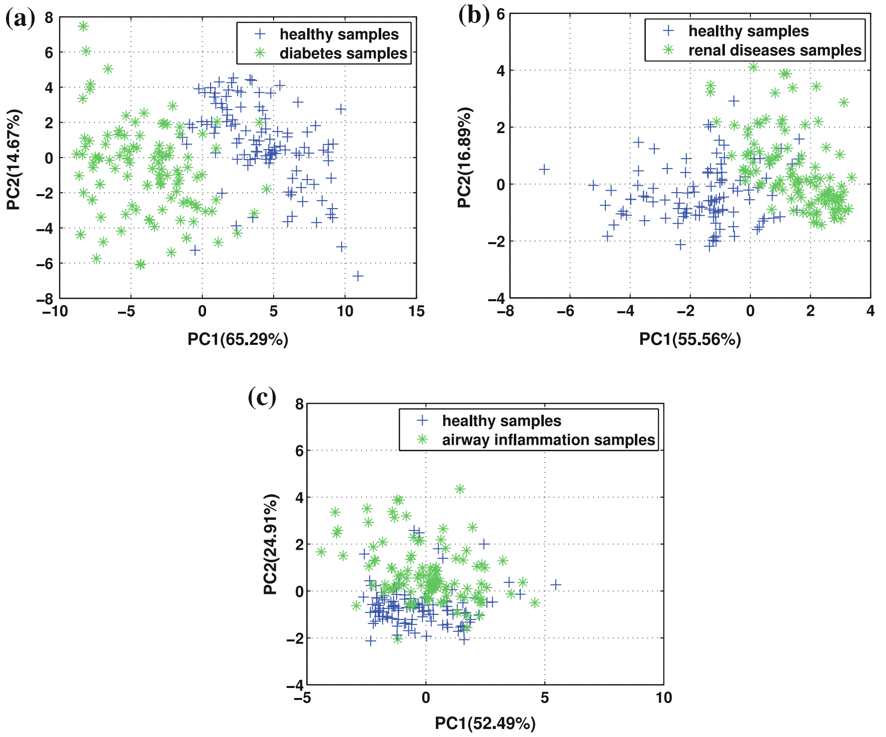


Fig. 3.11 PCA two-dimensional plot of the sensor signals corresponding to two classes: **a** healthy samples (+) and diabetes samples (∗), **b** healthy samples (+) and renal disease samples (∗), and **c** healthy samples (+) and airway inflammation samples (∗)

To compare the test results, we randomly selected 60 samples from each class as the test set and the remainder formed the training set. PCA was used to extract characteristic features of samples. Equation 3.4 and the condition $r_\lambda > 99\%$ were used to select the first 10 eigenvectors in all classes in every group. The KNN (K = 5) classifier as defined by Eq. 3.5 was then used to determine which class each test sample belonged to.

In medicine, the reliability of a diagnosis is measured in terms of sensitivity and specificity, with the outcome being either positive (unhealthy) or negative (healthy). In the classification, the number of genuine sick subjects is denoted *tp*; misidentified healthy subjects is *fp*; genuine healthy subjects is *tn*; the misdiagnosed sick subjects is denoted as *tn* (Blatt et al. 2007). Sensitivity and specificity are thus defined as in Table 3.10. Table 3.11 shows the classification results of all groups.

In the diabetes experiment, out of 60 samples in the test set, the system correctly diagnosed an average of 52.6 samples as diabetes and incorrectly diagnosed 7.4 samples as healthy. In the 60 healthy samples in the test set, an average of 52.12 samples

Table 3.10 The definition of sensitivity and specificity

		Test outcome		Sensitivity	Specificity
		Positive	Negative		
Actual condition	Positive	*tp*	*fp*	$\frac{tp}{tp+fn}$	$\frac{tn}{tn+fp}$
	Negative	*fn*	*tn*		

Table 3.11 The classification results defined by sensitivity and specificity

		Training/ Test sets	Test outcome		Sensitivity	Specificity
			Positive	Negative	(%)	(%)
Diabetes	Positive	57/60	52.6	7.4	86.97	87.57
	Negative	48/60	7.88	52.12		
Renal failure	Positive	50/60	51.94	8.06	83.96	86.14
	Negative	48/60	9.92	50.08		
Airway inflammation	Positive	50/60	42.12	17.88	73.79	71.58
	Negative	48/60	14.96	45.04		

were correctly diagnosed as healthy and 7.88 were incorrectly diagnosed as diabetes. The sensitivity of this diagnosis was thus 86.97% and the specificity was 87.57%.

In the renal disease experiment, an average of 51.94 disease samples were correctly diagnosed as renal disease and 8.06 disease samples were incorrectly diagnosed as healthy. In the 60 healthy samples in the test set, an average of 50.08 healthy samples were correctly diagnosed as healthy; while an average of 9.92 healthy samples were incorrectly diagnosed as renal disease. Consequently, the sensitivity and specificity of this diagnosis were 83.96% and 86.14% respectively.

Same as above, in the experiment of airway inflammation diagnosis, there were averagely 42.12 disease samples diagnosed correctly as airway inflammation and 17.88 disease samples diagnosed incorrectly as healthy, and there were averagely 14.96 healthy samples diagnosed incorrectly as airway inflammation and 45.04 healthy samples diagnosed correctly as healthy. The sensitivity of this diagnosis was thus 73.79% and the specificity was 71.58%.

3.6 Summary

This chapter proposed a breath analysis system that has a broad application in medicine, such as detecting diseases and monitoring the progress of related therapies. The system structure, working procedure, odor signal preprocessing, and pattern recognition method were introduced. To evaluate the system performance, breath samples were captured and two experiments were conducted on medical treatment evaluation and disease identification. The results showed that the system was not only able to distinguish between breath samples from subjects suffering from various diseases or conditions (diabetes, renal disease, and airway inflammation) and breath samples from healthy subjects, but in the case of renal failure was also helpful in evaluating the efficacy of hemodialysis.

Although the current pattern recognition method produced satisfactory results when we used integral data, it should still be possible to further improve the classification accuracy and speed by selecting proper features. Typically, the performance of an electronic olfaction system depends heavily on the features being provided to the odor classification algorithm. Therefore, in future work we will investigate how to select the most proper features for effective pattern classification. We also intend to extend the number of diseases/conditions that the system can analyze.

References

Amann A, Poupart G, Telser S, Ledochowski M, Schmid A, Mechtcheriakov S (2004) Applications of breath gas analysis in medicine. Int J Mass Spectrom 239(2–3):227–233

Ashutosh K (2000) Nitric oxide and asthma: a review. Curr Opin Pulm Med 6(1):21–25

Baraldi E, Carraro S (2006) Exhaled NO and breath condensate. Paediatr Respir Rev 7:20–22

Barker M, Hengst M, Schmid J, Buers H, Mittermaier B, Klemp D, Koppmann R (2006) Volatile organic compounds in the exhaled breath of young patients with cystic fibrosis. Eur Respir J 27(5):929–936

Blatt R, Bonarini A, Calabro E, Della Torre M, Matteucci M, Pastorino U (2007) Lung cancer identification by an electronic nose based on an array of MOS sensors. Int Jt Conf Neural Netw 2007:1423–1428

Brighenti F, Benini L, Del Rio D, Casiraghi C, Pellegrini N, Scazzina F, Jenkins D, Vantini I (2006) Colonic fermentation of indigestible carbohydrates contributes to the second-meal effect. Am J Clin Nutr 83(4):817–822

Cao W, Duan Y (2007) Current status of methods and techniques for breath analysis. Crit Rev Anal Chem 37(1):3–13

DAmico A, Di Natale C, Paolesse R, Macagnano A, Martinelli E, Pennazza G, Santonico M, Bernabei M, Roscioni C, Galluccio G, et al (2007) Olfactory systems for medical applications. Sens Actuators B Chem 130(1):458–465

Davies S, Spanel P, Smith D (1997) Quantitative analysis of ammonia on the breath of patients in end-stage renal failure. Kidney Int 52(1):223–228

Deng C, Zhang J, Yu X, Zhang W, Zhang X (2004) Determination of acetone in human breath by gas chromatography-mass spectrometry and solid-phase microextraction with on-fiber derivatization. J Chromatogr B 810(2):269–275

Deykin A, Massaro A, Drazen J, Israel E (2002) Exhaled nitric oxide as a diagnostic test for asthma: online versus offline techniques and effect of flow rate. Am J Respir Crit Care Med 165(12):1597–1601

Di Francesco F, Fuoco R, Trivella M, Ceccarini A (2005) Breath analysis: trends in techniques and clinical applications. Microchem J 79(1–2):405–410

Dragonieri S, Schot R, Mertens B, Le Cessie S, Gauw S, Spanevello A, Resta O, Willard N, Vink T, Rabe K et al (2007) An electronic nose in the discrimination of patients with asthma and controls. J Allergy Clin Immunol 120(4):856–862

Dweik R, Amann A (2008) Exhaled breath analysis: the new frontier in medical testing. J Breath Res 2(030):301

Greenberg A, Cheung A (2005) Primer on kidney diseases. WB Saunders Co

Gutierrez-Osuna R (2002) Pattern analysis for machine olfaction: a review. IEEE Sens J 2(3):189–202

Holt R, Hanley N, Brook C (2006) Essential endocrinology and diabetes

Horvath I, Loukides S, Wodehouse T, Kharitonov S, Cole P, Barnes P (1998) Increased levels of exhaled carbon monoxide in bronchiectasis: a new marker of oxidative stress. Brit Med J 53(10):867–870

Humad S, Zarling E, Clapper M, Skosey J (1988) Breath pentane excretion as a marker of disease activity in rheumatoid arthritis. Free Radic Res 5(2):101–106

Kharitonov S, Wells A, O'connor B, Cole P, Hansell D, Logan-Sinclair R, Barnes P, (1995) Elevated levels of exhaled nitric oxide in bronchiectasis. Am J Respir Crit Care Med 151(6):1889–1893

Le Marchand L, Wilkens L, Harwood P, Cooney R (2006) Breath hydrogen and methane in populations at different risk for colon cancer. Int J Cancer 55(6):887–890

Lin Y, Guo H, Chang Y, Kao M, Wang H, Hong R (2001) Application of the electronic nose for uremia diagnosis. Sens Actuators B Chem 76(1–3):177–180

Maziak W, Loukides S, Culpitt S, Sullivan P, Kharitonov S, Barnes P (1998) Exhaled nitric oxide in chronic obstructive pulmonary disease. Am J Respir Crit Care Med 157(3):998–1002

McGrath L, Patrick R, Mallon P, Dowey L, Silke B, Norwood W, Elborn S (2000) Breath isoprene during acute respiratory exacerbation in cystic fibrosis. Eur Respir J 16(6):1065–1069

Miekisch W, Schubert J (2006) From highly sophisticated analytical techniques to life-saving diagnostics: technical developments in breath analysis. Trends Anal Chem 25:665–673

Miekisch W, Schubert J, Noeldge-Schomburg G (2004) Diagnostic potential of breath analysis-focus on volatile organic compounds. Clin Chim Acta 347(1–2):25–39

Narasimhan L, Goodman W, Patel C (2001) Correlation of breath ammonia with blood urea nitrogen and creatinine during hemodialysis. Proc Nat Acad Sci 98(8):4617–4621

Olopade C, Zakkar M, Swedler W, Rubinstein I (1997) Exhaled pentane levels in acute asthma. Chest 111(4):862–865

Phillips M (1997) Method for the collection and assay of volatile organic compounds in breath. Anal Biochem 247(2):272–278

Phillips M, Sabas M, Greenberg J (1993) Increased pentane and carbon disulfide in the breath of patients with schizophrenia. J Clin Pathol 46(9):861–864

Phillips M, Herrera J, Krishnan S, Zain M, Greenberg J, Cataneo R (1999) Variation in volatile organic compounds in the breath of normal humans. J Chromatogr B Biomed Sci Appl 729(1–2):75–88

Phillips M, Cataneo R, Cummin A, Gagliardi A, Gleeson K, Greenberg J, Maxfield R, Rom W (2003a) Detection of lung cancer with volatile markers in the breath. Chest 123(6):2115–2123

Phillips M, Cataneo R, Ditkoff B, Fisher P, Greenberg J, Gunawardena R, Kwon C, Rahbari-Oskoui F, Wong C (2003b) Volatile markers of breast cancer in the breath. Breast J 9(3):184–191

Phillips M, Boehmer J, Cataneo R, Cheema T, Eisen H, Fallon J, Fisher P, Gass A, Greenberg J, Kobashigawa J et al (2004) Heart allograft rejection: detection with breath alkanes in low levels (the HARDBALL study). J Heart Lung Transplant 23(6):701–708

Phillips M, Altorki N, Austin J, Cameron R, Cataneo R, Greenberg J, Kloss R, Maxfield R, Munawar M, Pass H et al (2007a) Prediction of lung cancer using volatile biomarkers in breath. Cancer Biomark 3(2):95–109

Phillips M, Cataneo R, Condos R, Ring Erickson G, Greenberg J, La Bombardi V, Munawar M, Tietje O (2007b) Volatile biomarkers of pulmonary tuberculosis in the breath. Tuberculosis 87(1):44–52

Risby T, Solga S (2006) Current status of clinical breath analysis. Appl Phys B Lasers Opt 85(2):421–426

Rock F, Barsan N, Weimar U (2008) Electronic nose: current status and future trends. Chem Rev 108(2):705–725

Salazar M (2003) Breath markers of oxidative stress in patients with unstable angina. Heart Dis 5(2):95–99

Schubert J, Miekisch W, Geiger K, Nöldge-Schomburg G (2004) Breath analysis in critically ill patients: potential and limitations. Expert Rev Mol Diagn 4(5):619–629

Sedghi S, Keshavarzian A, Klamut M, Eiznhamer D, Zarling E (1994) Elevated breath ethane levels in active ulcerative colitis: evidence for excessive lipid peroxidation. Am J Gastroenterol 89(12):2217–2221

Sehnert S, Jiang L, Burdick J, Risby T (2002) Breath biomarkers for detection of human liver diseases: preliminary study. Biomarkers 7(2):174–187

Thaler E, Hanson C (2005) Medical applications of electronic nose technology. Expert Rev Med Devices 2(5):559–566

Turner A, Magan N (2004) Electronic noses and disease diagnostics. Nat Rev Microbiol 2(2):161–166

Van Berkel J, Dallinga J, Möller G, Godschalk R, Moonen E, Wouters E, Van Schooten F (2008) Development of accurate classification method based on the analysis of volatile organic compounds from human exhaled air. J Chromatogr B 861(1):101–107

Weitz Z, Birnbaum A, Sobotka P, Zarling E, Skosey J (1991) High breath pentane concentrations during acute myocardial infarction. Lancet 337(8747):933–935

Yu J, Byun H, So M, Huh J (2005) Analysis of diabetic patient’s breath with conducting polymer sensor array. Sens Actuators B Chem 108(1–2):305–308

Chapter 4
An LDA-Based Sensor Selection Approach

Abstract In the application of breath analysis, the redundant specificities of a sensor array often exceed the needs of the discrimination application. When recognizing types of diseases, not all sensors are required. However, it is difficult to decide which sensor is more useful for an unknown sample because some sensors are cross-sensitive to the biomarkers of the diseases. In this chapter, we propose a linear discriminant analysis (LDA)-based sensor selection technique (LDASS) which chooses an optimal configuration of sensors for a particular application from a whole set of available sensors. The proposed method finds the direction $\boldsymbol{w}$ via the LDA such that when data are projected onto this direction, the samples from two classes are as separate as possible. It is found that after projection, the difference of means of the two distinct sample classes can be expressed as the linear combination of the responses of all the sensors in the system, and $\boldsymbol{w}$ can be regarded as the weight vectors for these sensors which indicate the contribution weight of each sensor. Accordingly, it is possible to determine which sensor has a greater contribution in classifying the two classes. A series of experiments on different databases show that the proposed method outperforms other sensor selection techniques, such as the sequential forward selection (SFS) and genetic algorithm (GA) in recognition accuracy and processing time. This technique is not only applicable for breath analysis, but also useful in the general applications of e-noses.

Keywords Breath analysis · Disease identification · Sparse representation · Diabetes · Blood glucose levels

4.1 Introduction

Endogenous molecules in human breath, such as acetone, nitric oxide, hydrogen, and ammonia, are produced by metabolic processes and partitioned from blood via the alveolar pulmonary membrane and enter into the alveolar air (DAmico et al. 2007; Schubert et al. 2004; Miekisch et al. 2004). Changes in the concentration of these molecules could suggest various diseases or at least changes in metabolism (Amann et al. 2005). These molecules are therefore considered as biomarkers of the presence of diseases and clinical conditions. In comparison with other traditional methods such as blood and urine tests, breath analysis are noninvasive, real-time,

D. Zhang et al., *Breath Analysis for Medical Applications*,
DOI 10.1007/978-981-10-4322-2_4

and the least harmless to not only the subjects, but also the personnel who collect the samples (Van Berkel et al. 2008). Currently, there are increasing concerns about the applications of breath analysis in medicine and clinical pathology both as a diagnostic tool and as a way to monitor the progress of therapies (Di Francesco et al. 2005; Dweik and Amann 2008; Guo et al. 2010; D'Amico et al. 2010; Di Natale et al. 2003; Dragonieri et al. 2007, 2009; Yu et al. 2005; Shih et al. 2010).

In our previous work, we introduced a novel and portable system that is specific for breath analysis (Guo et al. 2010). In contrast to the broad panel of nonspecific sensors used in commercial e-noses, the sensors of our system are selected particularly to be sensitive to biomarkers and compositions in human breath. The system in Ref. Guo et al. (2010) has a sensor array, which is composed of twelve chemical sensors. Sensors 1–6, and 11 respond positively to volatile organic compounds (VOCs) with various sensitivities. They are used to detect pulmonary disease, diabetes, breast cancer, etc. (Deng et al. 2004; Phillips et al. 2003a, b, 2007). Sensor 7 is sensitive to sulfides, which are associated with liver diseases (Sehnert et al. 2002). Sensor 8 only detects carbon dioxide. Sensor 9 is used to ammonia, which is associated with renal diseases (Davies et al. 1997). Sensor 10 is sensitive to nitric oxide, which is associated with bronchiectasis, airway inflammation, and chronic obstructive pulmonary diseases (COPD) (Baraldi and Carraro 2006; Kharitonov et al. 1995; Horvath et al. 1998; Maziak et al. 1998). Finally, Sensor 12 is sensitive to hydrogen, which is used to detect gastrointestinal diseases (Brighenti et al. 2006; Le Marchand et al. 2006). A fundamental design concept for an array of sensors used in the system is that each sensor should have a different sensitivity profile over a range of compounds expected in the target application, e.g., the detection of an unknown disease. Therefore, the sensor array provides distinct response patterns to different analytes.

This kind of design offers system broad applications, but is problematic in practice. Since each sensor has a specific contribution in identifying a type of disease, not all sensors in the system are needed when we only want to detect one type of disease. For sensors that are not sensitive to the biomarkers of a given disease, they may only generate slightly different responses. These sensors would provide redundant information, which is not helpful, or even interfere with the identification results. Hence, it requires a method to select proper sensors. As shown in Fig. 4.1, breath gas collection is detected by a sensor array which converts the breath gas into electronic signals. After signal preprocessing, such as baseline manipulation and normalization, the signals become standard sample data. Then, the sensors involved in the standard samples are selected, and features in these sensors are extracted for classification. The classification task guides the execution of the sensor selection process.

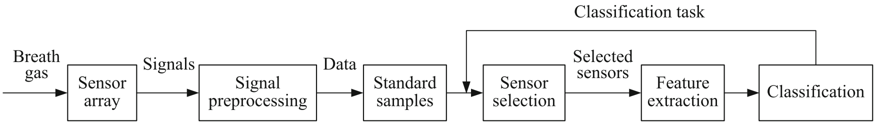


Fig. 4.1 The working flow of breath analysis system

There are many techniques used in sensor selection (Gardner et al. 2005; Phaisangittisagula et al. 2010; Zhang et al. 2009; Gualdron et al. 2007; Kermani et al. 1998; Pardo et al. 2001). Reference Pardo et al. (2001) made a good comparison between some common algorithms, such as sequential forward selection (SFS), sequential back selection (SBS), genetic algorithm (GA), and projection methods (PCA and LDA).

Even though the current techniques provide better recognition results than the methods that employ a full sensor set, they have several drawbacks. For example, SFS and SBS only explore a small fraction of the whole sensor set and can easily be trapped in local extrema while GA executes a global search in the whole set which incurs a heavy cost of computation. The most important is, all of these approaches fail to provide the weight information of the selected sensors, i.e., the significance of each sensor for a given recognition task. For example, in our system, Sensor 9 is specifically sensitive to ammonia, which is associated with renal diseases. Although other sensors, such as Sensors 3 and 12, also generate distinguishable responses to renal disease samples, they take on less important roles than Sensor 9 when recognizing renal diseases (Guo et al. 2010). In this case, the weight of Sensor 9 will be greater than the other sensors. However, the SFS, SBS, GA, and the projection methods provided by Refs. Kermani et al. (1998) and Pardo et al. (2001) cannot directly provide such weights for a given classification task.

Inspired by Refs. Kermani et al. (1998) and Pardo et al. (2001), which introduced the idea about the projection methods used in sensor selection, we propose a sensor selection approach in this chapter which is based on the LDA technique to choose an optimal configuration of sensors from a whole sensor set for given disease identification task. The approach finds the optimum projection direction $\boldsymbol{w}$ via the LDA such that when projecting the data in this direction, the samples from two classes are as separate as possible. We have found that after projection, the difference between the means of samples from the two classes is the linear combination of all sensors involved in the system, and $\boldsymbol{w}$ is a vector that can be divided into m sub-vectors. Each sub-vector is the scaling vector for the m sensors, indicating the weight of each sensor when recognizing a disease. A higher sub-vector implies a greater contribution of the sensor for the disease recognition. Hence, we not only can select the optimal sensor subset, but fix their weights, i.e., sub-vectors of $\boldsymbol{w}$. The sensors are multiplied by the norm of the sub-vectors to form the optimal configuration of sensors, which will be used in recognizing diseases. A series of experiments on different databases show that the proposed approach outperforms other sensor selection techniques, such as the SFS and GA, in recognition accuracy and processing time.

In fact, the proposed approach is not only effective in our breath analysis system, but also useful in commercial e-noses with an array of chemical sensors. Commercial e-noses, for the sake of broad application, such as wine and gasoline recognition, have just as many sensors involved. However, not all sensors are sensitive to wine or gasoline and redundant sensors may bring noises rather than improve the performance of e-noses. Therefore, sensor selection is also required and our technique can be helpful by identifying an appropriate sensor subset in the commercial e-noses.

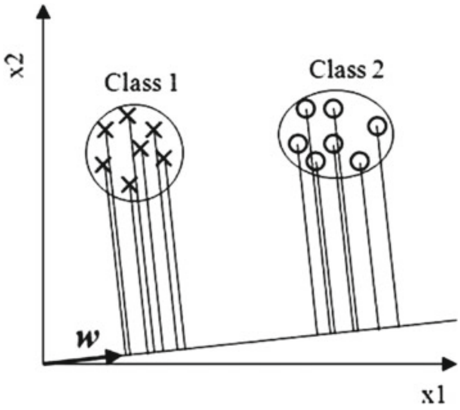


Fig. 4.2 Data from two classes that are projected onto *w*

The remainder of this chapter is organized as follows. Section 4.2 gives a detailed explanation about the proposed technique. Section 4.3 introduces the LDA-based approach (LDASS) in the application of breath analysis. Section 4.4 provides comparison experiments between our proposed approach and other two common methods, the SFS and GA. Section 4.5 offers our summaries.

4.2 LDA-Based Approach: Definition and Algorithm

This section details the LDA-based sensor selection approach (LDASS), which makes use of the LDA to model the difference of responses between two classes. The LDA can determine the optimum direction of projecting $\boldsymbol{w}$, such that when projecting onto $\boldsymbol{w}$, the samples from two different classes are as separate as possible and the samples from the same classes are as close as possible, which is shown by Fig. 4.2 (Alpaydin 2004). After projection, we deduced that the difference between the means of samples from two classes is the linear combination of all the sensors involved in the system with the entries of $\boldsymbol{w}$ as the coefficient of each sensor. Accordingly, in the linear combinations, the item with the largest value implies the corresponding sensor has the greatest contribution in classifying the two classes and is therefore selected for the classification. The sensor with the least contribution will be knocked out.

4.2.1 Data Expression

Assume that we have two types of disease samples. They are labeled as classes C_1 and C_2, respectively. Each class consists of n_i training samples, where $i = 1, 2$. Each sample comprises m sensor responses, i.e., $x_1, ..., x_m$, where $m = 12$ in our system. The response of each sensor is a discrete time signal with d dimensions, where

$d = 810$ according to our time sampling method. The response of the k-th sensor of the i-th sample from class C_1 is expressed as

$$\boldsymbol{s}_{i,k}^{c_1} = [t_1, t_2, \ldots, t_d]^{\mathrm{T}} \in \mathbf{R}^d. \tag{4.1}$$

Since each sample comprises m sensor responses,

$$\boldsymbol{x}_i^{c_1} = \begin{bmatrix} \boldsymbol{s}_{i,1}^{c_1} \\ \boldsymbol{s}_{i,2}^{c_1} \\ \vdots \\ \boldsymbol{s}_{i,m}^{c_1} \end{bmatrix} \tag{4.2}$$

is the i-th sample from class C_1, which is the l ($l = d \times m$)-dimensional column vector. So, the training set including n_1 samples from class C_1 is expressed as

$$\boldsymbol{X}^{c_1} = [\boldsymbol{x}_1^{c_1}, \boldsymbol{x}_2^{c_1}, \ldots, \boldsymbol{x}_{n_1}^{c_1}]. \tag{4.3}$$

Same as Eq. 4.3, the training set consisting of n_2 samples from class C_2 is expressed as

$$\boldsymbol{X}^{c_2} = [\boldsymbol{x}_1^{c_2}, \boldsymbol{x}_2^{c_2}, \ldots, \boldsymbol{x}_{n_2}^{c_2}]. \tag{4.4}$$

Hence, the training set from the two classes can be written as

$$\boldsymbol{X} = [\boldsymbol{X}^{c_1}, \boldsymbol{X}^{c_2}] \in R^{l \times n}, \tag{4.5}$$

where $n = n_1 + n_2$ is the total number of training samples.

4.2.2 Find Out the Optimum Direction by LDA

Given the samples introduced in Sect. 4.2.1,

$$y = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{X} \tag{4.6}$$

is the projection of $\boldsymbol{X}$ onto $\boldsymbol{w}$, where $\boldsymbol{w}$ is the l-dimensional column vector and $y = [y_1, \ldots, y_{n_1}, y_{n_1+1}, \ldots, y_n]$ is the n-dimensional row vector. Each entry of y represents a training sample.

The means of the samples with the two different diseases after projection are denoted by m_1 and m_2. The means of the samples with the two different diseases before projection are denoted by $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$, respectively. Thus, we have

$$m_1 = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{m}_1, \tag{4.7}$$

and

$$m_2 = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{m}_2. \tag{4.8}$$

After projection, we would like the samples with different diseases to be well separated, which implies that the means of the samples with different diseases is to be as far apart as possible and the samples with the same disease to be scattered in as small a region as possible, respectively. As we known, the LDA determines the $\boldsymbol{w}$ by maximizing

$$J(\boldsymbol{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}, \tag{4.9}$$

where s_1 and s_2 are the scatters of samples with two different diseases, respectively (Alpaydin 2004):

$$s_1^2 = \sum_{t=1}^{n_1} (y_t - m_1)^2, \tag{4.10}$$

and

$$s_2^2 = \sum_{t=n_1+1}^{n} (y_t - m_2)^2. \tag{4.11}$$

4.2.3 Difference Between Two Classes as the Linear Combination of Sensors

Since m is the number of sensors that is included in a sample, $\boldsymbol{w}$ is a vector that can be divided into m sub-vectors,

$$\boldsymbol{w} = \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \vdots \\ \boldsymbol{w}_m \end{bmatrix}. \tag{4.12}$$

Each sub-vector is a d-dimensional column vector, standing for the scaling vector for the m sensors and indicating the weight of each sensor when recognizing a disease.

From Eq. 4.6, the data before and after projection have the following relationship,

$$\begin{aligned} y &= \left[y_1, \ldots, y_{n_1}, y_{n_1+1}, \ldots, y_n\right] \\ &= \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \vdots \\ \boldsymbol{w}_m \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{s}_{1,1}^{c_1}, \ldots, \boldsymbol{s}_{n_1,1}^{c_1}, \boldsymbol{s}_{1,1}^{c_2}, \ldots, \boldsymbol{s}_{n_2,1}^{c_2} \\ \boldsymbol{s}_{1,2}^{c_1}, \ldots, \boldsymbol{s}_{n_1,2}^{c_1}, \boldsymbol{s}_{1,2}^{c_2}, \ldots, \boldsymbol{s}_{n_2,2}^{c_2} \\ \vdots \\ \boldsymbol{s}_{1,m}^{c_1}, \ldots, \boldsymbol{s}_{n_1,m}^{c_1}, \boldsymbol{s}_{1,m}^{c_2}, \ldots, \boldsymbol{s}_{n_2,m}^{c_2} \end{bmatrix}, \end{aligned} \tag{4.13}$$

where $y_1, \ldots, y_{n_1}, y_{n_1+1}, \ldots, y_{n_2}$ are the samples after projection, and the dimensions of the training set have been reduced from $l \times n$ to $1 \times n$.

By Eqs. 4.7 and 4.8, the mean of the training data from each class can be written as the linear combination of all sensors that are involved in the system,

$$m_1 = \frac{1}{n_1}\sum_{i=1}^{n_1} y_i = \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \vdots \\ \boldsymbol{w}_m \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \frac{1}{n_1}\sum\limits_{i=1}^{n_1} \boldsymbol{s}_{i,1}^{c_1} \\ \frac{1}{n_1}\sum\limits_{i=1}^{n_1} \boldsymbol{s}_{i,2}^{c_1} \\ \vdots \\ \frac{1}{n_1}\sum\limits_{i=1}^{n_1} \boldsymbol{s}_{i,m}^{c_1} \end{bmatrix} \tag{4.14}$$
$$= \boldsymbol{w}_1\bar{\boldsymbol{s}}_1^{c_1} + \boldsymbol{w}_2\bar{\boldsymbol{s}}_2^{c_1} +, \ldots, +\boldsymbol{w}_m\bar{\boldsymbol{s}}_m^{c_1},$$

and

$$m_2 = \frac{1}{n_2}\sum_{i=n_1+1}^{n} y_i = \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \vdots \\ \boldsymbol{w}_m \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \frac{1}{n_2}\sum\limits_{i=1}^{n_2} \boldsymbol{s}_{i,1}^{c_2} \\ \frac{1}{n_2}\sum\limits_{i=1}^{n_2} \boldsymbol{s}_{i,2}^{c_2} \\ \vdots \\ \frac{1}{n_2}\sum\limits_{i=1}^{n_2} \boldsymbol{s}_{i,m}^{c_2} \end{bmatrix} \tag{4.15}$$
$$= \boldsymbol{w}_1\bar{\boldsymbol{s}}_1^{c_2} + \boldsymbol{w}_2\bar{\boldsymbol{s}}_2^{c_2} +, \ldots, +\boldsymbol{w}_m\bar{\boldsymbol{s}}_m^{c_2}.$$

Finally, by subtracting Eq. 4.15 from Eq. 4.14, the difference between the means of the samples from C_1 and C_2 after projection can be expressed as the linear combination of all sensors involved in the sensor set and $\boldsymbol{w}$ could just be the coefficient of each sensor in the linear combination,

$$m_1 - m_2 = \sum_{k=1}^{m} \boldsymbol{w}_k^{\mathrm{T}}(\bar{\boldsymbol{s}}_k^{c_1} - \bar{\boldsymbol{s}}_k^{c_2}). \tag{4.16}$$

Each item on the right side of Eq. 4.16 can be positive or negative, but we will just consider the value of $\left|\boldsymbol{w}_k^{\mathrm{T}}(\bar{\boldsymbol{s}}_k^{c_1} - \bar{\boldsymbol{s}}_k^{c_2})\right|$, because the absolute value indicates the importance of each item. A larger value of $\left|\boldsymbol{w}_k^{\mathrm{T}}(\bar{\boldsymbol{s}}_k^{c_1} - \bar{\boldsymbol{s}}_k^{c_2})\right|$ implies a stronger effect of the k-th sensor on the difference of means of the two classes, and a smaller value of $\left|\boldsymbol{w}_k^{\mathrm{T}}(\bar{\boldsymbol{s}}_k^{c_1} - \bar{\boldsymbol{s}}_k^{c_2})\right|$ implies a weaker effect of the k-th sensor on the difference of means of the two classes.

Hence, if we remove a sensor with smaller value of $\left|\boldsymbol{w}_k^{\mathrm{T}}(\bar{\boldsymbol{s}}_k^{c_1} - \bar{\boldsymbol{s}}_k^{c_2})\right|$, $(m_1 - m_2)^2$ does not change greatly. However, since each item on the right sides of the both

Eqs. 4.10 and 4.11 consists of m sensors, we can see that $s_1{}^2$ and $s_2{}^2$ will decrease seriously (about $1/m$ of total value) if we remove one sensor from the whole sensor set. It means $J(\boldsymbol{w})$ is larger if we remove a sensor with smaller $\left|\boldsymbol{w}_k^{\mathrm{T}}(\bar{\boldsymbol{s}}_k^{c_1} - \bar{\boldsymbol{s}}_k^{c_2})\right|$. It also indicates we can obtain a better $\boldsymbol{w}$ which makes the two classes separate as far as possible.

Therefore, $\left|\boldsymbol{w}_k^{\mathrm{T}}(\bar{\boldsymbol{s}}_k^{c_1} - \bar{\boldsymbol{s}}_k^{c_2})\right|$ is regarded as the contribution of the k-th sensor to discriminate the two classes. The sensor with the highest associated $\left|\boldsymbol{w}_k^{\mathrm{T}}(\bar{\boldsymbol{s}}_k^{c_1} - \bar{\boldsymbol{s}}_k^{c_2})\right|$ is the most important one for classification and should be added to the selected sensor subset. The sensor can be selected one by one in accordance to the descending order of $\left|\boldsymbol{w}_k^{\mathrm{T}}(\bar{\boldsymbol{s}}_k^{c_1} - \bar{\boldsymbol{s}}_k^{c_2})\right|$ if they improve the classification result. The process ends when adding a new sensor does not improve the result.

4.2.4 Weight of Sensor

An advantage of this approach is that it provides the weight of a sensor for a given classification task. Since $\left|\bar{\boldsymbol{s}}_k^{c_1} - \bar{\boldsymbol{s}}_k^{c_2}\right|$ indicates the difference in the sensor response between the two given classes, $\boldsymbol{w}_k$ can be considered as the scaling factors of each sensor and each sensor is multiplied by the corresponding component of $\boldsymbol{w}_k$ to obtain the item $\left|\boldsymbol{w}_k^{\mathrm{T}}(\bar{\boldsymbol{s}}_k^{c_1} - \bar{\boldsymbol{s}}_k^{c_2})\right|$, making the magnitude of $\boldsymbol{w}_k$ indicative of the importance of the k-th sensor.

Equation 4.12 shows a series of scaling vectors for the m sensors. For any $\boldsymbol{w}_k(k = 1, 2, \ldots, m)$, it has t_d entries, $[w_{t_1}, w_{t_2}, \ldots, w_{t_d}]$. We use the following regularization item to exploit the scaling information of the sensor (Subrahmanya and Shin 2010),

$$W_k = \left\|\boldsymbol{w}_k\right\|_2, \tag{4.17}$$

which stands for the weight of the k-th sensor for a given classification task.

4.2.5 Algorithm Conclusion

Algorithm 4.2.1 summarizes the complete sensor selection procedure. In step 5, the classification accuracy is calculated by using the principal component analysis (PCA, to extract the characteristic features of samples) coupled with the K Nearest Neighbor (KNN, as a classifier) (Guo et al. 2010).

Algorithm 4.2.1 LDASS algorithm

```
Require: Two classes $\boldsymbol{X}^{c_1} = [\boldsymbol{x}_1^{c_1}, \boldsymbol{x}_2^{c_1}, \dots, \boldsymbol{x}_{n_1}^{c_1}]$ and $\boldsymbol{X}^{c_2} = [\boldsymbol{x}_1^{c_2}, \boldsymbol{x}_2^{c_2}, \dots, \boldsymbol{x}_{n_2}^{c_2}]$, where
    $\boldsymbol{x}_i^{c_1} = [s_{i,1}^{c_1}; s_{i,2}^{c_1}; \dots; s_{i,m}^{c_1}]$ and $\boldsymbol{x}_i^{c_2} = [s_{i,1}^{c_2}; s_{i,2}^{c_2}; \dots; s_{i,m}^{c_2}]$ (';' means the two columns are
    stacked).
1: Compute the eigenvector $\boldsymbol{w}$ by the LDA, where $\boldsymbol{w} = [\boldsymbol{w}_1; \boldsymbol{w}_2; \dots; \boldsymbol{w}_m]$.
2: Compute $\left|\boldsymbol{w}_k^{\mathrm{T}}(\bar{\boldsymbol{s}}_k^{c_1} - \bar{\boldsymbol{s}}_k^{c_2})\right|$ for each sensor and sort them in descending order.
3: Compute the corresponding $W_k = \|\boldsymbol{w}_k\|_2$ for each selected sensor.
4: Form the full configuration of sensors $[W_1\boldsymbol{s}_1, W_2\boldsymbol{s}_2, \dots, W_{12}\boldsymbol{s}_{12}]$.
5: Select the top $k$ items from the full sensor set to calculate the classification accuracy.
6: Check if the classification accuracy stops increasing. If Yes, go to the end. If No, $k = k + 1$ and
   go to Step 5.
Ensure: An optimal configuration of sensors $[W_1\boldsymbol{s}_1, W_2\boldsymbol{s}_2, \dots, W_k\boldsymbol{s}_k]$.
```

4.3 Sensor Selection in Breath Analysis System

This section introduces the application of the LDASS in breath analysis system. Two special cases are introduced. One identifies a disease sample from healthy ones (disease diagnosis), the other distinguishes the before and after treatment samples from the same type of disease (medical treatment evaluation).

4.3.1 Sensor Selection for Disease Diagnosis

This subsection describes the database that is used, which includes four classes: healthy subjects, and patients with diabetes, renal diseases, and airway inflammation, respectively. The breath collection, data sampling, and signal processing method have been introduced in our previous work (Guo et al. 2010). Subjects were assumed to be healthy on the basis of a recent health check; subjects were confirmed to be afflicted with diabetes, renal diseases, or airway inflammation whose conditions have been verified and correlated by comparing their levels with standard clinical blood markers for the relevant diseases and conditions. Table 4.1 details the composition of the subject database. The samples of each condition are divided into two group: training set and test set, which will be used in the following experiments.

Figure 4.3 shows the responses of the twelve different sensors (S1-S12) to the four different samples over a 90 s sampling period. Figure 4.3a–d are typical responses to

Table 4.1 Composition of the subject database

Type of subjects	Number	Male/Female	Age	Training set/Test set
Healthy subjects	135	62/73	21–65	65/70
Subjects with diabetes	158	71/87	31–70	78/80
Subjects with renal diseases	167	63/104	28–70	82/85
Subjects with airway inflammation	126	59/67	16–62	61/65

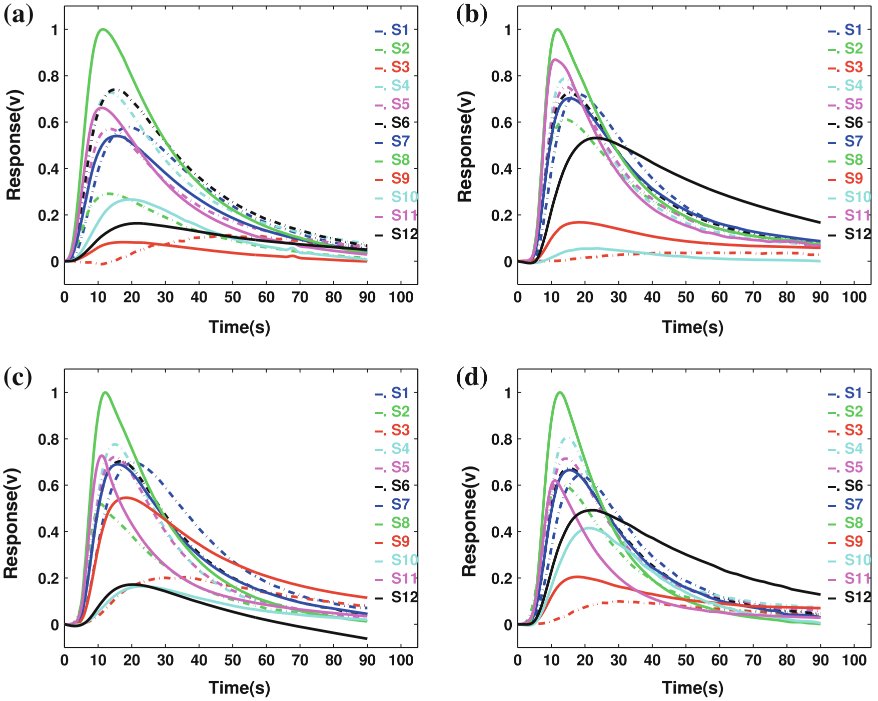


Fig. 4.3 Typical responses from four subject categories: **a** healthy subjects, **b** subjects with diabetes, **c** subjects with renal diseases, and **d** subjects with airway inflammation. The horizontal axis stands for the sampling time (0–90 s) and the vertical axis denotes the amplitude of sensor output in volts

healthy, diabetes, renal disease, and airway inflammation samples, respectively. As shown in Fig. 4.3, the responses of most sensors are not distinguishable between each pair of the four conditions. Figure 4.4 presents the mean responses of the twelve sensors which show the response of each sensor to the four types of samples. In each of the four categories, it is possible to find combinations of several sensors that could unambiguously identify each of the four conditions. For instance, the strongest responses to healthy samples come from the sixth, seventh, and eighth sensors while the strongest response to diabetes come from the second, fourth, fifth, and twelfth sensors. The strong response to the renal disease samples come from the first, third, ninth, and eleventh sensors, especially the ninth sensor, which is particularly sensitive to ammonia. The strongest response to airway inflammation comes from the tenth sensor, which detects nitric oxide. However, it also can be observed that most of sensors in the sensor array create overly similar responses in the classification of two of the four conditions except for these dominant sensors. This implies that not all sensors are useful in identifying an arbitrary sample. Therefore, not all of the sensors in the system are required and an effective sensor selection method is required for a given recognition problem.

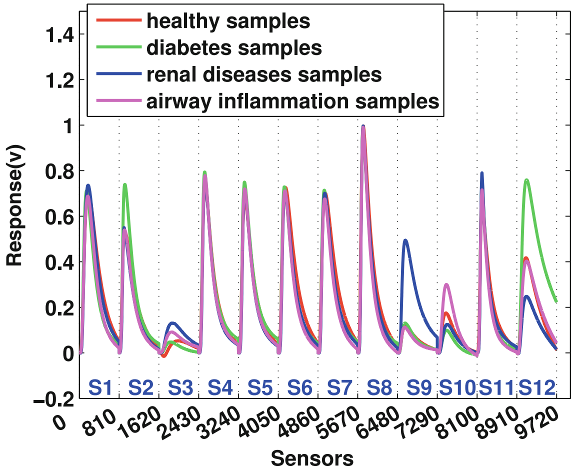


Fig. 4.4 Mean response of twelve sensors to four classes: healthy (*red line*), diabetes (*green line*), renal disease (*blue line*), and airway inflammation (*magenta line*). The horizontal axis denotes the twelve sensors and the vertical axis shows the responses of these sensors

In medicine, the reliability of a diagnosis is measured in terms of sensitivity and specificity, which are defined in Guo et al. (2010), with the outcome being either positive (unhealthy) or negative (healthy), which comes to a two class labeling problem. That is, the diagnosis of an unknown sample is either a type of disease or healthy. Hence, we divided the training samples into three groups. Each group was composed of two classes. The first group consisted of diabetes and healthy samples; the second group was made of renal disease and healthy samples; and the third group was formed with airway inflammation and healthy samples. We have selected three different sensor subsets in total to diagnose the three types of diseases, respectively, in this experiment.

Figure 4.5 shows the sub-vector $\boldsymbol{w}_k$ (solid line) computed by the LDA and the $(\bar{\boldsymbol{s}}_k^{c_1} - \bar{\boldsymbol{s}}_k^{c_2})$ (dash-dot line) by subtracting the mean of the responses of C_2 from the mean of responses of C_1. We express the response of the first sensor as $s_1 = [t_1, t_2 \ldots, t_{810}]^{\mathrm{T}}$, as shown in Eq. 4.1. Therefore, a sample that includes twelve sensors can be written as $x = [s_1; s_2; \ldots; s_{12}]$ (';' means the two columns are stacked), as shown by Eq. 4.2. The dimension of x is 9720 × 1. In these figures, the horizontal axis stands for the twelve sensors. The period 1–810 is the response of s_1, 811–1620 is the response of s_2, …, and 8911–9720 is the response of s_{12}. It can be seen from these figures that the curve of sub-vector $\boldsymbol{w}_k$ analogously varies to the differences between the two classes.

The vertical axis of Fig. 4.6 represents the value of $\left|\boldsymbol{w}_k^{\mathrm{T}}(\bar{\boldsymbol{s}}_k^{c_1} - \bar{\boldsymbol{s}}_k^{c_2})\right|$ of the corresponding sensor, indicating the contribution of the k-th sensor to distinguish the two classes. They are computed and sorted in descending order. As for the diabetes and healthy classes (Fig. 4.6a), the sensor that has the most contribution for distinguishing them, is sorted as {12, 11, 6, 8, 4, 5, 2, 7, 3, 10, 1, 9}. As for the renal disease and healthy classes (Fig. 4.6b), the order is {9, 12, 6, 3, 11, 1, 7, 8, 2, 10, 4, 5}, and in the airway inflammation and healthy classes (Fig. 4.6c), the order is {10, 6, 7, 1, 8, 12, 4, 5, 11, 2, 3, 9}, in which the numbers are the indexes of the sensors.

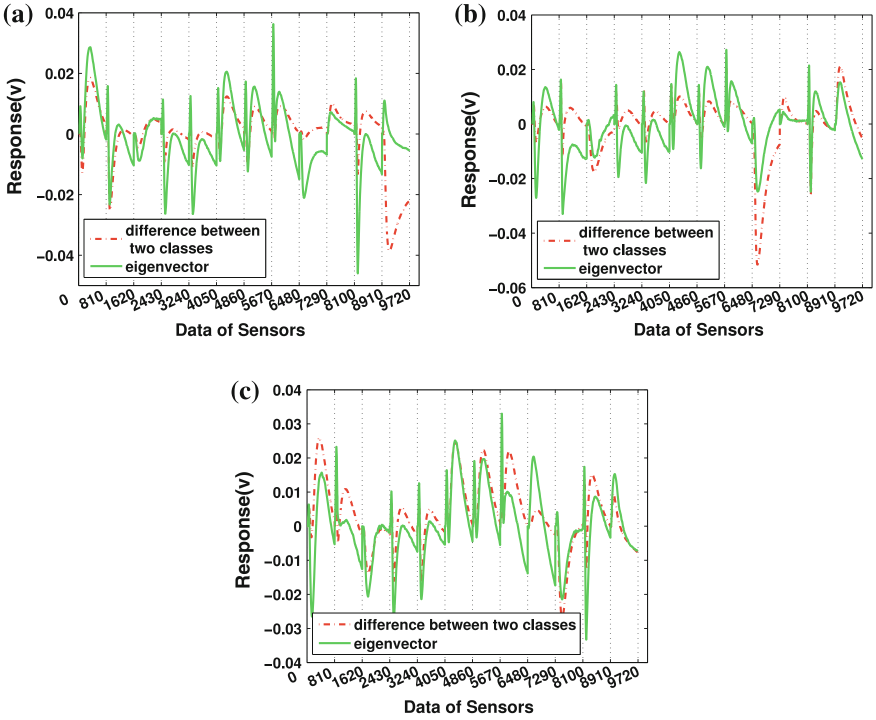


Fig. 4.5 The differences between the two classes (*dash-dot line*) and the eigenvectors computed by the LDA (solid line). **a** healthy samples and diabetes samples, **b** healthy samples and renal disease samples, and **c** healthy samples and airway inflammation samples. In the horizontal axis, each 810-point interval stands for one sensor. The vertical axis shows the differences between two classes and eigenvectors computed by LDA

To measure the classification accuracy of our proposed approach, we randomly selected a certain number of samples from each of the four subject categories as the test set and the remainder in each class formed the training set. The number of each test set and training set has been listed in Table 4.1. The PCA was used to extract the characteristic features of samples. The KNN ($K = 5$) classifier was then used to determine the class of each test sample. We ran this procedure 50 times and computed the results over all 50 runs.

Figure 4.7 shows the classification accuracy, which is given by sensitivity and specificity. In the horizontal axis, '1' indicates that only the most important sensor is used, '2' indicates that the top two important sensors are used, …, and '12' indicates that all of the sensors in the system are used. It can be seen from these figures that when only one sensor is used, the corresponding sensitivity and specificity are not satisfactory. When a number of sensors are added, the sensitivity and specificity gradually increase. It is worth noting that there is a maximum for both sensitivity and specificity, which means that there exists an optimal sensor subset when classifying two given classes.

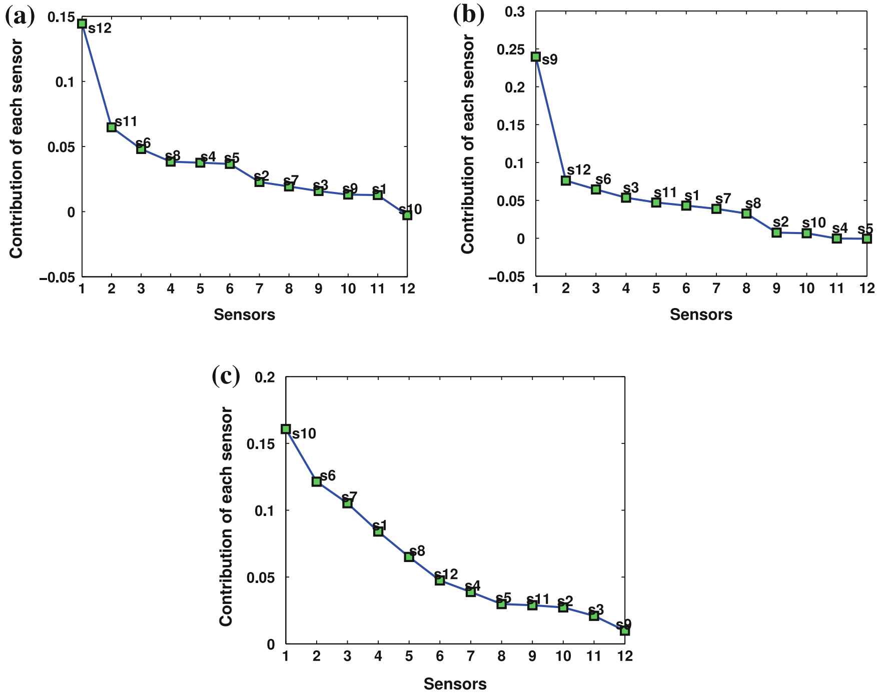


Fig. 4.6 The contributions of the sensors to discriminate two classes. **a** healthy samples and diabetes samples, **b** healthy samples and renal disease samples, and **c** healthy samples and airway inflammation samples. The horizontal axis stands for the weights of each sensor calculated by LDASS

From Fig. 4.7, we can also see that sometimes the sensitivity and specificity cannot reach the maximum at the same time. For example, in Fig. 4.7c, the sensitivity has reached its maximum when eight sensors are involved while the specificity is at its maximum when seven sensors are involved. In our experiments, we selected the optimal sensor subset by using the following criterion: use the sensor subset that allows the sum of the sensitivity and specificity to reach the maximum. When adding a new sensor did not improve the sum of sensitivity and specificity any more, we stopped the process. By using this criterion, for diagnosing diabetes, the optimal sensor subset is $[s_{12}, s_{11}, s_6, s_8, s_4, s_5]$ and the weights of these sensors are [0.5321 0.4662 0.3898 0.3166 0.2052 0.1804], respectively. For the diagnosis of renal diseases, the optimal sensor subset is $[s_9, s_{12}, s_6, s_3, s_{11}]$ and the, weight of these sensors are [0.6559 0.3745 0.3385 0.2888 0.2037], respectively. The optimal sensor subset to diagnose airway inflammation is $[s_{10}, s_6, s_7, s_1, s_8, s_{12}, s_4]$ and the weights of these sensors are [0.6805 0.4148 0.3125 0.2542 0.2083 0.1829 0.0534], respectively, as Table 4.2 shows.

We obtain the optimal configuration of sensors when these selected sensors are multiplied by the corresponding weights. They are $[0.5321s_{12}, 0.4662s_{11}, 0.3898s_6,$

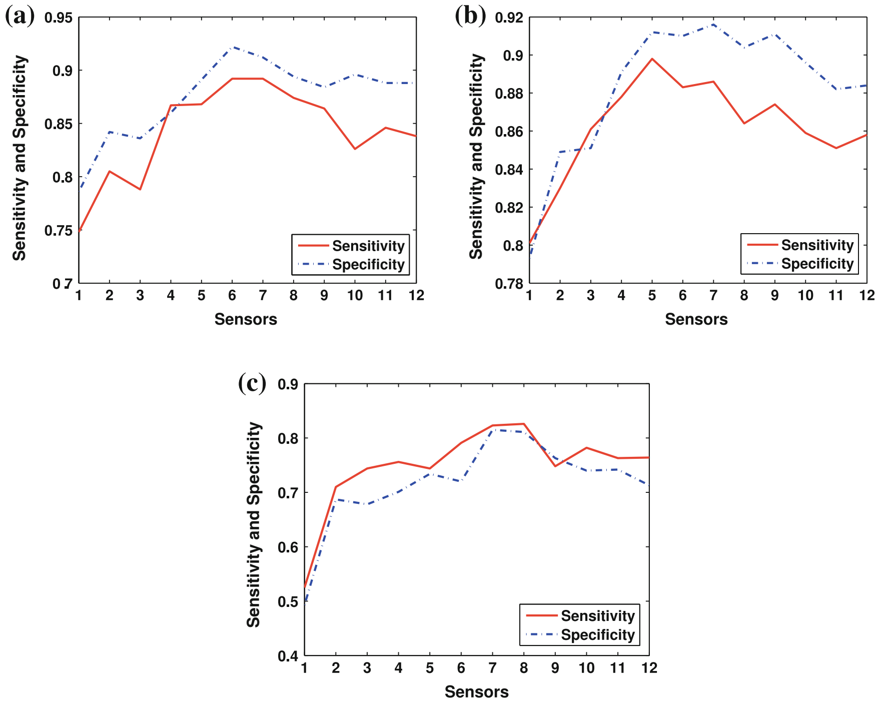


Fig. 4.7 Sensitivity (*dash-dot line*) and specificity (*solid line*). **a** healthy samples and diabetes samples, **b** healthy samples and renal disease samples, and **c** healthy samples and airway inflammation samples. The horizontal axis stands for the value of sensitivity and specificity, and the vertical axis indicates the sensors

$0.3166s_8, 0.2052s_4, 0.1804s_5]$, $[0.6559s_9, 0.3745s_{12}, 0.3385s_6, 0.2888s_3, 0.2037s_{11}]$, and $[0.6805s_{10}, 0.4148s_6, 0.3125s_7, 0.2542s_1, 0.2083s_8, 0.1829s_{12}, 0.0534s_4]$ for the three kinds of diseases, respectively. By using these configurations, coupling with PCA (to extract the features) and KNN (to classify the test samples), the classification results defined by sensitivity and specificity are given in Table 4.2. In the diabetes experiment, when all the sensors were used, the sensitivity and specificity are 86.11% and 86.25%, respectively. However, when our proposed approach (LDASS) was used, only six sensors were selected, with a sensitivity and specificity of 89.23% and 90.01%, respectively. In the renal disease experiment, when using all sensors, the sensitivity and specificity of this diagnosis are 83.62% and 84.17%, respectively. When LDASS was used, five sensors were selected, with a sensitivity and specificity of 88.35% and 89.16%, respectively. Similarly, for the airway inflammation diagnosis experiment, when using all sensors, the sensitivity and specificity of the diagnosis are 72.71% and 71.90%, respectively. But the sensitivity and specificity rise to 81.79% and 81.71%, respectively with seven sensors selected by LDASS.

From the results, we can see that after using the LDASS, the results are better than using the whole sensor set. The proposed approach is not only able to remove

Table 4.2 Classification results defined by sensitivity and specificity

Conditions	Use all sensors		LDASS			
	Sens (%)	Spec (%)	Sensors	Weight of sensors	Sens (%)	Spec (%)
Diabetes	86.11	86.25	12, 11, 6, 8, 4, 5	[0.5321 0.4662 0.3898 0.3166 0.2052 0.1804]	89.23	90.01
Renal diseases	83.62	84.17	9, 12, 6, 3, 11	[0.6559 0.3745 0.3385 0.2888 0.2037]	88.35	89.16
Airway inflammation	72.71	71.90	10, 6, 7, 1, 8, 12, 4	[0.6805 0.4148 0.3125 0.2542 0.2083 0.1829 0.0534]	81.79	81.71

Sens Sensitivity; *Spec* Specificity

the redundant sensors, but also enlarge the weights of the sensors which benefit the increasing of the classification accuracy.

4.3.2 Evaluating the Medical Treatment

In this subsection, our approach is tested when it is used to classify the same type of disease, i.e., distinguishing between before and after treatment breath samples from subjects with end stage renal failures. These samples were collected and classified for the purpose of evaluating the efficacy of treatment for renal failure. Detailed information about the sample collection and the disease treatment has been introduced in Guo et al. (2010). We collected the breath samples from 79 subjects with end-stage renal failure before and after they were treated by hemodialysis (a standard treatment for this condition). When classifying the two groups (before and after treatment) (Fig. 4.8), the accuracy is not satisfactory if all sensors are involved in the database. Since there is a reduction in the ammonia concentration in the exhaled breath of patients as the treatment proceeded (Narasimhan et al. 2001), the sensors that are sensitive to ammonia may show different responses between before and after treatment. However, for some sensors that are not sensitive to ammonia, only slightly distinct difference of responses may be generated between before and after treatment. Sensor selection is hence necessary for achieving high accuracy.

Figure 4.9 shows the eigenvector $\boldsymbol{w}_k$ (solid line) computed by the LDA and the $(\bar{\boldsymbol{s}}_k^{c_1} - \bar{\boldsymbol{s}}_k^{c_2})$ (dash-dot line) by subtracting the mean of the responses of sample before treatment from the mean of the responses of samples after treatment. The explanation for the horizontal axis is the same as in Sect. 4.3.1. Figure 4.10 gives the values of $\left|\boldsymbol{w}_k^{\mathrm{T}}(\bar{\boldsymbol{s}}_k^{c_1} - \bar{\boldsymbol{s}}_k^{c_2})\right|$, which indicates the contribution of the k-th sensor in distinguishing the two classes. They are computed and sorted in descending order as [9, 12, 10, 1, 7, 4, 5, 6, 3, 8, 2, 11] and the weights of these sensors are [0.4766 0.1327

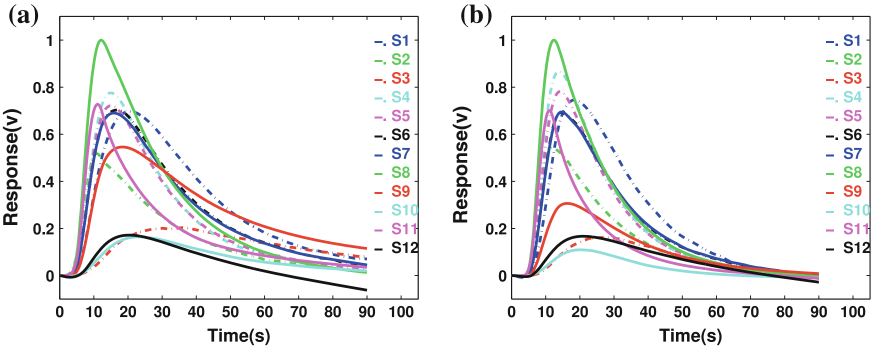
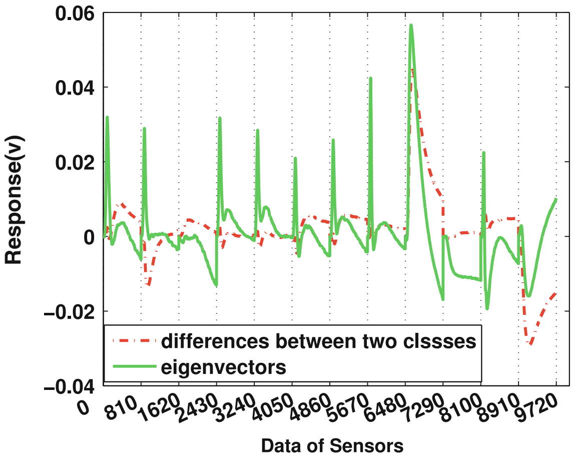


Fig. 4.8 Typical responses from the same patient: **a** before treatment, **b** after treatment. The horizontal axis stands for the sampling time (0–90 s) and the vertical axis denotes the amplitude of sensor output in volts

Fig. 4.9 The differences between two classes (*dash-dot line*) and the eigenvectors computed by the LDA (*solid line*). In the horizontal axis, each 810-point interval stands for one sensor. The vertical axis shows the differences between the two classes and the eigenvectors computed by the LDA



0.0206 0.0177 0.0106 0.0101 0.0099 0.0097 0.0075 0.0052 0.0044 0.0026], respectively. In these sensors, Sensor 9 has the greatest contribution, followed by Sensor 12.

In the 79 samples in each class, we randomly selected 39 samples as the training set and the remainder formed the test set to measure the classification accuracy of our proposed approach. The classification accuracy in this experiments is defined as the proportion of the sample amount of classified correctly against all samples. The PCA was used to extract the characteristic features of the samples. The KNN ($K = 5$) classifier was then used to determine the class of each test sample. We ran this procedure 50 times and computed the results over all 50 runs. The accuracies of the classification of both before and after treatment reach the maximum when we used the first three sensors (Fig. 4.11), i.e., [s_9, s_{12}, s_{10}]. Table 4.3 provides the classification results. In comparison to the results from the approach that used all

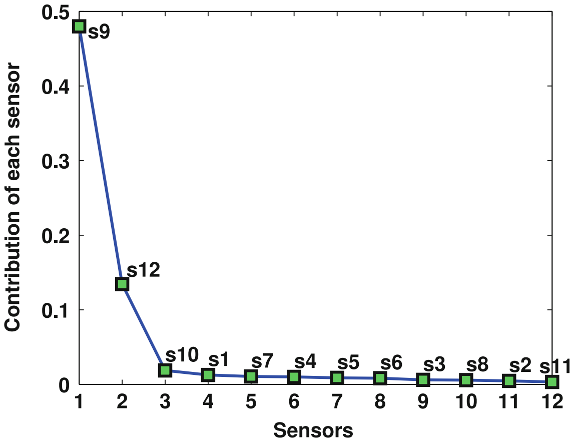


Fig. 4.10 The contributions of the sensors in discriminating the two classes

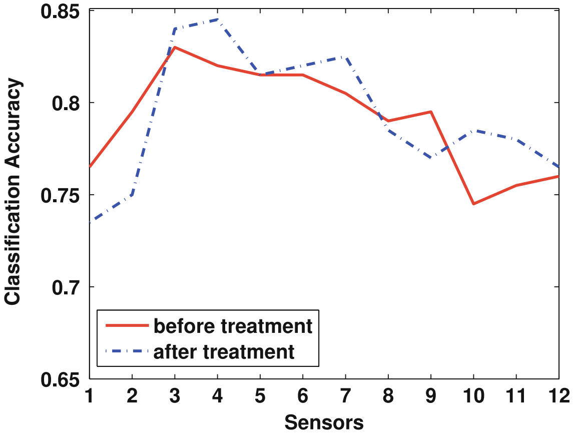


Fig. 4.11 The classification accuracy for before treatment and after treatment

Table 4.3 Classification results of two classes: renal failure samples before treatment and after treatment

Actual group member	Use all sensors	LDASS		
	Accuracy (%)	Sensors	Weight of sensors	Accuracy (%)
Before treatment	81.13	9, 12, 10	[0.4766 0.1327 0.0206]	82.52
After treatment	84.32			85.23

the sensors, the classification results of LDASS rise significantly, even though only three sensors were employed.

4.4 Comparison Experiment and Performance Analysis

In this section, the comparison experiments between our proposed LDASS method and common methods, such as the SFS and GA, will be discussed. Then, a performance analysis will be provided on these approaches.

4.4.1 *Sensor Selection for Disease Diagnosis*

To evaluate the proposed method, two other sensor selection methods, the SFS and GA, were tested by using the same database. The comparison results were listed in Table 4.4.

For the SFS, we first computed the accuracy by using all sensors one by one. s_l, the one with the highest classification accuracy was selected. The next step was to calculate all possibilities with two sensors, one of which was the s_l. The sensor, taken together with s_l, increased the classification accuracy mostly, was added to the sensor set. This process was iterated until the classification accuracy was not further increased by including a new sensor. Take diabetes diagnosis as an example, by using this approach, the selected optimal sensor subset was [s_{12}, s_{11}, s_2, s_8, s_5, s_4, s_6], according to the order of selection. In fact, the optimal sensor subset should be [s_{12}, s_{11}, s_6, s_8, s_4, s_5]. The SFS obtained the different sensor subset because it was trapped in local extrema at Sensor 2. The application of SFS on renal disease diagnosis has the same problem. The SFS selected [s_9, s_3, s_{11}, s_1, s_4, s_5] as the optimal sensor subset but the right one is [s_9, s_{12}, s_6, s_3, s_{11}] because the SFS was trapped in local extrema at Sensor 12 and Sensor 6. For airway inflammation diagnosis, the SFS selected the sensor subset correctly, same as LDASS. However, the classification results have a significant difference between the two approaches. The reason is that LDASS brings the weight of each sensor into the classification. The responses of sensors which have

Table 4.4 Classification results of the three approaches defined by sensitivity and specificity

	LDASS			SFS			GA		
	Sensors	Sens (%)	Spec (%)	Sensors	Sens (%)	Spec (%)	Sensors	Sens (%)	Spec (%)
1	12, 11, 6, 8, 4, 5	89.23	90.01	12, 11, 2, 8, 5, 4, 6	87.16	86.42	1, 2, 6, 8, 11, 12	89.74	88.41
2	9,12,6, 3, 11	88.35	89.16	9, 3, 11, 1, 4, 5	87.43	86.90	3, 6, 7, 9, 11, 12	87.24	88.53
3	10, 6, 7, 1, 8, 12, 4	81.79	81.71	10, 1, 6, 7, 8, 12, 4	80.21	79.19	1, 2, 4, 6, 7, 10, 12	80.14	79.61

1 Diabetes; *2* Renal diseases; *3* Airway inflammation

Sens Sensitivity; *Spec* Specificity

Table 4.5 Sensor selection time of three approaches

Conditions	Sensor selection time (s)		
	LDASS	SFS	GA
Diabetes	24.02	42.11	94.27
Renal diseases	24.90	43.20	93.73
Airway inflammation	25.63	45.69	101.50

positive impact on the classification were enlarged by being multiplied the scaling factors.

For the GA, we first selected four initial populations by randomly creating chromosomes, and then evaluated whether the classification was accurate by determining if each population has been reached. If the population has been reached, the chromosome of this population can be taken as the configuration of the sensor set and if not, a new population is generated by the GA. The parameters used for the evolution of the population are: population size: four chromosomes; initial population: random; selection: roulette-wheel; crossover: two points with probability 0.98; and mutation probability: 0.008. As for diabetes diagnosis, a created chromosome is 110001010011, which means that [s_1, s_2, s_6, s_8, s_{11}, s_{12}] are selected sensors. For renal disease diagnosis, a created chromosome is 001001101011, so [s_3, s_6, s_7, s_9, s_{11}, s_{12}] are selected sensors. And for airway inflammation diagnosis, a created chromosome is 110101100101, so [s_1, s_2, s_4, s_6, s_7, s_{10}, s_{12}] are selected sensors. Even though GA can obtain a global optimal solution, it has a large cost of computation. Besides, we found that the results of GA highly depend on the training samples. With different training samples, the selected sensors may be different, especially when the number of training samples is small. Therefore, the GA is not so robust as the LDASS in our experiments.

In the diabetes experiment, when using our proposed approach, only six sensors are selected, with a sensitivity and specificity of 89.23% and 90.01%, respectively. The result can be compared with the SFS approach, which has a sensitivity and specificity of 87.16% and 86.42%, respectively. Yet another comparison can be made with the GA, which has a sensitivity and specificity of 89.74% and 88.41%, respectively. In the renal disease experiment, our proposed approach selected five sensors with a sensitivity and specificity of 88.35% and 89.16%, respectively. In comparison with the result from the SFS approach, the sensitivity and specificity are 87.43% and 86.90%, respectively. In the GA, the sensitivity and specificity are 87.24% and 88.53%, respectively. Similarly, in the airway inflammation diagnosis experiment, our proposed approach selected five sensors with a sensitivity and specificity of 81.79% and 81.71%, respectively. In a comparison of the result from the SFS approach, the sensitivity and specificity are 80.21% and 79.19%, respectively. In the GA, the sensitivity and specificity are 80.14% and 79.61%, respectively.

Table 4.5 shows the sensor selection time of each method. The experiments were conducted on the computer with the configuration of 1.97 GB of RAM and E8200 @

2.66 GHz. The sensor selection times are 24.02, 24.90, and 25.63 s for the diabetes, renal diseases, and airway inflammation diagnoses, respectively, in our proposed approach. The sensor selection times are 42.11, 43.20, and 45.69 s for the diabetes, renal diseases, and airway inflammation diagnoses, respectively, for the SFS. The sensor selection times are 94.27, 93.73, and 101.50 s for the diabetes, renal diseases, and airway inflammation diagnoses, respectively, for the GA. It is obvious that the LDASS has a better performance in classification accuracy and processing time than the SFS and GA.

4.4.2 Evaluating the Medical Treatment

The proposed method is evaluated based on its performance in assessing the treatment of end stage renal failure. Two other sensor selection methods, the SFS and GA, are also tested by using the same database.

The experiment on the SFS is the same as the one in Sect. 4.4.1. By using this approach, the selected optimal sensor set is [s_9, s_8, s_{12}, s_1, s_7]. In comparison with the LDASS, the numbers in the selected sensor set have increased. After selecting Sensor 9, the SFS selected Sensor 8 and then Sensor 12. However, the optimal sensor subset should be [s_9, s_{12}, s_{10}]. There is an error in the selection because the SFS is easily trapped in local extrema.

The process of implementing the GA is also the same as that in Sect. 4.4.1. The final created chromosome is 000000001111, which means that [s_9, s_{10}, s_{11}, s_{12}] is selected as the optimal sensor subset. However, the GA can not provide the weight information of each sensor. By adding the weight of these sensors, the classification accuracy can be increased.

Table 4.6 shows the classification results of the three approaches, LDASS, SFS, and GA. When using the LDASS, the accuracy of classifying the before and after treatment test set is 84.32% and 85.23%., respectively. In comparison, two results are given from the SFS and GA. The accuracy of classifying the before and after treatment test set is 82.66% and 82.24%, and 84.13% and 84.40%, respectively.

The sensor selection time of each approach to evaluate the treatment of end stage renal failure is also given in Table 4.6. The experiments were conducted on the computer with the configuration of 1.97 GB of RAM and E8200 @ 2.66 GHz. Our proposed approach takes 9.30 s to evaluate the treatment. The SFS and GA require 21.72 and 87.01 s, respectively. It is obvious that the LDASS outperforms the SFS and GA in both classification accuracy and processing time.

4.5 Summary

This chapter first introduces breath analysis system for disease diagnosis and analyzes the problems that lie in the system. Since not all sensors can contribute to

Table 4.6 Comparison between three approaches for the classification of the renal failure samples

Actual group number	Accuracy		
	LDASS	SFS	GA
Before treatment (%)	84.32	82.66	82.24
After treatment (%)	85.23	84.13	84.40
Second selection time (s)	9.30	21.72	87.01

signal classification, it is necessary to provide a sufficiently large amount of sensors and select the most sensitive ones for applications. Therefore, this chapter has proposed an approach in the system to obtain the optimal configuration of sensors. Experiments show that this approach could significantly increase the classification accuracy and outperform other similar methods.

In spite of the many advantages, the proposed approach can still be improved. When seeking the best direction, the LDA does not work well with high-dimensional data and a small subset of samples. Some improvements (Xu et al. 2004a, b; Hu et al. 2009) could be investigated in the future to achieve better results.

References

Alpaydin E (2004) Introduction to machine learning. The MIT Press

Amann A, Schmid A, Scholl-Burgi S, Telser S, Hinterhuber H (2005) Breath analysis for medical diagnosis and therapeutic monitoring. Spectrosc Eur 17(3):18–20

Baraldi E, Carraro S (2006) Exhaled NO and breath condensate. Paediatr Respir Rev 7:20–22

Brighenti F, Benini L, Del Rio D, Casiraghi C, Pellegrini N, Scazzina F, Jenkins D, Vantini I (2006) Colonic fermentation of indigestible carbohydrates contributes to the second-meal effect. Am J Clin Nutr 83(4):817–822

D'Amico A, Di Natale C, Paolesse R, Macagnano A, Martinelli E, Pennazza G, Santonico M, Bernabei M, Roscioni C, Galluccio G, et al (2007) Olfactory systems for medical applications. Sens Actuators B Chem 130(1):458–465

D'Amico A, Pennazza G, Santonico M, Martinelli E, Roscioni C, Galluccio G, Paolesse R, Di Natale C (2010) An investigation on electronic nose diagnosis of lung cancer. Lung Cancer 68(2):170–176

Davies S, Spanel P, Smith D (1997) Quantitative analysis of ammonia on the breath of patients in end-stage renal failure. Kidney Int 52(1):223–228

Deng C, Zhang J, Yu X, Zhang W, Zhang X (2004) Determination of acetone in human breath by gas chromatography-mass spectrometry and solid-phase microextraction with on-fiber derivatization. J Chromatogr B 810(2):269–275

Di Francesco F, Fuoco R, Trivella M, Ceccarini A (2005) Breath analysis: trends in techniques and clinical applications. Microchem J 79(1–2):405–410

Di Natale C, Macagnano A, Martinelli E, Paolesse R, D'Arcangelo G, Roscioni C, Finazzi-Agrò A, D'Amico A (2003) Lung cancer identification by the analysis of breath by means of an array of non-selective gas sensors. Biosens Bioelectron 18(10):1209–1218

Dragonieri S, Schot R, Mertens B, Le Cessie S, Gauw S, Spanevello A, Resta O, Willard N, Vink T, Rabe K et al (2007) An electronic nose in the discrimination of patients with asthma and controls. J Allergy Clin Immunol 120(4):856–862

Dragonieri S, Annema J, Schot R, van der Schee M, Spanevello A, Carratú P, Resta O, Rabe K, Sterk P (2009) An electronic nose in the discrimination of patients with non-small cell lung cancer and COPD. Lung Cancer 64(2):166–170

Dweik R, Amann A (2008) Exhaled breath analysis: the new frontier in medical testing. J Breath Res 2(030):301

Gardner J, Boilot P, Hines E (2005) Enhancing electronic nose performance by sensor selection using a new integer-based genetic algorithm approach. Sens Actuators B Chem 106(1):114–121

Gualdron O, Brezmes J, Llobet E, Amari A, Vilanova X, Bouchikhi B, Correig X (2007) Variable selection for support vector machine based multisensor systems. Sensors Actuators B Chem 122(1):259–268

Guo D, Zhang D, Li N, Zhang L, Yang J (2010) A novel breath analysis system based on electronic olfaction. IEEE Trans Biomed Eng 57(11):2753–2763

Horvath I, Loukides S, Wodehouse T, Kharitonov S, Cole P, Barnes P (1998) Increased levels of exhaled carbon monoxide in bronchiectasis: a new marker of oxidative stress. Br Med J 53(10):867–870

Hu Q, Pedrycz W, Yu D, Lang J (2009) Selecting discrete and continuous features based on neighborhood decision error minimization. IEEE Trans Syst Man Cybern Part B Cybern 40(1):137–150

Kermani B, Schiffman S, Nagle H (1998) A novel method for reducing the dimensionality in a sensor array. IEEE Trans Instrum Meas 47(3):728–741

Kharitonov S, Wells A, O'connor B, Cole P, Hansell D, Logan-Sinclair R, Barnes P, (1995) Elevated levels of exhaled nitric oxide in bronchiectasis. Am J Respir Crit Care Med 151(6):1889–1893

Le Marchand L, Wilkens L, Harwood P, Cooney R (2006) Breath hydrogen and methane in populations at different risk for colon cancer. Int J Cancer 55(6):887–890

Maziak W, Loukides S, Culpitt S, Sullivan P, Kharitonov S, Barnes P (1998) Exhaled nitric oxide in chronic obstructive pulmonary disease. Am J Respir Crit Care Med 157(3):998–1002

Miekisch W, Schubert J, Noeldge-Schomburg G (2004) Diagnostic potential of breath analysis-focus on volatile organic compounds. Clin Chimi Acta 347(1–2):25–39

Narasimhan L, Goodman W, Patel C (2001) Correlation of breath ammonia with blood urea nitrogen and creatinine during hemodialysis. Proc Nat Acad Sci 98(8):4617–4621

Pardo A, Marco S, Calaza C, Ortega A, Perera A, Sundic T, Samitier J (2001) Methods for sensors selection in pattern recognition. In: Proceedings of the seventh international symposium on olfaction and electronic noses. Taylor & Francis, pp 83–88

Phaisangittisagula E, Nagle H, Areekul V (2010) Intelligent method for sensor subset selection for machine olfaction. Sens Actuators B Chem 145(1):507–515

Phillips M, Cataneo R, Cummin A, Gagliardi A, Gleeson K, Greenberg J, Maxfield R, Rom W (2003a) Detection of Lung Cancer With Volatile Markers in the Breath. Chest 123(6):2115–2123

Phillips M, Cataneo R, Ditkoff B, Fisher P, Greenberg J, Gunawardena R, Kwon C, Rahbari-Oskoui F, Wong C (2003b) Volatile markers of breast cancer in the breath. Breast J 9(3):184–191

Phillips M, Cataneo R, Condos R, Ring Erickson G, Greenberg J, La Bombardi V, Munawar M, Tietje O (2007) Volatile biomarkers of pulmonary tuberculosis in the breath. Tuberculosis 87(1):44–52

Schubert J, Miekisch W, Geiger K, Nöldge-Schomburg G (2004) Breath analysis in critically ill patients: potential and limitations. Expert Rev Mol Diagn 4(5):619–629

Sehnert S, Jiang L, Burdick J, Risby T (2002) Breath biomarkers for detection of human liver diseases: preliminary study. Biomarkers 7(2):174–187

Shih C, Lin Y, Lee K, Chien P, Drake P (2010) Real time electronic nose based pathogen detection for respiratory intensive care patients. Chemical, Sensors and Actuators B

Subrahmanya N, Shin Y (2010) Sparse multiple kernel learning for signal processing applications. IEEE Trans Pattern Anal Mach Intell 32(5):788–798

Van Berkel J, Dallinga J, Möller G, Godschalk R, Moonen E, Wouters E, Van Schooten F (2008) Development of accurate classification method based on the analysis of volatile organic compounds from human exhaled air. J Chromatogr B 861(1):101–107

Xu Y, Yang J, Jin Z (2004a) A novel method for Fisher discriminant analysis. Pattern Recogn 37(2):381–384

Xu Y, Yang J, Yang J (2004b) A reformative kernel Fisher discriminant analysis. Pattern Recogn 37(6):1299–1302

Yu J, Byun H, So M, Huh J (2005) Analysis of diabetic patient's breath with conducting polymer sensor array. Sens Actuators B Chem 108(1–2):305–308

Zhang S, Xie C, Zeng D, Li H, Liu Y, Cai S (2009) A sensor array optimization method for electronic noses with sub-arrays. Sens Actuators B Chem 142(1):243–252

Chapter 5
Sensor Evaluation in a Breath Acquisition System

Abstract Breath acquisition systems contain arrays of correlated chemical sensors. For such systems, sensor selection is needed. From the process of sensor selection, some insight behind the performance of different sensor arrays can be obtained. Thus, we can know more about the sensors, which could help us with the selection work in turn. In this chapter, we focus on the evaluation of sensor performance instead of particular sensor selection techniques. First, a breath acquisition system for diabetes diagnosis with 16 sensors is described. Based on this system, several methods are proposed to evaluate the importance, unique discriminant information, and redundancy of each sensor. They are based on the results of exhaustive sensor selection. These methods are made convenient to observe and draw intuitive conclusions. They are applied to the breath acquisition system and some useful discoveries about the sensors in the system are made accordingly.

Keywords Average accuracy improvement (AAI) • Cumulative sensor importance (CSI) • Sensor accuracy improvement (SAI) • Sensor accuracy similarity (SAS) • Sensor evaluation

5.1 Introduction

Researchers have developed many techniques to select useful sensors from the original sensor array. Wilks lambda-statistic coupled with elimination transform was used in Yin et al. (2013) to pick sensors with high discriminant ability. A rough set-based approach which could simultaneously classify the data as well as optimize the array was applied by Bag et al. (2011). In Szecowka et al. (2011), the authors tried to utilize neural network sensitivity analysis for this task. As introduced in the last chapter, linear discriminant analysis (LDA)-based sensor selection technique has also been proposed to estimate the contribution weight of each sensor, followed by subset selection (Guo et al. 2011). Traditional methods such as genetic algorithm (GA) are also feasible choices (Gardner et al. 2005).

D. Zhang et al., *Breath Analysis for Medical Applications*,
DOI 10.1007/978-981-10-4322-2_5

All of these methods have succeeded in providing optimized sensor arrays with higher accuracy and smaller size. Many of them could also estimate the importance of each sensor by either weights or an order of sensors (Bag et al. 2011; Guo et al. 2011; Szecowka et al. 2011; Yin et al. 2013). However, these importance indices do not have clear meanings. Sometimes questions such as "how much the accuracy will improve by adding this sensor" may be raised. Besides, we may also be interested in the inter-relationships between sensors, e.g., "given the presence of sensor S2, how much will the accuracy change by adding sensor S1?" "Does sensor S1 have very similar performance to S2?" From the answers to these questions, useful messages can be inferred about the importance, unique discriminant information, and redundancy of the sensors in our prediction task. These messages help us to select better array in turn.

To answer these questions, a direct and reliable way is to analysis the results of an exhaustive search among all possible sensor combinations. For an e-nose with N sensors, the number of experiments to carry out during searching is $2^N - 1$. For example, if $N = 13$, the experiment number is 8191. In fact, this number is affordable for modern computers if the time cost for each experiment is not too much. This exhaustive searching method is suitable for the cases in (Bag et al. 2011; Guo et al. 2011; Szecowka et al. 2011), since the e-noses have no more than 13 sensors. This method can also ensure to select the optimal array.

In this chapter, we propose several techniques based on exhaustive searching to evaluate the sensors so as to infer their importance, unique discriminant information, and redundancy (Yan and Zhang 2014). The experiments are carried out with a breath analysis system. It has 16 sensors to measure the volatile organic compounds (VOCs) in breath. 167 breath samples from healthy subjects and 151 from diabetics have been collected. This dataset is used to test the sensor evaluation methods. Several useful conclusions will be drawn after the evaluation. The conclusions have been used to develop a practical breath analysis system (Yan et al. 2014).

The remainder of this chapter is organized as follows: Sect. 5.2 describes the breath acquisition system in detail; Sect. 5.3 introduces the idea of our sensor evaluation methods. The analysis results and a few discussions will be provided in Sect. 5.4. Section 5.5 summarizes the chapter.

5.2 System Description

The proposed breath analysis system includes a device to measure breath and a set of data analysis algorithms. In this section, we will first introduce the framework of the breath analysis device. The key part of the device, the sensor array, will be described next. After that the process of breath collection and measurement will be shown. Finally, a brief introduction will be given about the data analysis algorithms.

5.2.1 Framework of the Device

Figure 5.1 shows the main framework of the breath acquisition device. The gas sensors in our sensor array include an electrochemical ammonia (NH_3) sensor, a photo ionization detector (PID) sensor, and 12 metal oxide semiconductor (MOS) sensors. The MOS sensors work in a relatively high temperature which is not suitable for the NH_3 and PID sensors, so separate small gas rooms were made for the NH_3 and PID sensors. Breath or fresh air is drawn from outside and pumped into the gas rooms by a micro vacuum pump. According to our experiments, the contamination and carryover in the pump is negligible.

The gas passes the gas room for the NH_3 sensor, the PID sensor, and the MOS sensors successively. The signal of the NH_3 and PID sensors are transmitted to the signal processing circuit through transmitting modules. The signal processing circuit magnifies and filters the responses of all the sensors. Finally, a data acquisition card digitizes the processed signals and transmits them to a computer using a USB cable. On the other way round, the computer sends control signals to the data acquisition card to control the on/off of the pump and the modulation voltage of the temperature modulated sensors. The whole device is powered by a 12 V power adapter. The power is sent to each unit by a power distribution circuit.

5.2.2 Sensor Array

There are 16 sensors in our device. Besides the NH_3 sensor, the PID sensor and the 12 MOS sensors, a humidity sensor and a MEMS mass flow sensor are also utilized. Table 5.1 is a list of these sensors. When choosing these candidate sensors, we focused on commercially available sensors because they are easier to acquire, more

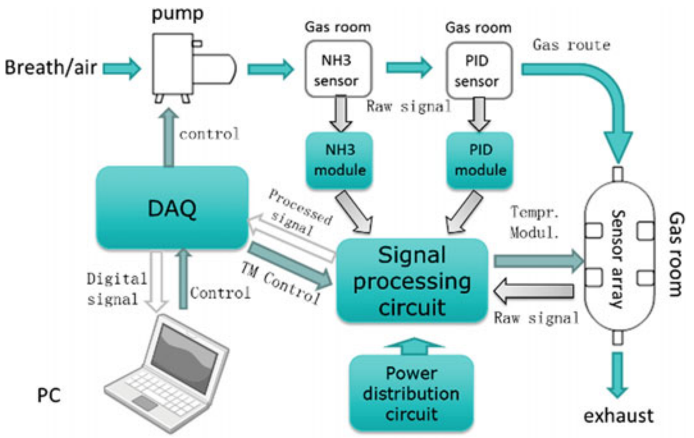


Fig. 5.1 Main framework of the proposed device

Table 5.1 Detailed sensor information

No.	Model	Function	Range	Company
1	D6F-P0001A1	Gas mass flow rate	0–100 (mL/min)	Omron Inc. Japan
2	NH_3 3E 100 SE	NH_3	0–100 (ppm)	City Inc. UK
3	piD-TECH 200	VOCs	1–200	Baseline-Mocon Inc. USA
4	TGS826	NH_3, VOCs	30–300	Figaro Inc. Japan
5	TGS2610-D00	VOCs, H_2	500–10,000	
6	TGS2602	VOCs, NH_3, H_2S	1–30	
7	TGS2600-TM	H_2, VOCs, CO	1–100	
8	GSBT11	VOCs	1–1000	Ogam Inc. Korea
9	TGS2602-TM	VOCs, NH_3, H_2S	1–30	Figaro Inc. Japan
10	WSP2111-TM	VOCs, H_2	5–40	Winsen, China
11	WSP2111	VOCs, H_2	5–40	
12	HTG3515CH	Humidity	10–95 (RH%)	Humirel Inc. France
13	TGS2610C	VOCs, H_2	500–10,000	Figaro Inc. Japan
14	TGS822	VOCs, H_2, CO	50–5000	
15	TGS2600	H_2, VOCs, CO	1–100	
16	SP3S-AQ2	VOCs, H_2, CO	1–1000	FIS Inc. Japan

robust and have a good variety. The diversity of the candidates was augmented by choosing sensors from different companies and of different types, sensitive spectrums and measurement ranges. We also paid attention to choose some sensors that have higher sensitivity to our target compounds such as acetone. Our references included sensor datasheets and previous studies (Guo et al. 2011; Yan et al. 2012).

Temperature modulation (TM) is a way of using the MOS sensors. Instead of giving the sensors a constant heating voltage as usual, this method periodically modulates the heating voltage. It has been proved that this method can increase the discriminability and selectivity of MOS sensor (Gutierrez-Osuna et al. 2003). Although it has been used a lot to classify different chemicals, TM has never been applied in breath acquisition systems. So three MOS sensors in our array (TGS2600, TGS2602, and WSP2111) were copied, with one copy of each sensor operated under TM. We wish to explore if they would outperform the original ones. In Table 5.1, the suffix "-TM" indicates that the sensor is a temperature modulated sensor.

5.2.3 Sampling Procedure

When collecting the breath sample from either a healthy or a diabetes subject, he/she is asked to exhale into a 600 mL Tedlar$^{®}$ gas bag through a disposable mouthpiece. After that, the gas bag is plugged onto the connector of the device to let the software control the device to finish the measurement of the breath. The measurement consists of 4 stages, including:

(1) *Baseline stage (1 s)*: The baseline values of the sensors are recorded for future data preprocessing.
(2) *Injection stage (7 s)*: The pump opens; breath is drawn from the gas bag to the gas room at a constant speed.
(3) *Reaction stage (56 s)*: The pump is off; the sensors continue reacting with the gas particles.
(4) *Purge stage (80 s)*: The pump opens again; fresh air is drawn into the gas room to push the breath gas out.

5.2.4 Data Analysis

In this chapter, sensors are evaluated based on their performance on the task of classification between healthy and diabetes samples. Before classification, the samples need to be preprocessed first. The preprocessing is made up of 3 steps: baseline removing, gas amount compensation, and humidity compensation. For each of the chemical sensors, the baseline value is estimated by its average response in the baseline stage. Then it is subtracted from the response of the corresponding sensor. Gas amount and humidity compensation is used to compensate the fluctuation of gas amount and humidity among samples. First, several acetone samples were collected with different gas amount and humidity. Then two linear models were built to describe each sensor's dependency on gas amount and humidity, respectively. Finally, the breath samples were compensated using these models and the responses of mass flow sensor and humidity sensor in the samples. The details of the compensation algorithm can be found in (Kashwan and Bhuyan 2005).

Next, principle component analysis (PCA) is used to extract a smaller set of features from the preprocessed responses of 14 chemical sensors (except the mass flow sensor and the humidity sensor). A critical coefficient of PCA is the ratio of variance that can be explained by the extracted features. In this chapter, the ratio is set to be 99.99%. Finally, support vector machine (SVM) with a Gaussian kernel is adopted to do the final classification. Details of these algorithms can be found in many related papers or textbooks.

5.3 Sensor Evaluation Methods

5.3.1 Cumulative Sensor Importance

Many sensor selection algorithms are capable of giving each sensor an estimation of importance/weight/contribution. For example, the estimation may come from some intermediate values of the classifier (Guo et al. 2011; Szecowka et al. 2011). However, if the prediction accuracy of every possible sensor array is available, we

Table 5.2 An example of the array rank list

Rank	Sensors	Accuracy
1	s7, s8, s9, s10, s11, s16	0.923
2	s7, s8, s9, s10, s11	0.922
3	s3, s4, s7, s8, s9, s10, s11	0.919
…	…	…

can get a more reliable importance evaluation. The question is, how to evaluate the sensor importance from these accuracy values?

When we have the prediction accuracy of every possible sensor array, it is easy to sort the arrays according to the descending order of the accuracy values. The first few items of this sorted list may be like Table 5.2. A reasonable intuition is that the sensor in the rank 1 array must be important. But because of the instability of the prediction model brought by the limited sample size, and the large correlation among the chemical sensors, some important sensor may not be present in the rank 1 array. Besides, what is the importance order of the sensors in the rank 1 array?

Following the evaluation criterion known as "cumulative match score" (Phillips et al. 2000), we propose a criterion named "cumulative sensor importance" (CSI). For each sensor, its occurrence in the top K ranked arrays is counted and denoted as CSI(K). A sensor with high importance should occur more in the top ranked arrays than those with low importance. So CSI can be used to indicate the importance of the sensors. Curves of CSI(K) with K as the x-axis can be plotted. Thus, by observing these curves' height, we can get the importance order of all the sensors.

5.3.2 Average Accuracy Improvement

The average accuracy improvement (AAI) intends to answer such a question: "how much will the accuracy improve by adding a certain sensor?" For instance, we already have an array A which does not contain the sensor S. The prediction accuracy for A is acc(A). So the accuracy improvement after adding sensor S is acc ($A \cup S$) − acc(A). Since A could be any array without S, it is better to average this improvement for all possible A's. So AAI(S) is defined by this average accuracy improvement of sensor S.

If we can use the prediction accuracy to estimate the discriminating information of an array, AAI(S) somehow reflects the average of unique discriminating information in S, subtracting the noise or redundancy in S. This is because the accuracy is likely to increase only if S contains some unique discriminating information that the original array does not have. Or, in other words, S should contain some complementary information. If compared to the discriminative information it can bring, S introduces more noise or redundancy to the prediction algorithm, and AAI (S) could even be negative.

As a result, we could also draw a scatter plot for all the sensors, with the accuracy of each single sensor as the x-axis, and the AAI of each sensor as the y-axis. Suppose the accuracy of each single sensor reflects the total discriminating information of the sensor, then the sensors in the right part of this plot are those with high discriminating information by themselves, while those in the upper part are those with more unique discriminating information. When selecting sensors, it is reasonable that we first select those in the upper right corner of the scatter plot, followed by the ones in the upper part. When discarding sensors, those in the lower left corner should be the first choices.

5.3.3 Sensor Inter-relationship

The AAI reflects the average of the unique discriminating information of a sensor. Sometimes we are interested in relationships between sensors pairs, e.g., “compared to sensor $S2$, does $S1$ contain any unique discriminating information?” This could be estimated by sensor accuracy improvement (SAI). We define SAI($S1$, $S2$) = acc ($S1 \cup S2$) − acc($S2$). The reason is that only if $S1$ has some unique discriminating information that the $S2$ does not have, acc($S1 \cup S2$) is likely to be larger than acc ($S2$). If compared to the discriminative information it can bring, $S1$ introduces more noise or redundancy to the prediction algorithm, SAI($S1$, $S2$) could even be negative.

In an opposite aspect, we may wonder if sensor $S1$ is very similar to $S2$. If so, one of them is likely to be redundant and can be discarded. The similarity could be estimated by algorithms such as correlation or mutual information between the features of two sensors, but the result is not task-specific. Our real goal is to evaluate if the two sensors contribute analogously to our prediction task. Therefore, it is better to use the prediction accuracy to judge the similarity. Suppose there is an array A which contains neither $S1$ nor $S2$. If acc($A \cup S1$) is close to acc($A \cup S2$) for every possible A, then $S1$ has similar performance to $S2$ for this prediction task. Thus, the sensor accuracy similarity (SAS) can be defined as SAS($S1$, $S2$) = 1 − mean(abs(acc($A \cup S1$) − acc($A \cup S2$))), where the operation “abs” means the absolute value and “mean” means averaging over all possible A’s.

Both SAI and SAS can be described with a matrix, where each row and each column correspond to a sensor. SAI(i, j) is an estimation of the unique discriminating information of sensor i that the sensor j does not have. SAS(i, j) is an estimation of the similarity between sensor i and j. The larger the SAS(i, j), the more similar. It is easy to know that SAI(i, i) = 0, SAS(i, i) = 1. SAI(i, j) $\neq$ SAI (j, i), but SAS(i, j) = SAS(j, i).

In this section, we introduced our methods to compute the cumulative sensor importance and average accuracy improvement for each sensor, as well as sensor accuracy improvement and sensor accuracy similarity between each pair of sensors. Each measurement can be plotted to a figure, which is convenient for us to observe and draw intuitive conclusions. While traditional sensor selection methods only tell us which sensors are important, these methods give us more insight and tell us why certain sensors are important and some should be discarded. These conclusions come from analyzing the exhaustive searching results of a certain task, followed by cumulating or averaging, so they are more reliable and more specific for the task. In the next chapter, these methods will be used to analyze the data collected by our breath acquisition system, followed by a detailed discussion.

5.4 Experiments and Discussion

5.4.1 Experiment Configuration

167 breath samples from healthy subjects and 151 from diabetics were collected. To classify between healthy and diabetes samples, half of samples were randomly picked from both class to form the training set. The rest of the samples formed the test set. We ran the classification 50 times and calculate the average accuracy for sensor performance evaluation.

There are 14 chemical sensors in our breath acquisition system. The NH_3 sensor is ignored since it is irrelevant to our task and generates little response for all the samples. Thus, there are 13 sensors to evaluate altogether. 8191 arrays need to be tested. The program was run on the Matlab v8.0 software on a computer with 2.4 GHz, 8 core CPU and 16 GB RAM. The parallel function of Matlab was used to accelerate the program. It took 8.8 h to test the 8191 arrays. This time cost is acceptable for our task. If the sensor number becomes larger, it will be impractical to test all the arrays. However, some filter algorithms can be used to exclude some sensors before testing.

5.4.2 Sensor Evaluation Results

Figure 5.2 shows the cumulative sensor importance of the evaluated 13 sensors. For clarity, the sensors' names have been replaced by their number. One can check Table 5.1 for correspondence. From Fig. 5.2, it is clear that sensors 9, 10, 7, 8, 11, and 4 are the most important ones. For the same K, their CSI(K) is higher, indicating that they appear more frequently in the top K arrays.

From Fig. 5.3, we can get similar conclusions. The average accuracy improvement (AAI) of sensors 9, 7, 10, 8, and 4 are the highest ones. Sensor 9 can improve the accuracy by 4% on average. This AAI list is not identical to CSI, since AAI averaged over the accuracy of all the arrays, while in the CSI figure only the top 1500 arrays are shown. The sensors 3, 13, 15, and 16 have negative AAI values, indicating that on average they introduce more noise and redundancy than discriminating information. The AAI is basically proportional to the accuracy of each single sensor. But for sensors 14 and 15, although they have higher single accuracy

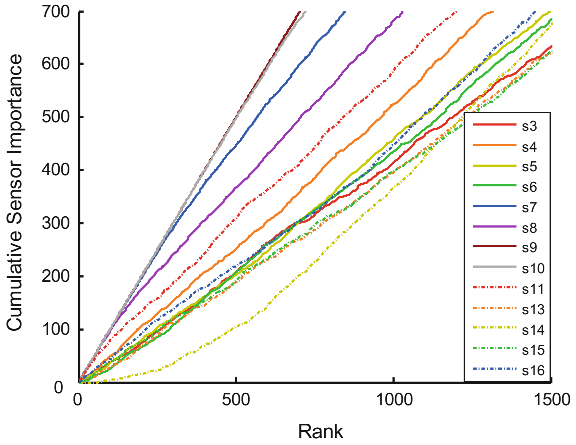


Fig. 5.2 Cumulative sensor importance (CSI) for all sensors

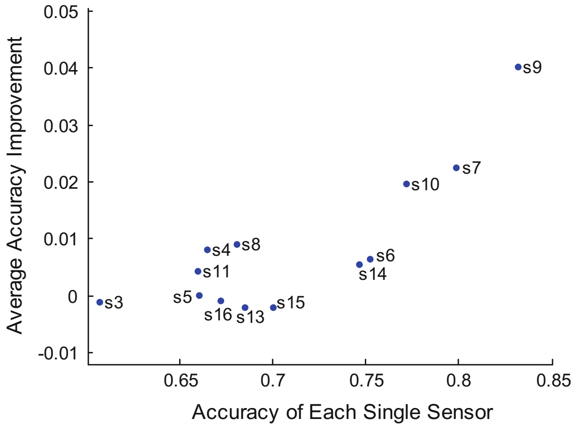


Fig. 5.3 Accuracy of each single sensor versus average accuracy improvement (AAI)

than sensors 8 and 4, their AAI is lower than sensors 8 and 4. This is probably because sensor 14 and 15 have some discriminating information, but much of it is overlapped with other sensors. Their unique discriminating information left is less than sensors 4 and 8.

The sensor accuracy improvement (SAI) matrix is illustrated in Fig. 5.4. For easy reading, pseudo-color is used to represent values. Red indicates large positive SAI values, whereas blue indicates large negative values. The *i*th row and *j*th column of this matrix is SAI(*i*, *j*), which means that adding sensor *i* to sensor *j*, the prediction accuracy will increase SAI(*i*, *j*). We can see that sensors 9, 7, 10 (the s9, s7, s10 row), and so on have relatively large SAI values along all the columns. This is a hint that they have some unique discriminating information that every other sensor does not have. This is consistent to our observation from CSI and AAI. Another finding is that SAI(s7, s15) > 0 but SAI(s15, s7) < 0. This is a hint that the discriminating information in s15 is a subset of that in s7.

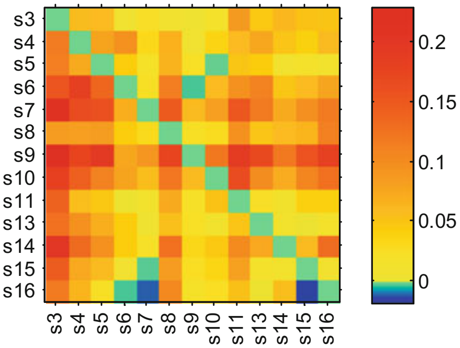


Fig. 5.4 Sensor accuracy improvement (SAI) matrix

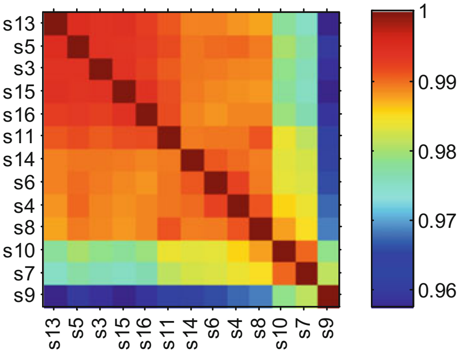


Fig. 5.5 Sensor accuracy similarity (SAS) matrix

To estimate the redundancy between sensors, we plot the sensor accuracy similarity (SAS) matrix in Fig. 5.5. A red-color grid (*i*, *j*) in this figure means that sensor *i* and *j* have very similar performance, thus one of them may be redundant and can be discarded. For clarity reason, the order of row and column has been rearranged to cluster sensors with close SAS values. As we can see, sensors 13, 5, 3, 15, and 16 have large SAS values between one another. This possibly means that they have highly correlated responses. An interesting fact is that from Fig. 5.3 we know none of them has much unique discriminating information and from Fig. 5.2 we know none of them is very important. Keeping one of them in the array is enough. Sensors 11, 14, 6, 4, and 8 also have some correlation between each other, but not as much as sensor 13 and so on. Sensors 10, 7, and 9 are really different from other sensors. They contain much unique discriminating information and adding them to the array will make big differences to the accuracy of this task.

5.4.3 Discussion

In this section, the evaluation results will be related with some prior knowledge about the sensors to see if it will bring more discoveries on the sensors.

From the four evaluation techniques it is found that sensors 9, 7, and 10 (particularly sensor 9) are the most important sensors, contain the most unique discriminant information than other sensors, and are not redundant at all. They are all temperature modulated sensors. Their performance is much better than their counterparts without TM, i.e., sensor 6, 15, and 11. SAI(15, 7) and SAS(6, 9) are even negative (see Fig. 5.4), which is to say that compared to the TM version, the non-TM version of the two sensors contains only noise and redundancy. In conclusion, temperature modulation can be applied to breath acquisition systems and it can greatly improve the performance of MOS sensors.

Sensors 8, 4, and 6 have medium importance, unique discriminant information, and moderate redundancy. Actually, in our pilot experiments, these 3 sensors have high sensitivity to acetone, which is the main biomarker of diabetes. This fact can explain their importance.

Sensors 13, 5, 3, 15, 16, and 11 contain high redundancy and low unique discriminant information. This is consistent to the fact that the sensitivity features in their datasheets are very similar. Sensor 5 and 13 have very similar performance according to Fig. 5.5, which is possibly because that they both belong to the TGS2610 series.

The four figures above show that the PID sensor (sensor 3) is not a good choice for our task. It is because that a PID sensor is sensitive to a wide range of analytes (Baseline-Mocon 2007), thus have poor selectivity.

From the discussions above, we have more knowledge on the sensor selection results now. Although the best sensor array consists of sensors 7, 8, 9, 10, 11, and 16 according to exhaustive searching, it is not the only choice. It can be inferred that sensors 7, 9, 10 are essential; sensor 16 could be replaced by another one in sensor 5, 13, 15; sensors 8, 4, 6 are better kept; sensors 11 and 14 are optional; sensor 3 should be discarded.

In its datasheet, sensor 14 has high sensitivity to acetone. The datasheet of sensors 8, 4, and 6 do not include any content about acetone. But in our experiments, sensor 14's sensitivity of acetone is not comparable to the latter three sensors. This is maybe because that the latter three sensors are not designed to sense acetone, so it is not written in their datasheet. This finding shows that when selecting sensors for e-noses, the datasheets are not completely reliable, since they do not show the sensors' sensitivity to every analyte.

5.5 Summary

In this chapter, a breath acquisition system for diabetes diagnosis was introduced. Four techniques were developed to evaluate the sensors' performance. The cumulative sensor importance shows the importance order of the sensors. The average accuracy improvement displays the accuracy improvement brought by a sensor. The sensor accuracy improvement tells us the accuracy gain by combining one sensor with another. The sensor accuracy similarity shows the likeness of performance between two sensors. Different from traditional sensor selection methods which focus on selecting the best array, the proposed statistics aim at judging the importance, unique discriminating information, and redundancy of each sensor based on the exhaustive search results of all the possible sensor arrays.

The key step of using these techniques lies in comparing the results to prior knowledge about the sensors. It can bring us more insight and tell us why certain sensors are important and some should be discarded, thus provide new aspects for selecting sensors. We have tested these techniques on our data and draw some useful conclusions. The conclusions themselves may not be very helpful for other researches since they are specific for our task. But the analysis methods are suitable to analyze all systems with an array of sensors, especially when different sensors have correlation. The heavy computation cost needed to do the exhaustive search is an obstacle. A possible solution is to filter out some irrelative or redundant sensors in advance.

References

Bag AK, Tudu B, Roy J et al (2011) Optimization of sensor array in electronic nose: a rough set-based approach. IEEE Sensors J 11:3001–3008

Baseline-Mocon (2007) piD-tech® plus: photoionization sensor user's manual. Baseline-MOCON, Inc.

Gardner J, Boilot P, Hines E (2005) Enhancing electronic nose performance by sensor selection using a new integer-based genetic algorithm approach. Sensors Actuators B: Chem 106:114–121

Guo D, Zhang D, Zhang L (2011) An LDA based sensor selection approach used in breath analysis system. Sensors Actuators B: Chem 157:265–274

Gutierrez-Osuna R, Gutierrez-Galvez A, Powar N (2003) Transient response analysis for temperature-modulated chemoresistors. Sensors Actuators B: Chem 93:57–66

Kashwan K, Bhuyan M (2005) Robust electronic-nose system with temperature and humidity drift compensation for tea and spice flavour discrimination. In: 2005 Asian conference on sensors and the international conference on new techniques in pharmaceutical and biomedical research. IEEE, pp 154–158

Phillips PJ, Moon H, Rizvi SA et al (2000) The FERET evaluation methodology for face-recognition algorithms. IEEE Trans Pattern Anal Mach Intell 22:1090–1104

Szecowka P, Szczurek A, Licznerski B (2011) On reliability of neural network sensitivity analysis applied for sensor array optimization. Sensors Actuators B: Chem 157:298–303

Yan K, Zhang D (2012) A novel breath analysis system for diabetes diagnosis. In: 2012 international conference on computerized healthcare, Hong Kong, China, pp 166–170

Yan K, Zhang D (2014) Sensor evaluation in a breath analysis system. In: 2014 international conference on medical biometrics (ICMB). IEEE, Shenzhen, pp 35–40

Yan K, Zhang D, Wu D et al (2014) Design of a breath analysis system for diabetes screening and blood glucose level prediction. IEEE Trans Biomed Eng 61:2787–2795

Yin Y, Yu H, Chu B et al (2013) A sensor array optimization method of electronic nose based on elimination transform of Wilks statistic for discrimination of three kinds of vinegars. J Food Eng

Part III
Breath Signal Pre-processing

Chapter 6
Improving the Transfer Ability of Prediction Models

Abstract Calibration transfer aims at making the prediction model trained on one e-nose transferable to other e-noses, which is important for the large-scale deployment of e-noses, especially when the cost of sample collection is high. In this chapter, the transfer ability of prediction models is improved in two simple yet effective steps. First, windowed piecewise direct standardization (WPDS) is used to standardize the slave device, i.e., to transform the variables from the slave device to match the master one. Then, data from the master device are used to develop prediction models with a novel strategy named standardization-error-based model improvement (SEMI). Finally, the standardized slave data can be predicted by the models with a better accuracy. The proposed WPDS is a generalization of the widely used PDS algorithm. To evaluate the algorithms, three e-noses specialized for breath analysis are adopted to collect a dataset, which contains pure chemicals and breath samples. Experiments show that WPDS outperforms previous methods in the sense of standardization error and prediction accuracy; SEMI consistently enhances the accuracy of the master model applied to standardized slave data.

Keywords Calibration transfer · Standardization-error-based model improvement (SEMI) · Tikhonov regularization · Variable standardization · Windowed piecewise direct standardization (WPDS)

6.1 Introduction

As increasing number of e-nose systems are being deployed in real-life applications, the problem of instrumental variation is receiving more and more attention. When two e-noses of the same model are used to measure the same gas sample, their responses are usually not identical, which is due to the variations in the manufacture of gas sensors, e-nose devices, and the change in operational condition (Bruins et al. 2013; Marco and Gutiérrez-Gálvez 2012; Zhang et al. 2011). Therefore, if the prediction model trained on one device (master device) is applied to other devices (slave devices), there will be a degradation in accuracy. However, it is often impractical to collect a set of labeled gas samples with each device to train prediction models,

D. Zhang et al., *Breath Analysis for Medical Applications*,
DOI 10.1007/978-981-10-4322-2_6

especially when the cost of sample collection is high. This problem limits the popularization of e-noses.

In order to make prediction models more applicable on slave devices, researchers have presented various calibration transfer methods. Many of them were originally proposed for spectroscopic data (Feudale et al. 2002; Park et al. 2001; Wang et al. 1991), but can also be applied to e-noses. This is because spectrometers generate 1D signals similar to e-noses, and also suffer from the instrumental variation problem. There are three typical ways of calibration transfer (Feudale et al. 2002; Marco and Gutiérrez-Gálvez 2012): transforming the data from the slave device to match the master one; updating the prediction model of the master device according to the slave data; and transforming the predicted values of the slave data. Transfer samples are often used in the algorithms. They usually consist of a group of standard gases which are reproducible and easy to acquire. The mapping information between devices can be obtained by analyzing the correspondence relationship between transfer sample groups.

In the field of e-noses, focuses have been paid on the first way (Balaban et al. 2000; Deshmukh et al. 2014; Marco and Gutiérrez-Gálvez 2012; Tomic et al. 2002; Zhang et al. 2011, 2013), since it is feasible in most situations and easy to implement. These kinds of methods are also known as variable standardization methods, which essentially deal with a regression problem. Common categories include univariate directstandardization (UDS), direct standardization (DS), and piecewise direct standardization (PDS), which differ mainly in the number of input variables. They can also be viewed as estimating a transformation matrix with different nonzero off-diagonal element constraints. Regression algorithms such as robust fitting (Deshmukh et al. 2014; Zhang et al. 2011), artificial neural network (ANN) (Balaban et al. 2000; Zhang et al. 2013), partial least squares (PLS) (Tomic et al. 2002), ordinary least square (OLS), and principal component regression (PCR) (Park et al. 2001) have been studied. Besides, in Peng et al. (2011), standardization was performed on a subspace obtained by spectral regression. The method is better than DS when the number of transfer samples is not less than 20. A calibration transfer approach based on alternating trilinear decomposition (ATLD) was proposed in Liu et al. (2014). With the method, the correction coefficients of multiple devices can be simultaneously derived. But the method may only be suitable when the changes between devices are restricted to relative intensity.

In the widely used PDS method, one variable in the master device is fitted by a group of variables around the corresponding variable in the slave device. All input variables are given the same weight (Wang et al. 1991). However, it is intuitive that the variables nearer to the corresponding variable should receive higher weights than the farther ones. With the constraint of the feature weights, the regression algorithm can be more stable. So we propose windowed piecewise direct standardization (WPDS) in this chapter, which allows us to give different weights to the input variables by assigning different penalty parameters. Experimental results show that WPDS outperforms UDS, PDS, and DS in the sense of test standardization error (the difference between standardized slave variables and the master variables) and prediction accuracy.

In current literatures, variable standardization and prediction model training are always considered separately. The transfer ability of prediction models is improved simply by minimizing the standardization error (SE). Nevertheless, we find that by incorporating some prior information obtained from standardization into the prediction models, the prediction accuracy of the slave data can also be enhanced (Yan and Zhang 2015). We call the strategy standardization-error-based model improvement (SEMI). The main idea is to make the models rely more on stable variables which have smaller SE. The strategy is combined with four popular prediction algorithms, i.e., logistic regression, support vector machine, ridge regression, and support vector regression. A weighted regularization term is included in the objective function of each algorithm. We impose larger penalty on the variables with larger SE, so as to reduce the weights of these variables in the trained model. Therefore, the model will be less sensitive to these unstable variables and have better transfer ability.

Calibration transfer is crucial in the application of clinical analysis because samples from patients are rather hard to collect. In this chapter, we will conduct experiments on a portable e-nose specialized for breath analysis (Yan et al. 2014; Yan and Zhang 2014), which will be introduced in Chap. 14. It achieves disease screening and monitoring through analyzing the biomarkers in breath, such as acetone, hydrogen, and ammonia. Three e-noses of this model are adopted to collect a gas sample dataset. Six pure chemical samples are chosen as transfer samples for variable standardization. Several prediction tasks are designed to evaluate the transfer ability of the models, includingclassification or regression of pure chemicals or breath samples. Experimental results show that the SEMI strategy consistently enhances the accuracy of the master model applied to standardized slave data, especially when the inconsistency between devices is large. In addition to its efficacy, SEMI can be easily extended to other prediction algorithms.

The chapter is organized as follows. Section 6.2 describes WPDS and SEMI in detail. Section 6.3 introduces the experimental configurations, including the e-nose module, the dataset, and the related data analysis procedure. Section 6.4 presents the results of the calibration transfer experiments and provides some discussion. Section 6.5 summarizes the chapter.

6.2 Design of Methods

The calibration transfer process in the chapter consists of two steps: (1) developing standardization models with WPDS to standardize the data from the slave device; (2) developing prediction models with the SEMI strategy to predict the standardized slave data. This section will describe the steps in detail.

6.2.1 Windowed Piecewise Direct Standardization (WPDS)

The objective of standardization is to model the difference between two devices and reduce it. To achieve this, a set of transfer samples are measured on both devices. Then, regression models are built based on the correspondence between the transfer samples, so as to transform each slave variable to match the corresponding master variable. Finally, the prediction models trained on master data can be applied to the standardized slave data and get a better accuracy.

In the simple univariate directstandardization (UDS) approach (Balaban et al. 2000), each master variable is fitted using the corresponding slave variable and obtain two coefficients: the slope and the intercept. When the device variation is large, the univariate approach cannot always model the master variables well. The direct standardization(DS) proposed in Wang et al. (1991) is a multivariate approach which fits each master variable using all slave variables. Some researchers (Balaban et al. 2000) reported that DS is better than UDS. However, when the number of variables is large and the number of transfer samples is limited, DS is prone to overfitting (Feudale et al. 2002). A trade-off approach between UDS and DS is piecewise direct standardization (PDS) (Wang et al. 1991). In PDS, each master variable is related to only a subset of slave variables, for example, neighboring wavelengths in near-infrared spectroscopy data. PDS is one of the most widely used standardization approaches in spectroscopic area. Its superiority is attributed to its local character and multivariate nature (Feudale et al. 2002). But it has not been well explored for e-nose data partially due to the feature extractionmethods used in previous studies. Commonly, only one steady response feature is extracted from each sensor before standardization, hence there are no "neighboring" variables. If multiple transient features are extracted from each sensors response curve, neighboring variables can be defined and PDS can be applied.

In PDS, the input variables are regarded as equally important. Intuitively, when fitting the kth master variable, the kth slave variable should be more important than the variables at some distance from k. Therefore, we propose a windowed PDS (WPDS) which gives different weights to input variables by assigning different penalty parameters in regression. The penalty parameters can be seen as a window around k. By changing the size and shape of the window, we can change the scope and weights of the input variables. Consequently, the original PDS turns out to be a special case of WPDS with a rectangular window (constant weights).

We adopt generalized ridge regression as the algorithm inside WPDS. Ridge regression (Hoerl and Kennard 1970; Hastie et al. 2009) is a well-known shrinkage method for linear regression. Suppose the problem is to find proper $\boldsymbol{\beta}$ and β_0 in

$$y^{(i)} = \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}^{(i)} + \beta_0 + \varepsilon^{(i)}, i = 1, 2, \ldots, N, \tag{6.1}$$

where y is the output variable; superscript (i) indicates the ith sample; N is the number of samples; $\boldsymbol{x}^{(i)} \in \mathbf{R}^M$ is a vector of M input variables; M is the window length of WPDS. $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_M]^{\mathrm{T}} \in \mathbf{R}^M$ and $\beta_0 \in \mathbf{R}$ are the regression coefficients to be estimated; and $\varepsilon^{(i)} \in \mathbf{R}$ is an error term. The problem formulation of ridge regression is

$$\min_{\boldsymbol{\beta}, \beta_0} \left\{ \sum_{i=1}^{N} (\boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{x}^{(i)} + \beta_0 - y^{(i)})^2 + \lambda \sum_{j=1}^{M} \beta_j^2 \right\}. \tag{6.2}$$

The second term is a regularization term, which imposes a penalty on the coefficients' size. It forces the coefficients to shrink towards zero. $\lambda \geq 0$ is a parameter controlling the amount of shrinkage. The larger λ, the greater the shrinkage. Note that the intercept β_0 is not included in the regularization term (Hastie et al. 2009).

In multivariate standardization approaches, the number of samples is limited and the input variables are often correlated. In such cases, the estimated coefficients can have large variance. Ridge regression is particularly useful in such problems. By introducing the regularization term, ridge regression reduces the variance and stabilizes the regression model (Hastie et al. 2009). The ridge regression model can be generalized by adding a penalty parameter w for each coefficient in the regularization term:

$$\min_{\boldsymbol{\beta}, \beta_0} \left\{ \sum_{i=1}^{N} (\boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{x}^{(i)} + \beta_0 - y^{(i)})^2 + \lambda \sum_{j=1}^{M} (w_j \beta_j)^2 \right\}. \tag{6.3}$$

Large w_j brings large penalty to β_j and shrinks it. As a result, the regression model will depend less on variable j. A triangular window is designed for w_j. Suppose the window length $M = 2L + 1$, which means that when fitting the kth master variable, from the $(k - L)$th to the $(k + L)$th slave variables will form the input vector $\boldsymbol{x}$. So x_{L+1} is the slave variable which corresponds to the master variable to be fitted. As shown in Fig. 6.1a, the penalty parameter w_{L+1} is set to be 0, and w grows linearly to 1 for the L adjacent variables on both sides. This means that we impose the minimum penalty to β_{L+1}; and the further the index is away from $L + 1$, the larger the penalty. When $k \leq L$ or $k > P - L$ (P is the total number of variables), the window should be truncated on one side since the number of input variables is less than $2L + 1$, as shown in Fig. 6.1b. It is worth noting that the proper shape of the window is dependent on the feature extraction method. The triangular window is suitable when the transient feature is utilized, either in time domain (the points on the sensors' response curves are used as features) or frequency domain (the Fourier transform coefficients of the response curves are used). In such cases, neighboring variables are closely related. If other feature extraction methods are applied before standardization, the window should be adjusted based on the relationships among the input variables.

To solve Eq. 6.3, the input and output variables are first centered to eliminate the intercept term β_0 (Hastie et al. 2009). Then it can be derived that

$$\boldsymbol{\beta} = \left(X^{\mathrm{T}} X + \lambda W\right)^{-1} X^{\mathrm{T}} \boldsymbol{y}, W = \mathrm{diag}(w_1^2, \ldots, w_M^2), \tag{6.4}$$

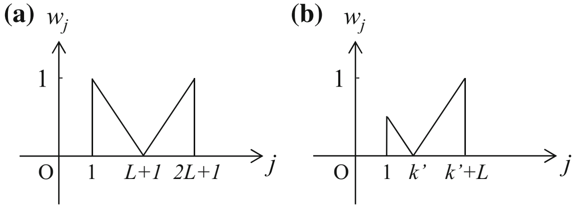


Fig. 6.1 Illustration of the triangular window used in WPDS. The x-axis is the index of the variables. The y-axis is the penalty parameter. Plot (**a**) shows a complete window; Plot (**b**) shows an example of a truncated window when the number of input variables is less than $2L + 1$ (there are less than L variables on either the *left* or the *right* side of the kth variable)

where $X = [\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}]^{\mathrm{T}} \in \mathbf{R}^{N \times M}$ and $\boldsymbol{y} = [y^{(1)}, \ldots, y^{(N)}]^{\mathrm{T}} \in \mathbf{R}^{N}$. In practice, we need to estimate a $\boldsymbol{\beta}$ for each master variable. X is from the slave transfer samples and $\boldsymbol{y}$ is from the master transfer samples.

6.2.2 Standardization-Error-Based Model Improvement (SEMI)

Compared with other calibration transfer methods, device standardization is widely used since it is easy to implement and feasible in most problems and prediction algorithms. Existing literatures have been treating standardization and prediction model training as two separate steps. In the latter step, the prediction algorithm focuses on fitting the master data, without trying to make the model adapt well to standardized slave data. In this section, we propose a strategy to connect these two steps. By incorporating some prior information obtained from standardization, the prediction models can be improved and have better transfer ability.

To enhance the prediction accuracy on standardized slave data, a direct way is to improve the standardization algorithm and minimize the standardization error (SE) of each variable. However, in practice, some sensors' responses are more stable than others when measuring certain gases. The signal-to-noise ratios of different feature extraction methods are also different. Therefore, some variables will have less inter-device variance after standardization, thus have less SE. If the prediction model can rely more on these variables, its performance on standardized slave data will be better.

We define the SE of a variable x to be the root mean square deviation (RMSD) between the standardized slave variable x^{SS} and the master variable x^{Ma}:

$$\mathrm{SE}(x) = \mathrm{RMSD}(x) = \sqrt{\frac{1}{N_t} \sum_{i=1}^{N_t} \left(x_i^{\mathrm{Ma}} - x_i^{\mathrm{SS}}\right)^2}, \qquad (6.5)$$

where the superscript "Ma" stands for "master" and "SS" stands for "standardized slave". N_t is the number of transfer samples.

To make the models more dependent on variables with less SE, we modify the objective functions of prediction algorithms to include a weighted regularization term. The method is based on Tikhonov regularization (Tikhonov and Arsenin 1977; Kalivas et al. 2009). The regularization term has the form $\lambda \sum_{j=1}^{M} (w_j \beta_j)^2$, which is similar to the one in generalized ridge regression for WPDS. β_j is the coefficient of variable j to be estimated in the model; w_j is set to be the SE of variable j; λ is a positive constant controlling the weight given to the term. By minimizing the objective function with this regularization term, the variables with larger SE are given larger penalty parameters. Hence, the coefficients (β_j) for these variables in the estimated model will be shrunk, and the model will be less dependent on them. We also tried to replace the L2-norm penalty with L1-norm ($\lambda \sum_{j=1}^{M} \left| w_j \beta_j \right|$). It can generate sparse models with some coefficients being exactly zero. But the estimated models are not stable (Park and Hastie 2007) and show worse transfer ability than the L2-norm one.

This standardization-error-based model improvement (SEMI) strategy is applicable on various prediction algorithms. We combine it with four popular classification or regression algorithms, i.e., logistic regression, support vector machine, ridge regression, and support vector regression. For logistic regression, a regularization term is added into its objective function. In the objective functions of the other three algorithms, a regularization term already exists. So we turn the term into a weighted version. These algorithms will be briefly reviewed in the following subsections. Hereinafter each input vector $\boldsymbol{x}$ is assumed to contain a constant component ($x_0 = 1$), so the intercept coefficient β_0 is merged into the coefficient vector $\boldsymbol{\beta}$.

6.2.2.1 Logistic Regression-Based Classification

In binary-class cases, the decision function of logistic regression (LR) is a sigmoid function

$$h_{\boldsymbol{\beta}}(\boldsymbol{x}) = \text{sigmoid}(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x})}. \tag{6.6}$$

A test sample $\boldsymbol{x}$ is classified into the positive class if $h_{\boldsymbol{\beta}}(\boldsymbol{x}) \geq 0.5$. Since $h_{\boldsymbol{\beta}}(\boldsymbol{x})$ is between 0 and 1, it can be viewed as the probability of the test sample belonging to the positive class, which is also the characteristic of LR. The coefficients $\boldsymbol{\beta}$ can be learned by maximum likelihood estimation, which seeks to maximize the log-likelihood function (Hastie et al. 2009):

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{N} y^{(i)} \log h_{\boldsymbol{\beta}}(\boldsymbol{x}^{(i)}) + (1 - y^{(i)}) \log(1 - h_{\boldsymbol{\beta}}(\boldsymbol{x}^{(i)})), \tag{6.7}$$

where $y^{(i)} = 1$ if $\boldsymbol{x}^{(i)}$ belongs to the positive class and 0 otherwise. Under the SEMI strategy, the LR problem can be formulated as

$$\min_{\boldsymbol{\beta}} \left\{ -\ell(\boldsymbol{\beta}) + \frac{\lambda}{2} \sum_{j=1}^{M} (\mathrm{SE}_j \cdot \beta_j)^2 \right\}, \tag{6.8}$$

which can be solved using numerical optimization methods. In K-class cases, K separate LR models are trained using the one-versus-all strategy and $\boldsymbol{x}$ is classified into the class whose decision function has the largest value.

6.2.2.2 Support Vector Machine-Based Classification

Support vector machine (SVM) is among the most popular techniques for classification. The main idea of the algorithm is finding a hyperplane to separate the training samples with a maximum margin. It has been proved to generalize well on test samples (Burges 1998). The detailed introduction of the algorithm can be found in (Burges 1998). The objective function of SVM has the form "loss + penalty" (Hastie et al. 2009), so we modify the penalty term in an L2-loss SVM model and formulate the problem of SVM + SEMI as:

$$\begin{gathered} \min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \xi_i^2 + \frac{\lambda}{2} \sum_{j=1}^{M} (\mathrm{SE}_j \cdot \beta_j)^2 \\ \text{s.t. } y^{(i)} \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{x}^{(i)} \geq 1 - \xi_i, \; \xi_i \geq 0, \; \forall i, \end{gathered} \tag{6.9}$$

where $y^{(i)} \in \{+1, -1\}$. The trust region Newton method in LIBLINEAR (Fan et al. 2008) is used in this chapter to solve Eq. 6.9 in the primal form. In K-class cases, the one-versus-all strategy is used.

6.2.2.3 Ridge Regression-Based Regression

Ridge regression generates regression models which are more robust than ordinary least squares, especially in ill-conditioned problems (Marco and Gutiérrez-Gálvez 2012; Hastie et al. 2009). Its formulation under the SEMI strategy and solution are similar to Eqs. 6.3 and 6.4:

$$\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} (\boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{x}^{(i)} - y^{(i)})^2 + \lambda \sum_{j=1}^{M} (\mathrm{SE}_j \cdot \beta_j)^2 \right\}. \tag{6.10}$$

$$\boldsymbol{\beta} = \left(X^{\mathrm{T}} X + \lambda W \right)^{-1} X^{\mathrm{T}} Y, W = \mathrm{diag}(0, \mathrm{SE}_1^2, \ldots, \mathrm{SE}_M^2). \tag{6.11}$$

The intercept coefficient β_0 is not penalized, so the first element of W is zero.

6.2.2.4 Support Vector Regression-Based Regression

Support vector regression (SVR) is a frequently used regression algorithm with good generalization ability (Marco and Gutiérrez-Gálvez 2012). The details of the algorithm can be found in (Smola and Schölkopf 2004). Similar to SVM, we modify the penalty term in an L2-loss SVR model and formulate the problem of SVR + SEMI as:

$$\begin{aligned} \min_{\boldsymbol{\beta}} \sum_{i=1}^{N} (\xi_i^2 + \xi_i^{*2}) + \frac{\lambda}{2} \sum_{j=1}^{M} (\mathrm{SE}_j \cdot \beta_j)^2 \\ \text{s.t. } y^{(i)} - \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{x}^{(i)} \le \varepsilon + \xi_i \\ \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{x}^{(i)} - y^{(i)} \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0, \ \forall i. \end{aligned} \tag{6.12}$$

The trust region Newton method in LIBLINEAR (Fan et al. 2008) is used in this chapter to solve Eq. 6.12 in the primal form.

6.3 Experimental Details

In order to assess the proposed methods, we used three e-noses to collect a gas sample dataset. In this section, the e-nose module, the composition of the dataset, and the related data analysis procedure will be described.

6.3.1 E-nose Module

The e-nose to be introduced in Chap. 14 is equipped with an array of 11 sensors. Among them, we focus on 9 metal oxide semiconductor (MOS) sensors for the detection of volatile organic compounds (VOCs). They have diverse sensitivity spectrums. For example, some sensors (TGS826, TGS2602, GSBT11) are proved to be sensitive to acetone, which is a biomarker of diabetes (Deng et al. 2004; Turner et al. 2009); some (TGS2610-D00, SP3S-AQ2, TGS822) are more sensitive to hydrogen, which has been used to detect functional intestinal disorders (Eisenmann et al. 2008); some (TGS826, TGS2602) are sensitive to ammonia, which is associated with renal failure (Davies et al. 1997). In our experiments, we found that GSBT11 is unstable and has a very large SE, so we discarded the sensor. The following analysis is performed on the remaining 8 MOS sensors. Examples of the signals measured using three e-noses of the same model are shown in Fig. 6.2. The inconsistency of the devices can be easily observed.

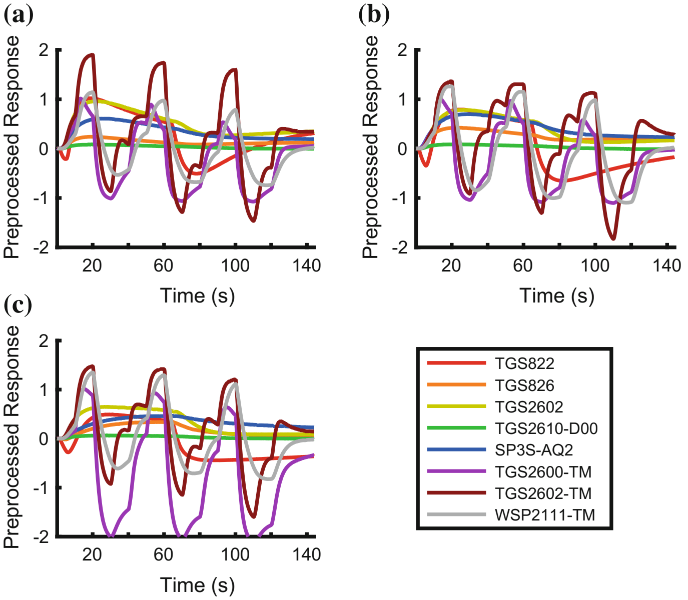


Fig. 6.2 Comparison of the baseline-removed responses of the same breath sample measured using the three e-noses. **a**–**c** correspond to devices 1–3, respectively. The legend shows the names of the sensors

6.3.2 Dataset

The e-noses were utilized to measure 7 groups of gas samples. Three groups of them are pure chemicals of different concentrations; one group is normal breath exhaled by healthy people; and the other three groups are chemicals blended with normal breath. Three kinds of chemicals were considered, namely acetone, hydrogen, and ammonia, since they are typical breath biomarkers of certain diseases. In the last three groups, the chemicals were diluted with exhaled breath instead of clean air. The aim is to make the prediction tasks more challenging, since normal breath already contains the three chemicals of different concentrations (Deng et al. 2004; Eisenmann et al. 2008; de Lacy Costello et al. 2008), as well as many other interfering VOCs (Phillips et al. 1999). They are also used to simulate the breath samples of patients. Details of the dataset are listed in Table 6.1. The ranges of concentrations of the chemicals were determined by their typical concentrations in breath (Deng et al. 2004; Turner et al. 2009; Eisenmann et al. 2008; Davies et al. 1997). A visualization of the data in groups 1–3 can be found in Fig. 6.3, which demonstrates the sensitivity characteristics of the sensors.

All 248 samples were measured using each of the three e-noses. The measurement procedure of each sample can be found in Chap. 14. Six pure chemical samples were empirically chosen as the transfer samples in this study. We chose two samples of

Table 6.1 Composition of the dataset

Group	Gas sample	#Samples	Notes
1	Acetone	16	8 concentrations (0.1, 0.2, 0.5, 1, 2, 5, 10, 20 ppm), 2 samples per concentration
2	Hydrogen	18	9 concentrations (0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50 ppm), 2 samples per concentration
3	Ammonia	14	7 concentrations (0.1, 0.2, 0.5, 1, 2, 5, 10 ppm), 2 samples per concentration
4	Normal breath	80	Collected from 24 healthy subjects in different days, 2–6 samples per subject
5	Acetone + breath	40	8 concentrations (0, 0.2, 0.3, 0.7, 1.7, 3.3, 5.0, 6.7 ppm), 5 samples per concentration, mixed with normal breath from 5 subjects
6	Hydrogen + breath	40	8 concentrations (0, 0.4, 0.7, 1.7, 4.2, 8.3, 12.5, 16.7 ppm), 5 samples per concentration, mixed with normal breath from 5 subjects
7	Ammonia + breath	40	8 concentrations (0, 0.3, 0.8, 1.7, 2.5, 3.3, 4.2, 5.0 ppm), 5 samples per concentration, mixed with normal breath from 5 subjects

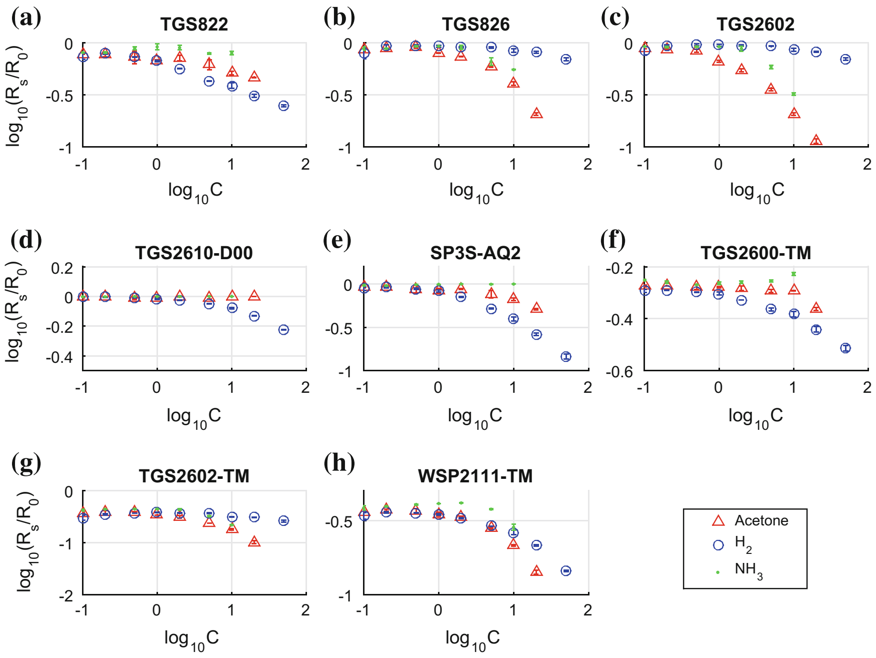


Fig. 6.3 Responses of the eight sensors to three gases. The *x*-axis is the logarithm of the concentration *C* of the target gas. The *y*-axis is the sensor resistance ratio R_S/R_0, where R_S is the sensor's resistance in the target gas and R_0 is that in fresh air. Each data point is the average of two tests of the same gas in the same concentration. The error bars show the standard deviation

each chemical, one with a low concentration and one with a high concentration, namely acetone (1 and 10 ppm), hydrogen (1 and 50 ppm), and ammonia (1 and 10 ppm). Experiments showed that with the proposed algorithms, these transfer samples can be used to standardize the breath samples with an acceptable accuracy.

6.3.3 Preprocessing and Feature Extraction

After a digitized gas sample is obtained, the baseline values are subtracted from the sensor responses to remove baseline drift. Then, discrete Fourier transform (DFT) features are extracted from the response curves. Compared with the traditional steady response feature, DFT contains the transient information of a curve, which will be helpful in prediction. Additionally, the energy of the typical response curve of a gas sensor lies mostly in low frequencies, so we can compress the curve using only a small number of variables in the frequency domain. In this chapter, we extract the modulus of the first 30 DFT coefficients from each curve, which has 1152 data points in the time domain. As a result, each feature vector contains 30×8 sensors $= 240$ variables.

The variables are further normalized with standard normal variate (SNV) to eliminate additive and multiplicative variations among different devices (Marco and Gutiérrez-Gálvez 2012). For each device, the mean and standard deviation values of each variable are calculated from the transfer samples. Then all samples from the device are centered and scaled by these values. This normalization step can be viewed as an initial standardization of the variables.

6.3.4 Data Analysis Procedure

The entire data analysis procedure for device standardization and prediction is shown in Fig. 6.4. After preprocessing and feature extraction, the gas samples from master and slave devices are represented by matrices X_{Ma} and X_{Sl}, respectively. The transfer samples from both devices are adopted to build standardization models. Then X_{Sl} is standardized using the models and the standardized slave sample matrix X_{SS} is obtained. Next, training samples from X_{Ma} are used to develop master prediction models. When the SEMI strategy is used, the standardization error will also join the model training. The trained model is applied to predict the test samples in X_{Ma} and X_{SS}, obtaining the accuracy $\mathrm{Acc}_{\mathrm{Ma}}$ and $\mathrm{Acc}_{\mathrm{SS}}$, respectively. We can also get $\mathrm{Acc}_{\mathrm{Sl}}$ when the training and test samples are both from X_{Sl}. Generally, $\mathrm{Acc}_{\mathrm{Ma}}$ and $\mathrm{Acc}_{\mathrm{Sl}}$ should be better than $\mathrm{Acc}_{\mathrm{SS}}$, since their training and test samples are from the same devices. The final goal is to enhance $\mathrm{Acc}_{\mathrm{SS}}$ to approach $\mathrm{Acc}_{\mathrm{Sl}}$, without degrading $\mathrm{Acc}_{\mathrm{Ma}}$. Note that for classification tasks, Acc is the classification precision (the larger the better); For regression tasks, Acc is the root mean square error (RMSE, the smaller the better).

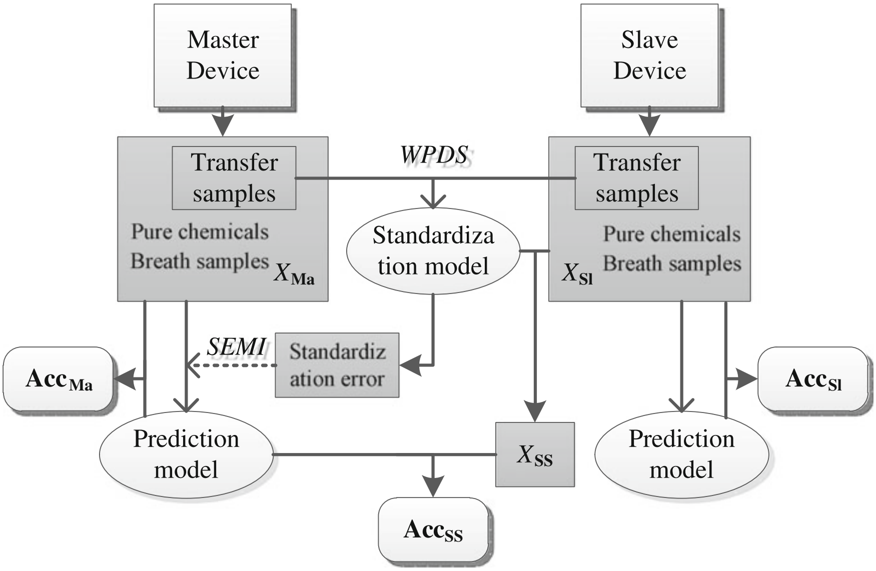


Fig. 6.4 Flowchart of the device standardization and prediction procedure. An unfilled *arrow* starts from an input, and a filled *arrow* points to an output. X_{Ma}, X_{Sl}, and X_{SS} represent the sample matrices of master, slave, and standardized slave device, respectively. Acc stands for prediction accuracy

Note that in the standardization step, if multivariate standardization methods are used, the input variables and the output variable should come from the same sensor. In the prediction step, four classification tasks and six regression tasks are executed to evaluate the algorithms. The classification tasks include a multi-class task (distinguishing the three chemicals) and three binary-class tasks (distinguishing normal breath samples from each group of blended ones). In each task, we randomly choose an equal number of samples from the each corresponding data group (class). Half of them are randomly chosen as training samples, leaving the rest as test ones. The two classification algorithms in Sect. 6.2.2 are applied. The process is repeated 20 times and an average accuracy is computed for each task. The regression tasks aim at predicting the concentration of chemicals in each data group except group 4. Within each group, the leave-one-out cross validation strategy is used. Ridge regression and support vector regression introduced in Sect. 6.2.2 are applied.

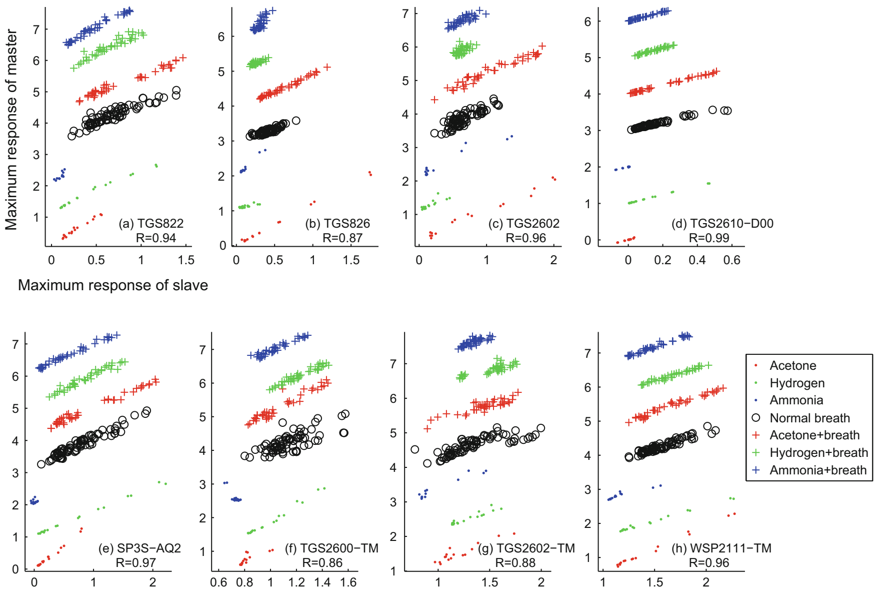


Fig. 6.5 Illustration of the relationship between the maximum responses of two devices. Plot (**a**)–(**i**) correspond to the eight sensors. The *x*-axis and *y*-axis correspond to the responses of the slave (device 3) and master device (device 1), respectively. Note that there is a manual shift of 1 on the *y*-axis between different data groups. The *R* values are the correlation coefficients of *x* and *y* (without shift) for the samples in all groups

6.4 Results and Discussion

6.4.1 Standardization

Among the three e-noses, we assigned device 1 as the master device, devices 2 and 3 as the slave ones. Figure 6.5 illustrates the relationship between the maximum responses of two devices. It is clear that the maximum responses of the two devices have a linear relationship.

Experiments were made to compare the performance of different standardization methods, including univariate direct standardization (UDS), direct standardization (DS),piecewise direct standardization (PDS), and the proposed windowed PDS (WPDS). The results are displayed in Table 6.2. In the method "only SNV," the slave variables are normalized with SNV but not standardized. For UDS, the robust fitting algorithm in (Zhang et al. 2011) was used. For DS and PDS, different regression algorithms were tested, such as OLS, PCR, PLS, and ridge regression (Eq. 6.2). It was found that generally the order of performance is: ridge > PLS > PCR > OLS. So only the results of ridge regression are listed in Table 6.2 for clarity. For DS, all 30 variables of each sensor in slave device were used to fit each master variable. The

Table 6.2 Test standardization error and prediction accuracy of different standardization methods. Bold values indicate the best results

Method	Device 2 as slave				Device 3 as slave			
	Test SE		Acc$_{\mathrm{SS}}$		Test SE		Acc$_{\mathrm{SS}}$	
	Chemical	Breath	Classifi. acc.	Regress. RMSE	Chemical	Breath	Classifi. acc.	Regress. RMSE
Only SNV	0.3668	0.6192	0.8125	3.4676	0.5492	0.6897	0.7045	3.6586
UDS	0.1905	0.4034	0.8762	1.8711	0.3776	0.5508	0.6159	2.7165
DS	0.1346	0.3139	0.7049	2.5188	0.2317	0.4280	0.6357	3.0510
PDS	0.1280	0.2848	0.7458	2.1887	0.2460	0.4235	0.6474	3.1444
PDS (neural)	0.7433	0.9093	0.6825	4.6332	0.8939	1.0148	0.6145	5.1032
WPDS	**0.0979**	**0.2713**	**0.8945**	**1.4658**	**0.2255**	**0.4189**	**0.7348**	**2.6188**

window length of PDS and WPDS was 15. The shrinkage parameter λ in Eqs. 6.2 and 6.3 was set to 10. The neural method was also tested with the same parameter setting as in Zhang et al. (2013).

In Table 6.2, the standardization error (SE) is not computed on transfer samples as in Eq. 6.5. Instead, it is computed on all chemical (groups 1–3) or breath (groups 4–7) samples and averaged over all variables. So we denote it as “test SE”. The Acc$_{\mathrm{SS}}$ is also listed, including the average accuracy of the four classification tasks and the average RMSE of the six regression tasks. The prediction algorithms are logistic regression and ridge regression without SEMI. WPDS has the lowest SE and best prediction accuracy. PDS and DS have the second lowest SE. It is found that good Acc$_{\mathrm{SS}}$ does not always occur together with low SE. It is possibly because high Acc$_{\mathrm{SS}}$ requires the variables important for prediction to have low SE. The displayed test SE, however, is averaged over all variables. So it does not strictly reflect the prediction results. This observation was also found in Park et al. (2001). The performance of PDS (neural) is not good, for the neural network algorithm in Zhang et al. (2013) has nonlinear hidden neurons and is easy to overfit the transfer samples, especially when the number of input variables is large. The triangular window adopted in WPDS brings effective prior information to the problem, which makes it outperform other methods.

It can be observed that the breath samples have larger SE than the chemical samples. Human breath is a complex mixture of gases. An average normal breath contains 204.2 VOCs (Phillips et al. 1999). It is not very similar to the transfer samples used in the chapter. But since human breath is not reproducible, it cannot be directly used as transfer sample. Experiments in this chapter show the possibility of using three chemicals to standardize breath samples. Further study is needed on the selection of transfer samples for breath analysis systems.

Figure 6.6 is a visual illustration of the effect of the standardization process. Linear discriminant analysis (LDA) is used to reduce the master samples in three classes

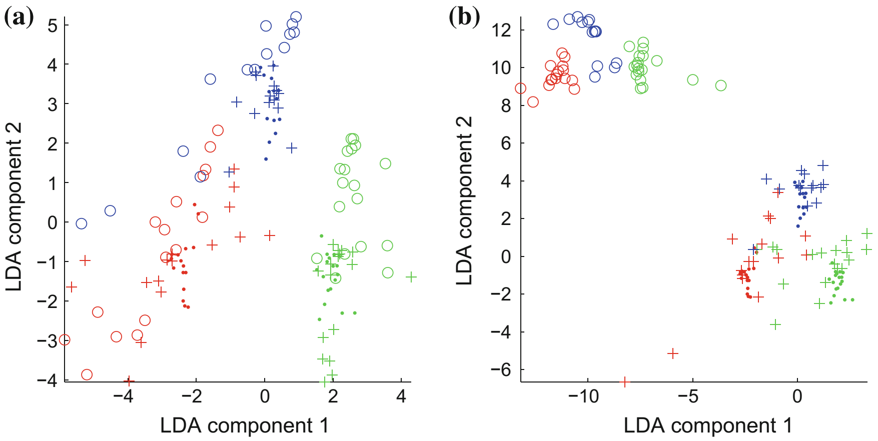


Fig. 6.6 Linear discriminant analysis (LDA) plot of the samples. The dots, circles, and plus signs represent the samples from the master device, the slave device before standardization, and the slave device after standardization, respectively. The red, green, and blue colors represent samples in three classes (acetone, hydrogen, and ammonia). The slave devices are devices 2 and 3 in plot (**a**) and (**b**), respectively

to a two-dimension subspace. Then the slave samples before and after standardization (WPDS) were projected onto the subspace. In Fig. 6.6, it is found that the raw samples from device 1 and 2 are similar. Meanwhile, the difference between device 1 and 3 is larger. It is probably because the gas route of device 3 is a bit different from the other two devices. Its sensor array is also more aged. These factors enlarge the difference between it and the master device. The standardization method reduces the difference, especially for device 3 (see Fig. 6.6b). It can make the prediction model trained on master data be applied to the standardized slave data without much accuracy loss.

6.4.2 *Prediction*

The performance of the SEMI strategy is evaluated in this section. In the strategy, the SEs in Eq. 6.5 are used as penalty parameters in various prediction models. In Fig. 6.7a, the SEs of the 240 variables are shown. Their mean is scaled to 1. Note that variables 1–30 are extracted from sensor 1, variables 31–60 are extracted from sensor 2, and so on. For each sensor, the first variable is the DFT feature of zero frequency and the frequency rises as the feature index increases. It can be seen that SEs of different sensors are different. Besides, the high frequency features generally have higher SEs. Figure 6.7b, c show the coefficients of a logistic regression model trained to distinguish three chemicals. Figure 6.7b shows the model under the SEMI strategy.

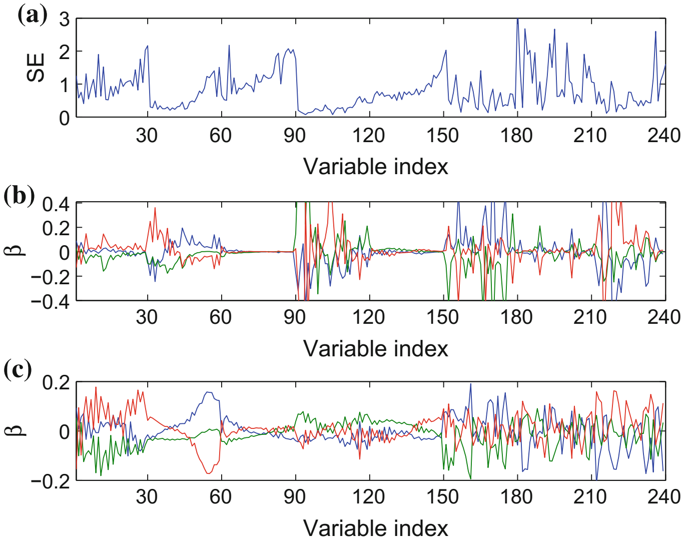


Fig. 6.7 **a** Standardization error of the variables. **b** The classification model trained by logistic regression with SEMI. The three *curves* are the coefficients of the decision functions of three chemicals. **c** The classification model trained by logistic regression without SEMI (the penalty parameters are all set to 1)

Compared with Fig. 6.7c, its coefficients have larger magnitude if the corresponding SE is small (e.g., variables 90–120).

The classification accuracies based on different configurations are demonstrated in Fig. 6.8. Both the two slave devices and the two classification algorithms are investigated. The shrinkage parameter λ is varied. When the master models are applied to slave data which is normalized but not standardized, the performance is not good. The accuracy is enhanced after standardization with WPDS (the curve with triangles). The SEMI strategy further enhances it and makes it closer to Acc_{Sl}, which is obtained by applying the slave model on slave data. The value of λ is related to Acc_{SS} with SEMI. If λ is too small, the weight of the regularization term is not enough to penalize the unstable variables. If λ is too large, the prediction algorithms will be too focused on penalization and cannot fit the training samples well. The results of the regression algorithms (Fig. 6.9) show similar trends.

Apart from the DFT feature, the traditional steady response feature was also tested. The SEMI strategy can also enhance its performance. But its overall performance is not as good as the DFT feature. It is probably because steady responses provide much less information than the DFT feature. In addition, SEMI is expected to work better if the dimension of the feature vector is relatively high. In such situations, the variables contain redundant information. When the coefficients of some variables are shrunk, the other variables can still provide sufficient information for prediction.

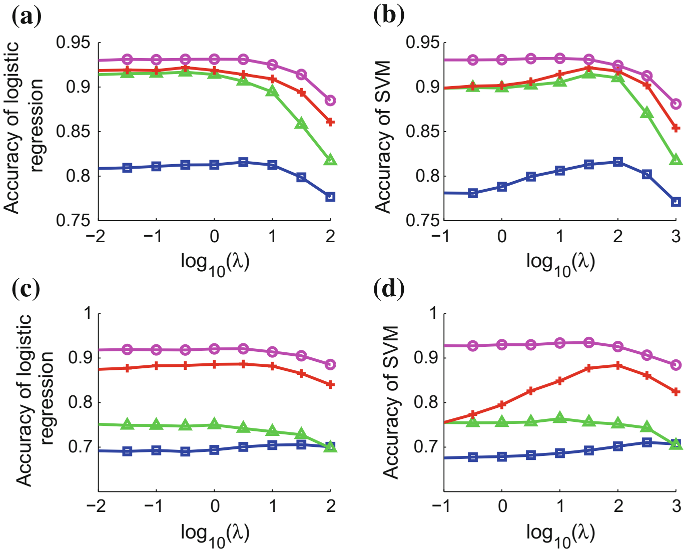


Fig. 6.8 Average accuracy of the classification tasks. Rows 1-2 correspond to devices 2 and 3, respectively. Columns 1-2 correspond to logistic regression and SVM, respectively. *Square* the master model is applied to slave data which is normalized but not standardized; *triangle* $\mathrm{Acc}_{\mathrm{SS}}$ without SEMI (the penalty parameters are all set to 1); *plus* $\mathrm{Acc}_{\mathrm{SS}}$ with SEMI (the mean of the penalty parameters (SEs) are scaled to 1); *circle* $\mathrm{Acc}_{\mathrm{Sl}}$

Detailed results for every prediction algorithm are listed in Tables 6.3, 6.4, 6.5, and 6.6. λ was searched among $\{10^{-4}, 10^{-3.5}, \ldots, 10^{4}\}$ for each result to find the best one. For device 2, the improvement of $\mathrm{Acc}_{\mathrm{SS}}$ brought by WPDS is relatively large. For device 3, however, the effect of device standardization is not good enough. But SEMI significantly improves $\mathrm{Acc}_{\mathrm{SS}}$ in this case. This proves that SEMI is especially effective when the inconsistency between devices is large. Surprisingly, $\mathrm{Acc}_{\mathrm{Ma}}$ and $\mathrm{Acc}_{\mathrm{Sl}}$ can also be slightly improved by applying the SEMI strategy (including the SE weighted regularization term in the prediction models), which implies that SEMI introduces helpful information about the variables (such as their stability) to the prediction algorithms. Among the four prediction algorithms, logistic regression and SVM are comparable; ridge regression performs relatively better than SVR. So we further list the results of logistic regression and ridge regression on all prediction tasks with the best λ settings in Tables 6.7 and 6.8. It can be observed that the proposed methods have made $\mathrm{Acc}_{\mathrm{SS}}$ close to $\mathrm{Acc}_{\mathrm{Sl}}$. $\mathrm{Acc}_{\mathrm{SS}}$ is even better than $\mathrm{Acc}_{\mathrm{Sl}}$ in some classification and regression tasks due to the helpful information introduced by SEMI.

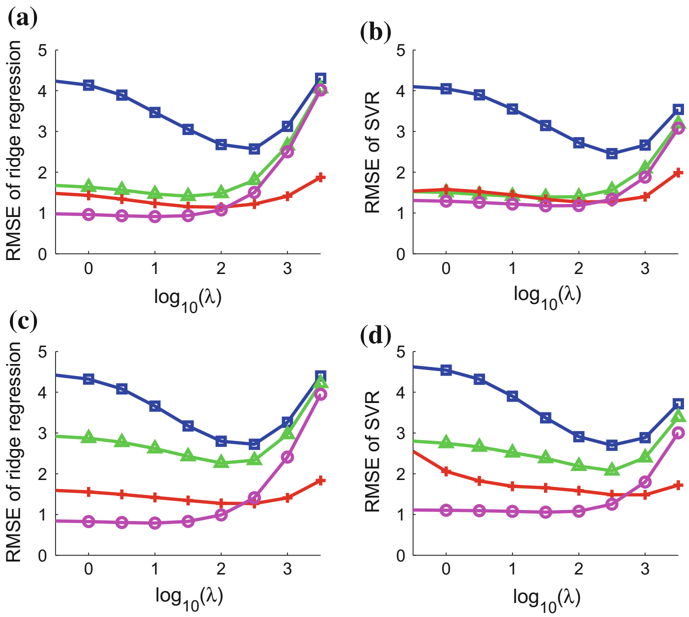


Fig. 6.9 Average RMSE of the regression tasks. Rows 1–2 correspond to devices 2 and 3, respectively. Columns 1–2 correspond to ridge regression and SVR, respectively. *Square* the master model is applied to slave data which is normalized but not standardized; *triangle* $\mathrm{Acc}_{\mathrm{SS}}$ without SEMI (the penalty parameters are all set to 1); *plus* $\mathrm{Acc}_{\mathrm{SS}}$ with SEMI (the mean of the penalty parameters (SEs) are scaled to 1); *circle* $\mathrm{Acc}_{\mathrm{Sl}}$

Table 6.3 Classification accuracy of logistic regression

Slave device	$\mathrm{Acc}_{\mathrm{Ma}}$		$\mathrm{Acc}_{\mathrm{Sl}}$		$\mathrm{Acc}_{\mathrm{SS}}$		
	Without SEMI	With SEMI	Without SEMI	With SEMI	Only SNV	WPDS	WPDS + SEMI
Device 2	0.9312	0.9324	0.9312	0.9410	0.8156	0.9167	0.9220
Device 3	0.9312	0.9336	0.9210	0.9314	0.7056	0.7532	0.8865

Table 6.4 Classification accuracy of SVM

Slave device	$\mathrm{Acc}_{\mathrm{Ma}}$		$\mathrm{Acc}_{\mathrm{Sl}}$		$\mathrm{Acc}_{\mathrm{SS}}$		
	Without SEMI	With SEMI	Without SEMI	With SEMI	Only SNV	WPDS	WPDS + SEMI
Device 2	0.9313	0.9331	0.9322	0.9395	0.8160	0.9144	0.9217
Device 3	0.9313	0.9343	0.9350	0.9364	0.7105	0.7637	0.8836

Table 6.5 Regression RMSE of ridge regression

Slave device	Acc_{Ma}		Acc_{Sl}		Acc_{SS}		
	Without SEMI	With SEMI	Without SEMI	With SEMI	Only SNV	WPDS	WPDS + SEMI
Device 2	0.8971	0.8810	0.9106	0.7716	2.5704	1.4119	1.1424
Device 3	0.8971	0.8012	0.7891	0.7873	2.7224	2.2629	1.2691

Table 6.6 Regression RMSE of SVR

Slave device	Acc_{Ma}		Acc_{Sl}		Acc_{SS}		
	Without SEMI	With SEMI	Without SEMI	With SEMI	Only SNV	WPDS	WPDS+ SEMI
Device 2	1.0522	1.0739	1.1748	1.0732	2.4545	1.3904	1.2690
Device 3	1.0494	1.0157	1.0559	1.0558	2.7004	2.0737	1.4824

Table 6.7 Accuracy of logistic regression on device 2 of all classification tasks (distinguishing the samples in different data groups)

	Group 1 versus 2 versus 3	4 versus 5	4 versus 6	4 versus 7	Average
Acc_{SS}: Only SNV	0.7286	0.8941	0.8824	0.7574	0.8156
Acc_{SS}: WPDS + SEMI	0.9762	0.8941	0.9206	0.8971	0.9220
Acc_{Sl}	0.9881	0.8779	0.9176	0.9412	0.9312

Table 6.8 RMSE of ridge regression on device 2 of all regression tasks (predicting the concentration of chemicals in different data groups)

	Group 1	2	3	5	6	7	Average
Acc_{SS}: Only SNV	2.8982	5.2685	1.4605	0.6872	3.9549	1.1529	2.5704
Acc_{SS}: WPDS + SEMI	1.1109	1.4875	0.9921	0.6679	1.6318	0.9641	1.1424
Acc_{Sl}	0.7763	0.7457	0.8403	0.3552	1.9783	0.7676	0.9106

6.5 Summary

This chapter is dedicated to making the prediction models of e-noses more transferable. Efforts were made in two aspects. First, the windowed piecewise direct standardization (WPDS) algorithm based on generalized ridge regression was proposed. Experiments showed that WPDS outperformed previous methods in the sense of test standardization error (SE) and prediction accuracy. It was also found that test SE is not proportional to prediction accuracy, which implies that standardization should be studied together with prediction in calibration transfer problems. Second, a novel strategy named standardization-error-based model improvement (SEMI) was applied in the prediction step. It incorporates a Tikhonov regularization term in the objective functions of prediction algorithms, so as to make the trained models more relied on stable variables, thus relatively not sensitive to the device inconsistency. It fills the gap between standardization and prediction by effectively combining the information obtained from the former step with the latter step. Experiments confirmed that it could enhance the accuracy of the master model applied to standardized slave data, especially when the inconsistency between devices is large.

The proposed methods are easily extensible. WPDS is also applicable for spectroscopic data. Its window can be adjusted to adapt to different feature extraction algorithms. The SEMI strategy can also be combined with various prediction algorithms other than the four explored in this chapter.

Clinical analysis is an important application of e-noses. However, literatures about the calibration transfer in this application are rare. In this chapter, we investigated the use of only six chemical samples as transfer samples in a breath analysis system. Experiments showed that with the algorithms proposed in this chapter, the accuracy after calibration transfer is largely improved. Further study is still needed to choose an optimized set of transfer samples and test the algorithms with real patients' breath under different temperature and humidity.

References

Balaban M, Korel F, Odabasi A, Folkes G (2000) Transportability of data between electronic noses: mathematical methods. Sens Actuators: B Chem 71(3):203–211

Bruins M, Gerritsen JW, van de Sande WW, van Belkum A, Bos A (2013) Enabling a transferable calibration model for metal-oxide type electronic noses. Sens Actuators: B Chem 188:1187–1195

Burges CJ (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Disc 2(2):121–167

Davies S, Spanel P, Smith D (1997) Quantitative analysis of ammonia on the breath of patients in end-stage renal failure. Kidney Int 52:223–228

Deng C, Zhang J, Yu X, Zhang W, Zhang X (2004) Determination of acetone in human breath by gas chromatography-mass spectrometry and solid-phase microextraction with on-fiber derivatization. J Chromatogr B 810(2):269–275

Deshmukh S, Kamde K, Jana A, Korde S, Bandyopadhyay R, Sankar R, Bhattacharyya N, Pandey R (2014) Calibration transfer between electronic nose systems for rapid in situ measurement of pulp and paper industry emissions. Anal Chim Acta 841:58–67

Eisenmann A, Amann A, Said M, Datta B, Ledochowski M (2008) Implementation and interpretation of hydrogen breath tests. J Breath Res 2(4):046,002

Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ (2008) Liblinear: a library for large linear classification. J Mach Learn Res 9:1871–1874

Feudale RN, Woody NA, Tan H, Myles AJ, Brown SD, Ferr J (2002) Transfer of multivariate calibration models: a review. Chemometr Intell Lab 64(2):181–192

Hastie T, Tibshirani R, Friedman J, Hastie T, Friedman J, Tibshirani R (2009) The elements of statistical learning, 2nd edn. Springer, New York

Hoerl AE, Kennard RW (1970) Ridge regression: biased estimation for nonorthogonal problems. Technometrics 12(1):55–67

Kalivas JH, Siano GG, Andries E, Goicoechea HC (2009) Calibration maintenance and transfer using Tikhonov regularization approaches. Appl Spectrosc 63(7):800–809

de Lacy Costello B, Ewen R, Ratcliffe NM (2008) A sensor system for monitoring the simple gases hydrogen, carbon monoxide, hydrogen sulfide, ammonia and ethanol in exhaled breath. J Breath Res 2(3):037,011

Liu Y, Cai W, Shao X (2014) Standardization of near infrared spectra measured on multi-instrument. Anal Chim Acta 836:18–23

Marco S, Gutiérrez-Gálvez A (2012) Signal and data processing for machine olfaction and chemical sensing: a review. IEEE Sens J 12(11):3189–3214

Park KS, Ko YH, Lee H, Jun CH, Chung H, Ku MS (2001) Near-infrared spectral data transfer using independent standardization samples: a case study on the trans-alkylation process. Chemometr Intell Lab 55(1):53–65

Park MY, Hastie T (2007) L1-regularization path algorithm for generalized linear models. J R Stat Soc Ser B Stat Methodol 69(4):659–677

Peng J, Peng S, Jiang A, Tan J (2011) Near-infrared calibration transfer based on spectral regression. Spectrochim Acta A 78(4):1315–1320

Phillips M, Herrera J, Krishnan S, Zain M, Greenberg J, Cataneo RN (1999) Variation in volatile organic compounds in the breath of normal humans. J Chromatogr B 729(1):75–88

Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14(3):199–222

Tikhonov AN, Arsenin VY (1977) Solutions of ill-posed problems

Tomic O, Ulmer H, Haugen JE (2002) Standardization methods for handling instrument related signal shift in gas-sensor array measurement data. Anal Chim Acta 472(1):99–111

Turner C, Walton C, Hoashi S, Evans M (2009) Breath acetone concentration decreases with blood glucose concentration in type i diabetes mellitus patients during hypoglycaemic clamps. J Breath Res 3(4):046,004

Wang Y, Veltkamp DJ, Kowalski BR (1991) Multivariate instrument standardization. Anal Chem 63(23):2750–2756

Yan K, Zhang D (2014) Blood glucose prediction by breath analysis system with feature selection and model fusion. In: Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE, IEEE, pp 6406–6409

Yan K, Zhang D (2015) Improving the transfer ability of prediction models for electronic noses. Sens Actuators B: Chem 220:115–124

Yan K, Zhang D, Wu D, Wei H, Lu G (2014) Design of a breath analysis system for diabetes screening and blood glucose level prediction. IEEE Trans Biomed Eng 61(11):2787–2795

Zhang L, Tian F, Kadri C, Xiao B, Li H, Pan L, Zhou H (2011) On-line sensor calibration transfer among electronic nose instruments for monitoring volatile organic chemicals in indoor air quality. Sens Actuators: B Chem 160(1):899–909

Zhang L, Tian F, Peng X, Dang L, Li G, Liu S, Kadri C (2013) Standardization of metal oxide sensor array using artificial neural networks through experimental design. Sens Actuators: B Chem 177:947–955

Chapter 7
Learning Classification and Regression Models Based on Transfer Samples

Abstract In this chapter, we introduce transfer-sample-based multitask learning (TMTL) to simultaneously address two problems in e-nose signals: instrumental variation and time-varying drift. Data collected with each device or in each time period define a domain. Transfer samples measured in every domain are used to share knowledge across domains. TMTL reduces the influence of drift in the target domains by aligning the transfer samples at the model level. Two paradigms, parallel and serial transfer, are designed to reflect different relationships between domains, which are dependent on the cause of drift. A dynamic model strategy is proposed to predict samples with known acquisition time and to handle noise in transfer samples. Classification and regression experiments on three real-world datasets confirm the efficacy of the proposed methods. They achieve good accuracies compared with traditional feature-level drift correction algorithms and typical labeled-sample-based MTL methods, with few transfer samples needed. TMTL is a practical algorithm framework which can greatly enhance the robustness of sensor systems with complex drift.

Keywords Drift correction · Instrumental variation · Multitask learning · Time-varying drift · Transfer learning

7.1 Introduction

Owing to the variations in the manufacture of gas sensors and e-noses, two e-noses of the same model can respond differently to the same gas sample. We have presented two algorithms in the last chapter to deal with this problem known as instrumental variation. Besides, because of aging and poisoning of gas sensors and change in operating conditions, gas sensors' responses change over time in a complex manner (Di Carlo and Falasconi 2012; Marco and Gutiérrez-Gálvez 2012). This problem is known as time-varying drift. From the perspective of machine learning, instrumental variation and time-varying drift are essentially a variation of the data distribution in the feature space. Following the terms in transfer learning

D. Zhang et al., *Breath Analysis for Medical Applications*,
DOI 10.1007/978-981-10-4322-2_7

(Pan and Yang 2010), we can refer to the data from the master device or before drift as the data from the source domain, and the data from the slave device or after drift as the data from the target domain. The goal is to transfer knowledge from the source to the target by leveraging information from both domains. In e-nose applications, we usually have enough labeled source data to train a prediction model. However, the labeled target data are scarce or hard to acquire, making it difficult to train a new model for the target domain.

Multitask learning (MTL) can be used to address this challenging problem. It is a type of inductive transfer learning method which has been successfully applied in several fields (Evgeniou and Pontil 2004; Pan et al. 2007; Yu and Ji 2015; Zhou et al. 2011b). It treats model learning in different domains as different but related tasks. Multiple models are learned simultaneously, so that information can be shared across them during the learning process to improve their accuracies (Caruana 1997). However, most MTL algorithms rely on labeled samples in the target domain, which are sometimes hard to acquire in real-world applications. In this situation, it is a good idea to use transfer samples to obtain knowledge from the target domain. Transfer-sample-based feature-level standardization methods introduced in the last chapter are easy to implement, but they are not able to correct the drift well when the distributional variation is complex. Note that the term "drift" often refers to time-varying drift (Marco and Gutiérrez-Gálvez 2012). In the following three chapters, we use the term to indicate the change in data distribution, which can be caused by instrumental variation, sensor aging, environmental change, and so on.

In this chapter, we propose a novel method named transfer-sample-based multitask learning (TMTL). It combines MTL with transfer samples, thus has strength in both accuracy and practical convenience. In the proposed algorithm, labeled source data and a group of transfer samples are exploited to learn the source and (multiple) target models jointly. The type of drift determines the relationship between domains, so we designed a parallel and a serial transfer paradigm for different drifts. To predict the sample measured in a specific time and handle the noise in transfer samples, a dynamic model strategy that uses combination of neighboring models is proposed. Like SEMI, TMTL is a framework that can be implemented using various loss functions. Two popular classification/regression loss functions, i.e., logistic and squared loss, are demonstrated in this chapter. We also compared three algorithms to select representative transfer samples.

We mainly focus on e-noses in this chapter, but the proposed methods have potential in fields such as spectroscopy (Yu and Ji 2015), indoor localization (Pan et al. 2007), and color correction of digital cameras (Wang and Zhang 2010). For these problems, data measured by sensors or devices contain drift, meanwhile transfer samples can be collected (e.g., the spectra of the same chemical, the WiFi signals at the same location, and the images of the same color).

The rest of this chapter is organized as follows. Section 7.2 briefly reviews the related works in drift correction and MTL. Section 7.3 describes the proposed

TMTL in detail. The transfer sample selection algorithms adopted in this chapter are introduced in Sect. 7.4. Section 7.5 presents the experimental configurations and results, along with some comprehensive analysis. Section 7.6 summarizes the chapter.

7.2 Related Work

In order to obtain knowledge from the target domain, some samples from the domain are needed. According to the type of the target samples, we classify drift correction methods into three categories, i.e., those based on labeled target samples, unlabeled target samples, and transfer samples.

In the setting of labeled-sample-based methods, some labeled data from the target domain is available, but not sufficient to retrain a target model. In this case, one intuitive idea is to use source and target data together to train a model, meantime increase the weights of the target samples to ensure the model's feasibility in the target domain. For instance, Zhang and Zhang (2015) combined e-nose data before and after drift into the objective function of an extreme learning machine. Although easy to implement, this kind of method often needs many target samples to capture the variance in the target domain. In the case of time-varying drift, drifted data comes in the form of streams. Concept drift adaptation methods make use of newly arrived labeled data to update the prediction models (Gama et al. 2014; Kadlec et al. 2011). As an example, Vergara et al. (2012) adopted an ensemble strategy to cope with time-varying drift in e-noses. Samples collected in different time were split into several batches. Then, a prediction model was trained on each batch. Finally, for a test sample in batch k, the outputs of models 1 to $k-1$ were fused by weighted majority voting, with the weights estimated from the prediction accuracies of the models on batch $k-1$. The method requires all samples in prior batches to be labeled, which is often impractical.

Multitask learning (MTL) uses a different strategy to fuse knowledge from different domains. Models for all domains are learned jointly. In the objective function of an MTL method, the prior knowledge about the relationship of the models and the features can be specified. Consequently, information can be shared properly among the tasks, so as to enhance the generalization ability of all models, especially for the target domain which has less labeled samples. Regularized MTL (RMTL) was proposed in Evgeniou and Pontil (2004), in which a regularization term was introduced to penalize the deviation among multiple models. Yu and Ji (2015) applied RMTL to transfer models between near-infrared spectra measured in different conditions (e.g., multiple devices) and achieved good results. In Zhou et al. 2011b, formulated disease progress prediction as a multitask regression problem, with learning the model at each time period as a task. Models at neighboring time periods were required to be close to capture the intrinsic temporal smoothness. Group Lasso regularization was also employed for feature selection.

The second category of methods is unlabeled-sample-based ones, whose main advantage is that unlabeled target samples are much easier to acquire in practice. Transductive transfer learning (Pan and Yang 2010) and semi-supervised learning algorithms can be adopted in this setting. A transfer learning approach based on weighted geodesic flow kernel and a semi-supervised classifier based on manifold regularization were used in Liu et al. (2014) to address sensor drift in e-noses. On the dataset introduced in Vergara et al. (2012), the prediction accuracy on drifted data was improved.

Transfer samples are more informative than unlabeled target samples, meanwhile more convenient to obtain than labeled target samples in many real-world applications. Most existing transfer-sample-based methods concentrate on feature-level correction. Algorithms based on variable standardization build regression models using the transfer samples. Each variable in the source domain is fitted with one or multiple variables in the target domain using regression algorithms, so as to transform the target data to the source domain. Then, the corrected data can be predicted by the source models (Yan and Zhang 2015). Algorithms based on component correction (CC) are also popular. CC-PCA (Artursson et al. 2000) finds the drift-related direction in the feature space by applying principal component analysis to the transfer samples. Then the component on the direction can be removed from all data. Orthogonal signal correction (OSC) (Padilla et al. 2010; Wold et al. 1998) is a CC-like method that relies on labeled target samples. It pools samples with and without drift and finds the undesired component by calculating the subspace that is orthogonal to the labels. One drawback of CC-like methods is that when the drift is complex, it may be difficult to accurately separate the directions of useful information and drift (Romain and Nicolas 2010).

7.3 Transfer-Sample-Based Multitask Learning (TMTL)

In this section, we will first consider the situation with only one source and one target domain. Transfer-sample-based coupled task learning (TCTL), the basic form of TMTL, is introduced for this situation. Then, we will extend TCTL to TMTL which involves multiple domains, and describe a parallel paradigm and a serial one to deal with different inter-domain relationships. Finally, we propose a combination of the two paradigms and a dynamic model strategy.

7.3.1 Transfer-Sample-Based Coupled Task Learning (TCTL)

A preliminary version of TCTL was introduced in our previous work (Yan and Zhang 2016). In order to depict the problem setup more concretely, we take calibration transfer as an example. Suppose an e-nose (the source device) was utilized to collect 50 breath samples from healthy subjects and 50 from diabetes patients. A classification model was trained on these data. Now we have made a new e-nose (the target device) of the same model for diabetes screening. A set of standard gas samples have been measured with both the old and the new e-nose. Then, TCTL can be used to learn the classification model of the new device.

Denote $X_S \in \mathbf{R}^{n \times m}$ as the matrix of source training data with each row as a feature vector; n is the number of labeled source samples; m is the number of variables; $y_S \in \mathbf{R}^n$ is the label vector; $T_S \in \mathbf{R}^{n_t \times m}$ and $T_T \in \mathbf{R}^{n_t \times m}$ are the matrices of the source and target transfer samples, respectively; n_t is the number of transfer samples; $\boldsymbol{\beta}_S, \boldsymbol{\beta}_T \in \mathbf{R}^m$ are the source and target prediction models to be estimated, respectively. The objective function of TCTL is presented as the following:

$$\begin{aligned} \min_{\boldsymbol{\beta}_S, \boldsymbol{\beta}_T} & \ell(X_S, y_S, \boldsymbol{\beta}_S) + \lambda_1 \|T_S \boldsymbol{\beta}_S - T_T \boldsymbol{\beta}_T\|_2^2 \\ & + \lambda_2 \|X_S \boldsymbol{\beta}_S - X_S \boldsymbol{\beta}_T\|_2^2 + \mu \sum_{j=1}^{m} w_j^2 \left(\beta_{S,J}^2 + \beta_{T,J}^2\right). \end{aligned} \tag{7.1}$$

In Eq. 7.1, the first term represents the empirical loss function for the source training samples. $\|T_S \boldsymbol{\beta}_S - T_T \boldsymbol{\beta}_T\|_2^2$ is the transfer sample term. It requires the corresponding source and target transfer samples to be close after they are respectively projected by the source and target models. The term $\|X_S \boldsymbol{\beta}_S - X_S \boldsymbol{\beta}_T\|_2^2$ encourages similar source and target models by requiring that they project the source training samples to similar values. The last term is a weighted shrinkage term. $\beta_{S,j}$ stands for the jth element of $\boldsymbol{\beta}_S$. The weights are defined as:

$$w_j = \sqrt{\sum_{i=1}^{n_t} \left(t_{S,ij} - t_{T,ij}\right)^2}, \tag{7.2}$$

where $t_{S,ij}$ means the element in the ith row (sample) and jth column (variable) of T_S. The shrinkage term penalizes the variables that have large deviation between the source and target transfer samples. λ_1, λ_2 and $\mu \geq 0$ are regularization parameters controlling the strength of the terms.

The transfer sample term is key for information transfer between domains. It aligns the transfer samples of the two domains in their respective projected spaces, so as to reduce the inter-domain drift. Thus, the discriminative information of the labeled source samples can be used in the target domain. The transfer sample term can also be regarded as an improvement on the conventional variable standardization (VS) method. In linear cases, the latter method is essentially estimating a matrix $M \in \mathbf{R}^{m \times m}$ to transform the target variables to the source space, i.e., to make $T_T M \approx T_S$. In the prediction step, the transformed target samples are projected by $\boldsymbol{\beta}_S$. So the goal actually boils down to reducing the difference between the two domains in the projected direction, in other words, minimizing $\|T_S \boldsymbol{\beta}_S - T_T M \boldsymbol{\beta}_S\|_2^2$. It is exactly the transfer sample term in TCTL if we set $\boldsymbol{\beta}_T = M\boldsymbol{\beta}_S$. M no longer needs to be estimated, which makes TCTL more efficient and less prone to overfitting compared with VS.

If we rely solely on the transfer sample term to infer $\boldsymbol{\beta}_T$ from $\boldsymbol{\beta}_S$, the control over $\boldsymbol{\beta}_T$ will be too weak. Because the number of transfer samples is often small, there will be infinite solutions to $\boldsymbol{\beta}_T$ that can minimize the transfer sample term and make it zero. Therefore, we add the model similarity term $\|X_S\boldsymbol{\beta}_S - X_S\boldsymbol{\beta}_T\|_2^2$ to introduce an inductive bias reflecting the prior belief that the models resemble each other. To reduce the inter-domain difference before applying TCTL, one can preprocess the source and target data separately with standard normal variate (SNV) (Marco and Gutiérrez-Gálvez 2012), i.e., each variable is centered and scaled by the mean and standard deviation calculated from the transfer samples of its domain. Additionally, many MTL algorithms (Evgeniou and Pontil 2004; Pan et al. 2007; Yu and Ji 2015; Zhou et al. 2011b) simply penalize the deviation between two models, e.g., minimizing $\|\boldsymbol{\beta}_S - \boldsymbol{\beta}_T\|_2^2$. This requirement is too strict when the inter-domain difference is large. Our model similarity term relaxes this requirement. The two models may not be identical, but their difference should be orthogonal to the space spanned by the source training samples. Experimental results show that the model similarity term in this form is better than that in the traditional form.

Moreover, we add the SEMI term in the last chapter into TCTL. Two implementations of the TCTL framework are as follows.

7.3.1.1 Classification: Logistic Loss

The proposed framework can be tied with various loss functions. Logistic loss function is demonstrated in this chapter because logistic regression (LR) is a popular and effective classifier. We denote $x^{(i)} \in \mathbf{R}^m$ as the ith training sample and $y^{(i)} \in \{0, 1\}$ as its label. $X = [x^{(1)}, \ldots, x^{(n)}]^T, y = [y^{(1)}, \ldots, y^{(n)}]^T$. In binary-class cases, the decision function of LR is a sigmoid function $h_{\boldsymbol{\beta}}(x) = 1/(1 + e^{-\boldsymbol{\beta}^{\mathrm{T}}x})$. A test sample x is classified into the positive class if $h_{\boldsymbol{\beta}}(x) \geq 0.5$. The logistic loss function can be written as:

$$\ell_L(X, y, \boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\left[y^{(i)}\log h_{\boldsymbol{\beta}}\left(x^{(i)}\right) + \left(1 - y^{(i)}\right)\log\left(1 - h_{\boldsymbol{\beta}}\left(x^{(i)}\right)\right)\right], \tag{7.3}$$

Combining Eq. 7.3 with Eq. 7.1, we formulate the objective function of TCTL with logistic loss as:

$$\begin{aligned} J_L(\boldsymbol{\beta}_S, \boldsymbol{\beta}_T) = {} & \ell_L(X_S, y_S, \boldsymbol{\beta}_S) + \frac{\lambda_1}{2n_t}\|T_S\boldsymbol{\beta}_S - T_T\boldsymbol{\beta}_T\|_2^2 \\ & + \frac{\lambda_2}{2n}\|X_S\boldsymbol{\beta}_S - X_S\boldsymbol{\beta}_T\|_2^2 + \frac{\mu}{2}\sum_{j=1}^{m} w_j^2\left(\beta_{S,J}^2 + \beta_{T,J}^2\right), \end{aligned} \tag{7.4}$$

whose gradient is given by:

$$\begin{aligned}
\frac{\partial J_L}{\partial \boldsymbol{\beta}_S} &= \frac{1}{n} X_S^T \left(h_{\boldsymbol{\beta}}(X_S) - y_S\right) + \frac{\lambda_1}{n_t} T_S^T (T_S \boldsymbol{\beta}_S - T_T \boldsymbol{\beta}_T) + \frac{\lambda_2}{n} X_S^T X_S (\boldsymbol{\beta}_S - \boldsymbol{\beta}_T) + \mu W \boldsymbol{\beta}_S, \\
\frac{\partial J_L}{\partial \boldsymbol{\beta}_T} &= -\frac{\lambda_1}{n_t} T_T^T (T_S \boldsymbol{\beta}_S - T_T \boldsymbol{\beta}_T) - \frac{\lambda_2}{n} X_S^T X_S (\boldsymbol{\beta}_S - \boldsymbol{\beta}_T) + \mu W \boldsymbol{\beta}_T, \\
W &= \mathrm{diag}(w_1^2, \ldots, w_m^2).
\end{aligned} \tag{7.5}$$

The problem above can be solved using numerical optimization methods such as conjugate gradient. In K-class cases, K LR models are trained using the one-versus-all strategy and x is classified into the class whose decision function has the largest value.

7.3.1.2 Regression: Squared Loss

For regression problems, the squared loss function is adopted in this chapter. The objective function of TCTL with squared loss is:

$$\begin{aligned}
J_S(\boldsymbol{\beta}_S, \boldsymbol{\beta}_T) = {} & \frac{1}{2n} \|X_S \boldsymbol{\beta}_S - y_S\|_2^2 + \frac{\lambda_1}{2n_t} \|T_S \boldsymbol{\beta}_S - T_T \boldsymbol{\beta}_T\|_2^2 \\
& + \frac{\lambda_2}{2n} \|X_S \boldsymbol{\beta}_S - X_S \boldsymbol{\beta}_T\|_2^2 + \frac{\mu}{2} \sum_{j=1}^{m} w_j^2 \left(\beta_{S,J}^2 + \beta_{T,J}^2\right).
\end{aligned} \tag{7.6}$$

By setting its gradient to zero, the closed-form solution to $\boldsymbol{\beta}_S$ and $\boldsymbol{\beta}_T$ can be derived:

$$\begin{pmatrix} \boldsymbol{\beta}_S \\ \boldsymbol{\beta}_T \end{pmatrix} = (A_1 + A_2 + A_3)^{-1} \boldsymbol{b}, \tag{7.7}$$

where

$$\begin{aligned}
A_1 &= \begin{pmatrix} P & 0 \\ 0 & 0 \end{pmatrix}, A_2 = \frac{\lambda_1}{n_t} \begin{pmatrix} T_S^T T_S & -T_S^T T_T \\ -T_T^T T_S & T_T^T T_T \end{pmatrix}, \\
A_3 &= \begin{pmatrix} \lambda_2 P + \mu W & -\lambda_2 P \\ -\lambda_2 P & \lambda_2 P + \mu W \end{pmatrix}, \boldsymbol{b} = \frac{1}{n} \begin{pmatrix} X_S^T \boldsymbol{y}_S \\ 0 \end{pmatrix}, \\
P &= \frac{1}{n} X_S^T X_S, W = \mathrm{diag}(w_1^2, \ldots, w_m^2).
\end{aligned}$$

7.3.2 TMTL-Parallel and TMTL-Serial

TCTL only exploits information from two domains. In reality, there are situations of multiple domains. If a number of new devices have been manufactured, each new device can be regarded as a target domain which is different but related with each other and the old device (source domain). In another situation, a device may have been used to collect data for a long time. Knowing its slow and irregular time-varying drift, we have collected transfer samples periodically. In this case, each period can be viewed as a target domain which has relatively small intra-domain drift. Each domain is different but related with its previous domain, i.e., the time period prior to it. TMTL shares information across many domains, which would probably be superior to TCTL. For instance, transfer samples in one domain may contain noises or outliers due to the uncertainty in the measurement process. In TCTL, the noises and outliers will mislead the model transfer process. However, in TMTL, the influence of noises and outliers to one model can be mitigated owing to the similarity requirements with all the other models.

Considering the relationship between domains, we have designed two paradigms, namely TMTL-parallel and TMTL-serial. TMTL-parallel is suitable for situations such as calibration transfer, where multiple domains are similar to each other. Here, we use a subscript k to denote the variable in the kth target domain, and a subscript 0 to denote the variable in the source domain for simplicity. The total number of target domains is d. The objective function of TMTL-parallel is expressed as:

$$
\begin{aligned}
\min_{\boldsymbol{\beta}_S, \boldsymbol{\beta}_T^{(1)}, \ldots, \boldsymbol{\beta}_T^{(d)}} \; & \ell(X_0, \boldsymbol{y}_{\mathbf{0}}, \boldsymbol{\beta}_0) + \lambda_1 \sum_{k=1}^{d} \| T_0 \boldsymbol{\beta}_0 - T_k \boldsymbol{\beta}_k \|_2^2 \\
& + \lambda_2 \sum_{k=0}^{d} \left\| X_0 \left(\boldsymbol{\beta}_k - \frac{1}{d+1} \sum_{r=0}^{d} \boldsymbol{\beta}_r \right) \right\|_2^2 \\
& + \mu \sum_{k=0}^{d} \sum_{j=1}^{m} w_{k,j}^2 \beta_{k,j}^2 .
\end{aligned} \tag{7.8}
$$

It is a natural extension of TCTL to multiple target domains. The transfer samples of each target domain are aligned to those in the source domain in their respective projected spaces. Each model is encouraged to resemble an average model (Evgeniou and Pontil 2004). In the SEMI term, the shrinkage weight for variable j in the source domain $\left(w_{0,j}\right)$ is the average of those in the target domain. By minimizing Eq. 7.8, we can obtain the prediction models for all devices efficiently.

TMTL-serial is specialized for situations such as time-varying drift. The difference between the parallel and serial TMTL is that the latter one encourages each model to be similar to its previous model:

$$\begin{aligned} \min_{\boldsymbol{\beta}_S, \boldsymbol{\beta}_T^{(1)}, \ldots, \boldsymbol{\beta}_T^{(d)}} \; & \ell(X_0, \boldsymbol{y}_{\mathbf{0}}, \boldsymbol{\beta}_0) + \lambda_1 \sum_{k=1}^{d} \| T_0 \boldsymbol{\beta}_0 - T_k \boldsymbol{\beta}_k \|_2^2 \\ & + \lambda_2 \sum_{k=1}^{d} \| X_0 (\boldsymbol{\beta}_k - \boldsymbol{\beta}_{k-1}) \|_2^2 \\ & + \mu \sum_{k=0}^{d} \sum_{j=1}^{m} w_{k,j}^2 \beta_{k,j}^2 . \end{aligned} \tag{7.9}$$

The intuition is to capture the temporal smoothness prior as in Zhou et al. (2011b). Note that there are two typical modes to analyze data streams. In the offline mode, data in all time periods are analyzed together, which implies that transfer samples collected in later periods can aid the model transfer process of former periods. In this mode, models of all periods can be obtained simultaneously by optimizing Eq. 7.9. The online mode, on the other hand, requires data in the current period to be analyzed in real time. This means that only the transfer samples collected before can be used. In this mode, we can optimize Eq. 7.9 each time a new group of transfer samples are collected, and use the latest model $(\boldsymbol{\beta}_d)$ obtained to predict recent samples.

7.3.3 TMTL-General and the Dynamic Model Strategy

In the most general case, samples can be collected by several devices in a long period of time, as illustrated in Fig. 7.1. So we can go one step further and combine the parallel and serial TMTL to simultaneously learn all models. In TMTL, each group of transfer samples corresponds to a model. First, the initial group of transfer samples measured by the oldest device (denoted as device 1) is selected as the overall reference. All the other groups should be aligned with it in their respective projected spaces, which form the transfer sample term in the objective function. Second, each model is expected to be similar to its previous model of the same device, while the first model of each device should resemble their average mean$(\boldsymbol{\beta}_{k,1})$, as shown in Fig. 7.1.

To deal with time-varying drift, the data stream of a device is split into discrete batches in most previous studies (Liu et al. 2014; Vergara et al. 2012; Zhang and Zhang 2015; Zhou et al. 2011b) and the discussions above. Each batch corresponds to one fixed model. This strategy loses the information carried in the exact acquisition time of the samples in the same batch. The drift within a batch cannot be modeled. Therefore, we propose a dynamic model strategy to exploit the information. Assuming that the time-varying drift of a device is smooth, it is intuitive to also let the model change smoothly over time. We set the model of device i at time t to be a function of all models of the same device. A straightforward method is to interpolate between neighboring models. We find it better to use a weighted combination as follows:

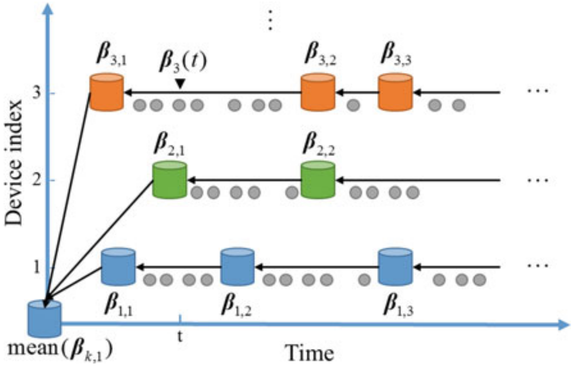


Fig. 7.1 Illustration of the sample collection process in the most general case. The *j*th cylinder located in the *i*th row represents the *j*th group of transfer samples measured by device *i*, which also corresponds to a model $\boldsymbol{\beta}_{i,j}$. The *circles* are ordinary samples measured by the device at a specific time. The *arrows* indicate the model similarity relationships: the model at the beginning of an *arrow* should resemble the model at the end

$$\begin{aligned}\boldsymbol{\beta}_i(t) &= \sum_j c_{i,j}(t)\boldsymbol{\beta}_{i,j},\\ c_{i,j}(t) &= \exp\left(-\sigma(t-t_{i,j})^2\right).\end{aligned} \tag{7.10}$$

$t_{i,j}$ is the acquisition time of the *j*th group of transfer samples of device *i*. The closer *t* is to $t_{i,j}$, the larger the weight $c_{i,j}(t)$ will be. σ is the window size parameter. Note that $c_{i,j}(t)$ should be normalized to keep a sum of 1. When using this dynamic model strategy, the model for every training and test sample should be calculated using Eq. 7.10. The mean and standard deviation values used to normalize variables in SNV should also be modified according to Eq. 7.10, which we will call dynamic SNV. These dynamic strategies can probably make the models more accurate. Another important function of the strategies is to deal with noises and outliers in transfer samples. They can further smooth the noise contained in individual models, which has similar insight to the ensemble strategy (Vergara et al. 2012). Details about TMTL with logistic or squared loss can be extended from Eqs. 7.5 to 7.7, thus will not be presented here for brevity.

7.4 Selection of Transfer Samples

The selection of transfer samples is also an important issue in TMTL. Transfer samples should be capable of representing one domain in order to effectively transfer knowledge between domains. Meanwhile, the number of transfer samples should be as small as possible to ease the burden of collecting them repeatedly (Rodionova and Pomerantsev 2008). Because only the source samples are accessible in the training stage, one often gathers a sufficient set of candidates from the source domain, then selects a compact and representative group from them. The selected transfer samples can then be measured by every new device and in each time period.

In the last chapter, we empirically selected six transfer samples. Nonetheless, this method requires strong prior knowledge of the dataset, thus we hope an algorithm can be applied to automatically choose transfer samples. In the field of machine olfaction (e-noses) and spectroscopy, the Kennard–Stone (KS) algorithm is the most intensively used (Feudale et al. 2002; Kennard and Stone 1969; Zhang et al. 2011b, 2015). Given a set of candidate transfer samples, KS aims to sequentially select the samples that capture the most variance of the candidates. First, the two samples farthest apart from each other are picked. The next sample selected should have the largest nearest distance from the existing selections. This procedure is repeated until desired number of samples have been chosen. One disadvantage of KS is that the selected samples may contain outliers.

Active learning methods (Yu et al. 2006; Zhang et al. 2011a) are also suitable for this problem. In this chapter, we explore two methods in this category. For easy controlling of the sample size, only sequential selection algorithms are considered. Transductive experimental design (TED) (Yu et al. 2006) selects samples that can be used to reconstruct the whole data set most precisely (Zhang et al. 2011a). Locally Linear Reconstruction (LLR) (Zhang et al. 2011a) further takes into account the local manifold structure. It requires that a data point can only be linearly reconstructed from its neighbors, then selects the samples that best reconstruct the whole data set. We will compare the three methods mentioned above in the next section. Another related issue is to choose the proper time to collect transfer samples. For this issue, one can refer to the change detection algorithms in Gama et al. (2014).

7.5 Experiments

In this section, we will conduct experiments on three datasets to evaluate the performance of the proposed algorithms. The datasets contain time-varying drift, instrumental variation, and both, respectively. Comparison will be made between our methods and other typical methods in the fields of drift correction and MTL. Different strategies in our methods will also be explored and analyzed.

7.5.1 Gas Sensor Array Drift Dataset

The gas sensor array drift dataset is a public dataset[1] introduced by Vergara et al. (2012, 2014). An e-nose with 16 gas sensors was utilized to collect the dataset over a course of 36 months. Six kinds of gases (ammonia, acetaldehyde, acetone, ethylene, ethanol, and toluene) at different concentrations were measured. The total number of samples is 13,910. Each sample is represented by a feature vector with 128 variables extracted from the sensors' response curves (Vergara et al. 2012). The dataset is split into 10 batches in chronological order. The period of collection and the number of samples in each batch can be found in Table 7.1. The goal is to classify the type of gases, despite their concentrations. We choose batch 1 (source domain) as the training set and test on batches 2–10 (target domains). This evaluation strategy was also used in Liu et al. 2014, Vergara et al. 2012, Zhang and Zhang 2015 and resembles the situation in real-world applications.

Figure 7.2 shows a scatter map for visual inspection of the time-varying drift across batches. The samples are projected to a 2D subspace using PCA. It can be found that the ammonia samples drift roughly to the $+x$ direction, whereas the drift of acetaldehyde is small. There are also some samples that do not follow the general trend of drift, which implies that the drifting pattern of the samples is complex and it is hard to directly compensate it (Zhang and Zhang 2015).

To explain the principle of proposed transfer-sample-based strategy, we depict the effect of TCTL in Fig. 7.3, in which transfer samples are leveraged to align the drifted samples in the projected subspace. An experiment was made with samples of three classes in two batches. Two classification models were trained to distinguish class 1 or 2 from the other two classes. Then, the samples were projected by the two models. The colored areas suggest the correct regions for the samples in each class. In plot (a), samples from both batches are projected by the source model. Therefore, some target samples (plus signs) fall into the wrong region because of the drift, thus will not be correctly classified. In plot (b), TCTL is applied to learn the source and the target models simultaneously. With the transfer samples in both domains aligned (black points), the drift is reduced in the model level and the target samples fall into correct regions.

The first step of our methods is choosing transfer samples. They are not directly provided in the dataset, hence need to be selected from a candidate set. The candidate set of batch k ($k = 2, \ldots, 10$) was defined as the overlapping samples in batch 1 and k, namely the samples of the same gas and concentration. Then, we used the three selection algorithms introduced in Sect. 7.4 to choose n_t transfer samples for each batch. After that, the samples in each batch were preprocessed with SNV. The models for batches 2–10 were learned using TCTL or TMTL. For TMTL, the serial paradigm and the online analysis mode were adopted. For a target batch k, the labeled training samples in batch 1 and the transfer sample groups of batches 1 to

[1] http://archive.ics.uci.edu/ml/datasets/Gas+Sensor+Array+Drift+Dataset+at+Different+Concentrations.

Table 7.1 Period of collection and number of samples in the gas sensor array drift dataset

Batch ID	Month	Ammonia (2.5–1000)[a]	Acetaldehyde (2.5–300)	Acetone (10–1000)	Ethylene (2.5–300)	Ethanol (2.5–600)	Toluene (1–230)	# Total
1	1, 2	83	30	70	98	90	74	445
2	3, 4, 8–10	100	109	532	334	164	5	1244
3	11–13	216	240	275	490	365	0	1586
4	14, 15	12	30	12	43	64	0	161
5	16	20	46	63	40	28	0	197
6	17–20	110	29	606	574	514	467	2300
7	21	360	744	630	662	649	568	3613
8	22, 23	40	33	143	30	30	18	294
9	24, 30	100	75	78	55	61	101	470
10	36	600	600	600	600	600	600	3600

[a] Numbers in the parenthesis show the range of concentration in parts per million (ppm)

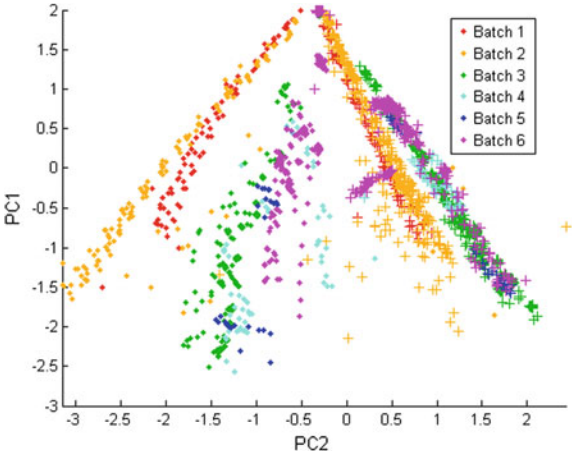


Fig. 7.2 Example of the drift across batches 1–6 in the gas sensor array drift dataset. *Dots* and *plus* signs represent ammonia and acetaldehyde samples, respectively. Different colors indicate different batches

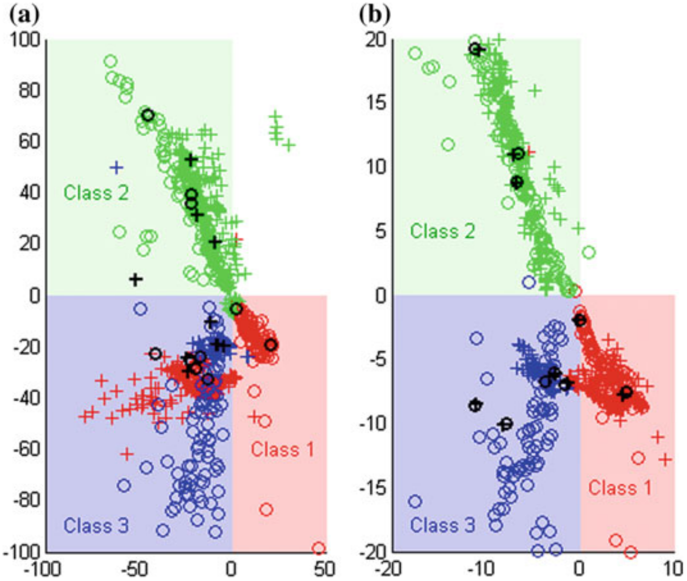


Fig. 7.3 Illustration of the effect of TCTL. Markers in different colors are samples from different classes, except the *black* ones, which represent the transfer samples. *Circles* are samples from batch 1 (source); *Plus* signs are those from batch 2 (target). In plot (**a**), samples from both batches are projected by the source model learned by LR. In plot (**b**), the source samples are projected by the source model learned by TCTL with logistic loss, whereas the target samples are projected by the target model learned by it

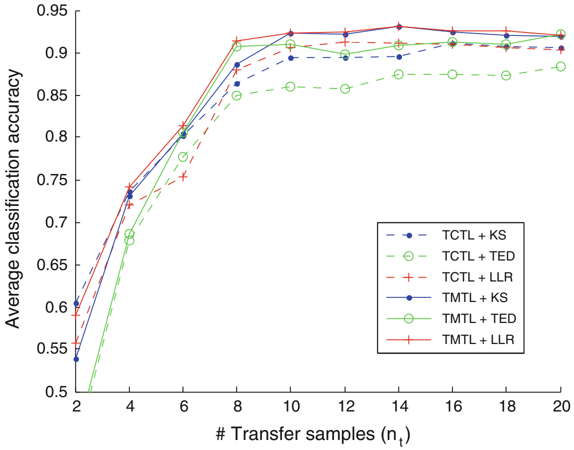


Fig. 7.4 Comparison of the three sample selection methods on the gas sensor array drift dataset

k were fed into Eq. 7.9 with logistic loss. After prediction, an average classification accuracy was computed.

Comparison of the sample selection algorithms is displayed in Fig. 7.4. The parameters for TED and LLR were set to be the same with those in the original papers (Yu et al. 2006; Zhang et al. 2011a). The parameters of TCTL and TMTL were optimized by grid search for each result, except that μ was empirically fixed to 10^{-3}. It can be found that the accuracy improves as n_t increases, until n_t reaches about 10. The overall order of performance is LLR > KS > TED. The effectiveness of the locally linear reconstruction strategy of LLR is proved. The traditional KS algorithm, although simple, shows performances close to LLR. Besides, TMTL is generally better than TCTL, which is because TMTL involves more tasks that can help each other. It makes use of all k groups of transfer samples for batch k, whereas TCTL only uses two groups. Furthermore, TMTL-serial is able to capture the temporal smoothness prior of the data. The influence of noises and outliers in transfer samples are better mitigated.

Figure 7.5 shows the average accuracy of TMTL-serial when parameters λ_1 and λ_2 are varied in $\{2^{-8}, 2^{-7}, \ldots, 2^2\}$. LLR was used to select 10 transfer samples in this experiment. μ was still fixed to 10^{-3}. We notice that the accuracy is the highest when λ_1 is neither too small nor too large. λ_1 controls the weight of the transfer sample term. If it is too small, the transfer samples cannot be aligned well. Meanwhile, putting too much emphasis on the transfer samples will cause overfitting. The accuracy degrades when λ_2 is large, indicating that the source and target models cannot be too similar because of the drift.

Figure 7.6 compares TCTL- and TMTL-serial with several other methods, including only preprocessing the features with SNV (Only SNV) (Marco and Gutiérrez-Gálvez 2012), variable standardization (last chapter), MTL based on temporal group Lasso (TGL) (Zhou et al. 2011a, b), and regularized MTL (RMTL) (Evgeniou and Pontil 2004). LLR was used to select transfer samples from the

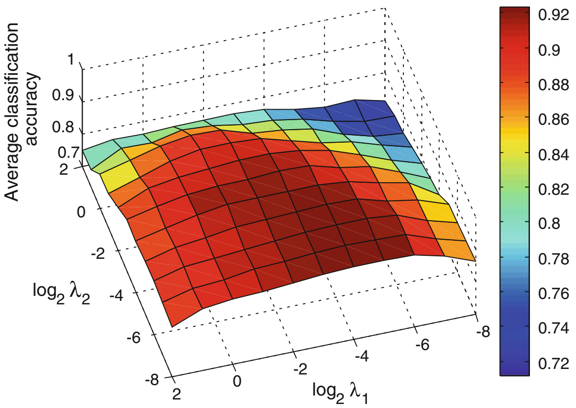


Fig. 7.5 The impact of the parameters λ_1 and λ_2 on the average classification accuracy of TMTL-serial

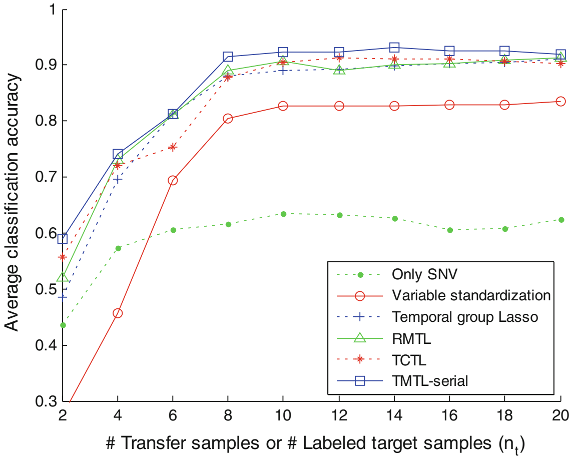


Fig. 7.6 Performance comparison on the classification task of the gas sensor array drift dataset

source domain or labeled target samples from the target domain. The first two methods standardize each target variable based on the transfer samples, then use the source models learned by LR to predict the standardized target samples. Their performances are not promising possibly because the drift is complex and the capacity of the feature-level correction methods is limited. The latter two are MTL methods (with logistic loss function and linear kernel) based on labeled target samples. The parameters were tuned by grid search for each result. Their performances are comparable with TCTL. TMTL-serial has the best accuracy for each n_t. Moreover, TCTL and TMTL have the advantage of not needing to select and label the target samples.

More results of existing methods are listed in Table 7.2. For "no transfer", data in batches 2–10 were directly predicted by the classification model trained on batch 1. Its accuracy is poor especially for batches with large IDs, which proves the influence of drift. The results of ensemble, DAELM-S, and ML-comGFK are

Table 7.2 Classification accuracy of various methods on the gas sensor array drift dataset

Target batch ID	2	3	4	5	6	7	8	9	10	Average
No transfer	88.59	66.96	40.99	54.82	43.22	44.40	31.63	45.74	39.11	50.61
CC-PCA	90.92	40.86	47.20	59.39	56.74	56.71	36.39	45.32	37.72	52.36
OSC	88.10	66.71	54.66	53.81	65.13	63.71	36.05	40.21	40.08	56.50
Ensemble	74.36	87.83	93.79	95.43	69.17	69.72	91.84	76.38	65.50	80.45
ML-comGFK	80.25	74.99	78.79	67.41	77.82	71.68	49.96	50.79	53.79	67.28
DAELM-S	87.98	95.74	85.16	95.99	94.14	83.51	86.90	100.00	53.62	87.00
TCTL	97.35	95.46	90.68	98.48	93.22	93.91	89.12	87.02	69.97	90.58
TMTL (sim2)	97.51	98.74	93.79	96.95	95.04	90.51	90.14	92.55	69.72	91.66
TMTL (no SEMI)	96.46	97.35	95.65	97.97	95.04	84.83	82.31	93.19	70.78	90.40
TMTL-parallel	97.35	97.16	93.17	97.46	95.96	91.70	90.14	90.85	73.28	91.90
TMTL-serial	97.35	98.80	90.06	98.48	95.35	91.50	91.84	96.38	71.56	**92.37**

Bold values indicate the best results

copied from the original papers. Although the ensemble method and DAELM-S achieve good results, they both need relatively large amount of auxiliary target samples. DAELM-S requires 30 selected labeled samples in each target batch. The ensemble method requires all samples in batches 1 to $k - 1$ to be labeled when predicting batch k. ML-comGFK needs only unlabeled target samples. But its accuracy is still not satisfactory.

In order to assess the strategies adopted in our methods, we have tested some possible alternatives, whose results are listed in the last five rows of Table 7.2. For "TMTL (sim2)", the proposed model similarity constraint $\left(\|X_S\boldsymbol{\beta}_1 - X_S\boldsymbol{\beta}_2\|_2^2\right)$ is replaced by $\|\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2\|_2^2$, which occurs in many MTL papers. For "TMTL (no SEMI)", the proposed weighted shrinkage term is replaced by an ordinary shrinkage term with uniform weights. TMTL-serial outperforms the two alternatives, indicating the superiority of the proposed strategies. Besides, TMTL-serial is slightly better than TMTL-parallel in this problem.

7.5.2 *Breath Analysis Dataset*

A breath analysis dataset was collected using two e-noses of the same model (Yan et al. 2014) as in Chap. 14. The collection process lasted for about 500 days starting from 2014. From the dataset, we select five diseases that have been proved to be related with certain biomarkers in breath, namely diabetes, chronical kidney disease (CKD), cardiopathy, lung cancer, and breast cancer (Wilson and Baietto 2011). Their sample sizes and days of collection are illustrated in Fig. 7.7, together with those of the healthy samples and transfer samples. Transfer sample groups were measured periodically, with eight pre-selected standard gas samples in each group.

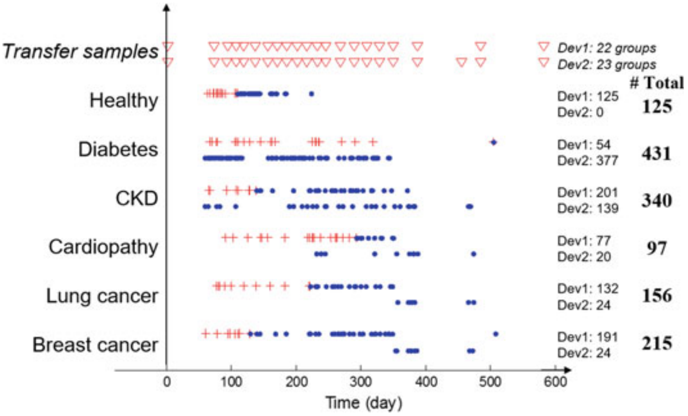


Fig. 7.7 Overview of part of the breath analysis dataset. Each *point* denotes a sample (or a group of transfer samples) collected in a specific time. The two rows of each class represent samples measured by the two devices, with the sample sizes labeled on the *right*. *Red plus* signs denote the training samples

Different from the simulation dataset used in the last chapter, this real-world dataset was collected in a loosely controlled environment in several hospitals in Guangzhou, China. Therefore, it suffers from a number of factors that will cause drift in data distribution, e.g., instrumental variation, sensor aging, temperature and humidity change, sensor damage and replacement, etc. As an example, we draw the steady-state responses of two sensors in Fig. 7.8. The sensitivity of the sensor in plot (a) gradually decayed over time, as can be observed from the trend of breath and transfer samples. For the sensor in plot (b), however, the decay was much faster, so we replaced it three times. It is worth noting that the transfer samples contain noise and outliers (e.g., in plot (a)), which cannot precisely reflect the true distribution of the data, thus will degrade the accuracy if we transfer knowledge based on them. One solution is to detect the outliers according to some prior knowledge. In this chapter, we use the dynamic model strategy in Eq. 7.10 to deal with it.

The experimental settings are as follows. Five binary-class classification tasks (healthy vs. disease) were executed. Because the classes are imbalanced, F-score was adopted as the accuracy metric. To simulate real-world applications, we used only the first 50 samples collected with device 1 in each class as training samples (see Fig. 7.7), others as test ones. Considering the complexity of the drift and the noise in transfer samples, we utilized the offline analysis mode, namely all groups of transfer samples were used to learn all models simultaneously. The 9D feature vector consists of steady-state responses of nine gas sensors, followed by dynamic SNV described in Sect. 7.3.3. Owing to the instability of the sensor GSBT11 (see Table 13.2), we have replaced it with TGS2603 from Figaro Inc.

Experimental results are listed in Table 7.3. Solutions based on labeled target samples are impractical in this case because of the difficulty in collecting breath

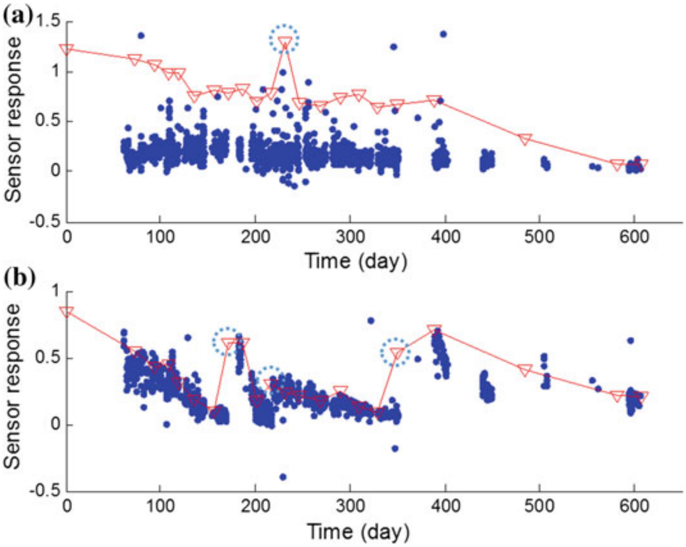


Fig. 7.8 Responses of two sensors in all breath samples (*blue dots*) and one transfer sample (*red triangles*). Each point represents the steady response of the sensor in one sample. *Dashed circles* mark the outlier in transfer samples (plot (**a**)) or the replacement of the sensor (plot (**b**))

samples from patients. Therefore, only transfer-sample-based methods were tested in this section. The parameters of each method were tuned by grid search. For methods except TMTL and "random train + TMTL", LR was adopted as the classifier. Multiplicative drift correction (MDC) (Artursson et al. 2000) is a simplified version of variable standardization which corrects each variable with a multiplicative factor. It performed better than variable standardization in this dataset. However, the two transfer-sample-based feature-level correction methods, MDC and CC-PCA, showed little improvement over "no transfer." For TMTL, TMTL-general with the dynamic model strategy was applied since the exact acquisition time of each sample is known. 45 models were learned simultaneously, as there were 45 groups of transfer samples altogether. The time-specific model for each training or test sample is a combination of neighboring models. The window

Table 7.3 F-score of the classification tasks on the breath analysis dataset

	Task 1	2	3	4	5	Average
No transfer	66.29	68.61	73.04	61.06	65.85	66.97
MDC	74.03	72.53	85.50	50.67	54.18	67.38
CC-PCA	70.09	68.00	76.40	64.90	74.97	70.87
TMTL	**76.17**	**84.38**	**87.80**	**77.79**	**82.77**	**81.78**
Random train	87.45	81.87	86.30	80.35	80.16	83.22
Random train + TMTL	95.80	87.70	89.11	83.88	84.84	88.27

Bold values indicate the best results

size parameter in Eq. 7.10 was empirically set to 10^{-4}. We find that this strategy is important for the dataset. If it is not used and each sample is predicted by an individual adjacent model, the accuracy will be poor. The noise in transfer samples could be the major cause. The combined model can smooth the noise. A minor drawback is that it cannot deal with "abrupt drift," e.g., sensor replacement. The accuracy of TMTL is close to "random train", in which the 50 training samples of each class were randomly selected from all devices and time periods to include the information of drift in the model. If we use TMTL with randomly select training samples, the accuracy can be further improved, indicating that TMTL can reduce the influence of drift effectively with the information contained in the transfer samples.

7.5.3 Corn Dataset

The corn dataset is a publicly available dataset in spectroscopy.[2] Three near-infrared spectrometers designated as m5, mp5, and mp6 were involved. Each device was adopted to measure the moisture, oil, protein, and starch contents of 80 corn samples. The ranges of the measured values are 9.377–10.993, 3.088–3.832, 7.654–9.711, and 62.826–66.472, respectively. The wavelength range is 1100–2498 nm at 2 nm intervals, resulting in 700 variables for each sample. Figure 7.9 illustrates the variation in distribution of the same samples measured by the three devices.

We follow the experimental setting in Yu and Ji (2015) and study the calibration transfer from m5 to the other two devices. A fourfold cross-validation was made by assigning every fourth sample to the test set. In each fold, the transfer samples or labeled target samples were selected by LLR from the training samples. Before training, each spectrum was first down-sampled to form a feature vector with 234 variables, followed by preprocessing with SNV. The four measured values were predicted separately and an average RMSE was computed. Table 7.4 lists the results on the two target devices when different number of transfer sample/labeled target samples were used. The parameters were tuned by grid search for each result. The RMSE of "RMTL (SVR)" are copied from Yu and Ji (2015), which only provided the results on mp6. TMTL-parallel was applied in the experiment. It achieves the best performance when the number of auxiliary samples (n_t) is small. RMTL (SVR) has smaller RMSE on mp6 when the n_t is larger than 15, which is probably because the labeled-sample-based method can extract more information from the additional labeled samples, whereas information brought by the additional transfer samples is marginal when n_t is large (can also be observed from Fig. 7.6). RMTL (SVR) also benefits from an ε-insensitive loss function with RBF kernel. It will be our future work to equip our methods with more powerful loss functions and

[2] http://www.eigenvector.com/data/Corn/.

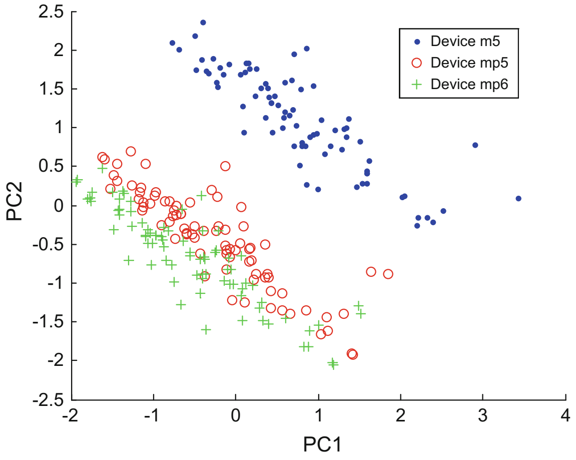


Fig. 7.9 Scatter plot of the samples measured by the three spectrometers. The samples are projected to a 2D subspace using PCA

Table 7.4 Average RMSE on the corn dataset with different number of auxiliary samples

# Auxiliary samples	Mp5 as target device				Mp6 as target device			
	7	10	15	20	7	10	15	20
No transfer	1.242				1.347			
Only SNV	0.220	0.216	0.227	0.224	0.231	0.224	0.237	0.231
Variable standardization	0.219	0.215	0.220	0.214	0.231	0.225	0.230	0.224
DAELM-S	0.213	0.217	0.200	0.206	0.222	0.227	0.207	0.216
RMTL (squared loss)	0.201	0.197	0.213	0.206	0.204	0.193	0.207	0.204
RMTL (SVR)	–				0.210	0.202	**0.181**	**0.177**
TCTL	0.196	0.189	0.196	**0.186**	0.194	0.184	0.190	0.181
TMTL-parallel	**0.186**	**0.183**	**0.190**	0.189	**0.188**	**0.182**	0.191	0.191
Train on target	0.185				0.189			

Bold values indicate the best results

kernels. For “train on target”, regression models were trained and tested on the same (slave) device. It can be regarded as an objective result for calibration transfer. We find that with the help of only 10 transfer samples, TMTL can actually outperform it.

7.6 Summary

We propose transfer-sample-based multitask learning (TMTL) to address the drift problem of e-noses. By drift we refer to the change of posterior data distribution caused by instrumental variation, sensor aging, environmental change, etc. Instead of using conventional methods to correct the drifted signals, our method handles drift under the framework of transfer learning and MTL. The key idea of our method is to reduce the influence of drift in the target domains by aligning the transfer samples in the model level. In this chapter, we have three observations:

(1) Different from existing MTL methods depending on labeled or unlabeled target samples, TMTL leverages transfer samples to transfer knowledge from source to target domains. In our experiments, it achieved better results, and the number of transfer samples needed for effective transfer was usually small (about 10). Besides, transfer samples are not required to be of the same type with the training and test samples. Thus, the proposed method is more convenient to use in many real-world applications.

(2) TMTL learns models for multiple target domains jointly. It is always better than its basic version, TCTL, which only involves one target domain. This confirms that TMTL has organized the models in a proper way so that they can improve each other.

(3) In the cases of time-varying drift, the serial transfer paradigm is better because it can capture the temporal smoothness prior. The dynamic model strategy is feasible when the acquisition time of each sample is known and when the transfer samples contain noise.

Overall, TMTL is a practical algorithm framework to predict data with complex drift caused by various factors. The robustness of e-noses can be greatly enhanced. Future works may include making more sophisticated assumptions on the structures of the models and features. It will also be beneficial to further exploit the information contained in unlabeled target samples.

References

Artursson T, Eklöv T, Lundström I et al (2000) Drift correction for gas sensors using multivariate methods. J Chemometr 14:711–723

Caruana R (1997) Multitask learning. Mach Learn 28:41–75

Di Carlo S, Falasconi M (2012) Drift correction methods for gas chemical sensors in artificial olfaction systems: techniques and challenges. In: Wang W (ed) Advances in chemical sensors. InTech, pp 305–326

Evgeniou T, Pontil M (2004) Regularized multi-task learning. In: Proceedings of ACM SIGKDD. ACM, Seattle, Washington, pp 109–117

Feudale RN, Woody NA, Tan H et al (2002) Transfer of multivariate calibration models: a review. Chemometr Intell Lab Syst 64:181–192

Gama J, Žliobaitė I, Bifet A et al (2014) A survey on concept drift adaptation. ACM Comput Surv (CSUR) 46:44

Kadlec P, Grbić R, Gabrys B (2011) Review of adaptation mechanisms for data-driven soft sensors. Comput Chem Eng 35:1–24

Kennard RW, Stone LA (1969) Computer aided design of experiments. Technometrics 11:137–148

Liu Q, Li X, Ye M et al (2014) Drift compensation for electronic nose by semi-supervised domain adaption. IEEE Sensors J 14:657–665

Marco S, Gutiérrez-Gálvez A (2012) Signal and data processing for machine olfaction and chemical sensing: a review. IEEE Sensors J 12:3189–3214

Padilla M, Perera A, Montoliu I et al (2010) Drift compensation of gas sensor array data by orthogonal signal correction. Chemometr Intell Lab Syst 100:28–35

Pan SJ, Kwok JT, Yang Q et al (2007) Adaptive localization in a dynamic WiFi environment through multi-view learning. In: The national conference on artificial intelligence. AAAI Press, MIT Press, Menlo Park, CA, Cambridge, MA, London, p 1108

Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22:1345–1359

Rodionova OY, Pomerantsev AL (2008) Subset selection strategy. J Chemometr 22:674–685

Rodriguez-Lujan I, Fonollosa J, Vergara A et al (2014) On the calibration of sensor arrays for pattern recognition using the minimal number of experiments. Chemometr Intell Lab Syst 130:123–134

Romain A-C, Nicolas J (2010) Long term stability of metal oxide-based gas sensors for e-nose environmental applications: an overview. Sensors Actuators B: Chem 146:502–506

Vergara A, Vembu S, Ayhan T et al (2012) Chemical gas sensor drift compensation using classifier ensembles. Sensors Actuators B: Chem 166:320–329

Wang X, Zhang D (2010) An optimized tongue image color correction scheme. IEEE Trans Inf Technol Biomed 14:1355–1364

Wilson AD, Baietto M (2011) Advances in electronic-nose technologies developed for biomedical applications. Sensors 11:1105–1176

Wold S, Antti H, Lindgren F et al (1998) Orthogonal signal correction of near-infrared spectra. Chemometr Intell Lab Syst 44:175–185

Yan K, Zhang D (2015) Improving the transfer ability of prediction models for electronic noses. Sensors Actuators B: Chem 220:115–124

Yan K, Zhang D (2016) Calibration transfer and drift compensation of e-noses via coupled task learning. Sensors Actuators B: Chem 225:288–297

Yan K, Zhang D, Wu D et al (2014) Design of a breath analysis system for diabetes screening and blood glucose level prediction. IEEE Trans Biomed Eng 61:2787–2795

Yu B, Ji H (2015) Near-infrared calibration transfer via support vector machine and transfer learning. Anal Methods 7:2714–2725

Yu K, Bi J, Tresp V (2006) Active learning via transductive experimental design. In: Proceedings of ICML. ACM, Pittsburgh, pp 1081–1088

Zhang L, Chen C, Bu J et al (2011a) Active learning based on locally linear reconstruction. IEEE Trans Pattern Anal Mach Intell 33:2026–2038

Zhang L, Tian F, Kadri C et al (2011b) On-line sensor calibration transfer among electronic nose instruments for monitoring volatile organic chemicals in indoor air quality. Sensors Actuators B: Chem 160:899–909

Zhang L, Zhang D (2015) Domain adaptation extreme learning machines for drift compensation in e-nose systems. IEEE Trans Instrum Meas 64:1790–1801

Zhou J, Chen J, Ye J (2011a) MALSAR: multi-task learning via structural regularization

Zhou J, Yuan L, Liu J et al (2011b) A multi-task learning formulation for predicting disease progression. In: Proceedings of ACM SIGKDD. ACM, San Diego, California, pp 814–822

Chapter 8
A Transfer Learning Approach for Correcting Instrumental Variation and Time-Varying Drift

Abstract In this chapter, we propose drift correction autoencoder (DCAE) to deal with instrumental variation and time-varying drift of e-noses. DCAE learns to model and correct these influential factors explicitly with the help of transfer samples. It generates drift-corrected and discriminative representation of the original data, which can then be applied to various prediction algorithms. Experimental results show that DCAE outperforms typical drift correction algorithms and autoencoder-based transfer learning methods. In particular, it is better than TMTL in the last chapter in datasets with complex drift, at the cost of longer training time and more hyper-parameters.

Keywords Autoencoder · Drift correction · Transfer learning

8.1 Introduction

To deal with instrumental variation and time-varying drift, an intuitive and commonly used idea is to transform the features in target domains to match those in the source one, so that the transformed target samples can be predicted by the models trained in the source domain (Zhang et al. 2011b; Yan and Zhang 2015; Feudale et al. 2002; Artursson et al. 2000; Padilla et al. 2010; Romain and Nicolas 2010). Here, following the terms in transfer learning (Pan and Yang 2010), we assume that the training samples are drawn from a source domain (e.g., collected with the old device or in the initial time period), whereas the test samples are drawn from target domains (e.g., collected with new devices or in later time periods). Efforts have also been made in the model level for drift correction. Prediction models suitable for the target samples were learned based on labeled source samples and a few labeled (Vergara et al. 2012; Zhang and Zhang 2015; Binfeng and Haibo 2015) or unlabeled (Liu et al. 2014) target samples or transfer samples (Yan and Zhang 2016a). These methods showed better accuracy than the feature-level ones.

Algorithms mentioned above operate on traditional hand-crafted features. In recent years, feature/representation learning methods based on deep networks have achieved promising results (Vincent et al. 2010; Bengio 2012; Bengio et al. 2013;

D. Zhang et al., *Breath Analysis for Medical Applications*,
DOI 10.1007/978-981-10-4322-2_8

Längkvist et al. 2014). These methods make use of the plentiful unlabeled data to learn representative features, whose discriminative power can be further enhanced by supervised fine-tuning. The adoption of nonlinear activation functions and multilayer stacking strategy enables the learned features to capture complex structures in the data. In the field of machine olfaction (e-noses), the pioneer works by Längkvist et al. (2013), Längkvist and Loutfi (2011) have shown that features learned by an autoencoder or restricted Boltzmann machines (RBMs) outperform traditional e-nose features. Transfer learning with deep networks has been discussed in Bengio (2012), Glorot et al. (2011), Zhou et al. (2014), Zhuang et al. (2015), Deng et al. (2014), Kandaswamy et al. (2014), Chopra et al. (2013). However, there is still no literature working on explicitly correcting instrumental variation and time-varying drift in sensor systems with deep networks. Furthermore, most transfer learning methods are designed for discrete source and target domains, whereas in the problem of time-varying drift, it is not easy to split data into such domains because the drift is continuous in time.

In this chapter, we propose drift correction autoencoder (DCAE) for joint representation learning and drift correction (Yan and Zhang 2016b). Besides the original features, the "domain features" are also inputted into DCAE, which contain the information about when and with which device the sample was collected. They make it convenient for DCAE to handle both the discrete drift among devices and the continuous drift over time. DCAE explicitly models the influence of these factors to the learned representation with the aid of transfer samples. A correction layer is used to further enhance DCAE's ability to correct complex time-varying drift. The hidden representation of DCAE is drift-corrected and can then be applied to various prediction algorithms. The supervised term in its objective function makes the representation to be discriminative as well.

The rest of the chapter is organized as follows. Related work on autoencoders is briefly reviewed in Sect. 8.2. Section 8.3 describes the proposed DCAE in detail. The experimental configurations and results are presented in Sect. 8.4, along with some discussions. Section 8.5 summarizes the chapter.

8.2 Related Work

8.2.1 Autoencoder

In this chapter, we will use the letter m to denote the number of features, h for the number of hidden units in the network, and n for the number of samples. The basic framework of an autoencoder is essentially a feed-forward neural network with one hidden layer. It takes an input vector $\boldsymbol{x} \in \mathbf{R}^m$, encodes it to a new representation $\boldsymbol{z} \in \mathbf{R}^h$, and then decodes $\boldsymbol{z}$ to $\hat{\boldsymbol{x}} \in \mathbf{R}^m$ in the original space. The encoding and decoding process can be formulated as:

$$z = \sigma(W\boldsymbol{x} + \boldsymbol{b}), \tag{8.1}$$

$$\hat{\boldsymbol{x}} = \sigma(W'z + \boldsymbol{b}'), \tag{8.2}$$

where $W \in \mathbf{R}^{h\times m}$ and $W' \in \mathbf{R}^{m\times h}$ are the weight matrices, $\boldsymbol{b} \in \mathbf{R}^{h}$ and $\boldsymbol{b}' \in \mathbf{R}^{m}$ are the bias vectors. σ is an activation function such as sigmoid, hyperbolic tangent (tanh), or linear (i.e. using an identity function). The objective of an autoencoder is to minimize the reconstruction error (Bengio et al. 2013):

$$J_{AE}(W, \boldsymbol{b}, W', \boldsymbol{b}') = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|^2, \tag{8.3}$$

where n is the number of training samples. When the number of hidden units h is less than the dimension of the original feature vector m, there will be a "bottleneck" in the network and a compressed representation of the original features will be learned. Note that one can use a tied weight (Bengio et al. 2013) by defining $W' = W^{\mathrm{T}}$, which can reduce the number of parameters to estimate.

One variant of the basic autoencoder is the denoising autoencoder (Vincent et al. 2010). It first corrupts the input $\boldsymbol{x}$ into $\tilde{\boldsymbol{x}}$ by randomly setting some of the features to zero. After obtaining the reconstruction of $\tilde{\boldsymbol{x}}$ from Eqs. 8.1 and 8.2, one minimizes the difference between it and the clean $\boldsymbol{x}$. It is expected that under this strategy, the learned representation will be more stable and robust to corruptions of the input, as well as capture more useful structures in the input distribution (Vincent et al. 2010). The single-layer autoencoders can also be stacked into deeper ones to obtain more abstract representation (Bengio et al. 2013). The hidden representation of an outer autoencoder serves as the input to an inner one. The hidden representation of the innermost one is regarded as the final learned representation, which can then be applied to various prediction algorithms.

8.2.2 Transfer Learning with Autoencoders

When the samples are collected from multiple domains, transfer learning with autoencoders can be considered (Bengio 2012; Glorot et al. 2011; Zhou et al. 2014; Zhuang et al. 2015; Deng et al. 2014; Kandaswamy et al. 2014; Chopra et al. 2013). For example, (Kandaswamy et al. 2014) proposed to reuse part of the weights of the network trained with source data and fine-tune the other part with labeled target samples. There are many other works that do not need label information in target domains. Glorot et al. (2011) and Bengio (2012) suggested that the learned features could generalize in multiple domains if one trains an autoencoder with unlabeled data from all these domains. Chopra et al. (2013) presented an interesting idea to learn an universal representation across domains. They constructed intermediate domains between the source and the target ones by merging different proportion of samples from the two domains. The final representation was the concatenation of the features learned from each domain, which contained the information of the distributional shift across domains. In the work by Deng et al. (2014), the knowledge in target domains is transferred to the source one by a similarity constraint of the weights.

Although these three methods can make the learned features capture the information in multiple domains, they may encounter difficulties in the scenario of drift correction, because the difference in distribution across domains is not explicitly reduced. The strategies used in Zhou et al. (2014), Zhuang et al. (2015), Kan et al. (2014) tried to reduce the difference. Zhou et al. (2014) first learned features from both domains separately with autoencoders, then computed a transformation matrix between the learned features according to a set of "cross-domain parallel data". This method is similar to variable standardization in machine olfaction. In Zhuang et al. (2015), the mean of the learned feature vectors of both domains were required to have small KL divergence. Kan et al. (2014) designed a network to learn pose-corrected features for face recognition. In the training process, face images with pose variations were used to reconstruct those without pose variations. The training images can actually be viewed as transfer samples.

8.3 Drift Correction Autoencoder (DCAE)

8.3.1 Domain Features

Most transfer learning algorithms (Pan and Yang 2010) split data into one or more source domains and one or more target ones. Labeled samples are sufficient in source domains, but scarce or not available in target ones. In drift correction problems, data without and with drift are often regarded as source and target domains, respectively (Zhang and Zhang 2015). For example, when the data are collected with different devices, each device defines a domain. In the case of time-varying drift, one can split the data into several batches in chronological order and treat each batch as a domain. Within each domain, drift is small. Prediction models need to be transferred from the source domain (the initial device or batch) to the target domains (the subsequent devices or batches).

However, in the most general cases, data are collected in continuous streams. The amount of drift in each sample is different. This information of continuity will be lost if we force the data into batches. Therefore, instead of putting data into different domains according to which device it is from and when it was collected (acquisition time), we consider all data as a whole and design "domain features" for each sample to describe this information conveniently. If we only consider the instrumental variation, a one-hot coding scheme can be used. Suppose there are n_{dev} devices. The domain feature vector is thus $\boldsymbol{d} \in \mathbf{R}^{n_{\text{dev}}}$, where $d_i = 1$ if the sample is from the ith device and 0 otherwise. If the time-varying drift is also considered, the acquisition time can be further added into $\boldsymbol{d}$. Suppose a gas sample is collected from the ith device at time t, then $\boldsymbol{d} \in \mathbf{R}^{2n_{\text{dev}}}$ and

$$d_j = \begin{cases} 1, & j = 2i - 1, \\ t, & j = 2i, \\ 0, & \text{otherwise}. \end{cases} \tag{8.4}$$

8.3.2 Basic Framework

In order to correct instrumental variation and time-varying drift, transfer samples should be collected from each device periodically. They are viewed as representatives or milestones of the samples collected with the same device and one time period. It is natural to assign the first group of transfer samples collected by the first device as the reference group (source). All other groups collected with other devices or in later time periods (target) should be aligned with the reference. If the difference of representation is small between the reference and the other groups, one can expect that the drift has been reduced in the learned representation.

Four sets of data are used in the learning procedure of DCAE. $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^{n}$ contains both labeled and unlabeled data. The labeled data and their labels are included in $\mathcal{D}_L = \{(\boldsymbol{x}_j^{(L)}, \boldsymbol{y}_j^{(L)})\}_{j=1}^{n_L}$. n and n_L are the total number of samples and the number of labeled samples, respectively. $\mathcal{D}_S = \{\boldsymbol{t}_p^{(S)}\}_{p=1}^{n_{\text{t,total}}}$ and $\mathcal{D}_T = \{\boldsymbol{t}_q^{(T)}\}_{q=1}^{n_{\text{t,total}}}$ denote the source and target transfer samples, respectively. $n_{\text{t,total}}$ is the total number of transfer sample pairs. $\boldsymbol{x}, \boldsymbol{x}^{(L)}, \boldsymbol{t}^{(S)}, \boldsymbol{t}^{(T)} \in \mathbf{R}^m$. $\boldsymbol{y}^{(L)} \in \mathbf{R}^c$, where c is the number of labels. Each sample $(\boldsymbol{x}_i, \boldsymbol{x}_j^{(L)}, \boldsymbol{t}_p^{(S)}, \boldsymbol{t}_q^{(T)})$ has a corresponding domain feature vector $(\boldsymbol{d}_i, \boldsymbol{d}_j^{(L)}, \boldsymbol{d}_p^{(S)}, \boldsymbol{d}_q^{(T)})$.

To clarify the composition of $\mathcal{D}_S$ and $\mathcal{D}_T$, let us assume that each group of transfer samples is made up of $n_{\text{t,gas}}$ different gas samples, and $n_{\text{t,group}}$ groups were collected altogether. Thus, $\mathcal{D}_T$ is the stack of samples in all groups, whereas $\mathcal{D}_S$ is the stack of $n_{\text{t,group}}$ repetitions of the samples in the reference group. Therefore, each target transfer sample $\boldsymbol{t}_p^{(T)}$ has a corresponding source one $\boldsymbol{t}_p^{(S)}$. The total number of transfer sample pairs is $n_{\text{t,total}} = n_{\text{t,gas}} \times n_{\text{t,group}}$. DCAE can thus incorporate the information from all available transfer sample groups into the learning process.

On the basis of an autoencoder, we utilize a new weight matrix to correct drift. It is denoted as $W_D \in \mathbf{R}^{h \times m_d}$, where m_d is the length of the domain feature vector and h is the number of hidden units of the original autoencoder. The component of drift is explicitly removed from the hidden representation by adding $W_D\boldsymbol{d}$ in the encoding process, and recovered in the decoding process by subtracting it:

$$z = f(\boldsymbol{x}, \boldsymbol{d}) = \sigma(W\boldsymbol{x} + W_D\boldsymbol{d} + \boldsymbol{b}), \tag{8.5}$$

$$\hat{\boldsymbol{x}} = g(z, \boldsymbol{d}) = \sigma(W'(z - W_D\boldsymbol{d}) + \boldsymbol{b}'). \tag{8.6}$$

Through the equations above, drift correction in the hidden representation and data reconstruction can both be accomplished. If a linear activation function is used, the reconstruction will be identical to that in the original autoencoder.

The objective function of DCAE is expressed as follows:

$$
\begin{aligned}
J_{DCAE}(W,\boldsymbol{b},W',\boldsymbol{b}',W_D,W_S,\boldsymbol{b}_S) = &\\
&\frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{x}_i-g(f(\boldsymbol{x}_i,\boldsymbol{d}_i),\boldsymbol{d}_i)\|^2\\
&+\frac{\lambda_1}{n_L}\sum_{j=1}^{n_L}\mathcal{L}\left(f(\boldsymbol{x}_j^{(L)},\boldsymbol{d}_j^{(L)}),\boldsymbol{y}_j^{(L)};W_S,\boldsymbol{b}_S\right)\\
&+\frac{\lambda_2}{n_{\mathrm{t,total}}}\sum_{p=1}^{n_{\mathrm{t,total}}}\|f(\boldsymbol{t}_p^{(S)},\boldsymbol{d}_p^{(S)})-f(\boldsymbol{t}_p^{(T)},\boldsymbol{d}_p^{(T)})\|^2
\end{aligned}
\tag{8.7}
$$

It can be decomposed into three parts: the first term is the reconstruction error of all data; the second term is the supervised loss of the labeled samples; the third term is the alignment error of the transfer samples. λ_1, λ_2 and μ are regularization hyperparameters. The weight matrix $W_S \in \mathbf{R}^{c\times h}$ and bias vector $\boldsymbol{b}_S \in \mathbf{R}^c$ are used in the supervised model. By minimizing Eq. 8.7, all weight matrices and bias vectors are learned simultaneously.

The first term of Eq. 8.7 is similar to the objective of the original autoencoder, except that the functions f and g are those defined in Eqs. 8.5 and 8.6, which take drift into consideration. The second term incorporates the label information into the function. The loss function $\mathcal{L}$ can be determined according to the problem. For example, the softmax function (Zhuang et al. 2015) can be used for classification problems, whereas the squared error function can be used for regression ones.

The third term is key for drift reduction in DCAE. It requires the corresponding source and target transfer samples to have similar hidden representations. The correction weight matrix W_D will be learned mainly base on this term. W and $\boldsymbol{b}$ will also be influenced to extract features that are more robust to drift. The term has similar insight with the transfer sample term of TMTL in the last chapter. An alternative strategy to learn W_D is using the drift-corrected hidden representation of the target transfer samples to reconstruct the corresponding source ones, which is similar to the strategy in Kan et al. (2014). The accuracy of this strategy is not as good as that of the proposed one, which is probably because an trained autoencoder only has low reconstruct error on certain data (Bengio et al. 2013), which are those in $\mathcal{D}$. However, transfer samples are not always of the same type with data in $\mathcal{D}$. For example, one is standard gas and the other is breath sample. Therefore, the transfer samples may not be well reconstructed.

8.3.3 Handling Complex Time-Varying Drift

For a sample with domain feature $\boldsymbol{d}$, the amount of correction received by the ith hidden unit is the ith element of $W_D\boldsymbol{d}$. This value is device-specific and linearly proportional to the acquisition time (see the coding scheme of $\boldsymbol{d}$ in Sect. 8.3.1). However, the relationship between time and the time-varying drift in most real-world applications is nonlinear. The sensitivity characteristic of gas sensors can be affected by many factors such as aging effect, humidity, temperature, background change, and sensor replacement (Marco and Gutiérrez-Gálvez 2012). Thus, the drift can display a complex pattern, which makes the linear correction insufficient. To enhance the correction ability, we further insert a correction layer between the domain feature vector and the hidden layer in the basic DCAE described in the last section. The encoding and decoding process thus becomes:

$$z = f(\boldsymbol{x}, \boldsymbol{d}) = \sigma(W\boldsymbol{x} + W_{D1}\,\sigma_{\text{cor}}(W_{D0}\boldsymbol{d}) + \boldsymbol{b}), \tag{8.8}$$

$$\hat{\boldsymbol{x}} = g(z, \boldsymbol{d}) = \sigma(W'(z - W_{D1}\,\sigma_{\text{cor}}(W_{D0}\boldsymbol{d})) + \boldsymbol{b}'). \tag{8.9}$$

In the equation above, $W_{D0} \in \mathbf{R}^{h_{\text{cor}} \times m_d}$ is the weight matrix from the domain features to the correction layer, where the number of units in the correction layer is h_{cor}. $W_{D1} \in \mathbf{R}^{h \times h_{\text{cor}}}$ is the weight matrix from the correction layer to the hidden layer of DCAE. σ_{cor} is the activation function of the correction layer. It should be set to a nonlinear one, as a linear one makes the correction linearly related to time, which is equivalent to the basic DCAE. Ideally, the correction output $W_{D1}\,\sigma_{\text{cor}}(W_{D0}\boldsymbol{d})$ should compensate the drift of each hidden unit with regard to time and device. There is no bias vector in the correction layer. The 1's in the domain features in Eq. 8.4 can be viewed as device-specific constant terms, so the bias vector is merged into W_{D0}. The number of units in the correction layer h_{cor} should be selected according to the complexity of the time-varying drift. A larger h_{cor} is preferable for more complex drift.

Both the basic DCAE and that with correction layer explicitly models the influence of instrumental variation and time-varying drift by the weight matrix W_D (or W_{D0} and W_{D1}). The parameters for different devices are stored in different columns of W_D or W_{D0}. Similar to the stacked autoencoders, the number of hidden layers in DCAE can be increased to extract more abstract representation. The correction output is added before the first hidden layer and removed after the last hidden layer, so that the representations learned in the hidden layers are all drift-corrected.

8.3.4 Summary

Figure 8.1 illustrates the architecture of a DCAE with correction layer and three hidden layers. The solid arrows indicate the flow of signals. The original features are input to the first layer of the main body of the network (in blue), whereas the domain features are input to the first layer of the correction part (in orange). The output of the correction layer is added to the input of the second layer of the main body, in the meantime subtracted from the output of the fourth (the second to the last) layer.

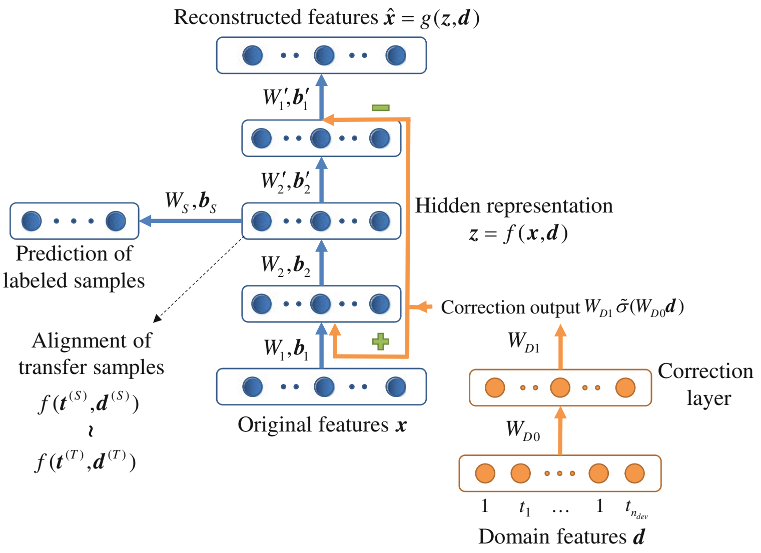


Fig. 8.1 Architecture of a DCAE with correction layer and three hidden layers

The hidden representation is applied to predict for the labeled samples and align the transfer samples, see the objective function Eq. 8.7.

It is important to carefully initialize the weights in deep networks. The greedy layerwise unsupervised pretraining strategy (Bengio et al. 2013) is widely used for stacked autoencoders. We first pretrain a stacked denoising autoencoder with all samples in $\mathcal{D}$. The fraction of corrupted input is set to 10% in this chapter. The weight tying strategy is used. Then, supervised fine-tuning (Vincent et al. 2010) is performed with the labeled data in $\mathcal{D}_L$. This step minimizes the supervised loss of the labeled data. Although a supervised loss term has been included in the objective function of DCAE, we find that this fine-tuning step can further improve the accuracy because it can make the initialized weights in DCAE closer to the optimal solution. An experiment will be made in Sect. 8.4.4 to compare the impact of different training procedures to the accuracy. Finally, the weights and biases in the fine-tuned network are used to initialize $W, \boldsymbol{b}, W', \boldsymbol{b}'$ in DCAE. W_S, W_{D1} are randomly initialized. $\boldsymbol{b}_S'$, W_{D0} are initialized to zeros. After that, Eq. 8.7 can be optimized using methods such as gradient descent or conjugate gradient. Note that the denoising strategy is not used in DCAE.

Once the optimization is done, the labeled samples and their domain features can be input to DCAE. Their hidden representations and labels are used to train a separate prediction model, which is then applied on the hidden representations of the unlabeled samples to obtain the predicted labels. The entire process is summarized in Algorithm 8.3.1.

Algorithm 8.3.1 Drift correction autoencoder (DCAE)

```
Input: The unlabeled dataset $\mathcal{D}$, the labeled dataset $\mathcal{D}_L$, and the transfer sample datasets $\mathcal{D}_S$, $\mathcal{D}_T$.
    The device index and acquisition time of each sample are known.
Output: The predicted labels of samples in $\mathcal{D}$.
Pretrain a stacked denoising autoencoder unsupervisedly with $\mathcal{D}$, and then fine-tune it with $\mathcal{D}_L$;
Initialize the weights in DCAE based on the weights of the autoencoder;
Create the domain feature vector for each sample according to Eq. 8.4;
Optimize Eq. 8.7 with samples in $\mathcal{D}$, $\mathcal{D}_L$, $\mathcal{D}_S$, $\mathcal{D}_T$ and their domain features to obtain the weights
    of DCAE;
Train a prediction model with the hidden representations of $\mathcal{D}_L$, and then apply it on the hidden
    representations of $\mathcal{D}$.
```

8.4 Experiments

In this section, we conduct experiments on the three datasets same with the last chapter. Locally linear reconstruction (LLR) (Zhang et al. 2011a) was employed for transfer sample selection, as it has been proved effective in the last chapter. The proposed method was implemented based on the Theano library (Bergstra et al. 2010). The optimization algorithm was conjugate gradient with the maximum iteration number set to 1000. No other strategies were used in the optimization process.

8.4.1 Gas Sensor Array Drift Dataset

Readers are directed to Sect. 7.5.1 for the detail of the gas sensor array drift dataset. Still, we assume that the labels in batch 1 are known, whereas those in batches 2–10 are to be predicted. The time-varying drift across batches can be visually inspected in Fig. 8.2a. Samples in two classes are projected to a 2D subspace using PCA. It can be found that there is an obvious drift for samples in both classes as time elapses. Therefore, if the prediction model trained on batch 1 is applied to classify samples in later batches, the accuracy will degrade. It is worth noticing that the direction in which the two classes can be discriminated is close to the direction of drift (along the y-axis). In such cases, correction methods that seeking to find a drift-free latent space or trying to remove the component of drift may suffer a loss of useful information.

We developed a DCAE with three hidden layers (one hidden layer between the input layer and the hidden representation layer). The numbers of units were 30, 20, and 30, respectively. Further increasing the number of layers or units cannot improve the accuracy. The tanh activation function was used. In the dataset, each sample is represented by 128 features extracted from the sensors' response curves (Vergara et al. 2012). The original features have unbounded values that cannot be well reconstructed by the bounded tanh function. Thus, before it was inputted into the network, each feature was normalized to have zero mean and unit variance, then divided by 2 to make most feature values range from -1 to 1. Directly mapping the minimum and

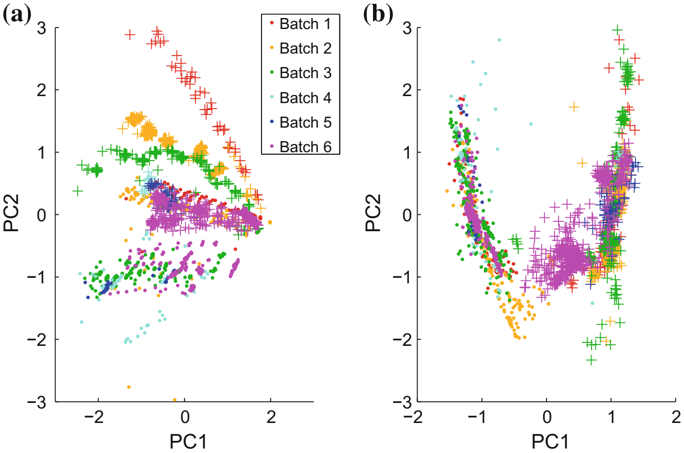


Fig. 8.2 Illustration of the drift across batches 1–6 in the gas sensor array drift dataset. Dots and plus signs represent ammonia and acetone samples, respectively. Different colors indicate different batches. The original features were used in plot (**a**) whereas the learned representation of DCAE were used in plot (**b**)

maximum values to −1 and 1 was not considered because of the disturbance of outlier samples. The domain features were created according to Eq. 8.4. In this dataset, the number of devices $n_{\text{dev}} = 1$, and the acquisition time t was defined as the batch index minus one, e.g., $t = 5$ for a sample in batch 6.

The transfer samples are not directly provided in the dataset, hence need to be selected for each batch. The candidate set of batch k ($k = 2, \dots, 10$) was defined as the overlapping samples in batch 1 and k, namely the samples of the same gas and concentration. Then, LLR was used to sequentially select $n_{\text{t,gas}}$ transfer samples from each candidate set. Following Algorithm 8.3.1, the weights of a pretrained and fine-tuned denoising stacked autoencoder were used to initialize the DCAE. Then, for each batch k ($k = 2, \dots, 10$), we used the labeled samples in batch 1 and the transfer sample groups from batch 1 to k to train the DCAE. After that, the hidden representations of batch 1 were adopted to train logistic regression classifiers with the one-vs-all strategy for multi-class classification. The hidden representations of the samples in batch k were predicted by the classifiers and an accuracy was computed. Plot (b) of Fig. 8.2 displays the scatter of the hidden representations of the same group of samples as Fig. 8.2a. We can find that the drift across batches has been greatly reduced and samples in each class are better clustered.

The proposed method has two variants: the basic DCAE (DCAE-basic) and DCAE with correction layer (DCAE-CL). In the latter variant, we set the activation function of the correction layer to be tanh as well. There are three major hyper-parameters in DCAE-CL, i.e., λ_1, λ_2, and the number of units in the correction layer h_{cor}. λ_1 controls the weight of the supervised loss term. λ_2 controls the weight of the transfer sample alignment error term. Larger h_{cor} brings higher capability in

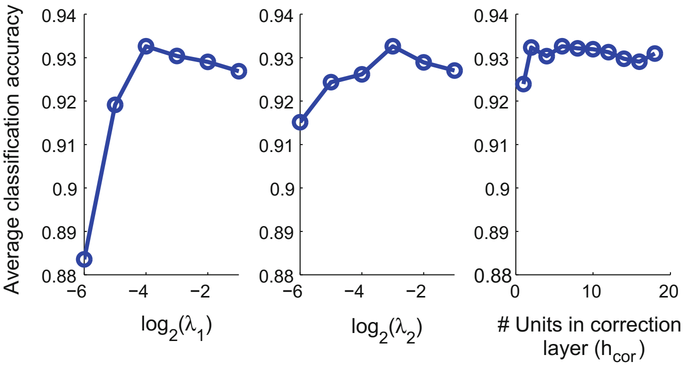
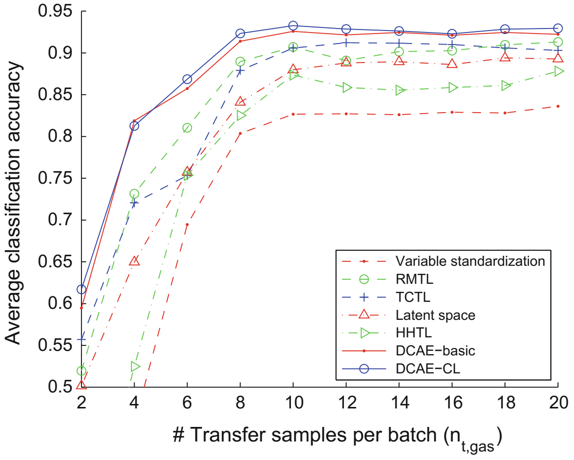


Fig. 8.3 Impact of the hyper-parameters on the average classification accuracy of DCAE-CL in the gas sensor array drift dataset

Fig. 8.4 Performance comparison on the gas sensor array drift dataset



correcting complex time-varying drift. Their influence to the performance of DCAE-CL is investigated in Fig. 8.3. The number of transfer samples in each batch is $n_{\mathrm{t,gas}} = 10$. When one hyper-parameter was tuned, the others were fixed. Owing to the random factor during the initialization of the networks, each experiment was repeated 10 times with 10 random seeds. The y-axis is the average accuracy of the 10 runs, where each accuracy is the average one on batches 2–10. It can be observed that when h_{cor} is not smaller than 2, the accuracy is relatively stable. Finally we adopt $\lambda_1 = 2^{-4}$, $\lambda_2 = 2^{-3}$ and $h_{\mathrm{cor}} = 6$.

The proposed methods are compared with typical existing methods in Fig. 8.4 with regard to different numbers of transfer samples ($n_{\mathrm{t,gas}}$). Variable standardization (Yan and Zhang 2016a) transforms the original features in batches k ($k = 2, \ldots, 10$) to batch 1 using ridge regression based on transfer samples. Regularized multitask learning (RMTL) (Binfeng and Haibo 2015) jointly learns two models for batches 1 and k respectively based on the labeled samples in both batches. The transfer samples in batch k were used as labeled samples in this method. On the other hand, transfer-sample-based coupled task learning (TCTL) (Yan and Zhang 2016a) is a multitask learning framework that seeks to align the transfer samples in the model level. Transfer learning with deep autoencoders (TLDA) (Zhuang et al. 2015) learns a common representation for the source and the target domains (batches 1 and k). It requires that the means of representations in the two domains are close. However, the method is suitable only when each domain has similar data composition. Moreover, transfer samples can provide much more information than the mean of each domain. Hence, we modify the method by replacing the mean of representations with transfer samples. We name the modified method as "latent space", because it actually assumes a latent representation space in which samples in different domains have similar distribution. The only difference between latent space and DCAE is that domain features are not used in the former method. Hybrid heterogeneous transfer learning (HHTL) (Zhou et al. 2014) first uses marginalized stacked denoising autoencoders (mSDAs) to learn representations separately in each domain, then transform the target representation to the source one in the same way as in variable standardization. The representations learned in different layers are finally concatenated to an augmented feature vector. In all methods, the transfer samples were selected with LLR. The parameters of the autoencoders in latent space and HHTL were the same as those in DCAE. For the autoencoder-based methods, we report the average accuracy of 10 random runs.

From Fig. 8.4, we can find that the accuracy improves as $n_{\mathrm{t,gas}}$ increases, but reaches a plateau when $n_{\mathrm{t,gas}} = 10$. The performance of variable standardization and HHTL is not promising, which is because the capacity of the feature-level correction methods is insufficient when facing complex drift. The fact that HHTL is better than variable standardization implies the augmented feature vector learned by mSDA is better than the original features. DCAE outperforms latent space, because it explicitly models and corrects drift, thus avoids information loss caused by the tangle between drift and useful information. Similar to the model-level methods like RMTL and TCTL, DCAE considers the discriminative information when correcting drift. Better yet, it can capture nonlinear and more abstract structures in data. The learned representation can be used in various prediction models, which makes it more convenient to use. The two DCAE variants outperform other methods especially when $n_{\mathrm{t,gas}}$ is smaller.

More results of existing methods are listed in Table 8.1. "No transfer" means the prediction model trained on batch 1 is directly applied on batches k ($k = 2, \ldots, 10$). The unsatisfactory performance proves the influence of drift. The two traditional methods based on component correction (CC-PCA and OSC) do not achieve large improvement because they rely on clear separation of drift and useful information in data. The results of ensemble, source domain adaptation extreme learning machine (DAELM-S), and manifold regularization with combination geodesic flow kernel (ML-comGFK) are copied from the original papers. DAELM-S achieves good results with the help of 30 selected labeled samples in each batch. Our proposed methods have a higher accuracy with only 10 transfer samples needed. The last three

Table 8.1 Classification accuracy (%) on the gas sensor array drift dataset. Bold values indicate the best results

	Batch 2	3	4	5	6	7	8	9	10	Average
No transfer	88.59	66.96	40.99	54.82	43.22	44.40	31.63	45.74	39.11	50.61
CC-PCA	90.92	40.86	47.20	59.39	56.74	56.71	36.39	45.32	37.72	52.36
OSC	88.10	66.71	54.66	53.81	65.13	63.71	36.05	40.21	40.08	56.50
Ensemble	74.36	87.83	**93.79**	95.43	69.17	69.72	91.84	76.38	65.50	80.45
ML-comGFK	80.25	74.99	78.79	67.41	77.82	71.68	49.96	50.79	53.79	67.28
DAELM-S	87.98	95.74	85.16	95.99	94.14	83.51	86.90	**100.0**	53.62	87.00
TMTL-serial	97.35	**98.80**	90.06	98.48	95.35	91.50	91.84	96.38	71.56	92.37
Joint train	59.41	56.33	58.63	37.26	44.77	43.17	20.72	34.26	35.55	43.34 ± 1.14
DCAE-basic	**97.58**	96.82	88.57	**98.73**	**95.40**	94.76	91.19	95.66	74.60	92.59 ± 0.61
DCAE-CL	97.47	96.13	90.50	98.58	94.94	**95.47**	**92.62**	96.40	**76.82**	**93.21** ± 0.52

rows are results obtained by deep networks. The standard deviation of the average accuracy of the 10 runs is also calculated. In “joint train”, samples in batch 1 and k are pooled together to train a stacked denoising autoencoder, as suggested by Glorot et al. (2011). The poor performance indicates that this strategy is not suitable in drift correction. Among the results of DCAE-basic and DCAE-CL, the former method is better in earlier batches whereas the latter wins in latter batches which have larger drift. It indicates that DCAE-CL is preferable when the time-varying drift is more complex. DCAE broke the record of TMTL in the last chapter possibly because it can learn more abstract and nonlinear representation from data.

8.4.2 Breath Analysis Dataset

A breath analysis dataset was collected using two e-noses of the same model (Yan et al. 2014). Please see Sect. 7.5.2 for the introduction of this dataset. Only transfer-sample-based methods will be tested in this section. Five binary-class classification tasks (healthy vs. disease) were carried out on this dataset. F-score was used as the accuracy criterion. In order to evaluate the drift correction algorithms, we chose the first 50 samples collected with device 1 in each class for training. Each sample was represented by the steady-state responses of nine gas sensors. The features were preprocessed with the normalization method similar to that in Sect. 8.4.1. Logistic regression was adopted as the final classifier.

The DCAE developed for this dataset has one hidden layer with four units. Because the numbers of features and classes are not large, further enlarging the network cannot improve the accuracy. Tanh activation function was used in the network including the correction layer. The domain features were created according to Eq. 8.4, where t was the exact acquisition time converted to years and the number of devices $n_{\text{dev}} = 2$. The impact of the hyper-parameters on the average accuracy is studied in Fig. 8.5. It is found that the change of accuracy is not large when λ_1 is varied. The best value of λ_2 is 2^{-6}, which is smaller than 2^{-3} in the gas sensor array drift dataset. We attribute it to the noisy transfer samples in this dataset. Larger λ_2 enforces the transfer samples to be aligned better in the representation space, which will cause overfitting if the transfer samples are noisy and their quantity is small. For h_{cor}, the accuracy is relatively stable when it is not smaller than 4. Finally we adopt $\lambda_1 = 2^{-4}$, $\lambda_2 = 2^{-6}$ and $h_{\text{cor}} = 4$.

The accuracy of various methods is compared in Table 8.2. Note that some results are not identical with Table 7.3 in the last chapter because the preprocessing strategy is different. If the prediction models trained on the original features are directly used for classification, the average accuracy is only 68.30%. Among all the tested methods, DCAE has the best performance. DCAE-CL outperforms DCAE-basic in all tasks. The last row of the table shows the results obtained by randomly selecting 50 training samples in each class. The experiment was repeated 20 times and the average accuracy is reported. No drift correction was done. Since the training

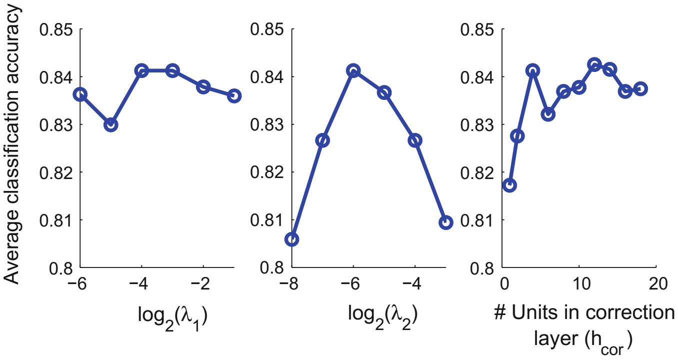


Fig. 8.5 Impact of the hyper-parameters on the average F-score of DCAE-CL in the breath analysis dataset

Table 8.2 Classification accuracy (%) on the breath analysis dataset. Bold values indicate the best drift correction results

	Task 1	2	3	4	5	Average
No transfer	63.20	71.49	75.87	64.36	66.55	68.30
Var. stdd.	52.13	49.58	65.85	47.88	45.47	52.18
CC-PCA	70.82	79.24	84.59	74.19	76.24	77.02
TMTL	76.17	**84.38**	87.80	77.79	82.77	81.78
Latent space	57.49	68.73	73.27	68.04	74.12	68.33 ± 2.37
DCAE-basic	74.44	82.20	89.36	81.15	82.04	81.84 ± 0.67
DCAE-CL	**82.16**	84.27	**89.94**	**81.34**	**82.92**	**84.13** ± 0.82
Random train	87.85	85.44	90.10	85.09	84.54	86.60

samples were from all devices and time periods, the trained model should be robust to drift. The accuracy of DCAE-CL is close to random train.

In the breath analysis dataset, the samples were collected in a stream. They drift in the data space continuously in both the training and the test set. TCTL and HHTL-Hybrid heterogeneous transfer learning (HHTL) are designed for problems with discrete source and target domains, so they are not feasible in this dataset. CC-PCA and latent space correct all data as a whole. Latent space tries to align each group of transfer samples to the reference group, which is not suitable when the transfer samples are noisy. This could be the reason why its performance is not good. We also tried to split the data into batches so that each batch had a group of transfer samples. A breath sample was assigned to the batch whose transfer sample group was closest to it in time. Then, variable standardization was applied for drift correction. The results are listed in the row “var. stdd.”. Its poor performance can also be caused by the

Table 8.3 Regression RMSE on the corn dataset. Bold values indicate the best calibration transfer results

	Mp5 as target domain					Mp6 as target domain				
	Moisture	Oil	Protein	Starch	Average	Moisture	Oil	Protein	Starch	Average
No transfer	1.499	0.288	1.097	2.086	1.242	1.632	0.343	1.350	2.063	1.347
Var. stdd.	0.269	0.106	0.147	0.354	0.219	0.303	0.110	0.153	0.356	0.231
DAELM-S	0.238	0.109	0.150	0.365	0.215 ± 0.001	0.266	0.113	0.154	0.368	0.225 ± 0.002
RMTL (SVR)	–					0.235	0.089	0.160	0.357	0.210
TCTL	**0.182**	0.104	0.148	0.351	0.196	0.193	0.104	0.143	**0.337**	0.194
TMTL	**0.182**	0.098	**0.127**	**0.339**	**0.186**	**0.190**	0.100	**0.130**	0.338	**0.189**
Latent space	0.191	0.095	0.162	0.370	0.205 ± 0.002	0.197	0.097	0.166	0.385	0.211 ± 0.001
HHTL	0.262	**0.092**	0.146	0.404	0.226	0.300	**0.091**	0.140	0.377	0.227
DCAE-basic	0.183	0.096	0.149	0.348	0.194 ± 0.002	0.199	0.096	0.148	0.350	0.198 ± 0.002
Train on target	0.137	0.101	0.140	0.363	0.185	0.139	0.099	0.149	0.369	0.189

noise in transfer samples, which makes the correction inaccurate. On the contrary, the correction output in DCAE is smooth over time as long as λ_2 is not too large. Two samples will be given similar correction output if their acquisition times are close. This is consistent with the prior knowledge that the time-varying drift usually changes slowly in time, which helps to smooth the noise in transfer samples. Besides, by using the domain features, DCAE can correct continuous drift intrinsically and without loss of the continuity information.

8.4.3 Corn Dataset

The dataset has been described in Sect. 7.5.3. The features were preprocessed with the normalization method similar to that in Sect. 8.4.1. We follow the experimental setting in Binfeng and Haibo (2015) and study the calibration transfer from m5 to the other two devices. A fourfold cross-validation was made by assigning every fourth sample to the test set. Regression models were trained on the training samples collected with m5 and tested on the test samples collected with mp5 and mp6. Linear ridge regression was adopted as the final regression algorithm. Root mean square error (RMSE) was used as the accuracy criterion.

DCAE-basic was used for this dataset since it contains no time-varying drift. Considering that the complexity of the dataset is relatively small, we adopted a network with one hidden layer and linear activation function. The number of units in the hidden layer was 15. The domain feature vectors were created according to the one-hot coding scheme in Sect. 8.3.1 with $n_{\mathrm{dev}} = 3$. Because the four values to predict have different variances, we normalized them to zero mean and unit variance before training DCAE, then transformed them back when computing RMSE. For this dataset, we found that small RMSE was obtained when setting $\lambda_1 = 2^1, \lambda_2 = 2^4$. The optimal hyper-parameters are larger than the previous datasets, which is because there is much less noise in this dataset. It is less prone to overfitting when the supervised loss and alignment error are given larger weights.

Results of various methods are compared in Table 8.3. For each calibration transfer algorithm except RMTL, LLR was used to select seven transfer samples (or labeled target samples) from the training data. The results of RMTL (SVR) are copied from Binfeng and Haibo (2015), which only provided the results on mp6. The “train on target” method trained and tested regression models on the same target device. Its results can be regarded as goals for calibration transfer. It can be found that TMTL won the first place and TCTL and DCAE-basic were second to it. It is easy to see why DCAE is not better than TMTL: this dataset is relatively small for a neural network with many parameters. Plus, the dataset only contains plain instrumental variation, whereas DCAE is more suitable for complex drift.

Table 8.4 Classification accuracy (%) or RMSE on the three datasets with different training procedures. Bold values indicate the best results

Procedure	Dataset 1	Dataset 2	Dataset 3 (RMSE)
Final(3000)	76.56 ± 2.36	81.28 ± 2.11	0.2096 ± 0.0014
Unsup.(1000) + final(2000)	75.83 ± 2.15	83.41 ± 2.94	0.2055 ± 0.0014
Sup.(1000) + final(2000)	93.04 ± 0.44	83.44 ± 0.89	0.1964 ± 0.0017
Unsup.(1000) + sup.(1000) + final(1000)	**93.21** ± 0.52	**84.13** ± 0.82	**0.1961** ± 0.0022

8.4.4 Impact of Different Training Procedures

According to Algorithm 8.3.1, the weights in DCAE are learned in three steps: unsupervised pretraining (unsup.), supervised fine-tuning (sup.), and final optimization of the objective function Eq. 8.7 (final). In order to explore the necessity of the two steps before the final optimization, we compare four different training procedures in Table 8.4. The number after the name of a step indicates its preset maximum iteration number. When one step is omitted, the maximum iteration number of the final optimization is increased for fair comparison.

As shown in the table, the proposed procedure achieves the best results in all three datasets. On the other side, directly optimizing the objective function is nearly always the worst one. Similar to the pretraining step in original autoencoders, the two steps before the final optimization drive the weights to approach a "good" local minimum, where "good" is in terms of generalization error (Bengio et al. 2013). Nevertheless, the weights obtained by directly optimizing Eq. 8.7 may have larger generalization error, even if it can obtain a smaller loss value. Besides, the actual iteration number of the proposed procedure is also the smallest among the four procedures, because both unsupervised pretraining and supervised fine-tuning are easier tasks than minimizing Eq. 8.7 and converge faster. In the experiment for batches 1 and 2 in the first dataset, the time costs for optimizing the three steps are 1.7, 7.3, and 48.1 s, respectively.

8.5 Summary

In this chapter, we proposed drift correction autoencoder (DCAE) for joint representation learning and drift correction in machine olfaction. The main difference between DCAE and the original stacked autoencoder lies in three aspects:

1. Domain features and the correction layer are introduced in DCAE to explicitly model instrumental variation and time-varying drift;
2. The component of drift is removed in the encoding process and recovered in the decoding one, so that the hidden representation of DCAE is drift-corrected;
3. Transfer samples are utilized to learn the correction weight matrices.

Compared with other drift correction algorithms, DCAE has several characteristics:

1. It tackles both discrete and continuous drift in the training and test data naturally without having to split the data into different domains;
2. The correction output in DCAE is intrinsically smooth over time, which is consistent with the temporal smoothness prior of the time-varying drift, so that the influence of noisy transfer samples can be diminished;
3. When correcting drift, DCAE keeps the learned representation to be representative and discriminative as well at the cost of two regularization hyper-parameters. According to our experiments, they can be set according to the noise level of the dataset and the transfer samples.

The superiority of DCAE is more obvious when the drift in the dataset is complex (e.g., Sects. 8.4.1 and 8.4.2). Besides, when the size of the dataset is not large (which is common in machine olfaction), it may be better to use a smaller network. The application scope of DCAE may be further extended. By simply modifying the domain features, other influential factors such as temperature and humidity can also be corrected as long as suitable transfer samples are collected. Apart from machine olfaction, it is also viable in fields such as spectroscopy, where data measured by sensors or devices contain drift and transfer samples can be collected. Future study may include deeper exploiting the information in the unlabeled samples to improve the effect of correction.

References

Artursson T, Eklöv T, Lundström I, Mårtensson P, Sjöström M, Holmberg M (2000) Drift correction for gas sensors using multivariate methods. J Chemometr 14(5–6):711–723

Bengio Y (2012) Deep learning of representations for unsupervised and transfer learning. Unsupervised Transf Learn Chall Mach Learn 7:19

Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–1828

Bergstra J, Breuleux O, Bastien F, Lamblin P, Pascanu R, Desjardins G, Turian J, Warde-Farley D, Bengio Y (2010) Theano: a CPU and GPU math expression compiler. In: Proceedings of the Python for scientific computing conference (SciPy), Austin, TX, vol 4, p 3

Binfeng Y, Haibo J (2015) Near-infrared calibration transfer via support vector machine and transfer learning. Anal Methods 7(6):2714–2725

Chopra S, Balakrishnan S, Gopalan R (2013) Dlid: deep learning for domain adaptation by interpolating between domains. In: ICML workshop on challenges in representation learning, vol 2, p 5

Deng J, Zhang Z, Eyben F, Schuller B (2014) Autoencoder-based unsupervised domain adaptation for speech emotion recognition. IEEE Signal Process Lett 21(9):1068–1072

Feudale RN, Woody NA, Tan H, Myles AJ, Brown SD, Ferré J (2002) Transfer of multivariate calibration models: a review. Chemometr Intell Lab 64(2):181–192

Glorot X, Bordes A, Bengio Y (2011) Domain adaptation for large-scale sentiment classification: a deep learning approach. In: Proceedings of the 28th international conference on machine learning (ICML-11), pp 513–520

Kan M, Shan S, Chang H, Chen X (2014) Stacked progressive auto-encoders (spae) for face recognition across poses. In: 2014 IEEE conference on computer vision and pattern recognition

Kandaswamy C, Silva LM, Alexandre LA, Santos JM, de Sá JM (2014) Improving deep neural network performance by reusing features trained with transductive transference. In: Artificial neural networks and machine learning–ICANN 2014. Springer, pp 265–272

Längkvist M, Loutfi A (2011) Unsupervised feature learning for electronic nose data applied to bacteria identification in blood. In: NIPS 2011 workshop on deep learning and unsupervised feature learning

Längkvist M, Coradeschi S, Loutfi A, Rayappan JBB (2013) Fast classification of meat spoilage markers using nanostructured Zno thin films and unsupervised feature learning. Sensors 13(2):1578–1592

Längkvist M, Karlsson L, Loutfi A (2014) A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recogn Lett 42:11–24

Liu Q, Li X, Ye M, Ge SS, Du X (2014) Drift compensation for electronic nose by semi-supervised domain adaption. IEEE Sens J 14(3):657–665

Marco S, Gutiérrez-Gálvez A (2012) Signal and data processing for machine olfaction and chemical sensing: a review. IEEE Sens J 12(11):3189–3214

Padilla M, Perera A, Montoliu I, Chaudry A, Persaud K, Marco S (2010) Drift compensation of gas sensor array data by orthogonal signal correction. Chemometr Intell Lab 100(1):28–35

Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359

Romain AC, Nicolas J (2010) Long term stability of metal oxide-based gas sensors for e-nose environmental applications: an overview. Sens Actuators B: Chem 146(2):502–506

Vergara A, Vembu S, Ayhan T, Ryan MA, Homer ML, Huerta R (2012) Chemical gas sensor drift compensation using classifier ensembles. Sens Actuators B: Chem 166:320–329

Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA (2010) Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 11:3371–3408

Yan K, Zhang D (2015) Improving the transfer ability of prediction models for electronic noses. Sens Actuators B: Chem 220:115–124

Yan K, Zhang D (2016a) Calibration transfer and drift compensation of e-noses via coupled task learning. Sens Actuators B: Chem 225:288–297

Yan K, Zhang D (2016b) Correcting instrumental variation and time-varying drift: a transfer learning approach with autoencoders. IEEE Trans Instrum Meas 65(9):2012–2022

Yan K, Zhang D, Wu D, Wei H, Lu G (2014) Design of a breath analysis system for diabetes screening and blood glucose level prediction. IEEE Trans Biomed Eng 61(11):2787–2795

Zhang L, Zhang D (2015) Domain adaptation extreme learning machines for drift compensation in e-nose systems. IEEE Trans Instrum Meas 64(7):1790–1801

Zhang L, Chen C, Bu J, Cai D, He X, Huang TS (2011a) Active learning based on locally linear reconstruction. IEEE Trans Pattern Anal Mach Intell 33(10):2026–2038

Zhang L, Tian F, Kadri C, Xiao B, Li H, Pan L, Zhou H (2011b) On-line sensor calibration transfer among electronic nose instruments for monitoring volatile organic chemicals in indoor air quality. Sens Actuators: B Chem 160(1):899–909

Zhou JT, Pan SJ, Tsang IW, Yan Y (2014) Hybrid heterogeneous transfer learning through deep learning. In: Twenty-eighth AAAI conference on artificial intelligence

Zhuang F, Cheng X, Luo P, Pan SJ, He Q (2015) Supervised representation learning: transfer learning with deep autoencoders. In: Proceedings of the 24th international conference on artificial intelligence. AAAI Press, pp 4119–4125

Chapter 9
Drift Correction Using Maximum Independence Domain Adaptation

Abstract Transfer samples are required by the drift correction algorithms in the last three chapters. When transfer samples are not available, we can resort to unsupervised domain adaptation approaches. Maximum independence domain adaptation (MIDA) is proposed in this chapter for unsupervised drift correction. MIDA borrows the definition of domain features in the last chapter and learns features which have maximal independence with them, so as to reduce the inter-domain discrepancy in distributions. A feature augmentation strategy is designed so that the learned subspace is background-specific. Semi-supervised MIDA (SMIDA) extends MIDA by exploiting the label information. The proposed algorithms are flexible and fast. The effectiveness of our approaches is verified by experiments on synthetic datasets and three real-world ones on sensors and measurement.

Keywords Dimensionality reduction · Domain adaptation · Drift correction · Hilbert–Schmidt independence criterion · Subspace learning · Unsupervised learning · Semi-supervised learning

9.1 Introduction

Transfer samples are usually needed when dealing with instrumental variation and time-varying drift. Methods in the last three chapters are methods of this class (Yan and Zhang 2015, 2016a, b). Despite the fact that their accuracy is often better, collecting transfer samples repeatedly is a demanding job especially for nonprofessional e-nose users. In such cases, domain adaptation techniques with unlabeled target samples are desirable (Pan and Yang 2010; Patel et al. 2015). An intuitive idea is to reduce the inter-domain discrepancy in the feature level, i.e., to learn domain-invariant feature representation (Pan et al. 2011; Shi and Sha 2012; Fernando et al. 2013; Cui et al. 2014; Gong et al. 2014; Shao et al. 2014; Blitzer et al. 2007; Chen et al. 2012; Jiang et al. 2016). For example, Pan et al. (2011) proposed transfer component analysis (TCA), which finds a latent feature space that minimizes the distributional difference of two domains in the sense of maximum mean discrepancy. More

D. Zhang et al., *Breath Analysis for Medical Applications*,
DOI 10.1007/978-981-10-4322-2_9

related methods will be introduced in Sect. 9.2.1. When applied to drift correction, however, existing domain adaptation algorithms are faced with two difficulties. First, they are designed to handle discrete source and target domains. In time-varying drift, however, samples come in a stream, so the change in data distribution is often continuous. One solution is to split data into several batches, but it will lose the temporal order information. Second, because of the variation in the sensitivity of chemical sensors, the same signal in different conditions may indicate different concepts. In other words, the conditional probability $P(Y|X)$ may change for samples with different backgrounds, where "background" means when and with which device a sample was collected. Methods like TCA project all samples to a common subspace, hence the samples with similar appearance but different concepts cannot be distinguished.

In this chapter, we present a simple yet effective algorithm called maximum independence domain adaptation (MIDA) (Yan et al. 2017). First, we apply the "domain features" in the last chapter to describe the background of each sample. Then, MIDA finds a latent feature space in which the samples and their domain features are maximally independent in the sense of Hilbert–Schmidt independence criterion (HSIC) (Gretton et al. 2005). Thus, the discrete and continuous change in distribution can be handled uniformly. In order to project samples according to their backgrounds, feature augmentation is performed by concatenating the original feature vector with the domain features. We also propose semi-supervised MIDA (SMIDA) to exploit the label information with HSIC. MIDA and SMIDA are both very flexible. (1) They can be applied in situations with single or multiple source or target domains thanks to the use of domain features. In fact, the notion "domain" has been extended to "background" which is more informative. (2) Although they are designed for unsupervised domain adaptation problems (no labeled sample in target domains), the proposed methods naturally allow both unlabeled and labeled samples in any domains, thus can be applied in semi-supervised (both unlabeled and labeled samples in target domains) and supervised (only labeled samples in target domains) problems as well. (3) The label information can be either discrete (binary- or multi-class classification) or continuous (regression).

To illustrate the effect of our algorithms, we first evaluate them on several synthetic datasets. Then, drift correction experiments are performed on two e-nose datasets and one spectroscopy dataset. The rest of the chapter is organized as follows. Related work on unsupervised domain adaptation and HSIC is briefly reviewed in Sect. 9.2. Section 9.3 describes domain features, MIDA, and SMIDA in detail. The experimental configurations and results are presented in Sect. 9.4, along with some discussions. Section 9.5 summarizes the chapter.

9.2 Related Work

9.2.1 Unsupervised Domain Adaptation

Two good surveys on domain adaptation can be found in Pan and Yang (2010) and Patel et al. (2015). In this section, we focus on typical methods that extract domain-invariant features. In order to reduce the inter-domain discrepancy while preserving useful information, researchers have developed many strategies. Some algorithms project all samples to a common latent space (Pan et al. 2011; Shi and Sha 2012; Shao et al. 2014). Transfer component analysis (TCA) (Pan et al. 2011) tries to learn transfer components across domains in a reproducing kernel Hilbert space (RKHS) using maximum mean discrepancy. It is further extended to semi-supervised TCA (SSTCA) to encode label information and preserve local geometry of the manifold. Shi and Sha (2012) measured domain difference by the mutual information between all samples and their binary domain labels, which can be viewed as a primitive version of the domain features used in this chapter. They also minimized the negated mutual information between the target samples and their cluster labels to reduce the expected classification error. The low-rank transfer subspace learning (LTSL) algorithm presented in Shao et al. (2014) is a reconstruction-guided knowledge transfer method. It aligns source and target data by representing each target sample with some local combination of source samples in the projected subspace. The label and geometry information can be retained by embedding different subspace learning methods into LTSL.

Another class of methods first project the source and the target data into separate subspaces, and then build connections between them (Gong et al. 2012; Fernando et al. 2013; Liu et al. 2014; Cui et al. 2014). Fernando et al. (2013) utilized a transformation matrix to map the source subspace to the target one, where a subspace was represented by eigenvectors of PCA. The geodesic flow kernel (GFK) method (Gong et al. 2012) measures the geometric distance between two different domains in a Grassmann manifold by constructing a geodesic flow. An infinite number of subspaces are combined along the flow in order to model a smooth change from the source to the target domain. Liu et al. (2014) adapted GFK to correct time-varying drift of e-noses. A sample stream is first split into batches according to the acquisition time. The first and the latest batches (domains) are then connected through every intermediate batch using GFK. Another improvement of GFK is domain adaptation by shifting covariance (DASC) (Cui et al. 2014). Observing that modeling one domain as a subspace is not sufficient to represent the difference of distributions, DASC characterizes domains as covariance matrices and interpolates them along the geodesic to bridge the domains.

9.2.2 Hilbert–Schmidt Independence Criterion (HSIC)

HSIC is used as a convenient method to measure the dependence between two sample sets X and Y. Let k_x and k_y be two kernel functions associated with RKHSs $\mathcal{F}$ and $\mathcal{G}$, respectively. p_{xy} is the joint distribution. HSIC is defined as the square of the Hilbert–Schmidt norm of the cross-covariance operator $\mathcal{C}_{xy}$ (Gretton et al. 2005):

$$\begin{aligned}&\mathrm{HSIC}(p_{xy},\mathcal{F},\mathcal{G}) = \|\mathcal{C}_{xy}\|^2_{\mathrm{HS}}\\ =&\mathbf{E}_{xx'yy'}[k_x(x,x')k_y(y,y')] + \mathbf{E}_{xx'}[k_x(x,x')]\mathbf{E}_{yy'}[k_y(y,y')]\\ &-2\mathbf{E}_{xy}[\mathbf{E}_{x'}[k_x(x,x')]\mathbf{E}_{y'}[k_y(y,y')]].\end{aligned}$$

Here $\mathbf{E}_{xx'yy'}$ is the expectation over independent pairs (x,y) and (x',y') drawn from p_{xy}. It can be proved that with characteristic kernels k_x and k_y, $\mathrm{HSIC}(p_{xy},\mathcal{F},\mathcal{G})$ is zero if and only if x and y are independent (Song et al. 2012). A large HSIC suggests strong dependence with respect to the choice of kernels. HSIC has a biased empirical estimate. Suppose $Z = X \times Y = \{(x_1,y_1),\dots,(x_n,y_n)\}$, $K_x, K_y \in \mathbf{R}^{n\times n}$ are the kernel matrices of X and Y, respectively, then (Gretton et al. 2005):

$$\mathrm{HSIC}(Z,\mathcal{F},\mathcal{G}) = (n-1)^{-2}\mathrm{tr}(K_xHK_yH), \tag{9.1}$$

where $H = I - n^{-1}\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}} \in \mathbf{R}^{n\times n}$ is the centering matrix.

Due to its simplicity and power, HSIC has been adopted for feature extraction (Song et al. 2007; Pan et al. 2011; Barshan et al. 2011) and feature selection (Song et al. 2012). Researchers typically use it to maximize the dependence between the extracted/selected features and the label. However, to our knowledge, it has not been utilized in domain adaptation to reduce the dependence between the extracted features and the domain features.

9.3 Proposed Method

9.3.1 Domain Feature

We aim to reduce the dependence between the extracted features and the background information. A sample's background information should (1) naturally exist, thus can be easily obtained; (2) have different distributions in training and test samples; (3) correlate with the distribution of the original features. According to these characteristics, the information clearly interferes the testing performance of a prediction model, and that is why we want to minimize the aforementioned dependence. In common domain adaptation problems, the domain label (which domain a sample belongs) is an example of such information.

From the perspective of drift correction, background information includes the device label (with which device the sample was collected) and the acquisition time (when the sample was collected). We use domain features $\boldsymbol{d}$ introduced in Sect. 8.3.1 to describe them. If we only consider the instrumental variation, a one-hot coding scheme can be used. Suppose there are n_{dev} devices. The domain feature vector is thus $\boldsymbol{d} \in \mathbf{R}^{n_{\text{dev}}}$, where $d_i = 1$ if the sample is from the ith device and 0 otherwise. If the time-varying drift is also considered, the acquisition time can be further added into $\boldsymbol{d}$. Suppose a gas sample is collected from the ith device at time t, then $\boldsymbol{d} \in \mathbf{R}^{2n_{\text{dev}}}$ and

$$d_j = \begin{cases} 1, & j = 2i - 1, \\ t, & j = 2i, \\ 0, & \text{otherwise.} \end{cases} \tag{9.2}$$

We can actually encode more information such as the place of collection, the operation condition, and so on, which will be useful in other applications. According to Eq. 9.1, the kernel matrix K_d of the domain features needs to be computed for HSIC. We apply the linear kernel. Suppose $D = [\boldsymbol{d}_1, \dots, \boldsymbol{d}_n] \in \mathbf{R}^{m_d \times n}$, m_d is the dimension of a domain feature vector. Then

$$K_d = D^{\mathrm{T}} D. \tag{9.3}$$

9.3.2 Feature Augmentation

Feature augmentation is used in this chapter to learn background-specific subspaces. In Daum III (2007), the author proposed a feature augmentation strategy for domain adaptation: if a sample $\boldsymbol{x} \in \mathbf{R}^m$ is from the source domain, then its augmented feature vector is $\hat{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x} \\ \mathbf{0}_m \end{bmatrix} \in \mathbf{R}^{3m}$; If it is from the target domain, then $\hat{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x} \\ \mathbf{0}_m \\ \boldsymbol{x} \end{bmatrix} \in \mathbf{R}^{3m}$. The augmented labeled source and target samples are then used jointly to train one prediction model. In this way, the learned model can be viewed as two different models for the two domains. Meanwhile, the two models share a common component. However, this strategy requires that data lie in discrete domains and cannot deal with time-varying drift. We propose a more general and efficient feature augmentation strategy: concatenating the original features and the domain features, i.e.,

$$\hat{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{d} \end{bmatrix} \in \mathbf{R}^{m+m_d}. \tag{9.4}$$

The role of this strategy can be demonstrated through a linear dimensionality reduction example. Suppose a projection matrix $W \in \mathbf{R}^{(m+m_d)\times h}$ has been learned for the augmented feature vector. h is the dimension of the subspace. W has two parts:

$W = \begin{bmatrix} W_x \\ W_d \end{bmatrix}$, $W_x \in \mathbf{R}^{m \times h}$, $W_d \in \mathbf{R}^{m_d \times h}$. The embedding of $\hat{\boldsymbol{x}}$ can be expressed as $W^{\mathrm{T}}\hat{\boldsymbol{x}} = W_x^{\mathrm{T}}\boldsymbol{x} + W_d^{\mathrm{T}}\boldsymbol{d} \in \mathbf{R}^h$, which means that a background-specific bias $(W_d^{\mathrm{T}}\boldsymbol{d})_i$ has been added to each dimension i of the embedding. From another perspective, the feature augmentation strategy maps the samples to an augmented space with higher dimension before projecting them to a subspace. It will be easier to find a projection direction in the augmented space to align the samples well in the subspace.

Take machine olfaction (e-noses) for example, there are situations when the conditional probability $P(Y|X)$ changes along with the background. For instance, the sensitivity of chemical sensors often decays over time. A signal that indicates low concentration in an earlier time actually suggests high concentration in a later time. In such cases, feature augmentation is important, because it allows samples with similar appearance but different concepts to be treated differently by the background-specific bias. The strategy also helps to align the domains better in each projected dimension. Its effect will be illustrated on several synthetic datasets in Sect. 9.4.1 and further analyzed in the complementary materials.

9.3.3 Maximum Independence Domain Adaptation (MIDA)

In this section, we introduce the formulation of MIDA in detail. Suppose $X \in \mathbf{R}^{m \times n}$ is the matrix of n samples. The training and the test samples are pooled together. More importantly, we do not have to explicitly differentiate which domain a sample is from. The feature vectors have been augmented, but we use the notations X and m instead of $\hat{X}$ and $m + m_d$ for brevity. A linear or nonlinear mapping function Φ can be used to map X to a new space. Based on the kernel trick, we need not know the exact form of Φ, but the inner product of $\Phi(X)$ can be represented by the kernel matrix $K_x = \Phi(X)^{\mathrm{T}}\Phi(X)$. Then, a projection matrix $\tilde{W}$ is applied to project $\Phi(X)$ to a subspace with dimension h, leading to the projected samples $Z = \tilde{W}^{\mathrm{T}}\Phi(X) \in \mathbf{R}^{h \times n}$. Similar to other kernel dimensionality reduction algorithms (Schlkopf et al. 1998; Scholkopft and Mullert 1999), the key idea is to express each projection direction as a linear combination of all samples in the space, namely $\tilde{W} = \Phi(X)W$. $W \in \mathbf{R}^{n \times h}$ is the projection matrix to be actually learned. Thus, the projected samples are

$$Z = W^{\mathrm{T}}\Phi(X)^{\mathrm{T}}\Phi(X) = W^{\mathrm{T}}K_x. \tag{9.5}$$

Intuitively, if the projected features are independent of the domain features, then we cannot distinguish the background of a sample by its projected features, suggesting that the inter-domain discrepancy is diminished in the subspace. Therefore, after omitting the scaling factor in Eq. 9.1, we get the expression to be minimized: $\mathrm{tr}(K_z H K_d H) = \mathrm{tr}(K_x W W^{\mathrm{T}} K_x H K_d H)$, where K_z is the kernel matrix of Z.

In domain adaptation, the goal is not only minimizing the difference of distributions, but also preserving important properties of data, such as the variance (Pan et al. 2011). It can be achieved by maximizing the trace of the covariance matrix of the project samples. The covariance matrix is

$$\mathrm{cov}(Z) = \mathrm{cov}(W^{\mathrm{T}}K_x) = W^{\mathrm{T}}K_xHK_xW, \tag{9.6}$$

where $H = I - n^{-1}\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}$ is the same as that in Eq. 9.1. An orthonormal constraint is further added on W. The learning problem then becomes

$$\begin{aligned} &\max_W -\mathrm{tr}(W^{\mathrm{T}}K_xHK_dHK_xW) + \mu\,\mathrm{tr}(W^{\mathrm{T}}K_xHK_xW), \\ &\text{s.t.} \quad W^{\mathrm{T}}W = I, \end{aligned} \tag{9.7}$$

where $\mu > 0$ is a trade-off hyper-parameter. Using the Lagrangian multiplier method, we can find that W is the eigenvectors of $K_x(-HK_dH + \mu H)K_x$ corresponding to the h largest eigenvalues. Note that a conventional constraint is requiring $\tilde{W}$ to be orthonormal as in Barshan et al. (2011), which will lead to a generalized eigenvector problem. However, we find that this strategy is inferior to the proposed one in both adaptation accuracy and training speed in practice, so it is not used.

When computing K_x, a proper kernel function needs to be selected. Common kernel functions include linear ($k(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}$), polynomial ($k(\boldsymbol{x},\boldsymbol{y}) = (\sigma\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} + 1)^d$), Gaussian radial basis function (RBF, $k(\boldsymbol{x},\boldsymbol{y}) = \exp(\frac{\|x-y\|^2}{2\sigma^2})$), and so on. Different kernels indicate different assumptions on the type of dependence in using HSIC (Song et al. 2012). According to Song et al. (2012), the polynomial and RBF kernels map the original features to a higher or infinite dimensional space, thus are able to detect more types of dependence. However, choosing a suitable kernel width parameter (σ) is also important for these more powerful kernels (Song et al. 2012).

The maximum mean discrepancy (MMD) criterion is used in TCA (Pan et al. 2011) to measure the difference of two distributions. Song et al. (2012) showed that when HSIC and MMD are both applied to measure the dependence between features and labels in a binary-class classification problem, they are identical up to a constant factor if the label kernel matrix in HSIC is properly designed. However, TCA is feasible only when there are two discrete domains. On the other hand, MIDA can deal with a variety of situations including multiple domains and continuous distributional change. The stationary subspace analysis (SSA) algorithm (Von Bünau et al. 2009) is able to identify temporally stationary components in multivariate time series. However, SSA only ensures that the mean and covariance of the components are stationary, while they may not be suitable for preserving important properties in data. Concept drift adaptation algorithms (Gama et al. 2014) are able to correct continuous time-varying drift. However, most of them rely on newly arrived labeled data to update the prediction models, while MIDA works unsupervisedly.

9.3.4 Semi-supervised MIDA (SMIDA)

MIDA aligns samples with different backgrounds without considering the label information. However, if the labels of some samples are known, they can be incorporated into the subspace learning process, which may be beneficial to prediction. Therefore, we extend MIDA to semi-supervised MIDA (SMIDA). Since we do not explicitly differentiate the domain labels of the samples, both unlabeled and labeled samples can exist in any domain. Similar to Song et al. (2007), Pan et al. (2011), Barshan et al. (2011), Song et al. (2012), HSIC is adopted to maximize the dependence between the projected features and the labels. The biggest advantage of this strategy is that all types of labels can be exploited, such as the discrete labels in classification and the continuous ones in regression.

The label matrix Y is defined as follows. For c-class classification problems, the one-hot coding scheme can be used, i.e., $Y \in \mathbf{R}^{c\times n}, y_{i,j} = 1$ if $\boldsymbol{x}_i$ is labeled and belongs to the jth class; 0 otherwise. For regression problems, the target values can be centered first. Then, $Y \in \mathbf{R}^{1\times n}, y_i$ equals to the target value of $\boldsymbol{x}_i$ if it is labeled; 0 otherwise. The linear kernel function is chosen for the label kernel matrix, i.e.,

$$K_y = Y^{\mathrm{T}} Y. \tag{9.8}$$

The objective of SMIDA is

$$\begin{aligned} &\max_W \operatorname{tr}(W^{\mathrm{T}} K_x(-HK_dH + \mu H + \gamma HK_yH)K_xW), \\ &\text{s.t.} \quad W^{\mathrm{T}}W = I, \end{aligned} \tag{9.9}$$

where $\gamma > 0$ is a trade-off hyper-parameter. Its solution is the eigenvectors of $K_x(-HK_dH + \mu H + \gamma HK_yH)K_x$ corresponding to the h largest eigenvalues. The outline of MIDA and SMIDA is summarized in Algorithm 9.3.1. The statements in brackets correspond to those specialized for SMIDA.

Algorithm 9.3.1 MIDA [or SMIDA]

```
Input: The matrix of all samples $X$ and their background information; [the labels of some samples];
    the kernel function for $X$; $h, \mu$, [and $\gamma$].
Output: The projected samples $Z$.
1: Construct the domain features according to the background information, e.g., Sect. 9.3.1.
2: Augment the original features with domain features Eq. 9.4.
3: Compute the kernel matrices $K_x, K_d$, $\left[\text{and } K_y\right]$.
4: Obtain $W$, namely the eigenvectors of $K_x(-HK_dH+\mu H)K_x$ [or $K_x(-HK_dH+\mu H+\gamma HK_yH)K_x$]
   corresponding to the $h$ largest eigenvalues.
5: $Z = W^{\mathrm{T}}K_x$.
```

Besides variance and label dependence, another useful property of data is the geometry structure, which can be preserved by manifold regularization (MR) (Belkin et al. 2006). The manifold structure is modeled by a data adjacency graph. MR can be conveniently incorporated into SMIDA by adding a regularization term $-\lambda\, \mathrm{tr}(W^{\mathrm{T}} K_x L K_x W)$ into Eq. 9.9, where L is the graph Laplacian matrix (Belkin et al. 2006), $\lambda > 0$ is a trade-off hyper-parameter. In our experiments, adding MR generally increases the accuracy slightly. However, it also brings three more hyper-parameters, including λ, the number of nearest neighbors, and the kernel width when computing the data adjacency graph. Consequently, the experimental results in the next section were obtained without MR. It can still be an option in applications where geometry structure is important.

9.4 Experiments

In this section, we first conduct experiments on some synthetic datasets to verify the effect of the proposed methods. Then, drift correction experiments are performed on the same three datasets as the last two chapters. Comparison is made between them and recent unsupervised domain adaptation algorithms that learn domain-invariant features.

9.4.1 Synthetic Dataset

In Fig. 9.1, TCA (Pan et al. 2011) and MIDA are compared on a 2D dataset with two discrete domains. The domain labels were used to construct the domain features in MIDA according to the one-hot coding scheme introduced in Sect. 9.3.1. The similar definition was used in synthetic datasets 3 and 4. For both methods, the linear kernel was used on the original features and the hyper-parameter μ was set to 1. In order to quantitatively assessing the effect of domain adaptation, logistic regression models were trained on the labeled source data and tested on the target data. The accuracies are displayed in the caption, showing that the order of performance is MIDA > TCA > original feature. TCA aligns the two domains only on the first projected dimension. However, we can find that the two classes have large overlap on that dimension. This is because the direction for alignment is different from that for discrimination. Incorporating the label information of the source domain (SSTCA) did no help. On the contrary, MIDA can align the two domains well in both projected dimensions, in which the domain-specific bias on the second dimension brought by feature augmentation played a key role. A 3D explanation is included in the supplementary materials. Thus, good accuracy can be obtained by using the two dimensions for classification.

In Fig. 9.2, SSA (Von Bünau et al. 2009) and MIDA are compared on a 2D dataset with continuous distributional change, which resembles time-varying drift in machine olfaction. Samples in both classes drift to the upper right. The chronological order of the samples was used to construct the domain features in MIDA, i.e. $d = 1$ for the first sample, $d = 2$ for the second sample, etc. The parameter setting

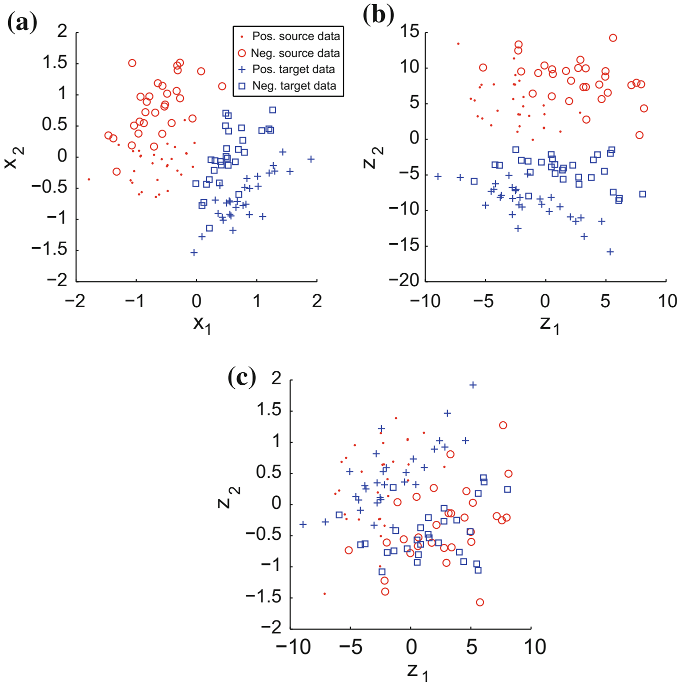


Fig. 9.1 Comparison of TCA and MIDA in a 2D synthetic dataset. Plots **a**–**c** show data in the original space and projected spaces of TCA and MIDA, respectively. The classification accuracies are 53, 70 (only using the first projected dimension z_1), and 88%

of MIDA was the same with that in Fig. 9.1, whereas the number of stationary components in SSA was set to 1. The classification accuracies were obtained by training a logistic regression model on the first halves of the data in both classes, and testing them on the last halves. SSA succeeds in finding a direction (z_1) that is free from time-varying drift. However, the two classes cannot be well separated in that direction. In plot (c), the randomly scattered colors suggest that the time-varying drift is totally removed in the subspace. MIDA first mapped the 2D data into a 3D space with the third dimension being time. Then, the augmented data were projected to a 2D plane that is orthogonal to the direction of drift in the 3D space. The projection direction was decided so that the independence between the projected data and time is maximized, meanwhile class separation was achieved by properly exploiting the background information.

No label information was used in the last two experiments. If keeping the label dependence in the subspace is a priority, SMIDA can be adopted instead of MIDA. In the 3D synthetic dataset in Fig. 9.3, the best direction (x_3) to align the two domains also mixes the two classes, which results in the output of MIDA in plot (b). For SMIDA, the weights for variance (μ) and label dependence (γ) were both set to 1. The labels in the source domain were used when learning the subspace. From plot

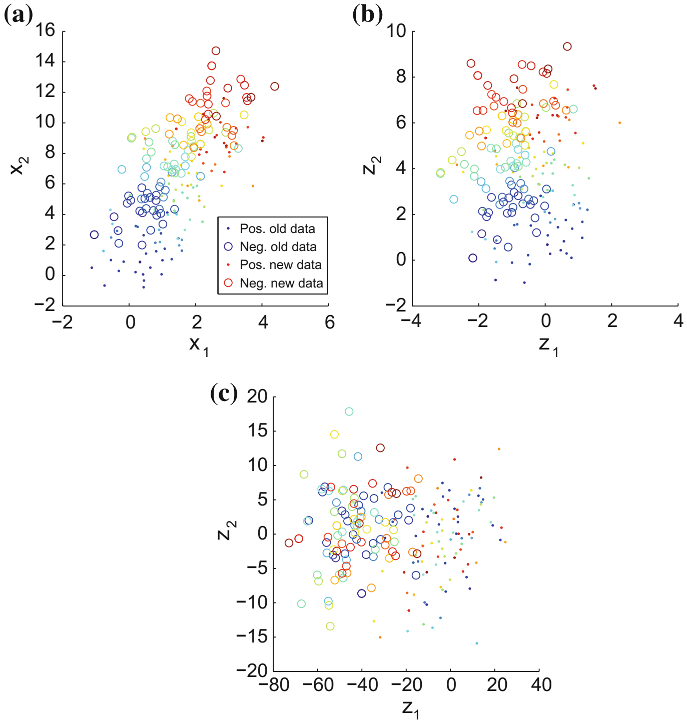


Fig. 9.2 Comparison of SSA and MIDA in a 2D synthetic dataset. Plots **a**–**c** show data in the original space, projected spaces of SSA and MIDA, respectively. The chronological order of a sample is indicated by color. The classification accuracies are 55, 74 (only using the first projected dimension z_1), and 90%

(c), we can observe that the classes are separated. In fact, class separation can still be found in the third dimension of the space learned by MIDA. However, for the purpose of dimensionality reduction, we generally hope to keep the important information in the first few dimensions.

Nonlinear kernels are often applied in machine learning algorithms when data is not linearly separable. Besides, they are also useful in domain adaptation when domains are not linearly "alignable", as shown in Fig. 9.4. As can be found in plot (a), the inter-domain changes in distributions are different for the two classes. Hence, it is difficult to find a linear projection direction to align the two domains, even with the domain-specific biases of MIDA. Actually, domain-specific rotation matrices are needed. Since the target labels are not available, the rotation matrices cannot be obtained accurately. However, a nonlinear kernel can be used to map the original features to a space with higher dimensions, in which the domains may be linearly alignable. We applied an RBF kernel with width $\sigma = 10$. Although the domains are not perfectly aligned in plot (c), the classification model trained in the source

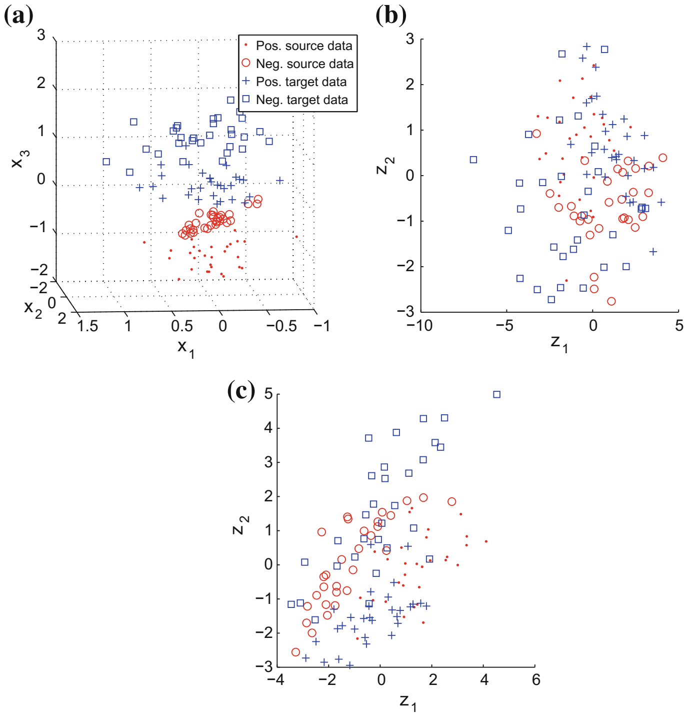


Fig. 9.3 Comparison of MIDA and SMIDA in a 3D synthetic dataset. Plots **a**–**c** show data in the original space and projected spaces of MIDA and SMIDA, respectively. The classification accuracies are 50, 55, and 82%

domain can be better adapted to the target domain. A comparison on different kernel and kernel parameters on two synthetic datasets is included in the supplementary materials.

9.4.2 Gas Sensor Array Drift Dataset

Readers are directed to Sect. 7.5.1 for the detail of the gas sensor array drift dataset. The original features have quite different dynamic ranges, which will interfere the learning process. Therefore, each feature was first normalized to have zero mean and unit variance within each batch. Next, the labeled samples in batch 1 were adopted as the source domain and the unlabeled ones in batch b ($b = 2, \dots, 10$) as the target domain. The proposed algorithms together with several recent ones were used to learn domain-invariant features based on these samples. Then, a logistic regression model was trained on the source domain and tested on each target one. For multi-class classification, the one-vs-all strategy was utilized.

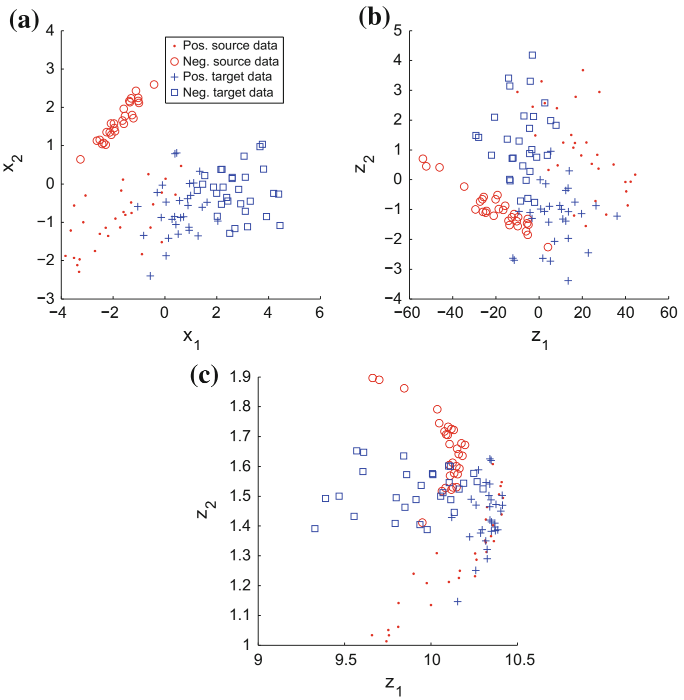


Fig. 9.4 Comparison of different kernels in a 2D synthetic dataset. Plots **a**–**c** show data in the original space and projected spaces of MIDA with linear and RBF kernels, respectively. The classification accuracies are 50, 57, and 87%

As displayed in Table 9.1, the compared methods include kernel PCA (KPCA), transfer component analysis (TCA), semi-supervised TCA (SSTCA) (Pan et al. 2011), subspace alignment (SA) (Fernando et al. 2013), geodesic flow kernel (GFK) (Gong et al. 2012), manifold regularization with combination GFK (ML-comGFK) (Liu et al. 2014), information-theoretical learning (ITL) (Shi and Sha 2012), structural correspondence learning (SCL) (Blitzer et al. 2007), and marginalized stacked denoising autoencoder (mSDA) (Chen et al. 2012). For all methods, the hyperparameters were tuned for the best accuracy. In KPCA, TCA, SSTCA, and the proposed MIDA and SMIDA, the polynomial kernel with degree 2 was used. KPCA learned a subspace based on the union of source and target data. In TCA, SSTCA, MIDA, and SMIDA, eigenvalue decomposition needs to be done on kernel matrices. If the number of samples is too large, this step can be time-consuming. In order to reduce the computational burden, we randomly chose at most n_t samples in each target domain when using these methods, with n_t being twice the number of the samples in the source domain. GFK used PCA to generate the subspaces in both source and target domains. The subspace dimension of GFK was determined according to the subspace disagreement measure in Gong et al. (2012). The results of ML-comGFK are copied from Liu et al. (2014). In SCL, the pivot features were binarized before training pivot predictors using logistic regression.

We also compared several variants of our methods. In Table 9.1, the notation "(discrete)" means that two discrete domains (source and target) were used in MIDA and SMIDA, which is similar to other compared methods. The domain feature vector of a sample was thus $[1, 0]^T$ if it was from the source domain and $[0, 1]^T$ if it was from the target. However, this strategy cannot make use of the samples in intermediate batches. An intuitive assumption is that the distributions of adjacent batches should be similar. When adapting the information from batch 1 to b, taking samples from batches 2 to $b - 1$ into consideration may improve the generalization ability of the learned subspace. Concretely, n_t samples were randomly selected from batches 2 to b instead of batch b alone. For each sample, the domain feature was defined as its batch index, which can be viewed as a proxy of its acquisition time. MIDA and SMIDA then maximized the independence between the learned subspace and the batch indices. The results are labeled as "(continuous)" in Table 9.1. Besides, the accuracies of continuous SMIDA without feature augmentation (no aug.) are also shown.

From Table 9.1, we can find that as the batch index increases, the accuracies of all methods generally degrade, which confirms the influence of the time-varying drift. Continuous SMIDA achieves the best average domain adaptation accuracy. The continuous versions of MIDA and SMIDA outperform the discrete versions, proving that the proposed methods can effectively exploit the chronological information of the samples. They also surpass ML-comGFK which uses the samples in intermediate batches to build connections between the source and the target batches. Feature augmentation is important in this dataset, since removing it in continuous SMIDA causes a drop of four percentage points in average accuracy. In Fig. 9.5, the average classification accuracies with varying subspace dimension are shown. MIDA and SMIDA are better than other methods when more than 30 features are extracted. Not surprisingly, the accuracy of unsupervised domain adaptation methods are not so good as the methods in the last two chapters which made use of the information in 10 transfer samples.

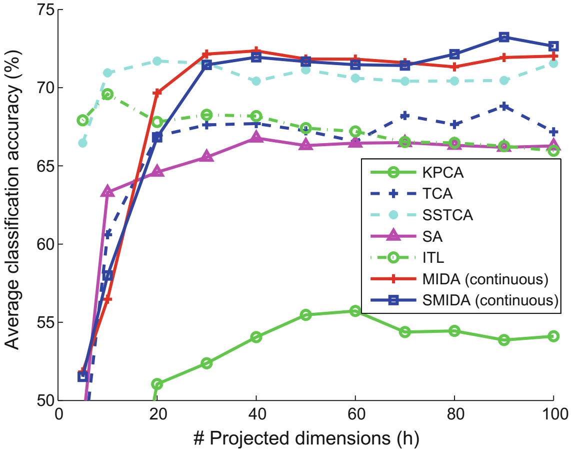
9.4.3 Breath Analysis Dataset

We have collected a breath analysis dataset in years 2014–2015 using two e-noses of the same model (Yan et al. 2014). Please see Sect. 7.5.2 for the introduction of this dataset. We performed five binary-class classification tasks to distinguish samples with one disease from the healthy samples. Each sample was represented by the steady state responses of nine gas sensors in the e-nose. The numbers of samples in the six classes (healthy and the five diseases mentioned above) are 125, 431, 340, 97, 156, and 215, respectively. We chose the first 50 samples collected with device 1 in each class as labeled training samples. Among the other samples, 10 samples were randomly selected in each class for validation, the rest for testing.

Table 9.1 Classification accuracy (%) on the gas sensor array drift dataset. Bold values indicate the best results

	Batch 2	3	4	5	6	7	8	9	10	Average
Original feature	80.47	79.26	69.57	77.16	77.39	64.21	52.04	47.87	48.78	66.30
KPCA	75.88	69.04	49.07	57.87	62.65	52.26	37.07	47.66	49.97	55.72
TCA	82.96	81.97	65.22	76.14	89.09	58.98	49.32	66.17	49.50	68.82
SSTCA	**84.57**	80.90	**80.12**	75.63	87.26	66.37	54.76	61.28	54.44	71.70
SA	80.79	80.01	71.43	75.63	78.35	64.68	52.04	48.51	49.58	66.78
GFK	77.41	80.26	71.43	76.14	77.65	64.99	36.39	47.45	48.72	64.49
ML-comGFK	80.25	74.99	78.79	67.41	77.82	**71.68**	49.96	50.79	53.79	67.28
ITL	76.85	79.45	59.63	**96.45**	78.00	60.95	49.32	**77.02**	48.58	69.58
SCL	77.57	82.03	68.32	82.74	77.22	65.18	53.74	48.51	48.08	67.04
mSDA	73.87	79.19	65.84	80.20	76.39	65.90	51.70	48.51	48.92	65.61
MIDA (discrete)	81.03	85.62	60.25	75.63	87.61	62.44	48.30	67.87	48.36	68.57
SMIDA (discrete)	80.47	**87.07**	65.22	75.63	90.04	59.20	50.00	62.77	44.81	68.36
MIDA (continuous)	84.32	81.59	68.32	75.63	91.74	63.13	78.91	62.34	45.14	72.35
SMIDA (no aug.)	82.23	83.17	67.70	75.13	85.22	61.67	51.02	61.49	**54.61**	69.14
SMIDA (continuous)	83.68	82.28	73.91	75.63	**93.00**	63.49	**79.25**	62.34	45.50	**73.23**

Fig. 9.5 Performance comparison on the gas sensor array drift dataset with respect to the subspace dimension h



The hyper-parameters were tuned in the validation sets. Logistic regression was adopted as the classifier. F-score was used as the accuracy criterion. Results are compared in Table 9.2.

In KPCA, TCA, SSTCA, MIDA, and SMIDA, the RBF kernel was used. Because methods other than stationary subspace analysis (SSA) (Von Bünau et al. 2009), MIDA, and SMIDA are not capable of handling the chronological information, we simply regarded each device as a discrete domain and learned device-invariant features with them. The same strategy was used in discrete MIDA and SMIDA. In continuous MIDA and SMIDA, the domain features were defined according to Eq. 9.2, where t was the exact acquisition time converted to years and the number of devices $n_{\text{dev}} = 2$. SSA naturally considers the chronological information by treating the sample stream as a multivariate time series and identifying temporally stationary components. However, SSA cannot deal with time series with multiple sources, such as the multi-device case in this dataset. Thus, the samples were arranged in chronological order despite their device labels.

From Table 9.2, we can find that the performance of the original features is not promising, which is caused by the instrumental variation and time-varying drift in the dataset. The domain adaptation algorithms can improve the accuracy. The improvement made by SSA is little, possibly because the stationary criterion is not suitable for preserving important properties in data. For example, the noise in data can also be stationary (Pan et al. 2011). MIDA and SMIDA achieved obviously better results than other methods. They can address both instrumental variation and time-varying drift. With the background-specific bias brought by feature augmentation, they can compensate for the change in conditional probability in this dataset. Similar to the gas sensor array drift dataset, it can be seen that the continuous MIDA and SMIDA that utilize the time information are better than the discrete ones. Feature augmentation can improve continuous SMIDA by six percentage points. SMIDA is better than MIDA because the label information of the first 50 samples in each class was better kept.

Table 9.2 Classification accuracy (%) on the breath analysis dataset. Bold values indicate the best results

	Task 1	2	3	4	5	Average
Original feature	34.34	63.67	73.71	43.17	42.93	51.57
KPCA	58.05	72.58	84.78	44.95	42.60	60.59
TCA	67.19	68.31	59.93	67.08	**68.17**	66.14
SSTCA	67.01	68.06	74.14	68.31	67.36	68.97
SA	29.95	72.42	72.74	42.19	44.54	52.37
GFK	41.49	68.50	58.96	75.63	70.16	62.95
ITL	68.59	66.53	74.75	66.67	68.03	68.91
SSA	49.77	72.10	33.49	52.64	55.38	52.68
SCL	32.52	61.16	75.43	35.35	51.86	51.26
mSDA	36.86	69.51	76.69	35.51	50.49	53.81
MIDA (discrete)	62.17	71.74	84.21	67.05	67.06	70.45
SMIDA (discrete)	80.16	**84.18**	88.47	68.45	52.41	74.73
MIDA (continuous)	68.30	67.54	74.01	73.04	69.63	70.50
SMIDA (no aug.)	82.80	72.57	72.61	**80.33**	70.05	75.67
SMIDA (continuous)	**85.29**	80.18	**91.67**	74.28	66.55	**79.59**

9.4.4 Corn Dataset

The dataset has been introduced in Sect. 7.5.3. This dataset resembles traditional domain adaptation datasets because there is no time-varying drift. Three discrete domains can be defined based on the three devices. We adopt m5 as the source domain, mp5 and mp6 as the target ones. In each domain, samples $4, 8, \ldots, 76, 80$ were assigned as the test set, the rest as the training set. For hyper-parameter tuning, we applied a threefold cross-validation on the training sets of the three domains. After the best hyper-parameters were determined for each algorithm, a regression model was trained on the training set from the source domain and applied on the test set from the target domains. The regression algorithm was ridge regression with the L2 regularization parameter $\lambda = 1$.

Table 9.3 displays the root mean square error (RMSE) of the four prediction tasks and their average on the two target domains. We also plot the overall average RMSE of the two domains with respect to the subspace dimension h in Fig. 9.6. ITL was not investigated because it is only applicable in classification problems. In KPCA, TCA, SSTCA, MIDA, and SMIDA, the RBF kernel was used. For the semi-supervised methods SSTCA and SMIDA, the target values were normalized to zero mean and unit variance during subspace learning. The domain features were defined according to the device indices using the one-hot coding scheme. We can find that when no domain adaptation was done, the prediction error is large. All domain adaptation algorithms managed to significantly reduce the error. KPCA also

Table 9.3 Regression RMSE on the corn dataset. Bold values indicate the best results

	Mp5 as target domain					Mp6 as target domain				
	Moisture	Oil	Protein	Starch	Average	Moisture	Oil	Protein	Starch	Average
Original feature	1.327	0.107	1.155	2.651	1.310	1.433	0.101	1.413	2.776	1.431
KPCA	0.477	0.165	0.215	**0.315**	0.293	0.396	0.164	0.238	**0.290**	0.272
TCA	0.539	0.322	0.217	0.402	0.370	0.398	0.145	0.259	0.572	0.343
SSTCA	0.343	0.093	**0.140**	0.366	0.235	0.367	0.088	0.186	0.318	0.240
SA	0.302	0.094	0.186	0.351	0.233	0.324	0.079	0.158	0.390	0.238
GFK	0.267	0.197	0.342	0.621	0.357	**0.263**	0.189	0.264	0.485	0.301
SCL	0.283	0.115	0.249	0.619	0.316	0.311	0.108	0.257	0.683	0.340
mSDA	**0.264**	0.107	0.211	0.446	0.257	0.285	0.097	0.198	0.471	0.263
MIDA	0.317	0.078	0.141	0.378	0.228	0.317	0.084	0.158	0.352	0.228
SMIDA	0.287	**0.072**	0.143	0.339	**0.210**	0.316	**0.073**	**0.152**	0.325	**0.217**
Train on target	0.176	0.094	0.201	0.388	0.215	0.182	0.108	0.206	0.414	0.228

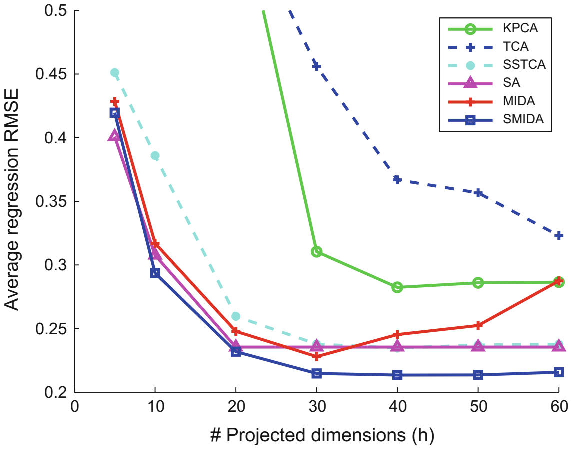


Fig. 9.6 Performance comparison on the corn dataset with respect to the subspace dimension *h*

has good performance, which is probably because the source and the target domains have similar principal directions, which also contain the most discriminative information. Therefore, source regression models can fit the target samples well. In this dataset, different domains have identical data composition. As a result, corresponding data can be aligned by subspaces alignment, which explains the small error of SA. However, this condition may not hold in other datasets.

MIDA and SMIDA obtained the lowest average errors in both target domains. Aiming at exploring the prediction accuracy when there is no instrument variation, we further trained regression models on the training set of the two target domains and tested on the same domain. The results are listed as “train on target” in Table 9.3. It can be found that SMIDA outperforms these results. This could be attributed to three reasons: (1) The inter-domain discrepancy in this dataset is relatively easy to correct; (2) The use of RBF kernel in SMIDA improves the accuracy; (3) SMIDA learned the subspace on the basis of both training and test samples. Although the test samples were unlabeled, they can provide some information about the distribution of the samples to make the learned subspace generalize better, which can be viewed as the merit of semi-supervised learning.

9.5 Summary

In this chapter, we introduced maximum independence domain adaptation (MIDA) to learn domain-invariant features. The main idea of MIDA is to reduce the inter-domain discrepancy by maximizing the independence between the learned features and the domain features of the samples. The domain features describe the background information of each sample, such as the device label and acquisition time. The feature augmentation strategy proposed in this chapter adds domain-specific biases to the learned features, which helps MIDA to align domains. It is also useful when there is a change in conditional probability. Finally, to incorporate the label information, semi-supervised MIDA (SMIDA) is presented.

MIDA and SMIDA are flexible algorithms. With the design of the domain features and the use of the HSIC criterion, they can be applied in all kinds of domain adaptation problems, including discrete or continuous distributional change, supervised/semi-supervised/unsupervised, multiple domains, classification or regression, etc. Thus, they have a wide range of potential applications. They are also easy to implement and fast, requiring to solve only one eigenvalue decomposition problem. Experimental results on various types of datasets proved their effectiveness. Although MIDA and SMIDA outperformed typical unsupervised domain adaptation algorithms, their accuracy still needs improvement compared with transfer-sample-based methods.

In the last four chapters, we have proposed five algorithms to mitigate the influence of instrumental variation and time-varying drift of e-noses, so as to enhance the robustness and practicability of breath analysis systems in real-world situations. Among them, windowed piecewise direct standardization (WPDS) is an improvement of the widely used PDS algorithm. It is easy to implement, but may not be the best choice when the drift is complex. Standardization-error-based model improvement (SEMI) is actually a regularization term that can be incorporated in existing prediction models. The remaining three algorithms, transfer-sample-based multitask learning (TMTL), drift correction autoencoder (DCAE), and maximum independence domain adaptation (MIDA), are complete algorithm frameworks. They have three features in common:

- They can handle instrumental variation and time-varying drift, i.e., discrete and continuous distributional change, at the same time;
- They consider drift correction and discriminability at the same time. There are supervision terms in their objective functions.
- They are flexible and extensible. For instance, labeled, unlabeled, and transfer (except MIDA) samples can occur in any domain; classification and regression problems are both allowed; the supervision term can be any loss function in TMTL and DCAE, etc.

The major differences between TMTL, DCAE, and MIDA are listed in Table 9.4.

In summary, drift correction is an important topic for e-noses. We have adapted and extended various transfer learning/domain adaptation algorithms into this field. The algorithms can be used on not only e-noses, but also other applications.

Table 9.4 Comparison of TMTL, DCAE, and MIDA

Feature	TMTL	DCAE	MIDA
Framework	Multitask learning	Autoencoder	Subspace learning
Principle	Aligning transfer samples in the model level	Aligning transfer samples in the representation level	Maximum independence criterion, adapting unlabeled samples with different background
Ensuring drift is sufficiently corrected	Learning one model for each domain	Applying different correction outputs	Feature augmentation strategy helps correction
Ensuring drift is not over-corrected	Model similarity constraint; Dynamic model strategy can smooth models	Correction outputs only relate with sample background and they change smoothly over time	Learning one subspace for samples with different background
Handling data stream	Dynamic model strategy	Using domain features	Using domain features
Suitable scenarios	General drift correction problems	Drift is complex; or more abstract and discriminative features are needed	Transfer samples are unavailable

References

Barshan E, Ghodsi A, Azimifar Z, Jahromi MZ (2011) Supervised principal component analysis: visualization, classification and regression on subspaces and submanifolds. Pattern Recogn 44(7):1357–1371

Belkin M, Niyogi P, Sindhwani V (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res 7:2399–2434

Blitzer J, Dredze M, Pereira F (2007) Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification. ACL 7:440–447

Chen M, Xu Z, Weinberger K, Sha F (2012) Marginalized denoising autoencoders for domain adaptation. In: 29th international conference on machine learning

Cui Z, Li W, Xu D, Shan S, Chen X, Li X (2014) Flowing on riemannian manifold: domain adaptation by shifting covariance. IEEE Trans Cybern 44(12):2264–2273

Daum III H (2007) Frustratingly easy domain adaptation. In: Proceedings of 45th annual meeting of the association for computational linguistics

Fernando B, Habrard A, Sebban M, Tuytelaars T (2013) Unsupervised visual domain adaptation using subspace alignment. In: Proceedings of the IEEE international conference on computer vision, pp 2960–2967

Gama J, Žliobaite I, Bifet A, Pechenizkiy M, Bouchachia A (2014) A survey on concept drift adaptation. ACM Comput Surv (CSUR) 46(4):44

Gong B, Shi Y, Sha F, Grauman K (2012) Geodesic flow kernel for unsupervised domain adaptation. In: 2012 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 2066–2073

Gong B, Grauman K, Sha F (2014) Learning kernels for unsupervised domain adaptation with applications to visual object recognition. Int J Comput Vis 109(1–2):3–27

Gretton A, Bousquet O, Smola A, Schlkopf B (2005) Measuring statistical dependence with hilbert-schmidt norms. In: Algorithmic learning theory. Springer, pp 63–77

Jiang M, Huang W, Huang Z, Yen G (2016) Integration of global and local metrics for domain adaptation learning via dimensionality reduction. IEEE Trans Cybern

Liu Q, Li X, Ye M, Ge SS, Du X (2014) Drift compensation for electronic nose by semi-supervised domain adaption. IEEE Sens J 14(3):657–665

Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359

Pan SJ, Tsang IW, Kwok JT, Yang Q (2011) Domain adaptation via transfer component analysis. IEEE Trans Neural Netw 22(2):199–210

Patel VM, Gopalan R, Li R, Chellappa R (2015) Visual domain adaptation: a survey of recent advances. IEEE Signal Process Mag 32(3):53–69

Schlkopf B, Smola A, Mller KR (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput 10(5):1299–1319

Scholkopft B, Mullert KR (1999) Fisher discriminant analysis with kernels. Neural Netw Signal Process IX:41–48

Shao M, Kit D, Fu Y (2014) Generalized transfer subspace learning through low-rank constraint. Int J Comput Vis 109(1–2):74–93

Shi Y, Sha F (2012) Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. In: Proceedings of the international conference on machine learning (ICML)

Song L, Gretton A, Borgwardt KM, Smola AJ (2007) Colored maximum variance unfolding. In: Advances in neural information processing systems, pp 1385–1392

Song L, Smola A, Gretton A, Bedo J, Borgwardt K (2012) Feature selection via dependence maximization. J Mach Learn Res 13(1):1393–1434

Von Bünau P, Meinecke FC, Király FC, Müller KR (2009) Finding stationary subspaces in multivariate time series. Phys Rev Lett 103(21):214,101

Yan K, Zhang D (2015) Improving the transfer ability of prediction models for electronic noses. Sens Actuators B: Chem 220:115–124

Yan K, Zhang D (2016a) Calibration transfer and drift compensation of e-noses via coupled task learning. Sens Actuators B: Chem 225:288–297

Yan K, Zhang D (2016b) Correcting instrumental variation and time-varying drift: a transfer learning approach with autoencoders. IEEE Trans Instrum Meas 65(9):2012–2022

Yan K, Zhang D, Wu D, Wei H, Lu G (2014) Design of a breath analysis system for diabetes screening and blood glucose level prediction. IEEE Trans Biomed Eng 61(11):2787–2795

Yan K, Kou L, Zhang D (2017) Learning domain-invariant subspace using domain features and independence maximization. IEEE Trans Cybern

Part IV
Feature Extraction and Classification

Chapter 10
Feature Selection and Analysis on Correlated Breath Data

Abstract Feature selection is a useful step in data analysis procedure. In this chapter, we study the classical support vector machine recursive feature elimination (SVM-RFE) algorithm and improve it by incorporating a correlation bias reduction (CBR) strategy into the feature elimination procedure. Experiments are conducted on a synthetic dataset and two breath analysis datasets. Large and comprehensive sets of transient features are extracted from the sensor responses. The classification accuracy with feature selection proves the efficacy of the proposed SVM-RFE + CBR. It outperforms the original SVM-RFE and other typical algorithms. An ensemble method is further studied to improve the stability of the proposed method. By statistically analyzing the features' rankings, some knowledge is obtained, which can guide future design of e-noses and feature extraction algorithms.

Keywords Correlation bias · Feature ranking · Feature selection · SVM-RFE · Transient feature

10.1 Introduction

Feature selection (FS) is a widely used technique in pattern recognition applications. By removing irrelevant, noisy, and redundant features from the original feature space, FS alleviates the problem of overfitting and improves the performance of the model. The time and space cost of the learning algorithm can also be reduced. More importantly, we can gain a deeper insight of the data by analyzing the importance of the features (Guyon and Elisseeff 2003; Saeys et al. 2007). Many researchers have explored the use of FS techniques in electronic nose (e-nose) systems and achieved good results (Paulsson et al. 2000; Llobet et al. 2007; Gualdrón et al. 2007; Pardo and Sberveglieri 2008; Cho and Kurup 2011; Kaur et al. 2012; Marco and Gutiérrez-Gálvez 2012).

In the context of classification, FS algorithms can be roughly divided into three categories: filters, wrappers and embedded methods, based on how they interact with classifiers (Guyon and Elisseeff 2003; Saeys et al. 2007). Filters evaluate each feature by predefined criteria, such as correlation criteria and information theoretic criteria

D. Zhang et al., *Breath Analysis for Medical Applications*,
DOI 10.1007/978-981-10-4322-2_10

(Guyon and Elisseeff 2003), which are independent from classifiers. Wrappers treat classifiers as black boxes and aim at finding a feature subset that has the minimum cross-validation error on the training data. Examples of wrappers include sequential forward selection (Paulsson et al. 2000; Yan and Zhang 2014a), genetic algorithms, and simulate annealing (Llobet et al. 2007). Embedded methods generally include two kinds of approaches. In some methods, such as a decision tree (Cho and Kurup 2011), the training of the classifier intrinsically selects a subset of features. Some methods estimate the importance of the features from the coefficients in the classifiers, e.g., the algorithm in (Gualdrón et al. 2007).

Support vector machine recursive feature elimination (SVM-RFE) is an embedded FS algorithm proposed by Guyon et al. (Guyon et al. 2002). It uses criteria derived from the coefficients in SVM models to assess features, and recursively removes features that have small criteria. It has both linear and nonlinear versions. The nonlinear SVM-RFE uses a special kernel strategy (Guyon et al. 2002; Rakotomamonjy 2003) and is preferred when the optimal decision function is nonlinear. As a backward elimination method, SVM-RFE is able to model the dependencies among features. Compared to wrappers, SVM-RFE does not use the cross-validation accuracy on the training data as the selection criterion, thus is (1) less prone to overfitting; (2) able to make full use of the training data; (3) much faster, especially when there are a lot of candidate features. As a result, it has been successfully applied in many problems, especially in gene selection (Guyon et al. 2002; Rakotomamonjy 2003; Duan et al. 2005; Tang et al. 2007; Yoon and Kim 2009; Mundra and Rajapakse 2010).

However, there is still one problem in SVM-RFE that has not been addressed. When some of the candidate features are highly correlated, the assessing criteria of these features will be influenced, and their importance will be underestimated. Inspired by (Toloşi and Lengauer 2011), we call this phenomenon “correlation bias”. It is a crucial problem especially for gas sensor features that are often correlated. In this chapter, a simulated experiment is first employed to illustrate this problem. Then a novel strategy, correlation bias reduction (CBR), is proposed to reduce this potential bias in both linear and nonlinear SVM-RFE. Finally, an ensemble method is suggested to improve the stability of the feature selection results. (Yan and Zhang 2015)

The proposed method is evaluated on two breath analysis datasets. The first breath analysis dataset was collected by an e-nose with 10 gas sensors (Yan et al. 2014). The dataset contains 295 samples from healthy subjects and 279 from diabetics. It will also be investigated in Chap. 14. The second dataset was collected by the e-nose in Chap. 3 with 12 MOS sensors (Guo et al. 2010). The breath samples were from healthy subjects and also subjects with diabetes, renal disease, and airway inflammation, respectively. Over 1000 features are extracted from the gas sensors’ responses. The comprehensive feature set contains seven kinds of transient features. Experimental results show that the Gaussian SVM-RFE is better than the linear one, as well as other typical algorithms. The proposed CBR strategy further enhances the accuracy. The ensemble method is proved to have better stability. Furthermore, systematic statistical analysis on the features’ rankings reveals useful information about which sensors, feature types and TM voltages are more important. For example, TM sensors significantly outperform the ones operated under constant temperature. Phase feature extracted from TM sensors is proved to be the most effective feature. The information provides guidance for future e-nose and feature designing.

The chapter is organized as follows. Section 10.2 describes the details of linear and nonlinear SVM-RFE algorithm. Section 10.3 investigates the correlation bias problem and proposes SVM-RFE + CBR. Section 10.4 introduces the breath analysis datasets and feature extraction methods. Section 10.5 shows the results of the FS experiments and provides the feature analysis results. Section 10.6 summarizes the chapter.

10.2 SVM-RFE

10.2.1 Linear SVM-RFE

The output of SVM-RFE is a ranked feature list. Feature selection can be achieved by choosing a group of top-ranked features. The ranking criterion of SVM-RFE is closely related to the SVM model. SVM is a popular algorithm for classification partially due to its high accuracy and good generalization ability. It has been successfully applied in many e-nose applications (Marco and Gutiérrez-Gálvez 2012).

The intuition of SVM is to find a separating hyperplane with the largest margin. In linear separable cases, the margin is twice the distance between the separating hyperplane and the training sample closest to it (Burges 1998). Given a set of training samples $\{\boldsymbol{x}_i, y_i\}$, $\boldsymbol{x}_i \in \mathbf{R}^m$, $y_i \in \{-1, 1\}$, $i = 1, \ldots, n$, the decision function of a linear SVM is

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b. \tag{10.1}$$

It can be proved that the margin M is simply $2/\|\boldsymbol{w}\|$, thus maximizing the margin is equivalent to minimizing $\|\boldsymbol{w}\|^2$ under constraints. The dual form of the Lagrangian formulation of the problem can be written as (Burges 1998):

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j, \tag{10.2}$$

where α_i are the Lagrange multipliers. Solutions of α_i can be found by maximizing L_D under constraints $\alpha_i \geq 0$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$. The samples corresponding to nonzero α's are known as support vectors. Then the weight vector $\boldsymbol{w}$ can be obtained by

$$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i. \tag{10.3}$$

The ranking criterion for feature k is the square of the kth element of $\boldsymbol{w}$,

$$J(k) = w_k^2. \tag{10.4}$$

In each iteration of the recursive feature elimination (RFE), a linear SVM model is trained. The feature with the smallest ranking criterion is removed since it has the least effect on classification (Tang et al. 2007). The remaining features are kept for the SVM model in the next iteration. This process is repeated until all the features have been removed. Then the features are sorted according to the order of removal. The later a feature is removed, the more important it should be. When the feature dimension is high, removing features one by one will be time-consuming. In such cases, more than one feature can be removed in each iteration (Guyon et al. 2002). However, this strategy may influence the precision (Tang et al. 2007) and cause the correlation bias problem, which will be described in Sect. 10.3.1.

10.2.2 Nonlinear SVM-RFE

Most gene selection problems have much more features (several thousand) than samples (less than 100), so linear SVM-RFE is more suitable in these cases to avoid overfitting. But in many other situations where the number of samples is larger, nonlinear SVM-RFE can be expected to outperform the linear one since it can fit the data with less bias.

Nonlinear SVM considers to map the features into a new space with higher dimension:

$$\boldsymbol{x} \in \mathbf{R}^m \mapsto \Phi(\boldsymbol{x}) \in \mathbf{R}^h. \tag{10.5}$$

In the new space, the samples are expected to be linearly separable. Thus Eq. 10.2 can be rewritten as

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j). \tag{10.6}$$

Note that the only form that $\Phi(\boldsymbol{x})$'s are involved in the training algorithm is their inner product. So we can replace $\Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j)$ with a kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ without knowing the explicit form of Φ. This is a particularly useful trick because it is hard to determine the form of Φ in real-world problems. There are several choices for kernel functions, though, one common choice being the Gaussian kernel

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}. \tag{10.7}$$

Since the form of Φ is unknown, the weight vector $\boldsymbol{w}$ cannot be obtained. However, linear SVM-RFE can be extended to nonlinear cases via a special strategy. If the removal of a feature causes only small changes in the objective function Eq. 10.6, the feature should be removed (Guyon et al. 2002; Rakotomamonjy 2003). This leads to the following ranking criterion for feature k:

$$J(k) = \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i^{(-k)}, \boldsymbol{x}_j^{(-k)}). \tag{10.8}$$

The notation $(-k)$ means the feature k has been removed, i.e., $\boldsymbol{x}^{(-k)} \in \mathbf{R}^{m-1}$. The above criterion is the difference of Eq. 10.6 before and after removing feature k while keeping the α's unchanged. The features with small J's will be eliminated in each iteration of RFE. This criterion is applicable for all kinds of kernels. When the linear kernel is used ($K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i \cdot \boldsymbol{x}_j$), it is equivalent to the linear SVM-RFE. This nonlinear version of SVM-RFE costs a little more time than the linear version, but some techniques can be applied to accelerate it, which will be introduced in Sect. 10.3.3.

10.3 Improved SVM-RFE with Correlation Bias Reduction

10.3.1 Correlation Bias

Some classification applications contain highly correlated features. For example, in gene classification, features represented by probes that either have similar molecular functions or genomic locations are highly correlated (Toloşi and Lengauer 2011). In e-nose applications, gas sensors are known to be cross-sensitive. Besides, when multiple transient features (e.g., the magnitude, derivative and integral at different time points) are extracted from a gas sensor's response, high correlation often exists among these features. The correlation brings adverse impacts to some feature selection algorithms. Toloşi and Lengauer (2011) evaluated feature importance based on Lasso penalized logistic regression and random forest. They discovered that the evaluation was biased in highly correlated feature groups. Concretely, the features in the groups received smaller weights due to the shared responsibility in the classification models. Therefore, the importance of the features will be underestimated even if they are highly relevant. The larger the group size, the larger the underestimation. This phenomenon is called "correlation bias" (CB) in (Toloşi and Lengauer 2011).

The phenomenon has not been studied in SVM-RFE. However, both linear and nonlinear SVM-RFE are affected due to the similar reason for Lasso penalized logistic regression and random forest. We have conducted a simulated experiment to illustrate the phenomenon. A synthetic dataset is generated with 500 samples and 100 features. There are 22 latent variables $z_1, \ldots, z_{22}$ that contribute to the class label y. They are all drawn from the normal distribution $\mathcal{N}(0, 1)$. The class label y is decided by

Table 10.1 Weights and sizes of the feature groups in the synthetic dataset

Parameter	Real feature index								
	1–20	21–30	31–36	37–40	41–60	61–70	71–76	77–80	81–100
Weight w	1	1	1	1	0.5	0.5	0.5	0.5	0
Group size k	10	5	2	1	10	5	2	1	1

$$y = \operatorname{sign}\left(\sum_{i=1}^{22} w_i z_i + \epsilon\right), \tag{10.9}$$

where ϵ is a perturbation term drawn from $\mathcal{N}(0, 0.01)$. $w_1, \ldots, w_{22}$ are predefined weights. From each latent variable, a group of real features x are generated:

$$x_{ij} = z_i + \epsilon, j = 1, \ldots, k_i; i = 1, \ldots, 22. \tag{10.10}$$

k_i is the size of the ith group, $\epsilon \sim \mathcal{N}(0, 0.01)$. So the features in each group are highly correlated. The 22 groups of features are concatenated and constitute $x_1, \ldots, x_{80}$. $x_{81}, \ldots, x_{100}$ are pure noise features drawn from $\mathcal{N}(0, 1)$. The weights and sizes of the feature groups are listed in Table 10.1. For example, x_1–x_{20} belong to two groups with size 10 and $w = 1$.

The 100 features are evaluated using the feature ranking criteria introduced in Sect. 10.2. Note that the RFE procedure is not performed. In order to compare the criteria among groups, they are normalized to [0, 1]. The results for linear and nonlinear SVM-RFE are displayed in Fig. 10.1. It is clear that both the group size and the weight influence the criteria. Ideally, the criteria should solely depend on the weight, so the criteria for feature 1–40, 41–80 and 81–100 should be about 1, 0.5 and 0, respectively. However, because of CB, the features in larger groups receive smaller criteria. The features with group size 5 and 10 receive criteria comparable to that of the noise features. In this case, if a batch of features are removed in one iteration of RFE, features in large groups are likely to be removed entirely.

10.3.2 Correlation Bias Reduction

Highly correlated features bring wrong estimations to several embedded FS algorithms including SVM-RFE. They also affect regression applications, causing large variance of the estimates and inaccurate prediction (Park et al. 2007). In order to deal with the problem, some methods have been proposed. An intuitive method is to replace each group of highly correlated features with one representative before selection or regression. For example, Park et al. (2007) performed hierarchical clustering on features and used the cluster centroids for regression. Another idea is to perform

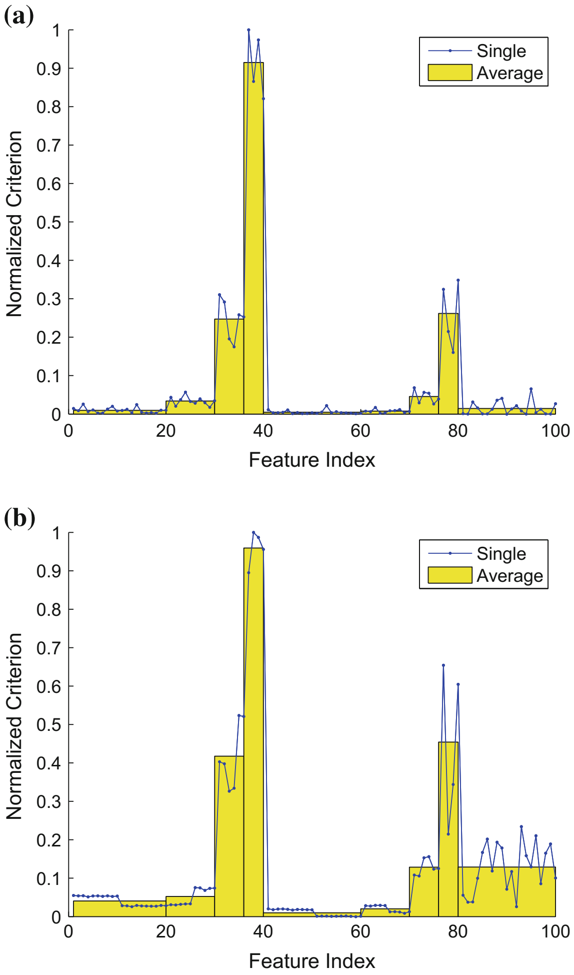


Fig. 10.1 Normalized feature ranking criteria of **a** linear (Eq. 10.4) and **b** nonlinear (Eq. 10.8) SVM-RFE in the synthetic dataset. The curves represent the criteria of each single feature. The bars show the average criteria of the features in the groups with the same size and weight (see Table 10.1). It can be found that if the group size increases or the weight decreases, the criterion decreases

selection or regression on feature groups instead of single features. For example, Sharma et al. (2013) proposed to automatically group and select correlated features based on a penalization scheme. However, the scheme only applies to linear models.

Our method differs from the methods described above. It makes use of the RFE procedure to reduce the influence brought by CB. For efficiency, it is impractical for the RFE procedure to remove one feature each time if the feature dimension is high. When a batch of features are removed in one iteration of RFE, a group of correlated features may be removed entirely. This may either because the features are truly irrelevant, or because their ranking criteria have been incorrectly underestimated. In both conditions, we can move a representative feature of the group back to the surviving feature list. Then it can be evaluated again in the next iteration without the influence of CB. The group representative can be chosen as the feature with the highest criterion in this iteration. This strategy does not change the candidate feature set or the ranking criterion, but monitors and corrects the potentially wrong decisions due to CB.

The details of the correlation bias reduction (CBR) algorithm are shown in Algorithm 10.3.1. $\mathcal{F}^{\text{out}}$ is the list of features to be removed in one iteration of RFE. $\mathcal{F}^{\text{in}}$ is the list of features that survives. The purpose of the algorithm is to move potentially useful features from $\mathcal{F}^{\text{out}}$ back to $\mathcal{F}^{\text{in}}$. In order to identify highly correlated feature groups in $\mathcal{F}^{\text{out}}$, two thresholds T_c and T_g are used. We start from examining the feature with the highest criterion in $\mathcal{F}^{\text{out}}$ and denote it as feature k. If there exist more than T_g features (including k) whose absolute correlation coefficient with k is larger than T_c, they are identified as a group. If none of the group members are in $\mathcal{F}^{\text{in}}$, k should be moved to $\mathcal{F}^{\text{in}}$ since it has the highest criterion in the group. This operation is repeated on all features in $\mathcal{F}^{\text{out}}$. In Algorithm 10.3.1, $|\mathcal{F}|$ represents the number of elements in $\mathcal{F}$.

Algorithm 10.3.1 CBR($\mathcal{F}^{\text{in}}$, $\mathcal{F}^{\text{out}}$)

```
Require: $\mathcal{F}^{\text{out}}$: the list of features to be removed in one iteration of RFE;
    $\mathcal{F}^{\text{in}}$: the list of features that survives;
    Thresholds $T_c$ and $T_g$.
Ensure: Modified $\mathcal{F}^{\text{out}}$ and $\mathcal{F}^{\text{in}}$.
Sort $\mathcal{F}^{\text{out}}$ according to the descending order of the ranking criteria.
for $p = 1$ to $\left|\mathcal{F}^{\text{out}}\right|$ do
  Suppose feature $k$ is the $p$th element of the sorted $\mathcal{F}^{\text{out}}$, let
  $\mathcal{G}^{\text{out}} \leftarrow \left\{i \in \mathcal{F}^{\text{out}} \middle| |\text{corr}(i,k)| > T_c\right\}$;
  $\mathcal{G}^{\text{in}} \leftarrow \left\{j \in \mathcal{F}^{\text{in}} \middle| |\text{corr}(j,k)| > T_c\right\}$.
  if $\left|\mathcal{G}^{\text{out}}\right| > T_g$ and $\left|\mathcal{G}^{\text{in}}\right| == 0$ then
    $\mathcal{F}^{\text{out}} \leftarrow \mathcal{F}^{\text{out}} - k$;
    $\mathcal{F}^{\text{in}} \leftarrow \mathcal{F}^{\text{in}} \cup k$.
  end if
end for
```

The larger T_g, the fewer groups will be identified. In practice, we find that setting T_g to 1 or 2 achieves comparable accuracy. Larger values of T_g will degrade the accuracy since some groups of correlated features are eliminated too early. T_c is the correlation threshold. We will explore the effect of different T_c values on the accuracy in Sect. 10.5.2. The experimental results in Sect. 10.5 prove that the CBR strategy improves the performance of SVM-RFE.

10.3.3 Efficient Implementation of SVM-RFE with CBR

This section describes some details on the implementation of the proposed algorithm. First, the number of features that are removed in each iteration of RFE should be determined. In this chapter, a method that simultaneously considers the time cost and precision is used (Rakotomamonjy 2003). At the beginning of the algorithm, one half of the remaining features are removed in each iteration. When the number of the remaining features is less than an elimination number threshold T_e, they are removed one by one in the following iterations for better precision.

A technique can be applied to accelerate the calculation of the ranking criterion for nonlinear SVM-RFE (Eq. 10.8) with Gaussian kernel. First, Eq. 10.8 is expressed in a matrix form:

$$J(k) = \frac{1}{2}\left(\boldsymbol{\beta}^{\mathrm{T}} H \boldsymbol{\beta} - \boldsymbol{\beta}^{\mathrm{T}} H^{(-k)} \boldsymbol{\beta}\right). \tag{10.11}$$

Here, $\boldsymbol{\beta}$ is the column vector of signed α's, i.e., $\beta_i = \alpha_i y_i$. Only the nonzero α's are included. H is the kernel matrix, $H_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Only the support vectors are included. For the Gaussian kernel, we have $H_{ij} = e^{-\gamma S_{ij}}$, where $S_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$. It is easy to prove that

$$S_{ij}^{(-k)} = \|\boldsymbol{x}_i^{(-k)} - \boldsymbol{x}_j^{(-k)}\|^2 \tag{10.12}$$

$$= S_{ij} - \left(x_i^{(k)} - x_j^{(k)}\right)^2, \tag{10.13}$$

where $x_i^{(k)} \in \mathbf{R}$ is the kth feature of the ith support vector. When computing $S_{ij}^{(-k)}$, we can use Eq. 10.13 to replace the original Eq. 10.12, since the matrix S can be cached and reused, and computing the scalar operation in Eq. 10.13 is much easier than the vector norm in Eq. 10.12. According to experiments using MATLAB, Eq. 10.13 is about 5 times faster.

The complete algorithm of SVM-RFE with CBR is summarized in Algorithm 10.3.2.

10.3.4 Stability Improvement with Ensemble Method

The stability of an FS algorithm is a topic of recent interest (Bhondekar et al. 2011; Awada et al. 2012; Somol and Novovicova 2010; Saeys et al. 2008; Kalousis et al. 2007). A stable FS algorithm is important for data mining applications such as bioinformatics. Stability describes the sensitivity of a method to variations in the training set (Kalousis et al. 2007). If the training set is perturbed, the difference in selected features should not be too large. The Jaccard index is a widely used criterion to measure the difference between two selected feature subsets A and B (Saeys et al. 2008; Somol and Novovicova 2010; Kalousis et al. 2007):

$$S(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{10.14}$$

Its value ranges from 0 to 1, with 0 meaning that the two subsets have no overlap and 1 meaning that they are identical. When evaluating the stability of an FS algorithm,

Algorithm 10.3.2 SVM-RFE with CBR

```
Require: A set of training samples with feature dimension $m$;
    An SVM training algorithm (linear or nonlinear); $T_e$.
Ensure: A ranked list of features $\mathcal{F}^{\text{ranked}} = \mathcal{F}^{\text{out}}$, the most important feature in the first place.
Initialize the list of surviving features $\mathcal{F}^{\text{in}} \leftarrow \{1, \dots, m\}$;
    the list of eliminated features $\mathcal{F}^{\text{out}} \leftarrow \emptyset$.
while $\mathcal{F}^{\text{in}} \neq \emptyset$ do
    Train an SVM model with the features in $\mathcal{F}^{\text{in}}$.
    Calculate the features' ranking criteria with Eq. 10.4, or Eqs. 10.11 and 10.13.
    Sort $\mathcal{F}^{\text{in}}$ according to the descending order of the ranking criteria.
    if $|\mathcal{F}^{\text{in}}| > T_e$ then
        $r = \min(\text{floor}(|\mathcal{F}^{\text{in}}|/2), |\mathcal{F}^{\text{in}}| - T_e)$,
    else
        $r = 1$.
    end if
    $\mathcal{F}^{\text{removing}} \leftarrow$ the last $r$ elements in $\mathcal{F}^{\text{in}}$;
    $\mathcal{F}^{\text{in}} \leftarrow$ the first $|\mathcal{F}^{\text{in}}| - r$ elements in $\mathcal{F}^{\text{in}}$.
    if $r > 1$ then
        Call Algorithm 10.3.1: $(\mathcal{F}^{\text{in}}, \mathcal{F}^{\text{removing}}) \leftarrow \text{CBR}(\mathcal{F}^{\text{in}}, \mathcal{F}^{\text{removing}})$.
    end if
    $\mathcal{F}^{\text{out}} \leftarrow [\mathcal{F}^{\text{removing}}, \mathcal{F}^{\text{out}}]$.
end while
```

we can use the N-fold cross-validation strategy. After N subsets have been selected based on N training sets, the Jaccard index is computed for all $N(N-1)/2$ pairs of subsets. The final stability is the average over all pairs (Kalousis et al. 2007).

To improve the stability of FS algorithms, one of the popular ideas is to use an ensemble method (Awada et al. 2012; Saeys et al. 2008), i.e., to aggregate the outputs of the single feature selectors. In this chapter, we apply this method to SVM-RFE + CBR. Part (9/10 in this chapter) of the training samples are randomly picked to generate a ranked feature list. The process is repeated M times, then the average rank of each feature is used to determine its final rank. In this way, the features with stably good performance are more likely to rank higher. Note that the stability issues and performance of the ensemble method will be separately discussed in Sect. 10.5.5. The results described in Sects. 10.5.1–10.5.4 are obtained without the ensemble method.

10.4 Datasets and Feature Extraction

10.4.1 Dataset 1

10.4.1.1 Description

Dataset 1 will also be investigated in Chap. 14. It consists of breath samples from healthy and diabetic subjects. A breath analysis system was proposed by Yan et al.

Table 10.2 Summary of the sensor array used in dataset 1

No.	Model	Manufacturer	Function
1	TGS4161	Figaro Inc., Japan	CO_2
2	TGS822		VOCs (e.g., acetone), H_2, CO, NH_3, H_2S, etc.
3	TGS826		
4	TGS2610-D00		
5	SP3S-AQ2	FIS Inc., Japan	
6	GSBT11	Ogam Inc., Korea	
7	WSP2111	Winsen Inc., China	
8	TGS2600-TM	Figaro Inc., Japan	
9	TGS2602-TM		
10	WSP2111-TM	Winsen Inc., China	
11	HTG3515CH	Humirel Inc., France	Temperature
12			Humidity

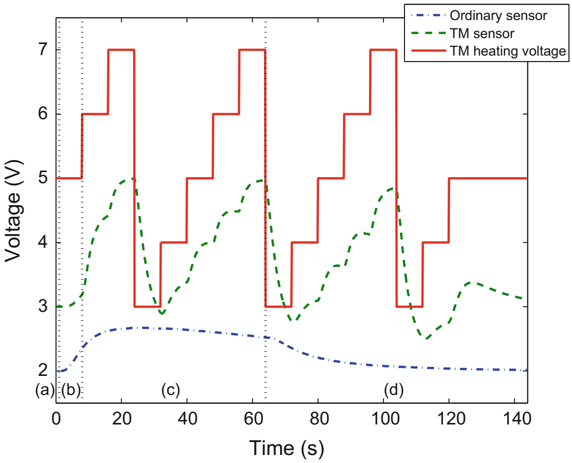


Fig. 10.2 Typical curves in dataset 1. *Solid line* the heating voltage of temperature modulated (TM) sensors. Dashed line: a typical response curve of a TM sensor. *Dash dot line* a typical response curve of an ordinary sensor. The *vertical dotted lines* separate the 4 stages of the sampling procedure: **a** baseline stage; **b** injection stage; **c** reaction stage; **d** purge stage

(2014) to measure breath samples of healthy people and diabetics. They developed an e-nose with a carbon dioxide sensor, a temperature-humidity sensor and 9 metal oxide semiconductor (MOS) sensors. The details of the sensor array are listed in Table 10.2. The carbon dioxide sensor was utilized to compensate for the difference in proportion of alveolar air. The MOS sensors were carefully selected for better accuracy in diabetes identification (Yan and Zhang 2014b). It is worth noting that three of the MOS sensors were operated under temperature modulation (sensor 8-10 with the notation "-TM" in Table 10.2). They were heated by a staircase voltage oscillated between 3 V and 7V. Figure 10.2 illustrates the waveform of the heating voltage and compares typical responses of a TM sensor and an ordinary sensor. More details about the e-nose will be revealed in Chap. 14.

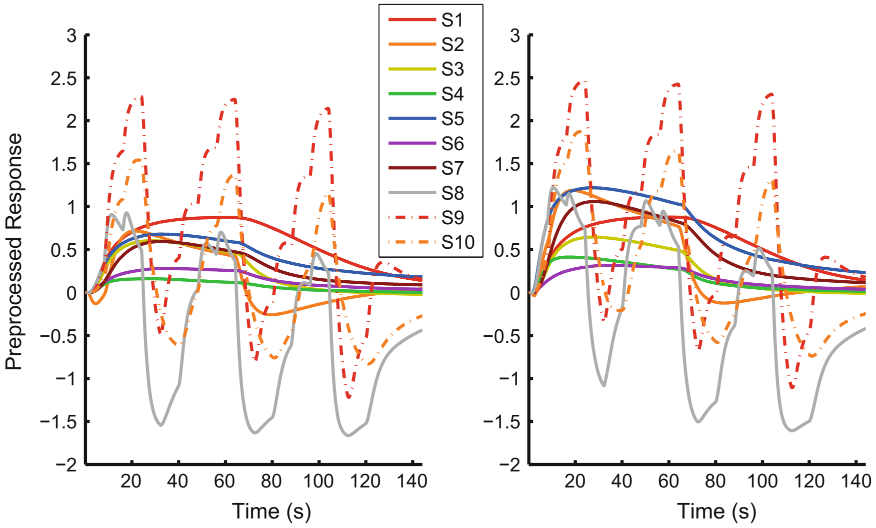


Fig. 10.3 Average preprocessed responses of the two classes in dataset 1. *Left* healthy; *right* diabetes. The sensor models can be found in Table 10.2

A total of 295 healthy and 279 diabetes breath samples were collected. Before feature extraction, the samples are preprocessed. First, the baseline values are subtracted from the sensor responses. Humidity compensation is necessary because breath samples contain water vapor. Linear humidity response models are built for each sensor, then applied to rectify the samples (Yan et al. 2014). Figure 10.3 exhibits the average preprocessed samples of healthy and diabetic subjects. Only the carbon dioxide sensor (S1) and MOS VOC sensors (S2-S10) are drawn. For S2-S10, the responses in diabetes samples are larger than that in healthy samples, showing that the concentration of VOCs in breath of diabetics is higher than that of healthy subjects. Besides, the curve shape of S8 is significantly different between the two classes.

10.4.1.2 Transient Feature Extraction

Traditional features of gas sensors are their steady-state responses. However, additional useful information is carried in the transient responses (Hierlemann and Gutierrez-Osuna 2008; Marco and Gutiérrez-Gálvez 2012). Transient responses are often related to the change of gas flow (injection/purge) or temperature (for TM sensors). In this chapter, 1712 transient features are extracted from sensors 1–10 in dataset 1. The feature set includes magnitude, difference, derivative, second derivative, integral, time constant and phase features. It is a larger and more comprehensive feature set than previous studies (Paulsson et al. 2000; Pardo and Sberveglieri 2008; Cho and Kurup 2011), which enables us to (1) enhance the classification accuracy, for the best feature subset in a large candidate set should be better than that in a

Table 10.3 Feature description for ordinary sensors in dataset 1

Feature type	Description	#Features
Magnitude	Down-sampled values of the curve's magnitude M	21
	The maximum magnitude	1
	Down-sampled values of the normalized magnitude $\tilde{M}, \tilde{M} = M/\max(M)$	21
Difference	The difference F of magnitude M, $F_i = M(t_{i+1}) - M(t_i), t = [0, 8, 36, 64, 92, 120], i = 1, \ldots, 5$	5
Derivative	Down-sampled values of the curve's derivative D	21
	The maximum and minimum derivative	2
2nd derivative	The maximum and minimum 2nd derivative in both injection and purge stage	4
Integral	The integral of the 5 intervals of the curve, the intervals are the same with the difference feature	5
Time constant	The time when the magnitude reaches 30, 60, 90, and 100% of its maximum value ($T_{30}, T_{60}, T_{90}, T_{\max}$), and 90, 60, and 30 of its maximum value in the purge stage ($T_{-90}, T_{-60}, T_{-30}$)	7
	The time when the derivative reaches its maximum and minimum values	2
	The time when the 2nd derivative reaches its maximum and minimum values in both injection and purge stage	4
Phase feature	The phase feature is proposed in (Martinelli et al. 2003). First, the response is transformed to the phase space, which is spanned by its magnitude and derivative. Then, the phase features are defined by $P_i = \int_{M(t_i)}^{M(t_{i+1})} D \, \mathrm{d}M$, t is the same with the difference feature	5

small set; (2) Perform a systematic statistical analysis on the features; (3) Testify the performance of the proposed FS algorithm in a large correlated feature set.

The features extracted from each ordinary sensor are described in Table 10.3. There are 98 features altogether. The difference, integral, and phase features are calculated on 5 intervals of the curve (1 in injection stage, 2 in reaction stage, and 2 in purge stage), which are illustrated in Fig. 10.4a. The shape of a TM sensor's response is more complex and informative. However, the features defined in Table 10.3 can still be used to describe the transient of the curve. Because the response of a TM sensor has 18 "stairs", transient features are extracted on every stair. The features are similar to those in Table 10.3, but the features related to the 2nd derivative in the purge stage and the time constant features $T_{30}, T_{60}, T_{90}, T_{-90}, T_{-60}, T_{-30}$ are not included. The feature dimension for each TM sensor is $18 \times 19 = 342$.

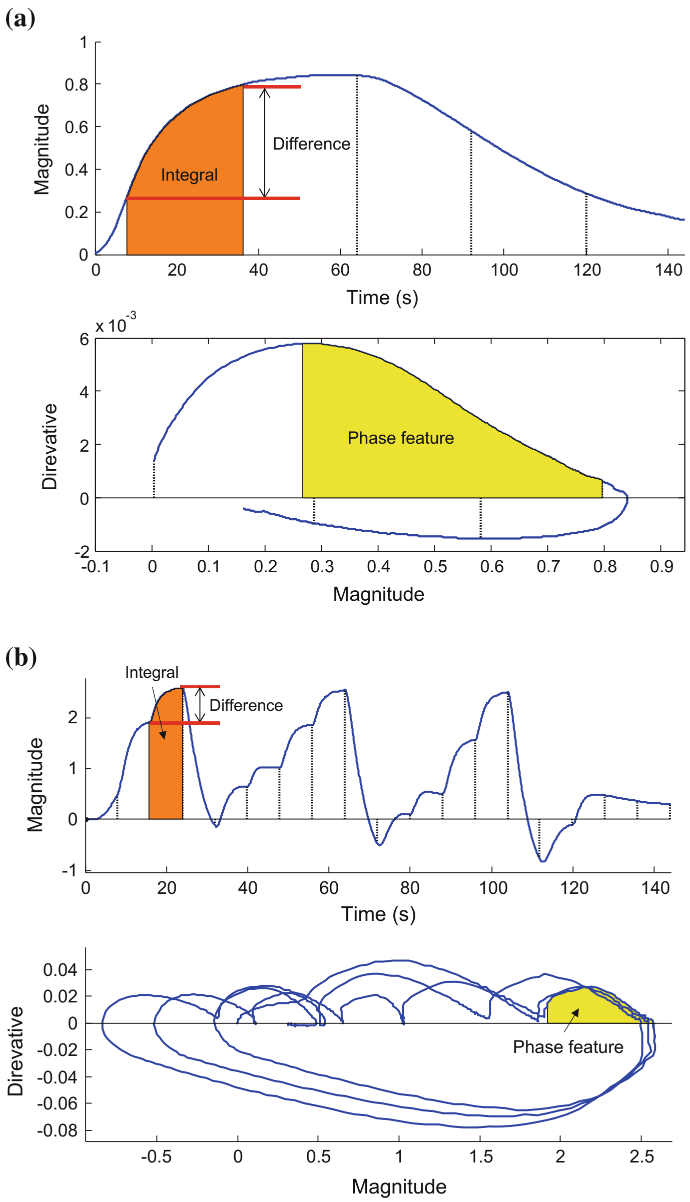


Fig. 10.4 Examples of the difference, integral, and phase feature for **a** ordinary sensors and **b** TM sensors

Table 10.4 Summary of the sensor array used in dataset 2

No.	Model	Gas	Sensitivity (ppm)
1	TGS2600	H_2, CO and VOCs	1–30
2	TGS2602	VOCs	1–30
3	TGS2611-C00	VOCs	500–10000
4	TGS2610-C00	VOCs	500–10000
5	TGS2610-D00	VOCs	500–10000
6	TGS2620	VOCs and CO	50–5000
7	TGS825	H_2S	5–100
8	TGS4161	CO_2	350–10000
9	TGS826	NH_3	30–300
10	TGS2201	NO and NO_2	0.1–10
11	TGS822	VOCs	50–5000
12	TGS821	H_2	10–1000

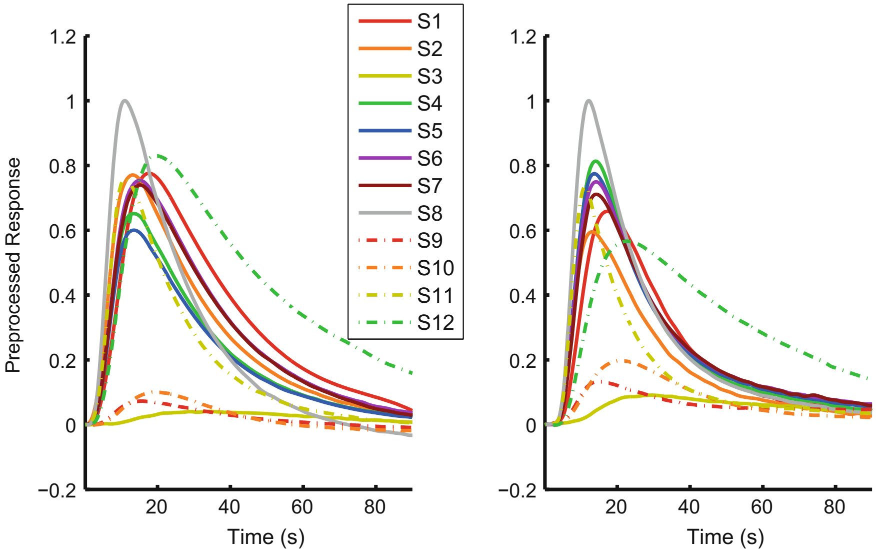


Fig. 10.5 Typical samples in dataset 2. *Left* healthy; *right* airway inflammation. The sensor models are in Table 10.4

10.4.2 Dataset 2

The second dataset, which has been described in Chap. 3, was collected by a breath analysis system designed by Guo et al. (2010). It was equipped with a carbon dioxide sensor and 11 MOS sensors. The details of the sensor array are listed in Table 10.4. All sensors are commercially available from Figaro Inc., Japan. They were operated under constant heating voltage. The sensor array is able to detect biomarkers of several kinds of diseases. There are 135 healthy samples, 181 diabetes samples, 167 renal disease samples, and 126 airway inflammation samples in the dataset. Typical samples in the dataset are displayed in Fig. 10.5. The duration of each breath sample was 90 s. The data preprocessing algorithm includes baseline subtraction and signal normalization (Guo et al. 2010). After preprocessing, 1140 transient features are extracted. The features are similar to the ones for ordinary sensors in dataset 1, but not as many since the sample duration of dataset 2 is shorter.

10.5 Results and Discussion

The performance of the proposed method (without ensemble) on the three datasets will be described and compared in Sects. 10.5.1–10.5.4. Section 10.5.5 will discuss the stability of the FS methods and study the performance of SVM-RFE + CBR with ensemble. Some useful information will be provided in Sect. 10.5.6 by analyzing the feature importance.

10.5.1 Synthetic Dataset

The synthetic dataset has been described in Sect. 10.3.1. It contains several groups of highly correlated features. A tenfold cross-validation was conducted on the dataset to evaluate the algorithms. First, feature ranking was performed on the training sets. Then, linear SVM classifiers based on the top-ranked features were used to classify separate test sets. Finally, the average classification accuracy and the standard deviation are calculated. Because the relationship between the features and the class label is linear, we adopted linear SVM-RFE to rank the features. The penalty parameter C of the SVM models was empirically set to 2^3 for both SVM-RFE and classification. For the RFE procedure, the elimination number threshold is $T_e = 22$. For the CBR strategy, the group size threshold is $T_g = 1$; the correlation threshold is $T_c = 0.9$.

In Fig. 10.6, three algorithms are compared. Besides the linear SVM-RFE with or without CBR, the "slowest" SVM-RFE is also explored, which removes the features one by one in the RFE procedure (equivalent to setting $T_e = \infty$). Theoretically, this method is not affected by correlation bias. The left figure shows how many latent variables have been included in the top-ranked features. Recall that the 80 relevant features in the dataset are generated from 22 latent variables. We can see that the original SVM-RFE fails to include all the variables in the top 30 features, probably because some of the feature groups are eliminated too early due to CB. Both SVM-RFE + CBR and the slowest SVM-RFE succeed to include all the variables in the top 30 features. The right figure compares the accuracy of the three algorithms. This result shows that SVM-RFE + CBR is comparable to the slowest SVM-RFE and better than the original SVM-RFE in the synthetic dataset. It proves the ability of

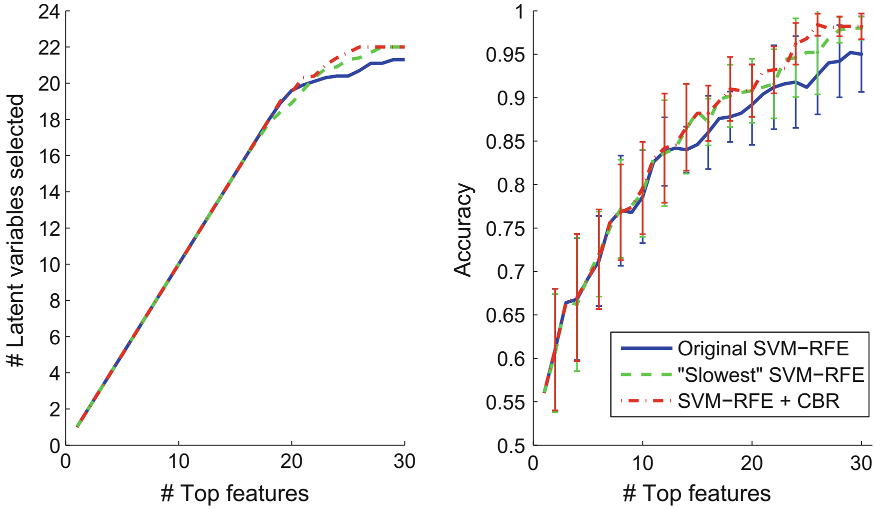


Fig. 10.6 FS results on the synthetic dataset. *Left* the number of latent variables identified in the *top*-ranked features. *Right* average classification accuracy of the *top*-ranked features. The error bars represent the standard deviations

the CBR strategy to reduce the influence of CB. Besides, the slowest SVM-RFE becomes impractical to use when the feature dimension is high. When running the experiment on dataset 1 using MATLAB, SVM-RFE + CBR needed 100 s to rank the 1712 features in one cross-validation, while the slowest SVM-RFE did not finish it in 9 h. So we will not compare the result of the latter method in the breath analysis datasets.

10.5.2 Dataset 1

Similar to the synthetic dataset, tenfold cross-validation was carried out for dataset 1. Gaussian SVM was adopted for classification with parameters $C = 2^3$ and $\gamma = 2^{-6}$. In both linear and nonlinear SVM-RFE algorithms, we set $C = 2^3$ since it has good performance. The kernel parameter γ was searched among $\{2^{-3}, 2^{-4}, \ldots, 2^{-10}\}$ for nonlinear SVM-RFE with or without CBR. Finally, we found that the best accuracy is achieved in both situations when $\gamma = 2^{-8}$. Other parameters for RFE and CBR were: $T_e = 60, T_g = 2$.

In Fig. 10.7, performance of linear and nonlinear SVM-RFE are compared. The shown performance is the best accuracy among the top 60 feature subsets. The x-axis of the figure is the correlation threshold T_c in CBR strategy. When $T_c = 1$, the algorithm is equivalent to the original SVM-RFE without CBR. Table 10.5 shows the classification accuracy of several feature extraction/selection methods. The parame-

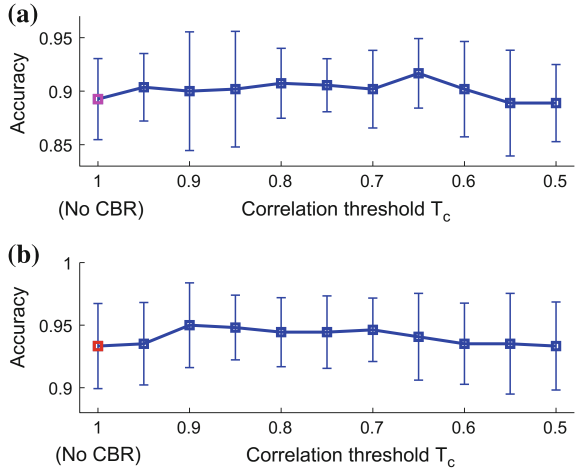


Fig. 10.7 Average accuracy of **a** linear and **b** nonlinear SVM-RFE with CBR in dataset 1 with varying correlation threshold T_c. When $T_c = 1$, the algorithm is equivalent to the original SVM-RFE without CBR. The error bars represent the standard deviations

ters of the methods have been optimized. In the principal component analysis (PCA) method, the ratio of variance (Yan et al. 2014) was searched from 80% to 99.9%. In the min-redundancy max-relevance (mRMR) method, the features were discretized to three levels to compute the mutual information (Peng et al. 2005). When hierarchical clustering (HC) was applied before SVM-RFE, the number of clusters was searched between 800 to 1700. In clinical applications, sensitivity and specificity measures are important, so they are also listed in Table 10.5. Sensitivity is the proportion of correctly classified patients in all the patients, while specificity is the proportion of correctly classified healthy subjects in all the healthy subjects. The accuracy is presented as “mean ± standard deviation”.

10.5.3 Dataset 2

There are three subproblems in dataset 2: discriminating healthy samples from diabetes, renal disease, and airway inflammation samples, respectively. tenfold cross-validation was carried out for each problem. The experimental configurations were similar to those in Sect. 10.5.2. The kernel parameter was also separately tuned in nonlinear SVM-RFE with or without CBR. The value of T_c was searched among $0.5, 0.55, \ldots, 0.95$ for better accuracy. The results are shown in Tables 10.6, 10.7 and 10.8.

Table 10.5 Performance comparison of various methods in dataset 1. The stability is the average Jaccard index

Algorithm	#Features	Sensitivity (%)	Specificity (%)	Average acc. (%)	Stability
All transient	1712	87.04 ± 6.36	90.74 ± 5.59	88.89 ± 4.86	–
PCA + transient	58	90.74 ± 5.01	90.00 ± 4.95	90.37 ± 2.44	–
PCA + magnitude	40	92.59 ± 4.62	90.74 ± 6.11	91.67 ± 4.56	–
mRMR	40	90.37 ± 4.68	91.85 ± 4.55	91.11 ± 3.24	**0.5183**
SFS	28	83.70 ± 6.10	88.15 ± 10.88	85.93 ± 6.25	0.1088
Original linear SVM-RFE	57	90.37 ± 7.03	88.15 ± 4.20	89.26 ± 3.79	0.2201
Linear SVM-RFE + CBR ($T_c = 0.65$)	**17**	90.74 ± 7.66	92.60 ± 3.51	91.67 ± 3.24	0.2238
Original nonlinear SVM-RFE	30	93.33 ± 5.74	93.33 ± 4.20	93.33 ± 3.40	0.4996
Nonlinear SVM-RFE + HC	36	94.07 ± 5.30	94.07 ± 4.95	94.07 ± 3.50	0.4228
Nonlinear SVM-RFE + CBR ($T_c = 0.9$)	31	**94.44** ± **6.11**	**95.56** ± **3.40**	**95.00** ± **3.39**	0.4572

Table 10.6 Performance comparison of various SVM-RFE strategies in dataset 2: distinguishing between healthy and diabetes samples

Algorithm	#Features	Sensitivity (%)	Specificity (%)	Average acc. (%)	Stability
Linear	59	90.00 ± 8.15	90.00 ± 8.92	90.00 ± 6.07	0.0531
Linear + CBR ($T_c = 0.7$)	**43**	90.00 ± 7.30	90.77 ± 8.73	90.38 ± 6.35	0.0525
Nonlinear	52	97.69 ± 3.72	**99.23** ± **2.43**	98.46 ± 2.69	**0.4131**
Nonlinear + CBR ($T_c = 0.9$)	56	**99.23** ± **2.43**	**99.23** ± **2.43**	**99.23** ± **1.62**	0.4087

10.5.4 Discussion

Figure 10.7 shows that the CBR strategy improves the average accuracy of both linear and nonlinear SVM-RFE when T_c is not less than 0.6. The improvement seems not very obvious because the standard deviation (SD) is relatively large. However, when observing the detailed results of dataset 1 and 2 (Tables 10.5, 10.6, 10.7 and 10.8), we find that SVM-RFE + CBR has comparable or lower SD than the original SVM-RFE. Moreover, the sensitivity, specificity, and average accuracy of SVM-RFE + CBR are consistently better than that of SVM-RFE. So the CBR strategy is effective in terms of accuracy. The SD of other feature extraction/selection methods such as PCA, SFS, and mRMR is comparable to SVM-RFE, which is probably caused by the

Table 10.7 Performance comparison of various SVM-RFE strategies in dataset 2: distinguishing between healthy and renal disease samples

Algorithm	#Features	Sensitivity (%)	Specificity (%)	Average acc. (%)	Stability
Linear	60	92.31 ± 8.11	86.92 ± 8.92	89.62 ± 5.75	0.0494
Linear + CBR ($T_c = 0.7$)	50	92.31 ± 5.13	90.00 ± 7.30	91.15 ± 5.14	0.0490
Nonlinear	40	96.92 ± 7.43	97.96 ± 5.19	97.31 ± 4.23	**0.4077**
Nonlinear + CBR ($T_c = 0.85$)	**32**	**98.46** ± **3.24**	**98.46** ± **3.24**	**98.46** ± **1.99**	0.3510

Table 10.8 Performance comparison of various SVM-RFE strategies in dataset 2: distinguishing between healthy and airway inflammation samples

Algorithm	#Features	Sensitivity (%)	Specificity (%)	Average acc. (%)	Stability
Linear	54	83.33 ± 11.11	78.33 ± 11.25	80.83 ± 7.14	0.0473
Linear + CBR ($T_c = 0.8$)	52	86.67 ± 8.96	79.17 ± 10.58	82.92 ± 7.47	0.0453
Nonlinear	**45**	93.33 ± 8.29	92.50 ± 7.30	92.92 ± 5.49	**0.4844**
Nonlinear + CBR ($T_c = 0.8$)	**45**	**95.00** ± **5.83**	**93.33** ± **5.27**	**94.17** ± **2.91**	0.4322

fact that the number of training samples is limited. We also find that the accuracy of nonlinear SVM-RFE is always better than the linear one, which is because of the nonlinear nature of the data. The best T_c value varies between 0.65 and 0.9 depending on the dataset and the algorithm (linear or nonlinear).

In Table 10.5, when all the transient features are used, the accuracy is not very good. Although it contains useful features for classification, the transient feature set also contains irrelevant and redundant features, which will hinder the training of the classifier. PCA can reduce the redundancy within features, so the accuracy is improved in the method "PCA + transient features." In the "PCA + magnitude" method, PCA is applied to the magnitude of the whole curve as presented in Chap. 14. Its accuracy is even better, possibly because there are less irrelevant features in the magnitude.

A proper FS method can be used to identify the effective features and discard the irrelevant and redundant ones. We compared a few FS methods that are able to handle redundancy in candidate features. The mRMR method proposed by Peng et al. (Peng et al. 2005) is a popular filter method which selects relevant and nonredundant features based on a mutual information criteria. The widely used sequential forward selection (SFS) algorithm (Paulsson et al. 2000) iteratively examines each feature and selects the one that maximizes the cross-validation accuracy in the training set. If there is redundancy in features, the time cost of SFS will increase, but the accuracy will not be affected. The drawback of the two methods is that they use greedy strategies, thus are prone to be trapped in local optima. As a wrapper method, SFS often overfits the training set, especially when the sample size is much smaller than the feature dimension, which results in a low accuracy in Table 10.5. Additionally, it is often impractical to use wrapper methods such as SFS and genetic algorithm on high-dimensional FS problems due to the large time cost.

Besides the proposed CBR strategy, there is an alternative way to deal with the correlation bias problem in SVM-RFE. The intuitive idea is to filter the redundant features before FS. Following (Park et al. 2007), we implemented a method which uses hierarchical clustering to group the correlated features. The feature closest to the group center is kept in each group. Then the filtered features (about 1500 in dataset 1) are ranked using nonlinear SVM-RFE. Table 10.5 shows that this method generates a higher accuracy (94.07%) than the original nonlinear SVM-RFE (93.33%). However, the proposed nonlinear SVM-RFE + CBR achieves the best accuracy 95.00% with fewer features.

10.5.5 Stability Analysis and the Ensemble Method

The stability of the FS algorithms listed in Tables 10.5, 10.6, 10.7 and 10.8 is the average Jaccard index (see Sect. 10.3.4) of the top 60 features. The overall stability is not high, which is mainly because there are many highly correlated features, hence the same accuracy can be achieved by different feature subsets. In Table 10.5, mRMR achieves the highest stability. The other algorithms all depend on the training of the SVM classifier, thus will be more sensitive to the perturbation of the training set. This result is consistent with (Kalousis et al. 2007), where SVM-RFE was found to be less stable than univariate filter methods. The stability of nonlinear SVM-RFE is better than the linear one due to the nonlinear nature of the data. The stability of SVM-RFE + CBR is slightly lower than SVM-RFE. The possible reason is that the CBR strategy moves several features back to the surviving feature list in each RFE iteration, which increases the uncertainty of the algorithm, since deciding which feature to move can be sensitive to sample perturbation if several features are highly correlated and have similar ranking criteria.

In order to improve the stability of SVM-RFE + CBR, the ensemble method introduced in Sect. 10.3.4 was investigated. Figure 10.8 shows the accuracy and stability of the ensemble method as the ensemble size (the number of ranking processes to be aggregated) changes. The results were obtained by tenfold cross-validation followed by averaging over 10 repetitions. It can be seen that as the ensemble size increases, the average accuracy and standard deviation do not change obviously, but the stability has significant improvement. When the ensemble size is greater than 9, the stability of nonlinear SVM-RFE + CBR is better than mRMR. So the ensemble method is able to improve the stability at the cost of more computation time. It is

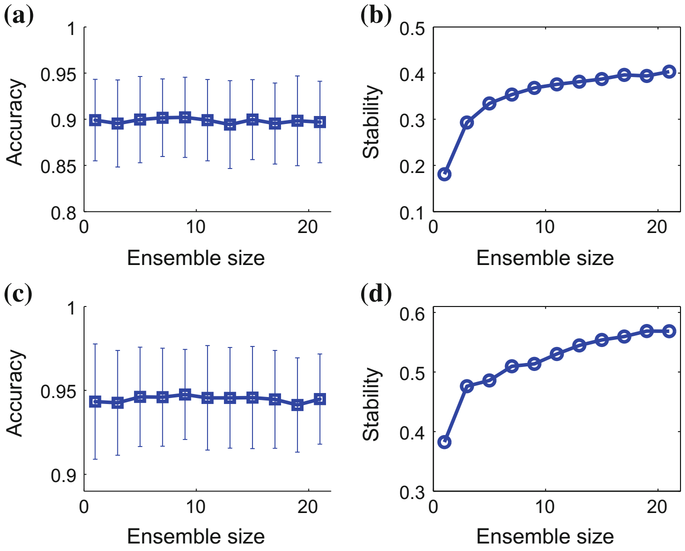


Fig. 10.8 Average accuracy and stability of SVM-RFE + CBR with ensemble in dataset 1. Plot **a** and **b** correspond to linear SVM-RFE + CBR with $T_c = 0.65$. Plot **c** and **d** correspond to nonlinear SVM-RFE + CBR with $T_c = 0.9$. The error bars represent the standard deviations

worth noting that higher accuracy can be coupled with relatively lower stability particularly in the presence of highly correlated features (Kalousis et al. 2007). In FS applications where stability and accuracy are both important, ensemble methods can be considered.

10.5.6 Analysis of the Ranked Features

FS techniques can help us understand the data better. By analyzing the importance of the features, useful information about the sensors and feature extraction algorithms can be obtained. Dataset 1 is studied since it contains TM sensors. We wish to find the answers to several questions: In the application of diabetes identification, which sensors are important? What kinds of features are suitable for ordinary/TM sensors? What heating voltage is a better choice for TM sensors?

The output of SVM-RFE + CBR is a ranked feature list. The ranking of a feature indicates its importance. However, a sensor or a type of feature (such as the magnitude feature) is made up of a group of features. Their importance needs to be estimated according to a group of rankings. Simply averaging the rankings may be improper, because we are more interested in whether the feature group contains useful features. The irrelevant features in the group may degrade its ranking and mislead our judgement. As a result, we use the "average top rank" criterion, namely the average of the top 5 rankings of the features in the group, to evaluate the feature groups. The smaller the criterion, the more important the feature group. Note that the ten ranking lists of the tenfold cross-validation are pooled together. Using this criterion, some helpful conclusions can be summarized:

- **Sensors**. The importance order of the sensors is TGS2600-TM, TGS2602-TM, TGS2610-D00, TGS826, WSP2111-TM, TGS4161, GSBT11, SP3S-AQ2, TGS822, WSP2111. The result is not identical with that in Chap. 5, but some similar trends can be observed. The two most important sensors are both TM sensors. The models of WSP2111-TM and WSP2111 are the same, but the former one is operated under TM, which makes it turn from the least important sensor to the 5th one. To the best of our knowledge, (Yan et al. 2014) is the first literature that applied TM technique to breath analysis systems. The results prove the effectiveness of TM in such applications. The carbon dioxide sensor (TGS4161) has an average top rank of 11.6, showing that it is useful for the application.
- **Feature types**. The average top ranks of the seven types of transient features are displayed in Table 10.9. The phase feature extracted from TM sensors is the most effective. The time constant (especially the $T_{\max}$ feature) and derivative are effective for both types of sensors, while the 2nd derivative and integral are the least effective ones. The normalized magnitudes show slightly smaller average top ranks than the magnitudes without normalization.
- **TM heating voltages**. We find that the average top rank is smaller when the heating voltage is in the interval of 6V-7V-3V (see Fig. 10.2). It implies that when detecting breath biomarkers such as acetone, the TM sensors' responses are more discriminative when the temperature is close to or higher than normal range. According to (Hosseini-Golgoo and Hossein-Babaei 2011), responses at low temperatures contained mostly redundant or indiscriminative information. This is consistent with our study. But it needs further investigation whether the high heating voltage will increase sensor drift.

Table 10.9 Average top ranks of the seven types of transient features extracted from ordinary and TM sensors, respectively. The smaller the better

Feature type	Ordinary sensor	TM sensor
Magnitude	7.2	6.4
Difference	20.4	4.6
Derivative	6.2	2.6
2nd derivative	80.4	35.4
Integral	46.0	15.6
Time constant	**4.0**	2.6
Phase feature	11.8	**1.0**

10.6 Summary

In this chapter, the linear and nonlinear support vector machine recursive feature elimination (SVM-RFE) algorithms were studied. The correlation bias problem in SVM-RFE was raised, which will affect the accuracy of the feature selection result. The correlation bias reduction (CBR) algorithm was proposed to solve the problem by improving the feature elimination strategy. A synthetic dataset and two breath analysis datasets with large sets of correlated features were used to evaluate the algorithms. The nonlinear SVM-RFE + CBR was proved to be effective. It outperformed the original SVM-RFE and other typical algorithms. A complete and efficient implementation of the proposed method was also presented. The stability of the proposed algorithm can be improved by applying the ensemble method.

In this study, the comprehensive feature set included seven types of transient features. By analyzing the features' rankings, some useful knowledge was obtained. Three representative conclusions for dataset 1 are:

- MOS sensors with temperature modulation (TM) are significantly more effective than those without TM.
- Phase feature is the best feature for TM sensors and time constant is the best for ordinary sensors.
- The TM sensors' responses are more discriminative when the temperature is close to or higher than the normal range.

These information will be helpful when making new breath analysis systems or designing new features. For example, the low-ranking sensors may be discarded to lower the cost without precision loss. Features related to TM sensors or the phase space can be further studied. In summary, the proposed FS algorithm is a promising method for accuracy enhancement, dimension reduction and data interpretation. Future works may include investigation of more kinds of features (Zhang et al. 2008; Gutierrez-Osuna et al. 2003).

References

Awada W, Khoshgoftaar TM, Dittman D, Wald R, Napolitano A (2012) A review of the stability of feature selection techniques for bioinformatics data. In: 2012 IEEE 13th international conference on information reuse and integration (IRI). IEEE, Las Vegas, USA, pp 356–363

Bhondekar AP, Kaur R, Kumar R, Vig R, Kapur P (2011) A novel approach using dynamic social impact theory for optimization of impedance-tongue (itongue). Chemom Intell Lab 109(1):65–76

Burges CJ (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Disc 2(2):121–167

Cho JH, Kurup PU (2011) Decision tree approach for classification and dimensionality reduction of electronic nose data. Sens Actuators: B Chem 160(1):542–548

Duan KB, Rajapakse JC, Wang H, Azuaje F (2005) Multiple SVM-RFE for gene selection in cancer classification with expression data. IEEE T NanoBiosci 4(3):228–234

Gualdrón O, Brezmes J, Llobet E, Amari A, Vilanova X, Bouchikhi B, Correig X (2007) Variable selection for support vector machine based multisensor systems. Sens Actuators: B Chem 122(1):259–268

Guo D, Zhang D, Li N, Zhang L, Yang J (2010) A novel breath analysis system based on electronic olfaction. IEEE Trans Biomed Eng 57(11):2753–2763

Gutierrez-Osuna R, Gutierrez-Galvez A, Powar N (2003) Transient response analysis for temperature-modulated chemoresistors. Sens Actuators: B Chem 93(1):57–66

Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182

Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46(1–3):389–422

Hierlemann A, Gutierrez-Osuna R (2008) Higher-order chemical sensing. Chem Rev 108(2):563–613

Hosseini-Golgoo S, Hossein-Babaei F (2011) Assessing the diagnostic information in the response patterns of a temperature-modulated tin oxide gas sensor. Meas Sci Technol 22(3):035, 201

Kalousis A, Prados J, Hilario M (2007) Stability of feature selection algorithms: a study on high-dimensional spaces. Knowl Inf Syst 12(1):95–116

Kaur R, Kumar R, Gulati A, Ghanshyam C, Kapur P, Bhondekar AP (2012) Enhancing electronic nose performance: a novel feature selection approach using dynamic social impact theory and moving window time slicing for classification of kangra orthodox black tea (camellia sinensis (l.) o. kuntze). Sens Actuators B: Chem 166:309–319

Llobet E, Gualdrón O, Vinaixa M, El-Barbri N, Brezmes J, Vilanova X, Bouchikhi B, Gomez R, Carrasco J, Correig X (2007) Efficient feature selection for mass spectrometry based electronic nose applications. Chemom Intell Lab 85(2):253–261

Marco S, Gutiérrez-Gálvez A (2012) Signal and data processing for machine olfaction and chemical sensing: a review. IEEE Sens J 12(11):3189–3214

Martinelli E, Falconi C, D'Amico A, Di Natale C (2003) Feature extraction of chemical sensors in phase space. Sens Actuators: B Chem 95(1):132–139

Mundra PA, Rajapakse JC (2010) SVM-RFE with MRMR filter for gene selection. IEEE Trans NanoBiosci 9(1):31–37

Pardo M, Sberveglieri G (2008) Random forests and nearest shrunken centroids for the classification of sensor array data. Sens Actuators: B Chem 131(1):93–99

Park MY, Hastie T, Tibshirani R (2007) Averaged gene expressions for regression. Biostatistics 8(2):212–227

Paulsson N, Larsson E, Winquist F (2000) Extraction and selection of parameters for evaluation of breath alcohol measurement with an electronic nose. Sens Actuators: A Phys 84(3):187–197

Peng H, Long F, Ding C (2005) Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 27(8):1226–1238

Rakotomamonjy A (2003) Variable selection using SVM based criteria. J Mach Learn Res 3:1357–1370

Saeys Y, Inza I, Larrañaga P (2007) A review of feature selection techniques in bioinformatics. Bioinformatics 23(19):2507–2517

Saeys Y, Abeel T, Van de Peer Y (2008) Robust feature selection using ensemble feature selection techniques. In: Machine learning and knowledge discovery in databases. Springer, pp 313–325

Sharma DB, Bondell HD, Zhang HH (2013) Consistent group identification and variable selection in regression with correlated predictors. J Comput Graph Stat 22(2):319–340

Somol P, Novovicova J (2010) Evaluating stability and comparing output of feature selectors that optimize feature subset cardinality. IEEE Trans Pattern Anal Mach Intell 32(11):1921–1939

Tang Y, Zhang YQ, Huang Z (2007) Development of two-stage SVM-RFE gene selection strategy for microarray expression data analysis. IEEE ACM T Comput Bi 4(3):365–381

Toloşi L, Lengauer T (2011) Classification with correlated features: unreliability of feature ranking and solutions. Bioinformatics 27(14):1986–1994

Yan K, Zhang D (2014a) Blood glucose prediction by breath analysis system with feature selection and model fusion. In: 2014 36th Annual international conference of the IEEE engineering in medicine and biology society (EMBC). IEEE, pp 6406–6409

Yan K, Zhang D (2014b) Sensor evaluation in a breath analysis system. In: 2014 International Conference on medical biometrics (ICMB). IEEE, pp 35–40

Yan K, Zhang D (2015) Feature selection and analysis on correlated gas sensor data with recursive feature elimination. Sens Actuators B: Chem 212:353–363

Yan K, Zhang D, Wu D, Wei H, Lu G (2014) Design of a breath analysis system for diabetes screening and blood glucose level prediction. IEEE Trans Biomed Eng 61(11):2787–2795

Yoon S, Kim S (2009) Mutual information-based SVM-RFE for diagnostic classification of digitized mammograms. Pattern Recogn Lett 30(16):1489–1495

Zhang S, Xie C, Hu M, Li H, Bai Z, Zeng D (2008) An entire feature extraction method of metal oxide gas sensors. Sens Actuators: B Chem 132(1):81–89

Chapter 11
Breath Sample Identification by Sparse Representation-Based Classification

Abstract It has been discovered that some compounds in human breath can be used to detect some diseases and monitor the development of the conditions. A sensor system in tandem with certain data evaluation algorithm offers an approach to analyze the compositions of breath. Currently, most algorithms rely on the generally designed pattern recognition techniques rather than considering the specific characteristics of data. They may not be suitable for odor signal identification. This chapter proposes a Sparse Representation-based Classification (SRC) method for breath sample identification. The sparse representation expresses an input signal as the linear combination of a small number of the training signals, which are from the same category as the input signal. The selection of a proper set of training signals in representation, therefore, gives us useful cues for classification. Two experiments were conducted to evaluate the proposed method. The first one was to distinguish diabetes samples from healthy ones. The second one aimed to classify these diseased samples into different groups, each standing for one blood glucose level. To illustrate the robustness of this method, two different feature sets, namely, geometry features and principle components were employed. Experimental results show that the proposed SRC outperforms other common methods, such as Support Vector Machine (SVM) and K-Nearest Neighbor (KNN), irrespective of the features selected.

Keywords Breath analysis · Disease identification · Sparse representation · Diabetes · Blood glucose levels

11.1 Introduction

Endogenous molecules in human breath, such as acetone, nitric oxide, hydrogen and ammonia, are produced by metabolic processes. They are separated from blood and enter into the alveolar air via the alveolar pulmonary membrane (DAmico et al. 2007; Schubert et al. 2004; Miekisch et al. 2004). Variation in the concentration of these molecules can suggest various diseases or at least changes in metabolism (Amann et al. 2005). These molecules are considered as biomarkers of the presence of diseases and clinical conditions. For instance, nitric oxide in breath can be measured

D. Zhang et al., *Breath Analysis for Medical Applications*,
DOI 10.1007/978-981-10-4322-2_11

as an indicator of asthma or other conditions characterized by airway inflammation (Deykin et al. 2002). Breath isoprene is significantly lower in cystic fibrosis patients with acute respiratory exacerbation (McGrath et al. 2000). Increased pentane and carbon disulfide have been observed in the breath of patients with schizophrenia (Phillips et al. 1993). Breath concentration of volatile organic compounds (VOCs) such as cyclododecatriene, benzoic acid, and benzene are much higher in lung cancer patients than in control groups (Phillips et al. 2007). Acetone has been found to be more abundant in the breath of diabetics (Fleischer et al. 2002; Deng et al. 2004), and breath ammonia is significantly elevated in patients with renal diseases (Davies et al. 1997).

By detecting the molecules in breath, one can identify the diseases in an early stage and monitor their development. Compared with other traditional methods, such as blood and urine tests, breath analysis is noninvasive, real time, and the least harmful to not only the subjects but also the personnel who collect the samples (Berkel et al. 2008). Increasing interest has been expressed about the applications of breath analysis in medicine and clinical pathology both as a diagnostic tool and as a way to monitor the progress of therapies (Francesco et al. 2005; Dweik and Amann 2008). With the development of sensor technology, sensor systems promise a number of advantages in breath analysis compared with the traditional analytical instruments, like the gas chromatograph, because of its low-cost and ease of operation property. In recent years, much work has been reported about the application of one kind of sensor system, namely, electronic noses (e-noses), to breath analysis. To take some examples, D'Amico et al. have reported a diagnosis method for lung cancer by the analysis of breath by means of an e-nose with eight quartz microbalance (QMB) gas sensors (Natale et al. 2003; D'Amico et al. 2009). Dragonieri et al. have introduced the application of an e-nose with 32 polymer sensors to lung diseases identification, such as asthma, CPOD, pneumonia, lung cancer, and the postoperative state of lung cancer (Dragonieri et al. 2007, 2009). Yu et al. have detected diabetes by analyzing the acetone in patients' breath using an array with four conducting polymer sensors (Yu et al. 2005). Shih et al. have used an e-nose with an array of 24 individual transducers to detect and identify bacterial infections of the lungs and airways (Shih et al. 2010).

While these methods work satisfactorily in some applications, the results could possibly be improved. That is because, on the one hand, the analysis tools they have utilized were commercial e-noses. Current commercial e-noses, for the sake of their marketing concerns, have to provide some versatility in applications, such as coffee, wine, and fragrances identification. The versatility, in contrast, limits their performance in disease detection since their sensor selection has to match broad applications. On the other hand, the data evaluation methods they have used are generally designed pattern recognition algorithms, such as Principal Component Analysis (PCA) plus Euclidean distance and Canonical Discriminant Analysis (CDA) plus Mahalanobis distance, which rarely consider the specific characteristics of samples and may not be very suitable for odor signal identification.

It has been claimed that the sensor array itself, the reference data set, and the data evaluation algorithms are all responsible for the system performance (Rock et al.

2008). The improvements taken in these three aspects would help to enhance the disease identification accuracy. In our previous work, we have developed a system especially for breath analysis (Guo et al. 2010b). The system employs 12 chemical sensors to detect the abnormal concentration of the molecules in breath. In contrast to the broad panel of nonspecific sensors used in commercial e-noses, the sensors in our system are selected particularly to be sensitive to the disease biomarkers in human breath. Previous work has shown that when used in tandem with general pattern recognition methods, for example, PCA plus KNN classifier, the system performed well in detecting diseases, monitoring the progress of related therapies, and evaluating the efficiency of medical treatment (Guo et al. 2010a, b, c).

Although the mentioned system works well, its performance can be further improved when a comparatively appropriate pattern recognition method is employed. Breath signals have the following characteristics: on the one hand, these data are with high dimensionality; on the other, the number of samples is limited, due to the high cost of data collection. This means the number of samples utilized to train the classifier is too small relative to the dimensionality of data in each sample. In this case, the traditional statistical pattern recognition method may not work well, since the efficiency of these methods is highly dependent on the interrelationship between sample sizes, number of features, and structure of classifiers (Jain and Mao 2000).

In this chapter, we propose a Sparse Representation-based Classification (SRC) method for the breath signal identification to improve the classification accuracy of the system. Sparse representation has shown great promise in face recognition (Wright et al. 2008). The basic idea is to search for the most compact representation of an input signal in terms of linear combination of atoms in an overcomplete dictionary (Huang and Aviyente 2007). We apply this method in disease identification in this chapter. For semiconducting metal oxide sensor, the transient response is related to the gas type, and the amplitude of the response is associated with gas concentration (Liess 2002). Hence, the data of the samples from the same class, which only have the difference in amplitude, are highly correlated and can be linearly represented by each other. And the data of the samples from different classes, whose transient responses and amplitudes are different, are independently distributed and cannot be linearly represented by each other. We represent a test sample as the linear combination of a set of training samples. In the linear combination, ideally, the training samples that are from the same class as the test sample have nonzero coefficients, while those from the different class as the test sample have zero coefficients. Accordingly, we can assign a test sample to the class whose training samples hold higher linear combination coefficients. The coefficients can be obtained by solving an optimization problem with the constraint of l_1 norm minimization. To our knowledge, there is no similar odor signal identification system that using SRC. The SRC method is not only effective in the signal analysis of our system, but useful and applicable to the performance enhancement of other current existing electronic noses.

Two experiments were conducted to evaluate the performance of this method. The first aimed to distinguish samples with diabetes from healthy ones (disease diagnosis). The second was to classify these diseased samples into different groups, each standing for a distinct blood glucose level (condition monitoring). Two different feature sets, namely, geometry features and principle components were employed to illustrate the robustness of this method. As classical classifiers, SVM and KNN have been widely used in odor signal identification (Brudzewski et al. 2004; Pardo and Sberveglieri 2005; Pardo et al. 2005; Zhang et al. 2008a). In this chapter, we compared the proposed method with SVM and KNN in terms of classification accuracy. Experimental results show that the proposed SRC outperforms SVM and KNN, irrespective of the features selected for classification.

The rest of this chapter is organized as follows: Sect. 11.2 describes in detail the proposed SRC method and its application to odor sample identification. Section 11.3 summarizes the overall disease identification procedure. Section 11.4 explains the experimental details and presents the results and discussion. Finally, Sect. 11.5 offers some conclusions.

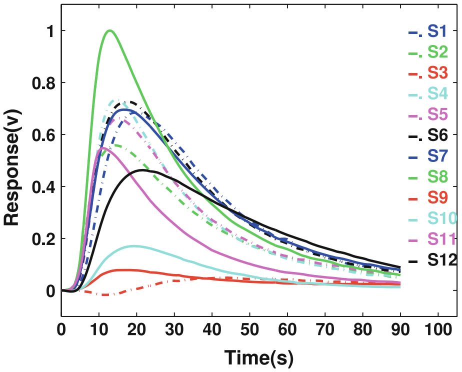
11.2 Sparse Representation Classification

This section describes the SRC method and its application to odor signal identification in detail.

11.2.1 Data Expression

Figure 11.1 is a typical output of the breath analysis system. The horizontal axis stands for the sampling time (0–90 s) and the vertical axis shows the amplitude of the sensor output in volts. The output is composed of l sensors' responses, where $l = 12$ since there are twelve chemical sensors involved in the system. The response of each sensor, i.e., each curve in Fig. 11.1, is a discrete time series with d dimensions ($d = 810$ according to the sampling rule). The response of the k-th sensor in the sample is:

Fig. 11.1 The typical output of the system



$$s = [t_{1,k}, t_{2,k}, \dots, t_{d,k}]^{\mathrm{T}}. \tag{11.1}$$

The sample with l sensors can be expressed as a matrix:

$$\mathbf{S} = \begin{bmatrix} t_{1,1}, t_{1,2}, \dots, t_{1,l} \\ t_{2,1}, t_{2,2}, \dots, t_{2,l} \\ \vdots \\ t_{d,1}, t_{d,2}, \dots, t_{d,l} \end{bmatrix}. \tag{11.2}$$

We can express the sample matrix as a vector by stacking its columns,

$$\mathbf{v} = [\mathrm{t}_{1,1}, \dots, \mathrm{t}_{\mathrm{d},1}, \mathrm{t}_{1,2}, \dots, \mathrm{t}_{\mathrm{d},2}, \dots, \mathrm{t}_{1,\mathrm{l}}, \dots, \mathrm{t}_{\mathrm{d},\mathrm{l}}]^{\mathrm{T}} \in \mathbf{R}^{m}. \tag{11.3}$$

Assume there are k classes, the i-th class includes n_i samples. All training samples from the i-th object class can be expressed as a matrix,

$$\mathbf{A}_i = [\mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \dots, \mathbf{v}_{i,n_i}] \in \mathbf{R}^{m \times n_i}. \tag{11.4}$$

All samples from k object classes form a new matrix:

$$\begin{aligned} \mathbf{A} &= [\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_k] \\ &= [\mathbf{v}_{1,1}, \dots, \mathbf{v}_{1,n_1}, \mathbf{v}_{2,1}, \dots, \mathbf{v}_{2,n_2}, \dots, \mathbf{v}_{k,1}, \dots, \mathbf{v}_{k,n_k}]. \end{aligned} \tag{11.5}$$

$\mathbf{A} \in \mathbf{R}^{m \times n}$ is called a dictionary matrix. m is the dimension of each sample and n is the number of all training samples in matrix $\mathbf{A}$.

11.2.2 Test Sample Representation by Training Samples

One of the test samples $\mathbf{y}$ can be expressed as a linear combination of all training samples from all object classes:

$$\mathbf{y} = \alpha_{1,1}\mathbf{v}_{1,1} + \alpha_{1,2}\mathbf{v}_{1,2} +, \dots, \alpha_{k,n_k}\mathbf{v}_{k,n_k} = \mathbf{A}\mathbf{x} \in \mathbf{R}^{m}, \tag{11.6}$$

where $\mathbf{x} = [\alpha_{1,1}, \dots, \alpha_{1,n_1}, \dots, \alpha_{k,1}, \dots, \alpha_{k,n_k}]^{\mathrm{T}} \in \mathbf{R}^{n}$ is a coefficient vector.

As have mentioned, the samples from the same class are highly correlated and hence can be linearly represented by each other. While those from different classes are independently distributed and therefore cannot be linearly represented by each other. Ideally, the training samples from the same class as the test sample have nonzero coefficients in the linear combination, whereas those from the different class as the test sample have zero coefficients. For example, if a test sample is from the i-th class, the coefficient vector of the training samples should be:

$$\mathbf{x} = [0, \dots, 0, \alpha_{i,1}, \dots, \alpha_{i,n_i}, 0, \dots, 0]. \tag{11.7}$$

In general, the behavior of a linear system is determined by the relationship between the columns of **A** (the number of equations) and the rows of **A** (the number of unknowns). When the system has fewer equations than unknowns, say $m < n$ in dictionary $\mathbf{A} \in \mathbf{R}^{m \times n}$, it may have an infinite number of solutions. As a result, in all solutions of $\mathbf{y} = \mathbf{A}\mathbf{x}$, it is possible to arrive at the best solution ($\mathbf{x}_1$), which is infinitely close to the ideal solution showed in Eq. 11.7. The sparsest solution of $\mathbf{y} = \mathbf{A}\mathbf{x}$ is defined as the following optimization problem:

$$\hat{\mathbf{x}}_1 = \arg\min(\|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2 + \lambda \|\mathbf{x}\|_1), \tag{11.8}$$

where the columns of **A** are the training samples, **y** is the test sample, and λ is the regularization parameter to control the trade-off between the least squares error of representation and the sparsity of the coefficients. The solution can be obtained by using the MATLAB package provided by Koh et al. (2007).

As mentioned earlier, only when $m < n$ in dictionary **A** is it possible to discuss the optimal problem. In most cases, however, the number of features in breath samples is larger than the number of training samples; that is, m is always bigger than n. Due to that fact, feature extraction for the purpose of data dimensionality reduction is required. Additionally, feature extraction has the benefit of reducing computational cost when searching for the optimal solution. In the experiments, two feature extraction methods were employed. The first one was to extract geometry features from the response curve directly; the second one was to obtain the principal components by PCA. Both of the two methods will be described in next section. The features extracted from one sample form a vector. A test vector stands for the set of features extracted from a test sample, and a training vector contains the features extracted from a training sample.

11.2.3 Sampling Errors

Practical application requires considering sampling errors since odor signals are quite susceptible to environmental contamination. The correlation between the samples from the same class may be weakened by sampling errors. To solve this problem, we add an error vector **e** to the test vector **y**, and the real test vector becomes:

$$\mathbf{y}_0 = \mathbf{y} + \mathbf{e}. \tag{11.9}$$

Ideally, all entries of **e** should be zeros. However, because of sampling errors, the nonzero entry of **e** indicates that the response is miss-sampled or corrupted in that position by accident. The linear representation is hence improved as follows:

$$\mathbf{y} = [\mathbf{A}, \mathbf{I}] \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix} = \mathbf{B}\mathbf{w}, \tag{11.10}$$

where $\mathbf{B} = [\mathbf{A}, \mathbf{I}] \in \mathbf{R}^{m\times(m+n)}$ and $\mathbf{w} = [\mathbf{x}; \mathbf{e}]$ (';' means the two columns are stacked). And as such, it becomes an l^1-minimization problem:

$$\hat{\mathbf{w}}_1 = \arg\min(\|\mathbf{B}\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|_1). \tag{11.11}$$

The solution to the minimization problem can also be obtained by using the MATLAB package provided by Koh et al. (2007).

11.2.4 Voting Rules

There are two voting rules to make the classification decision. One measures the coefficients of representation and another measures the residue between the original signal and reconstructed signal.

11.2.4.1 Coefficients Based Voting Rule

The entries of the optimal solution $\hat{\mathbf{x}}_1$ are the representation coefficients of training vectors from all classes. Figure 11.2 is an example to show the coefficients. There are two classes A and B. We represent a test vector by using a set of training vectors from Class A and B, respectively. The blue solid line stands for the coefficients of training vectors from Class A, and the red dash-dot line stands for the coefficients of training vectors from Class B. It is observed that the training vectors from Class A hold higher coefficients than those from Class B. The test vector can be assigned to the class whose coefficients are much higher.

In the experiment, the following vote rule is used: assuming there are k classes, the i-th class includes n_i training vectors. When representing an unknown test vector, the training vectors from the i-th class obtain such coefficients: $C_i = [\hat{\alpha}_{i,1}, \ldots, \hat{\alpha}_{i,n_i}]$, where $i = 1, ..., k$, the classification result is given by

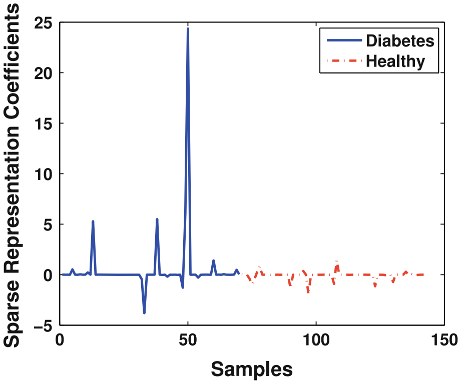


Fig. 11.2 Example of sparse representation coefficients

$$c = \arg\max_{i=1,\ldots,k} \|C_i\|^2, \tag{11.12}$$

where c is the label of the predicted class. The test vector is then predicted using Eq. 11.12.

11.2.4.2 Residue Based Voting Rule

Another voting rule for signal prediction is by computing the residue between the original signal and signal reconstructed by the atoms in the dictionary. To make the explanation clearer, we introduce a new vector $\delta_i(\hat{\mathbf{x}}_1)$, whose nonzero entries are the entries in $\hat{\mathbf{x}}_1$ that are associated with the i-th class (Wright et al. 2008). For example, assume $\hat{\mathbf{x}}_1 = [\hat{\alpha}_{1,1}, \ldots, \hat{\alpha}_{1,n_1}, \ldots, \hat{\alpha}_{i,1}, \ldots, \hat{\alpha}_{i,n_i}, \ldots, \hat{\alpha}_{k,1}, \ldots, \hat{\alpha}_{k,n_k}]^{\mathrm{T}}$, where $\hat{\alpha}_{i,1}, \ldots, \hat{\alpha}_{i,n_i}$ are coefficients of vectors from class i, respectively, and $\delta_i(\hat{\mathbf{x}}_1) = [0, \ldots, 0, \ldots, \hat{\alpha}_{i,1}, \ldots, \hat{\alpha}_{i,n_i}, \ldots, 0, \ldots, 0]^{\mathrm{T}}$. The test vector $\mathbf{y}$ can be represented by the training vectors from class i:

$$\tilde{\mathbf{y}}^{(i)} = \mathbf{A}\delta_i(\hat{\mathbf{x}}_1). \tag{11.13}$$

The residue between $\mathbf{y}$ and $\tilde{\mathbf{y}}^{(i)}$ is:

$$r_i(\mathbf{y}) = \left\|\mathbf{y} - \mathbf{A}\delta_i(\hat{\mathbf{x}}_1)\right\|_2, \tag{11.14}$$

which indicates how well the training vectors represent $\mathbf{y}$. The smaller the value of $r_i(\mathbf{y})$, the more likely $\mathbf{y}$ belongs to class i. So the test vector $\mathbf{y}$ can be assigned to the object class whose $\tilde{\mathbf{y}}$ has the minimal residue with $\mathbf{y}$.

11.2.5 Identification Steps

The identification procedure can be summarized as follows. The voting rule is based on the residue.

1. Divide all samples into a training set and a test set;
2. Select one sample from the test set as a test sample;
3. Extract the features from each training and test sample as training vectors and test vector, respectively;
4. The training vectors form $\mathbf{A}$ and the test vector is $\mathbf{y}$;
5. Normalize the column of $\mathbf{A}$ to have unit l^2-norm;
6. $\mathbf{B} = [\mathbf{A}, \mathbf{I}]$;
7. Solve $\hat{\mathbf{w}}_1 = \arg\min(\|\mathbf{Bw} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|_1)$ and obtain $\hat{\mathbf{w}}_1$, where $\hat{\mathbf{w}}_1 = [\hat{\mathbf{x}}_1; \hat{\mathbf{e}}_1]$;
8. Reconstruct the test vector $\mathbf{y}$ by $\tilde{\mathbf{y}}^{(i)} = \mathbf{A}\delta_i(\hat{\mathbf{x}}_1) + \hat{\mathbf{e}}_1$, $i = 1, 2, \ldots, k$, $\delta_i(\hat{\mathbf{x}}_1)$ stands for the coefficient vector associated with class i;

9. Compute residue $\left\|\mathbf{y}-\tilde{\mathbf{y}}^{(i)}\right\|_2$ of each class, respectively;
10. Assign test vector **y** to the class with the least residue;
11. Iterate Step 2–10 until all test vectors are identified;
12. Compute the total identification accuracy.

11.3 Overall Procedure

This section introduces the disease identification procedure, including the feature extraction and classification.

11.3.1 Disease Identification

The identification procedure is presented in Fig. 11.3, which has been introduced in detail in Guo et al. (2010b).

In breath sample acquisition phase, breath gas is injected into the device containing twelve chemical sensors. The signal measurement module measures the responses of sensors and converts them into analog electrical signals. After filtering, amplifying, and digitizing, the signals are transmitted to a computer for future analysis.

In the phase of signal processing, original signals are processed by de-noising, baseline manipulation, and normalization and then stored in the database as standard samples. When an unknown sample is delivered, it undergoes the same processing. The characteristic features are extracted from both the training samples and the test sample. Finally, the classifier decides which class the unknown sample belongs to.

Among these phases, feature extraction and classifier selection play critical roles in disease identification, which will be introduced in the following sections.

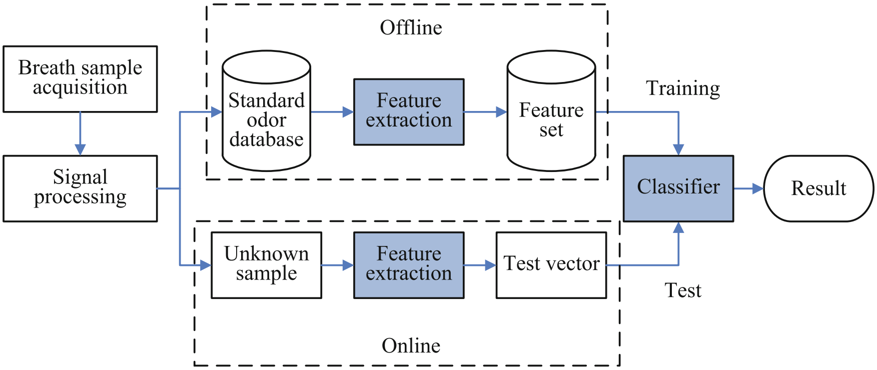


Fig. 11.3 Working flow of disease identification

11.3.2 Feature Extraction

Feature extraction reduces the measurement cost and increases the classification accuracy. A feature set that describes the original data perfectly can make the classifier work efficiently. A range of feature extraction methods has been applied in odor signals. Instances of those include extracting parameters of fitting model for odor signals (Carmel et al. 2003), extracting geometry features from signal curves (Paulsson et al. 2000), extracting global features by PCA or Linear Discriminant Analysis (LDA) (Wang et al. 2009), and extracting frequency features by Discrete Wavelet Transform (DWT) (Distante et al. 2002). The purpose of this chapter is to demonstrate that SRC is the best for odor signal classification, irrespective of the features selected. As a result, two different sets of features, i.e., geometry features from transient curves and principal components obtained by PCA, are employed in the experiments to illustrate the robustness of the proposed method.

11.3.2.1 Geometry Features

Geometry features are the parameters such as rise times, maximum response, slope, and curve integrals extracted directly from the transient response curve. They have been utilized in several odor signal identifications (Paulsson et al. 2000; Distante et al. 2002; Martinelli et al. 2003; Zhang et al. 2008b; Mirmohseni et al. 2007; Pardo and Sberveglieri 2007). These features consume the least time to extract and are suitable for real-time and on-site data analysis.

It has been observed that signal amplitudes, such as maximal amplitude and amplitude at given time, are related to the concentration of analytes; signal curvature, curve integrals, and response time, are associated with the type of analytes. These features are extracted from the transient response curves, as Fig. 11.4 shows. The labeled features are explained in Table 11.1. There are 12 features extracted from one sensor response curve. The feature vector of sensor i is:

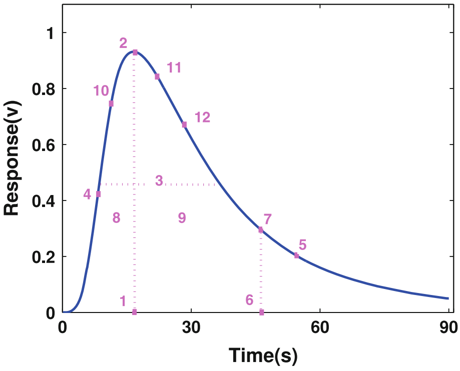


Fig. 11.4 Geometry features extracted from one response curve

Table 11.1 Composition of the subject database

Feature label	Detailed explanation
1	Rise times
2	Maximum response
3	Length of median
4	Response at the half of rise times
5	Response at the half of drop times
6	Time when curvature is maximal in drop period
7	Maximal curvature in drop period
8	Curve integrals in rise period
9	Curve integrals in drop period
10	Response at 5 s before drop time
11	Response at 5 s after drop time
12	Response at 10 s after drop time

$$v_i = [f_{i,1},f_{i,2},\cdots,f_{i,12}]. \tag{11.15}$$

A breath sample is composed of the responses of l chemical sensors ($l = 12$), the feature matrix of this sample is:

$$\mathbf{V} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_l \end{bmatrix} = \begin{bmatrix} f_{1,1},f_{1,2},\cdots,f_{1,p} \\ f_{2,1},f_{2,2},\cdots,f_{2,p} \\ \vdots \\ f_{l,1},\, f_{l,2},\cdots,f_{l,p} \end{bmatrix}, \tag{11.16}$$

where $p = 12$ is the number of features.

Feature normalization is required as the values of these features have huge diversities in magnitude. A proposed method is to normalize one kind of feature from the l sensors together. A new vector is defined including the j-th feature from all sensors, $c_j = [f_{1,j},f_{2,j},\cdots,f_{l,j}], j = 1,\ldots,p$. A new matrix $\tilde{\mathbf{V}}$ after normalization is:

$$\tilde{V} = \begin{bmatrix} f_{1,1}/\left\|c_1\right\|_2,f_{1,2}/\left\|c_2\right\|_2,\cdots,f_{1,p}/\left\|c_p\right\|_2 \\ f_{2,1}/\left\|c_1\right\|_2,f_{2,2}/\left\|c_2\right\|_2,\cdots,f_{2,p}/\left\|c_p\right\|_2 \\ \vdots \\ f_{l,1}/\left\|c_1\right\|_2,\, f_{l,2}/\left\|c_2\right\|_2,\cdots,f_{l,p}/\left\|c_p\right\|_2 \end{bmatrix}. \tag{11.17}$$

Consequently, the same features in all sensors range between 0 and 1, but their relative distances remain unchanged.

11.3.2.2 Principal Component Analysis

The Principal Component Analysis is also utilized to extract the principal components of each sample. The feature extraction by PCA has been described in Guo et al. (2010b). s ($s = 144$) principle components were selected to guarantee they count for more than 99.9% variability. One of reasons for selecting 144 eigenvectors is to keep both the geometry features and the principal components as the same dimensionality. It is possible to compare the two feature extraction methods together.

11.3.3 Classifier

Finally, an appropriate classifier is applied to assign the unknown sample to the tagged class. At present, odor signals, captured by chemical sensors or e-noses, can be identified via many methods. For example, Wang et al. applied Relevance Vector Machines (RVM) to classify coffee sampled by a commercial e-nose (Wang et al. 2009). Brudzewski et al. used SVM network to classify milk measured by a chemical sensor array (Brudzewski et al. 2004). Lozano et al. employed ANN to identify wine aromas captured by an e-nose (Lozano et al. 2006). In our experiments, SRC is proposed to classify human breath. Additionally, SVM and KNN are also used to provide comparative results, which are described briefly in the following sections.

11.3.3.1 SVM

The basic idea of SVM is to construct a hyperplane in classifying the data such that the gap between the classified data set is maximized (Cristianini and Shawe-Taylor 2000). For any training data $\{\mathbf{x}_i, y_i\}, i = 1, \ldots, n$, $\mathbf{x}_i \in R^d$ is the training sample and y_i is either 1 or -1, indicating the class to which the sample $\mathbf{x}_i$ belongs to. The optimization problem in SVM is to maximize:

$$\begin{aligned} L(\alpha) &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), \\ s.t. \quad & \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0. \end{aligned} \tag{11.18}$$

$K(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel of SVM, which can be defined by user. The solution is expressed in terms of linear combination of the training vectors:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i. \tag{11.19}$$

In the experiment, the Gaussian kernel $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-v\|\mathbf{x}_i - \mathbf{x}_j\|^2}$ is used to train the classifier with a predefined v. The learnt classifier is used for unknown sample classification.

11.3.3.2 KNN

KNN classifies an unlabeled test sample by finding the K nearest neighbors in the training set using Euclidean distance and assigning the label of that class represented by a majority among the K neighbors (Gutierrez-Osuna 2002). The vote rule is: assuming there are m classes and one sample has $K_1, K_2, ..., K_m$ nearest neighbors for the m classes, where $\sum_{i=1}^{m} K_i = K$, the classification result is given by

$$c = \arg\max_{i=1,...,m} \{\frac{K_i}{K}\}, \tag{11.20}$$

where c is the label of the predicted class. The training vectors are classified in advance into m classes, labeled as either healthy or diseased. The test vector is then predicted using Eq. 11.20.

11.4 Experiments and Results

This section presents two experiments to test the performance of the proposed SRC. The first one is to identify diabetes from a database including healthy samples and diabetes samples. The second one aims to classify the diabetics' breath samples into four levels by measuring the concentration of acetone in breath. The classification results are judged by the subjects' blood glucose levels provided by simultaneous blood test.

11.4.1 Diabetes Identification

It has been claimed that the abnormal concentration of acetone in exhaled air indicates the existence of diabetes (Deng et al. 2004). Chemical sensors can respond to the diabetes samples distinctively due to the abundant acetone in patients' breath. The typical healthy breath sample and diabetics' breath sample are shown in Fig. 11.5. The difference between the two kinds of samples have been discussed in Guo et al. (2010b).

Table 11.2 details the composition of the database, which contains 104 diabetes samples and 108 healthy samples. In each class, two thirds of the samples is selected randomly as training samples and the rest as test samples. Totally, the training set

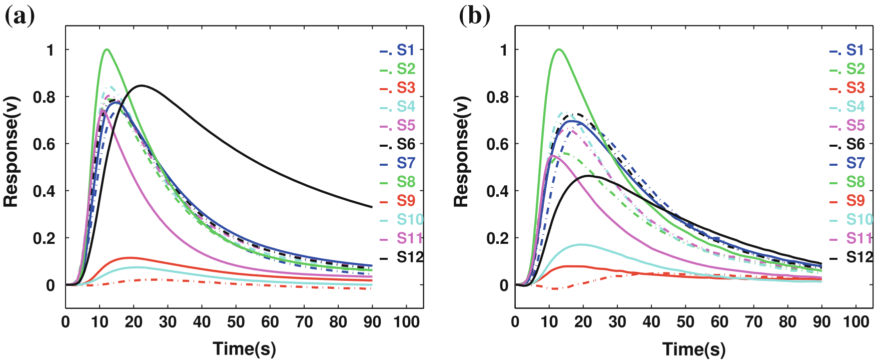


Fig. 11.5 Typical responses from one healthy sample and one diabetes sample: **a** healthy sample, **b** diabetes sample

Table 11.2 Composition of the subject database

Type of subjects	Number	Male/Female	Age
Healthy subjects	108	58/50	23–60
Subjects with diabetes	104	62/42	25–77

included 142 samples, 70 labeled as diabetes and 72 labeled as healthy. The test set was composed of 70 samples, 34 labeled as diabetes, and 36 labeled as healthy.

Each sample includes 12 responses of sensors. Each response is a 810 dimensional discrete time series. According to Eq. 11.3, one sample was expressed as a $12 \times 810 = 9720$ dimensional vector. Geometry features and principle components were extracted at the same time. Finally, each sample formed a 144-dimensional feature vector by using either method.

All training vectors formed the dictionary **A**. One test vector was selected randomly from the test set, which was represented by each column of **A**. The sparse representation coefficients of the training vectors were obtained by Eq. 11.11, as shown in Fig. 11.6. The first 70 training vectors (blue solid line) were from the diabetes class and the rest of the training vectors (red dash-dot line) were from the healthy class. When the test vector was from the diabetes class, the training vectors from diabetes samples held far larger sparse coefficients than those from healthy samples (Fig. 11.6a), whereas when the test vector was from the healthy class, the healthy training vectors possessed much larger sparse coefficients than those from diseased class (Fig. 11.6b).

The residue between the test vector and the vector reconstructed by the training vectors from each class presented the same result as what have been shown by coefficients. Figure 11.7 provided the residues of two cases: test sample from the diabetes class (Fig. 11.7a) and the test sample from the healthy class (Fig. 11.7b). In Fig. 11.7a, the blue solid line stood for the residue between the test vector and the vector reconstructed by only the training vectors from the diabetes class, while

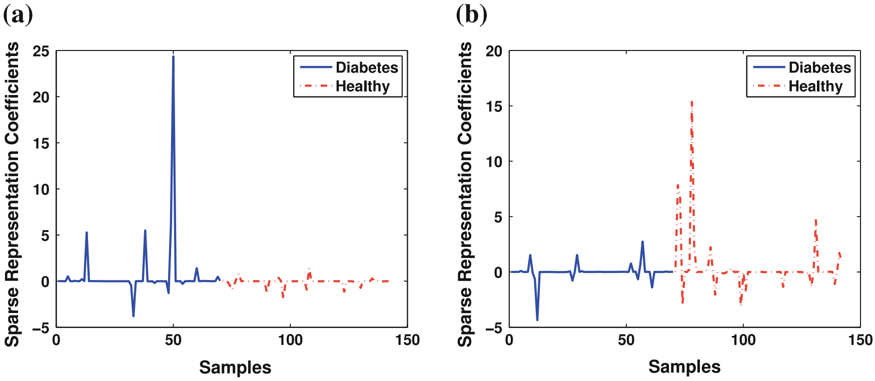


Fig. 11.6 Sparse representation coefficients of the training vectors from the classes: diabetes samples (*blue solid line*) and healthy samples (*red dash-dot line*). **a** When representing a diabetes test vector, **b** when representing a healthy test vector. The horizontal axis stands for the number of training vectors and the vertical axis shows the vectors' sparse representation coefficients

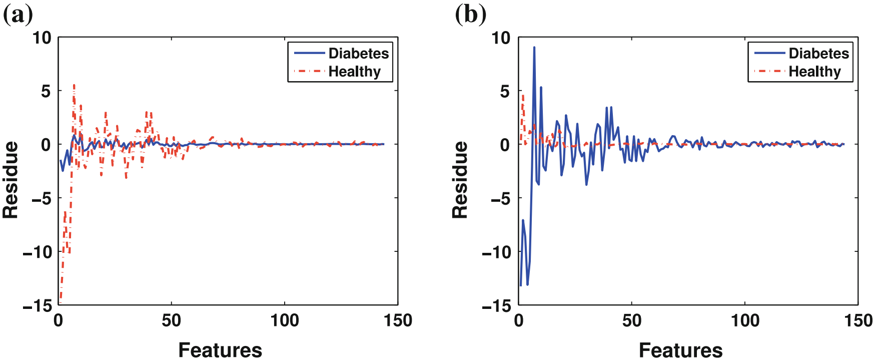


Fig. 11.7 Residues between test vector and the vector reconstructed by training vectors. The test vector is from **a** diabetes class, **b** healthy class. The horizontal axis stands for the number of features and the vertical axis shows the residues

the red dash-dot line represented the residue between the test vector and the vector reconstructed by only the training vectors from the healthy class. Comparing the two lines shows that when the test vector and the training vectors were from the same class, the residue was much smaller than when the test vector and the training vectors were from different classes. The same held good for the second case that the test sample was from the healthy class, as shown in Fig. 11.7b. The residue between the test vector and the vector reconstructed by only the healthy training vectors was much smaller than the residue between the test vector and the vector reconstructed by only the training vectors of diabetes.

The two cases indicated that the training vectors could successfully represent the test vectors in the same class but could hardly represent the test vectors in the

Table 11.3 Recognition results defined by sensitivity and specificity

No.	Methods	Sensitivity(%)	Specificity(%)
1	Geometry features + KNN	78.07	77.61
2	Geometry features + SRC	**88.55**	**87.28**
3	Geometry features + SVM	87.62	86.81
4	PCA + KNN	81.63	79.85
5	PCA + SRC	**92.10**	**91.24**
6	PCA + SVM	91.81	89.17

different class. By computing the residue, we could possibly determine which class the test sample belongs to: the class possessing the least residue was the one the test vector belonged.

The comparative experimental results were provided by SVM and KNN. In all of these experiments, the classification procedure was run 30 times and the average classification rate over all runs was computed.

Table 11.3 provides the identification results defined by sensitivity and specificity. There were six identification results listed in the table. It is obvious that SRC provided the best results, while KNN gave the worse results whatever features used. It means that the classification accuracy can be increased by selecting an appropriate classifier.

11.4.2 Blood Glucose Measurement

Diabetics need to check their blood glucose levels several times each day by drawing blood samples. This process is invasive and unsafe and requires considerable skill. Hence, not everyone is suited to this approach. An alternative is to collect a sample of exhaled breath with the breath analysis system (Melker et al. 2006). This technology is likely to increase the acceptance of frequent blood glucose monitoring and reduce the danger posed by drawing blood samples.

There is a linear correlation between the mean group acetone and the mean group blood glucose level of diabetics (Wang et al. 2010). Consequently, attempt was made in this chapter to find out if the diabetics' breath samples captured by the device can be grouped based on their blood glucose levels. The breath samples of diabetes and their simultaneous blood glucose levels were collected at the same time. Table 11.4 lists the subjects' blood glucose levels which is defined in (Wang et al. 2010) and their corresponding number.

Figure 11.8 shows the responses of the twelve different sensors (S1–S12) to the samples of four diabetics over the 90 s sampling period. Figure 11.8a–d are the responses from four diabetics' breath samples, their blood glucose levels increased gradually from Fig. 11.8a to d. From the figures we can see the difference between

Table 11.4 The blood glucose levels and the corresponding number

Level stage	Name	Blood glucose level (mg/dL)	Number
Level 1	Low level	81–100	18
Level 2	Borderline	101–150	49
Level 3	High level	151–200	20
Level 4	Very high level	201–421	17
Total		81–421	104

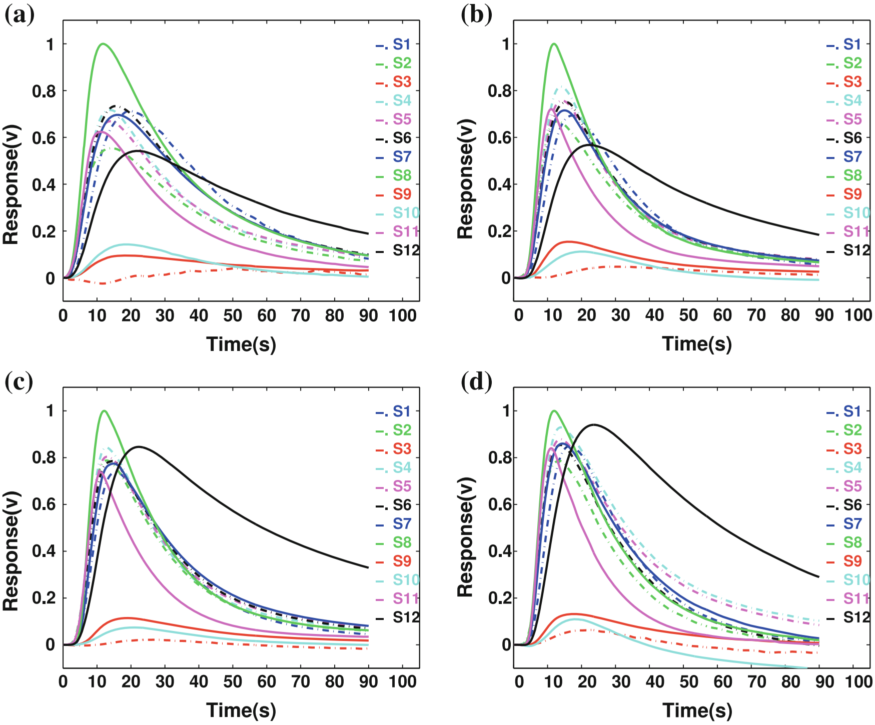


Fig. 11.8 Responses from four diabetics with different simultaneous blood glucose levels: **a** *Level* 1, **b** *Level* 2, **c** *Level* 3, and **d** *Level* 4

the four samples. The amplitudes of the sensors' responses increase with the blood glucose levels. As a result, it is possible to measure the blood glucose levels of diabetics by classifying their breath gas samples.

The feature extraction method used in this experiment is PCA. Figure 11.9 presents the sparse coefficients of the training vectors. Figure 11.9a–d indicate a test vector was from Level 1 to Level 4, respectively when it was represented by all training vectors. When representing one vector from Level 1, the training vectors in the

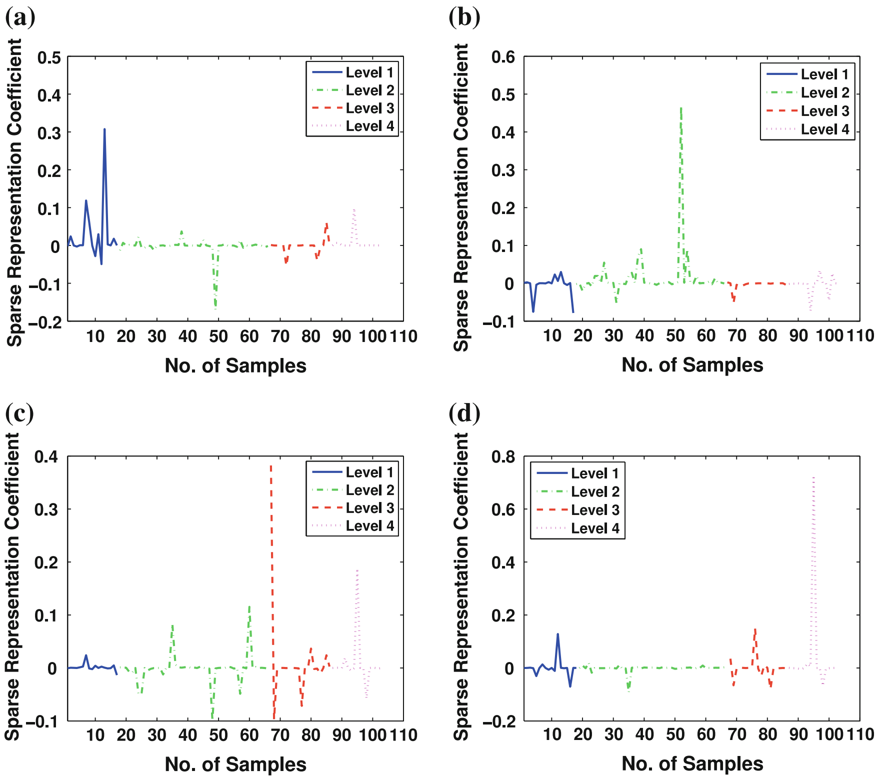


Fig. 11.9 Sparse representation coefficients of the training vectors from the four classes: *Level* 1 (*blue solid line*), *Level* 2 (*green dash-dot line*), *Level* 3 (*red dash line*), and *Level* 4 (*magenta dot line*). **a** The test vector is from *Level* 1 class, **b** the test vector is from *Level* 2 class, **c** the test vector is from *Level* 3 class, and **d** the test vector is from *Level* 4 class. The horizontal axis stands for the number of training vectors and the vertical axis shows the vectors' sparse representation coefficients

same Level obtained larger sparse coefficients than others (Fig. 11.9a). As with Level 1 vectors, when representing one vector from Level 2, the training vectors in Level 2 obtained larger sparse coefficients than other training vectors (Fig. 11.9b). The same held good for Level 3 and Level 4 vectors. When representing one vector of the same class, the coefficients of the training vector are larger.

Figure 11.10 shows the residues between the test vector and the vector reconstructed by the training vectors from the four classes. Figure 11.10a–d show the test vector was from Level 1 to Level 4, respectively. In each figure, the blue solid line, green dash-dot line, red dash line, and magenta dot line indicates the training vectors were from Level 1 to Level 4, respectively. When the test vector and the training vectors were from the same classes, the smallest residue was achieved. The test vector was therefore assigned to the class with the least residue.

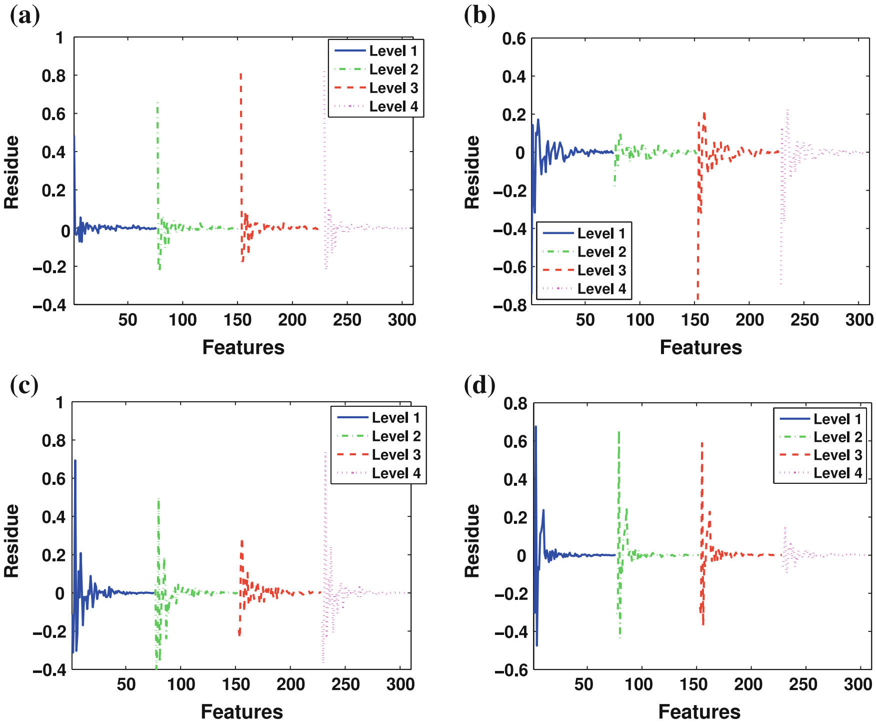


Fig. 11.10 Residues between test vector and the vector reconstructed by training vectors. The test vector is from **a** *Level* 1, **b** *Level* 2, **c** *Level* 3, and **d** *Level* 4. The horizontal axis stands for the number of features and the vertical axis shows the residues

Since the first experiment showed that the results provided by the PCA + SRC were better than geometry features + SRC, this experiment only employed the PCA + SRC to evaluate the classification. The feature dimension was 76. We also provided the result given by the PCA + KNN for comparison and use the leave-one-out cross-validation to obtain the classification results. In this process, firstly selected one sample as the test sample and the rest as the training samples to compute whether the test sample was classified correctly. The process was reiterated until every sample was tested once. The results of SRC were out of parenthesis, and the results of KNN were in parenthesis, as Table 11.5 shows. The results of SRC were given out of parenthesis and the results of KNN were given in parenthesis. From the table, we can see that the results provided by SRC were much better than KNN.

Table 11.5 The classification results calculated by sparse representation

	Number	Test outcome of SRC (KNN)				Accuracy(%)
		Level 1	Level 2	Level 3	Level 4	
Level 1	18	**13 (8)**	3 (10)	2 (0)	0 (0)	72.22 (44.44)
Level 2	49	2 (3)	**41 (27)**	5 (13)	1 (6)	83.67 (55.1)
Level 3	20	0 (0)	4 (8)	**14 (9)**	2 (3)	70 (45)
Level 4	17	0 (0)	2 (2)	2 (3)	**13 (12)**	76.47 (70.59)

11.5 Summary

This chapter has proposed a novel SRC method for the breath sample identification. Two experiments were conducted to measure the proposed method. The experimental results showed that the proposed method outperforms SVM and KNN, irrespective of the features selected. The results achieved here are promising if not entirely satisfactory. Clearly, more samples are required to test the method. Improvement of the feature extraction and classification method is also necessary. From the classifier optimization point of view, we can improve the current result by optimizing the l^1-minimization problem (Eq. 11.8) and learning proper dictionary (matrix A in Eq. 11.8), which will be part of our future work.

References

Amann A, Schmid A, Scholl-Burgi S, Telser S, Hinterhuber H (2005) Breath analysis for medical diagnosis and therapeutic monitoring. Spectrosc Eur 17(3):18–20

Brudzewski K, Osowski S, Markiewicz T (2004) Classification of milk by means of an electronic nose and SVM neural network. Sens Actuators B: Chem 98(2–3):291–298

Carmel L, Levy S, Lancet D, Harel D (2003) A feature extraction method for chemical sensors in electronic noses. Sens Actuators B: Chem 93(1–3):67–76

Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press

DAmico A, Di Natale C, Paolesse R, Macagnano A, Martinelli E, Pennazza G, Santonico M, Bernabei M, Roscioni C, Galluccio G, et al. (2007) Olfactory systems for medical applications. Sens Actuators: B Chem 130(1):458–465

D'Amico A, Pennazza G, Santonico M, Martinelli E, Roscioni C, Galluccio G, Paolesse R, Di Natale C (2009) An investigation on electronic nose diagnosis of lung cancer. Lung Cancer 68:170–176

Davies S, Spanel P, Smith D (1997) Quantitative analysis of ammonia on the breath of patients in end-stage renal failure. Kidney Intern 52(1):223–228

Deng C, Zhang J, Yu X, Zhang W, Zhang X (2004) Determination of acetone in human breath by gas chromatography-mass spectrometry and solid-phase microextraction with on-fiber derivatization. J Chromatogr B 810(2):269–275

Deykin A, Massaro A, Drazen J, Israel E (2002) Exhaled nitric oxide as a diagnostic test for asthma: online versus offline techniques and effect of flow rate. Am J Respir Crit Care Med 165(12):1597–1601

Di Francesco F, Fuoco R, Trivella M, Ceccarini A (2005) Breath analysis: trends in techniques and clinical applications. Microchem J 79(1–2):405–410

Di Natale C, Macagnano A, Martinelli E, Paolesse R, D'Arcangelo G, Roscioni C, Finazzi-Agrò A, D'Amico A (2003) Lung cancer identification by the analysis of breath by means of an array of non-selective gas sensors. Biosens Bioelectr 18(10):1209–1218

Distante C, Leo M, Siciliano P, Persaud K (2002) On the study of feature extraction methods for an electronic nose. Sens Actuators B: Chem 87(2):274–288

Dragonieri S, Schot R, Mertens B, Le Cessie S, Gauw S, Spanevello A, Resta O, Willard N, Vink T, Rabe K et al. (2007) An electronic nose in the discrimination of patients with asthma and controls. J Allergy Clin Immunol 120(4):856–862

Dragonieri S, Annema J, Schot R, van der Schee M, Spanevello A, Carratú P, Resta O, Rabe K, Sterk P (2009) An electronic nose in the discrimination of patients with non-small cell lung cancer and COPD. Lung Cancer 64(2):166–170

Dweik R, Amann A (2008) Exhaled breath analysis: the new frontier in medical testing. J Breath Res 2(030):301

Fleischer M, Simon E, Rumpel E, Ulmer H, Harbeck M, Wandel M, Fietzek C, Weimar U, Meixner H (2002) Detection of volatile compounds correlated to human diseases through breath analysis with chemical sensors. Sens Actuators B: Chem 83(1–3):245–249

Guo D, Zhang D, Li N (2010a) Monitor blood glucose levels via breath analysis system and sparse representation approach. In: Sensors, 2010 IEEE. IEEE, pp 1238–1241

Guo D, Zhang D, Li N, Zhang L, Yang J (2010b) A novel breath analysis system based on electronic olfaction. IEEE Trans Biomed Eng 57(11):2753–2763

Guo D, Zhang D, Li N, Zhang L, Yang J (2010c) Diabetes identification and classification by means of a breath analysis system. Int Conf Med Biom 52–63

Gutierrez-Osuna R (2002) Pattern analysis for machine olfaction: a review. IEEE Sens J 2(3):189–202

Huang K, Aviyente S (2007) Sparse representation for signal classification. Adv Neural Inf Process Syst 19:609–616

Jain A, Mao R (2000) Statistical pattern recognition: a review. IEEE Trans Pattern Anal Mach Intell 22(1):4–37

Koh K, Kim S, Boyd S (2007) l1 ls: a matlab solver for large-scale l1-regularized least squares problems

Liess M (2002) Electric-field-induced migration of chemisorbed gas molecules on a sensitive film-a new chemical sensor. Thin Solid Film 410(1–2):183–187

Lozano J, Santos J, Aleixandre M, Sayago I, Gutierrez J, Horrillo M (2006) Identification of typical wine aromas by means of an electronic nose. IEEE Sens J 6(1):173–178

Martinelli E, Falconi C, D'Amico A, Di Natale C (2003) Feature extraction of chemical sensors in phase space. Sens Actuators B: Chem 95(1–3):132–139

McGrath L, Patrick R, Mallon P, Dowey L, Silke B, Norwood W, Elborn S (2000) Breath isoprene during acute respiratory exacerbation in cystic fibrosis. Eur Respir J 16(6):1065–1069

Melker R, Bjoraker D, Lampotang S (2006) System and method for monitoring health using exhaled breath. US Patent App. 11/512,856

Miekisch W, Schubert J, Noeldge-Schomburg G (2004) Diagnostic potential of breath analysis-focus on volatile organic compounds. Clinica Chimica Acta 347(1–2):25–39

Mirmohseni A, Abdollahi H, Rostamizadeh K (2007) Analysis of transient response of single quartz crystal nanobalance for determination of volatile organic compounds. Sens Actuators B: Chem 121(2):365–371

Pardo M, Sberveglieri G (2005) Classification of electronic nose data with support vector machines. Sens Actuators B: Chem 107(2):730–737

Pardo M, Sberveglieri G (2007) Comparing the performance of different features in sensor arrays. Sens Actuators B: Chem 123(1):437–443

Pardo M, Kwong L, Sberveglieri G, Brubaker K, Schneider J, Penrose W, Stetter J (2005) Data analysis for a hybrid sensor array. Sens Actuators B: Chem 106(1):136–143

Paulsson N, Larsson E, Winquist F (2000) Extraction and selection of parameters for evaluation of breath alcohol measurement with an electronic nose. Sens Actuators A: Phys 84(3):187–197

Phillips M, Sabas M, Greenberg J (1993) Increased pentane and carbon disulfide in the breath of patients with schizophrenia. J Clin Pathol 46(9):861–864

Phillips M, Altorki N, Austin J, Cameron R, Cataneo R, Greenberg J, Kloss R, Maxfield R, Munawar M, Pass H et al. (2007) Prediction of lung cancer using volatile biomarkers in breath. Cancer Biomark 3(2):95–109

Rock F, Barsan N, Weimar U (2008) Electronic nose: Current status and future trends. Chem Rev 108(2):705–725

Schubert J, Miekisch W, Geiger K, Nöldge-Schomburg G (2004) Breath analysis in critically ill patients: potential and limitations. Expert Rev Mol Diagn 4(5):619–629

Shih C, Lin Y, Lee K, Chien P, Drake P (2010) Real time electronic nose based pathogen detection for respiratory intensive care patients. Chem Sens Actuators B, pp 153–157

Van Berkel J, Dallinga J, Möller G, Godschalk R, Moonen E, Wouters E, Van Schooten F (2008) Development of accurate classification method based on the analysis of volatile organic compounds from human exhaled air. J Chromatogr B 861(1):101–107

Wang C, Mbi A, Shepherd M (2010) A study on breath acetone in diabetic patients using a cavity ringdown breath analyzer: exploring correlations of breath acetone with blood glucose and glycohemoglobin a1c. IEEE Sens J 10(1):54–63

Wang X, Ye M, Duanmu C (2009) Classification of data from electronic nose using relevance vector machines. Sens Actuators B: Chem 140(1):143–148

Wright J, Yang A, Ganesh A, Sastry S, Ma Y (2008) Robust face recognition via sparse representation. IEEE Trans Pattern Anal Mach Intell 210–227

Yu J, Byun H, So M, Huh J (2005) Analysis of diabetic patient's breath with conducting polymer sensor array. Sens Actuators B: Chem 108(1–2):305–308

Zhang Q, Zhang S, Xie C, Fan C, Bai Z (2008a) Sensory analysis' of Chinese vinegars using an electronic nose. Sens Actuators B: Chem 128(2):586–593

Zhang S, Xie C, Hu M, Li H, Bai Z, Zeng D (2008b) An entire feature extraction method of metal oxide gas sensors. Sens Actuators B: Chem 132(1):81–89

Chapter 12
Monitor Blood Glucose Levels via Sparse Representation Approach

Abstract It has been reported that the abnormal concentration of acetone in exhaled air is an indicator of diabetes and the concentration rises progressively with the blood glucose level of patients. Therefore, the acetone in human breath can be used to monitor the development of diabetes. In this chapter, we introduce a breath analysis system to measure acetone in human breath, and therefore to evaluate the blood glucose levels of diabetics. The system structure, breath collection method, and signal preprocessing method are introduced. To enhance the system performance, we use a novel classification approach, i.e., Sparse Representation based Classification (SRC), to classify diabetics' breath samples into different blood glucose levels. Experimental results show that coupling with SRC, the system is able to classify these levels with satisfactory accuracy.

Keywords Breath analysis · Disease identification · Sparse representation · Diabetes · Blood glucose levels

12.1 Introduction

Changes in the concentration of components in human breath such as acetone, nitric oxide, and ammonia could suggest various diseases or at least changes in the metabolism (Amann et al. 2005). These molecules are therefore considered as biomarkers of the presence of diseases and clinical conditions. It is well known that the abnormal concentration of acetone in exhaled air is an indicator of diabetes (Deng et al. 2004). Additionally, the mean acetone concentration in exhaled air rises progressively with the blood glucose of the diabetics, especially when they are with high blood glucose levels (Tassopoulos et al. 1969). These patients should check their blood glucose levels several times each day by drawing their blood samples. The process is invasive and unsafe and requires considerable skill. Therefore, not everyone is suitable for this approach. An alternative is to collect a breath sample and measure the concentration of acetone (Melker et al. 2006). This technology will increase the acceptance of frequent blood glucose monitoring and reduce the danger caused during drawing blood samples.

D. Zhang et al., *Breath Analysis for Medical Applications*,
DOI 10.1007/978-981-10-4322-2_12

In this chapter, we propose a breath analysis system to monitor the levels of blood glucose. The system uses chemical sensors that are particularly sensitive to the biomarkers, triggering responses to the breath sample when the biomarkers are detected. The response signals serve to subsequent processing and then become the standard odor samples. In fact, this system is not only used in monitoring the blood glucose levels, but also used in disease detection, such as diabetes, renal diseases, and airway inflammation effectively (Guo et al. 2010a).

Comparing with detecting diseases, monitoring the blood glucose levels is much more difficult for the breath analysis system because the responses of different blood glucose levels are not very distinguishable. Additionally, the samples are high dimensional and the data set formed by these samples is with a small size. Hence, the traditional pattern recognition method such as KNN and RBF neural networks do not work well in our case (Gutierrez-Osuna 2002). In this chapter, we use a Sparse Representation based Classification (SRC) approach to classify the levels of blood glucose. This approach have shown good performance in face recognition (Wright et al. 2008). In this chapter, we use it in classifying the diabetes samples with different blood glucose levels. The experimental result show that coupling with SRC, the system has satisfying performance in this application.

The remainder of this chapter is organized as follows. Section 12.2 describes the breath analysis system. Section 12.3 explains the approach of SRC and its application in odor signal. Section 12.4 explains the experimental details and gives the result and discussion. Section 12.5 offers our summary.

12.2 System Description and Breath Signals Acquisition

The system, operates in three phases (Fig. 12.1), breath collection, sample process, and pattern recognition.

In the first phase, subject is requested to breath into a Tedlar gas sampling bag. The collected gas is then injected into a chamber containing a sensor array. The sensor array is composed of 12 metal oxide semiconductor gas sensors (from FIGARO Engineering Inc.) set in a stainless steel chamber. The resistances of the sensors

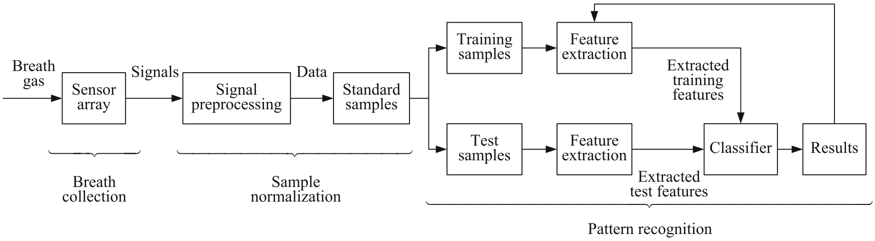


Fig. 12.1 The working flow defined in our system

changes when they are exposed to sampled gas and the output voltage changes correspondingly. Then, the signal measurement module measures the output voltage and converts it into analog electrical signal. The analog signal is subsequently conditioned by signal filtering and amplifying. Finally, the signal is sampled and transmitted through a USB interface to a computer for future analysis.

In the second phase, the original signals are preprocessed by denoising, baseline manipulation, and normalization. Denoising is to remove the noise from the original signals. Baseline manipulation is implemented for drift compensation, contrast enhancement, and scaling. Normalization is used to compensate for sample-to-sample variations caused by analyte concentration and pressure of oxygen (PO2). For denoising, we use a low-pass filter to remove the noise since it is just with high frequency. For baseline manipulation, we subtract the initial value of each sensor from the whole sensor response curves to guarantee all signals starts from the same baseline. For normalization, we firstly find the maximum sensor response in the whole sensor array and set it as 1 by multiplying a coefficient, other data in the whole sensor array are also multiplied by the same coefficient. Therefore, the responses of all sensors are set in [0, 1] but their relationship is unchanged. After these steps, we obtain the standard samples, which are stored in data base for future analysis.

In the third phase, these samples, from several classes, are divided into two sub-databases: training samples and test samples for blood glucose levels classification. Before these samples are sent to classifier, feature extraction from both training samples and test samples are required for reducing data dimension and computational cost. Then, the training samples are used to train the parameters of selected classifier. In the end, the classifier decides which class the test sample belongs to. In Sect. 12.3, we will introduce the proposed classification approach based on SRC in detail.

12.3 Sparse Representation Classification

The basic idea of sparse representation classification is to represent a test sample as the linear combination of a set of training samples. The coefficients of the linear combination are restricted to be sparse. Ideally, the coefficients are all zero except those associated with the same class as the test sample and by using these coefficients these training samples can represent the test sample accurately. Therefore, we can assign the test sample to the class that have smallest residual between the representation and the test sample. This approach has been reported by Wright et al., for face recognition (Wright et al. 2008). In this chapter, we introduce its application in odor signal classification.

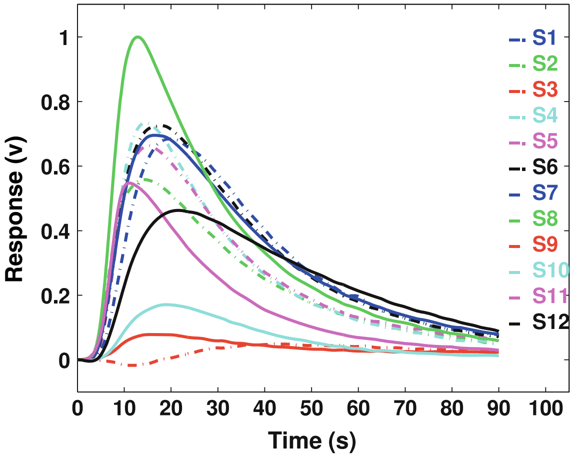


Fig. 12.2 The typical response of our system to a healthy person

12.3.1 Data Expression

Figure 12.2 is a typical response from a healthy breath sample, which includes l sensors' responses, where $l = 12$. The response of each sensor is $s = [t_1, t_2, \dots, t_d]^T$, where $d = 810$. Therefore, the sample can be expressed as a column vector $\mathbf{v} = [\mathrm{t}_{1,1}, \dots, \mathrm{t}_{\mathrm{d},1}, \mathrm{t}_{1,2}, \dots, \mathrm{t}_{\mathrm{d},2}, \dots, \mathrm{t}_{1,\mathrm{l}}, \dots, \mathrm{t}_{\mathrm{d},\mathrm{l}}]^{\mathrm{T}} \in \mathbf{R}^m$, where $m = d \times l$. Assume we have k classes, the i-th class includes n_i samples, $\sum_{i=1}^{k} n_i = n$. Express all training samples form the i-th object class as,

$$A_i = [\mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \dots, \mathbf{v}_{i,n_i}] \in \mathbf{R}^{m \times n_i}, \tag{12.1}$$

All of the samples from the k object classes is expressed as

$$A = [A_1, A_2, \dots, A_k] = [\mathbf{v}_{1,1}, \mathbf{v}_{1,2}, \dots, \mathbf{v}_{k,n_k}] \in \mathbf{R}^{m \times n}. \tag{12.2}$$

12.3.2 Sparse Representation of Odor Signals

One of the test sample $\mathbf{y}$ can be expressed as a linear combination of all of training samples from all object classes:

$$\mathbf{y} = \alpha_{1,1}\mathbf{v}_{1,1} + \alpha_{1,2}\mathbf{v}_{1,2} +, \dots, \alpha_{k,n_k}\mathbf{v}_{k,n_k} = A\mathbf{x} \in \mathbf{R}^m, \tag{12.3}$$

where $\mathbf{x} = [\alpha_{1,1}, \dots, \alpha_{1,n_1}, \dots, \alpha_{k,1}, \dots, \alpha_{k,n_k}]^T \in \mathbf{R}^n$ is a coefficient vector. The intra-class samples, which are highly correlated, can be linearly represented by the training samples; the inter-class samples, however, which are independently distributed, cannot be linearly represented by the training set, and therefore have zero coefficients in the linear model. Ideally, the best solution ($\mathbf{x}_1$) of $\mathbf{y} = \mathbf{Ax}$ should be sparse enough to satisfy: the entries of $\mathbf{x}_1$ should be zeros except those associated with the same class as the test sample. The sparse representation coefficients can be obtained by solving the following l^1-minimization problem:

$$\hat{\mathbf{x}}_1 = \arg\min(\|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2 + \lambda \|\mathbf{x}\|_1), \tag{12.4}$$

where the columns of $\mathbf{A}$ are the training samples, $\mathbf{y}$ is the test sample, and λ is the regularization parameter to control the trade-off between the reconstructed error and coefficients' sparsity. We can obtain the solution of the minimization problem by using the matlab package provided by Koh et al. (2007).

In practical application, we have to consider the sampling errors since the odor signals are quite susceptible to environment. The correlation between the samples from the same class may be weakened by the sampling errors. Therefore, an error vector $\mathbf{e}$ is added to the test sample $\mathbf{y}$, the real test sample becomes $\mathbf{y}_0 = \mathbf{y} + \mathbf{e}$. Ideally, all of entries of $\mathbf{e}$ should be zero. However, because of sampling errors, there are the nonzero entries of $\mathbf{e}$, indicating which part of the response is mis-sampled or corrupted by incident. Therefore, we improve the linear representation as follows:

$$\mathbf{y} = [\mathbf{A}, \mathbf{I}] \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix} = \mathbf{B}\mathbf{w}, \tag{12.5}$$

where $\mathbf{B} = [\mathbf{A}, \mathbf{I}] \in \mathbf{R}^{m\times(m+n)}$ and $\mathbf{w} = [\mathbf{x}; \mathbf{e}]$ (';' means the two columns are stacked). So Eq. 12.4 becomes such an l^1-minimization problem:

$$\hat{\mathbf{w}}_1 = \arg\min(\|\mathbf{B}\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|_1). \tag{12.6}$$

We can also obtain the solution of the minimization problem by using the MATLAB package provided by Koh et al. (2007).

12.3.3 Feature Extraction

In fact, before all samples are used in the sparse representation, feature extraction is necessary for reducing computational cost. In our experiments, we use principal components analysis (PCA) to extract features of samples. The dimension of each sample is therefore reduced from m to m' ($m' \ll m$). As a result, in Eq. 12.6, $\mathbf{B} \in \mathbf{R}^{m'\times(m'+n)}$ and $\mathbf{e}, \mathbf{y} \in \mathbf{R}^{m'}$.

12.3.4 Classification Steps

The following summarizes the classification procedure:

1. In the database, randomly select one sample as the test sample and the rest are as the training samples;

2. Extract the features from each training and test samples;
3. Form the extracted features of each training samples as $\mathbf{A}$, form the extracted features of the test sample as $\mathbf{y}$;
4. Normalize the column of $\mathbf{A}$ to have unit l^2-norm;
5. Form $\mathbf{B} = [\mathbf{A}, \mathbf{I}]$;
6. Solve $\hat{\mathbf{w}}_1 = \arg\min(\|\mathbf{Bw} - \mathbf{y}\|^2 + \lambda\,\|\mathbf{w}\|_1)$ and obtain $\hat{\mathbf{w}}_1$, where $\hat{\mathbf{w}}_1 = [\hat{\mathbf{x}}_1; \hat{\mathbf{e}}_1]$;
7. Reconstruct the test sample $\tilde{\mathbf{y}}^{(i)}$ by $\tilde{\mathbf{y}}^{(i)} = \mathbf{A}_i\hat{\mathbf{x}}_1^i + \hat{\mathbf{e}}_1$, $i = 1, 2, \ldots, k$, $\hat{\mathbf{x}}_1^i$ stands for the coefficient vector associated with class i;
8. Compute residual $\|\tilde{\mathbf{y}} - \tilde{\mathbf{y}}^{(i)}\|_2$ of each class respectively;
9. Throw test sample $\mathbf{y}$ to the class with the least residual;

12.4 Experiments and Results

In this section, we present experiment on blood glucose levels measurement. Table 12.1 list the subjects' blood glucose levels and the corresponding number. The samples were grouped by different blood glucose levels defined in (Wang et al. 2010).

Figure 12.3 shows the responses of the 12 sensors (S1–S12) to the samples of four diabetics with four blood glucose levels. In our system, S1–S6, S11, and S12 are specially sensitive to diabetics' breath (Guo et al. 2010a). Hence, these sensors have obvious responses to diabetes samples comparing with the responses of healthy sample (Fig. 12.2), especially for Level 4 samples. When the blood glucose of one patient is in low level (Fig. 12.3a), the response is not distinguishable from the healthy response (Fig. 12.2). However, when the patient is with high level blood glucose (Fig. 12.3c, d), the differences between the response of healthy subject and the patient is quite discernible.

In our experiment, we selected one sample as test sample and the rest as training samples and iterated the process until all samples are tested. One sample can be expressed as a $12 \times 810 = 9720$ dimensional column vector and there are 89 training samples, so $A \in R^{9720 \times 89}$. PCA was used to reduce the dimension of each sample from 9720 to 189 with 100% information kept. So $A \in R^{189 \times 89}$ after feature extraction.

Table 12.1 The blood glucose level and the corresponding number

Level stage	Name	Blood glucose level (mg/dL)	Number
Level 1	Low level	81–100	4
Level 2	Borderline	101–150	49
Level 3	High level	151–200	20
Level 4	Very high level	201–421	17
Total		81–421	90

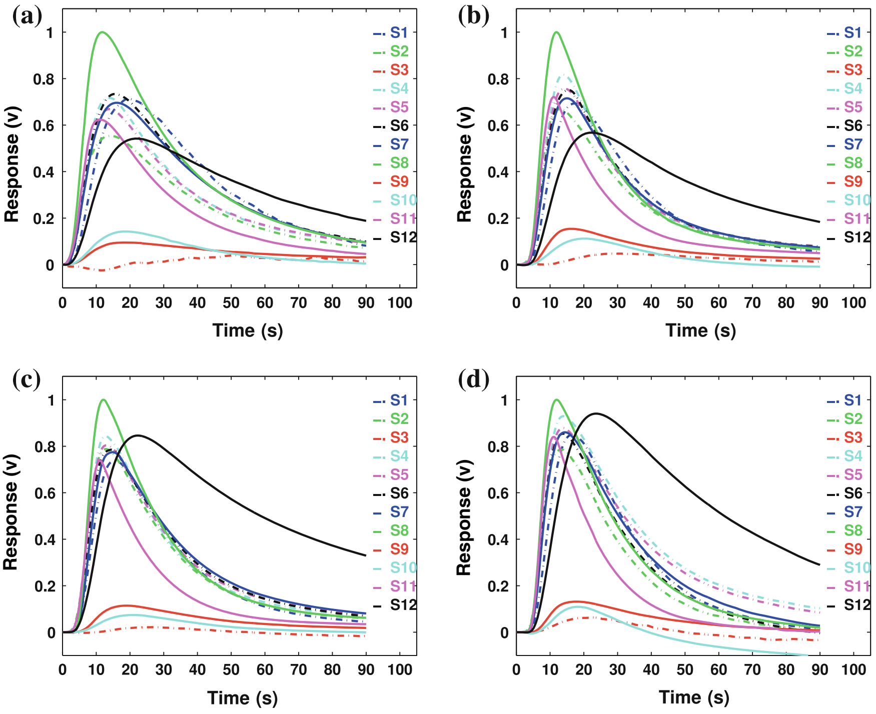


Fig. 12.3 Responses from four diabetics with different simultaneous blood glucose levels: **a** *Level 1*, **b** *Level 2*, **c** *Level 3*, and **d** *Level 4*

Figure 12.4 shows the sparse representation coefficients of samples from four classes: Level 1 (blue solid line), Level 2 (green dash-dot line), Level 3 (red dash line), and Level 4 (magenta dot line). The horizontal axis stands for the index of the training samples. Figure 12.4a–d represents a test sample from Level 1, Level 2, Level 3, and Level 4, respectively when representing this test sample by using all training samples. Since $\mathbf{w} = [\mathbf{x}; \mathbf{e}]$ (';' means the two columns are stacked), these figures just show the entries of $\mathbf{x}$ because $\mathbf{x}$ consists of the coefficients of sparse representation. When representing one sample from Level 1 class, the training samples in Level 1 obtain larger sparse coefficients than other training samples (Fig. 12.4a). Same as Level 1 samples, when representing one sample from Level 2, the training samples in Level 2 obtain larger sparse coefficients than other training samples (Fig. 12.4b). And so do Level 3 and Level 4 samples. Therefore, we can say that when representing one sample from the same class, the corresponding coefficients will be larger.

In fact, we can get a good result by computing the absolute value of the sparse coefficients. However, the residuals of the test sample and the representation (reconstructed sample by using the training samples) can achieve a better result when classifying the odor signals (Table 12.2). A comparative result is given by PCA+KNN

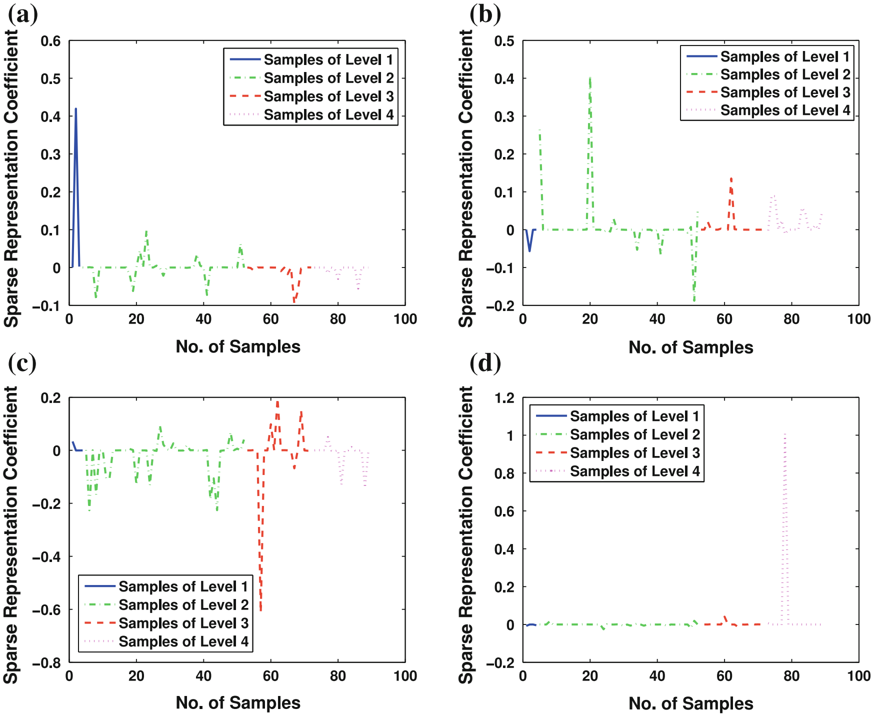


Fig. 12.4 Sparse representation coefficients of the samples from the four classes: *Level 1* (*blue solid line*), *Level 2* (*green dash-dot line*), *Level 3* (*red dash line*), and *Level 4* (*magenta dot line*)

Table 12.2 The classification results of SRC

Level	Number	Test outcome				Accuracy (%)
		Level 1	Level 2	Level 3	Level 4	
Level 1	4	**2**	2	0	0	50
Level 2	49	2	**41**	5	1	83.67
Level 3	20	0	4	**12**	3	60
Level 4	17	0	2	2	**13**	76.47

to show our proposed method outperforms it (Table 12.3). From the two tables we can see that the result given by KNN is quite poor. However, our proposed method provides a better result, especially when classifying Level 2 samples. Another comparative result is given by building a mathematical model to fit the responses of training samples (Guo et al. 2010b). The method is also not as good as our proposed approach except when classifying Level 1 samples (it gives the accuracy of 75% for Level 1). However, the result of SRC can be increased by improving the optimization equation (Eq. 12.4) and learning proper dictionary (matrix A in Eq. 12.4), which will be our future work.

Table 12.3 The classification results of KNN

Level	Number	Test outcome				Accuracy (%)
		Level 1	Level 2	Level 3	Level 4	
Level 1	4	**1**	3	0	0	25
Level 2	49	3	**27**	13	6	55.1
Level 3	20	0	8	**9**	3	45
Level 4	17	0	2	3	**12**	70.59

12.5 Summary

This chapter proposes a breath analysis system and a novel classification method for diabetes monitoring. The system structure, signal processing method, and the SRC approach are introduced in detail. The experimental results on blood glucose levels measurement shows the system has good performance in this application.

References

Amann A, Schmid A, Scholl-Burgi S, Telser S, Hinterhuber H (2005) Breath analysis for medical diagnosis and therapeutic monitoring. Spectrosc Eur 17(3):18–20

Deng C, Zhang J, Yu X, Zhang W, Zhang X (2004) Determination of acetone in human breath by gas chromatography-mass spectrometry and solid-phase microextraction with on-fiber derivatization. J Chromatogr B 810(2):269–275

Guo D, Zhang D, Li N, Zhang L, Yang J (2010a) A novel breath analysis system based on electronic olfaction. IEEE Trans. Biomed Eng 57(11):2753–2763

Guo D, Zhang D, Li N, Zhang L, Yang J (2010b) Diabetes identification and classification by means of a breath analysis system. In: International conference on medical biometrics, pp 52–63

Gutierrez-Osuna R (2002) Pattern analysis for machine olfaction: a review. IEEE Sens J 2(3):189–202

Koh K, Kim S, Boyd S (2007) l1_ls: a matlab solver for large-scale *l*1-regularized least squares problems

Melker R, Bjoraker D, Lampotang S (2006) System and method for monitoring health using exhaled breath. US patent application 11/512, 856

Tassopoulos C, Barnett D, Russell Fraser T (1969) Breath-acetone and blood-sugar measurements in diabetes. The Lancet 293(7609):1282–1286

Wang C, Mbi A, Shepherd M (2010) A study on breath acetone in diabetic patients using a cavity ringdown breath analyzer: exploring correlations of breath acetone with blood glucose and glycohemoglobin A1C. IEEE Sens J 10(1):54–63

Wright J, Yang A, Ganesh A, Sastry S, Ma Y (2008) Robust face recognition via sparse representation. IEEE Trans Pattern Anal Mach Intell 210–227

Part V
Medical Applications

Chapter 13
Breath Signal Analysis for Diabetics

Abstract Much attention has been focused on the noninvasive blood glucose monitoring for diabetics. It has been reported that diabetics' breath includes acetone with abnormal concentrations and the concentrations rise gradually with patients' blood glucose values. Therefore, the acetone in human breath can be used to monitor the development of diabetes. This chapter investigates the potential of breath signals analysis as a way for blood glucose monitoring. We employ a specially designed chemical sensor system to collect and analyze breath samples of diabetic patients. Blood glucose values provided by blood test are collected simultaneously to evaluate the prediction results. To obtain an effective classification results, we apply a novel regression technique, SVOR, to classify the diabetes samples into four ordinal groups marked with "well controlled", "somewhat controlled", "poorly controlled", and "not controlled", respectively. The experimental results show that the accuracy to classify the diabetes samples can be up to 68.66%. The current prediction correct rates are not quite high, but the results are promising because it provides a possibility of noninvasive blood glucose measurement and monitoring.

Keywords Breath analysis · Diabetes detection · Blood glucose levels · Support vector ordinal regression · Probabilistic output

13.1 Introduction

Blood glucose monitoring is particularly important for diabetic patients to control their conditions. Typically, they measure their blood glucose levels by piercing a finger to obtain a blood sample. This method is accurate, but is painful, invasive, and unsafe. Therefore, it does not suit everyone, especially in the case where it needs several samplings each day. As a result, it highly requires the noninvasive continuous glucose monitoring.

Recently, several noninvasive glucose monitoring approaches were studied, including spectrophotometric (IR light, fluorescence-bases, etc.), electrical impedance, photoacoustic, light scattering, and iontophoretic (Ferrante do Amaral and Wolf 2008; Tura et al. 2007). These approaches can provide a painless and con-

D. Zhang et al., *Breath Analysis for Medical Applications*,
DOI 10.1007/978-981-10-4322-2_13

venient procedure. However, all of them work through skin measurements. They are susceptible to environmental variations and subjects' physical and chemical parameters, such as changes in temperature and humidity, variation in subjects' blood pressure, skin hydration, and skin pigmentation (Tura et al. 2007).

Breath analysis is supposed to provide useful information about blood glucose levels of diabetic patients. The analysis of breath acetone associated with diabetes was the most extensively studied area in the field of breath analysis since 1960s (Rooth and Ostenson 1966; Levey et al. 1964; Tassopoulos et al. 1969). The key reason is that diabetes is the most common pathological cause of elevated blood ketone (Laffel 1999), and, plasma acetone has been proven to be linearly related to breath acetone (Sulway and Malins 1970). Therefore, we can smell the "sweet odor" of the breath of diabetics due to the elevated presence of acetone in blood and breath. As early as 1969, Tassopoulos et al. (1969) used GC to measure the breath acetone of 251 diabetic patients after overnight fasting and the patients' blood glucose values. The results showed that the concentration of breath acetone has quite high correlation with blood glucose values. More recently, Galassetti et al. (2005) discovered that the breath ethanol and acetone are highly correlated with the corresponding blood glucose values. Wang et al. (2010) observed that a linear correlation between the mean concentration of breath acetone and the mean of blood glucose levels exists. Consequently, breath analysis may offer a solution to non-invasive monitoring of blood glucose levels.

Conventional breath analysis is typically carried out by using the highly sensitive gas chromatography (GC). However, GC has the disadvantages of slow response, relatively high cost, lack of portability, and complicated operation. These disadvantages limit their applications in both the household and clinic. A less expensive and more portable alternative is the chemical sensor system, the so-called electronic nose (e-nose). It has been increasingly used in medicine for the diagnosis of renal disease (Lin et al. 2001), lung cancer (Blatt et al. 2007; D'Amico et al. 2009; Dragonieri et al. 2009), airway inflammation (Shih et al. 2010), and asthma (Fleischer et al. 2002; Dragonieri et al. 2007) and presents satisfactory performance. Recently, several research groups have applied this technique to the diagnosis of diabetes with promising results (Wang et al. 1997; Zhang et al. 2000; Mohamed et al. 2002). Despite a large amount of researches on diabetes diagnosis, there is still no report about using chemical sensor system to evaluate the blood glucose levels of diabetic patients. It is a more difficult tasks than diabetes diagnosis since the latter just distinguishes diabetic samples from healthy samples and other diseased samples, while the former attempts to make a classification intra diabetic samples.

In this chapter, we employ a specially designed chemical sensor system to collect and analyze breath samples of diabetic patients and to test the possibility of chemical-sensor-based blood glucose monitoring. Breath samples from diabetics and their blood glucose values provided by blood test are collected simultaneously. To obtain an effective classification results, we use a novel regression technique, SVOR, to classify the diabetes samples into four ordinal groups marked with "well controlled", "somewhat controlled", "poorly controlled", and "not controlled", respectively. Finally, we map the outputs of SVOR to probabilities to decide which levels the input sample belongs to. The experimental results show that the accuracy to classify the diabetes samples can be up to 68.25%. To our knowledge, this is the first attempt at adopting a low-cost chemical sensor system to measure the blood glucose values of diabetes. Frankly speaking, current prediction correct rates are not quite high comparing with the blood test, but the results are promising because it does provide a possibility of noninvasive blood glucose measurement and further investigation on this topic will continue.

The remainder of this chapter is organized as follows. Section 13.2 introduces how the breath analysis system works, how to collect breath samples, and data processing. SVOR and probability outputs are described in Sect. 13.3. Section 13.4 introduces the experiments. Section 13.5 presents the results and discussion. Finally, Sect. 13.6 offers our summaries.

13.2 Breath Analysis System

In this section, we will describe the detection mechanism of breath analysis system briefly since it has been introduced in (Guo et al. 2010). Then, we will present the breath collection, signal sampling, and data preprocessing.

13.2.1 Chemical Sensor Array

The most critical component of the system is the sensor array. During system design, our main concern is about the choice of sensors since the function and performance of the breath analysis system highly depend on the capabilities of sensors. A fundamental design concept is that each sensor should have a distinct sensitivity profile over a range of compounds expected in the target application, e.g., the detection of an unknown disease (Rock et al. 2008). In our system, we selected 12 metal oxide semiconducting sensors (from FIGARO Engineering Inc.) to form a sensor array. Sensors 1–6, and 11 respond positively to VOCs with various sensitivities. Since the biomarker of diabetes is acetone, a kind of VOCs, these sensors in our system have specially significant responses to the breath of diabetic patients. Sensor 7 is sensitive to sulfide, which is associated with liver diseases (Sehnert et al. 2002). Sensor 8 only detects carbon dioxide. Sensor 9 is used to detect ammonia, which is associated with renal diseases (Davies et al. 1997). Sensor 10 is sensitive to nitric oxide, which is associated with bronchiectasis, airway inflammation, and chronic obstructive pulmonary diseases (COPD) (Baraldi and Carraro 2006; Kharitonov et al. 1995; Horvath et al. 1998; Maziak et al. 1998). Finally, Sensor 12 is sensitive to hydrogen, which is used to detect gastrointestinal diseases (Brighenti et al. 2006; Le Marchand et al. 2006). These sensors are not specially sensitive to acetone, but they are useful when detecting the complications of diabetes.

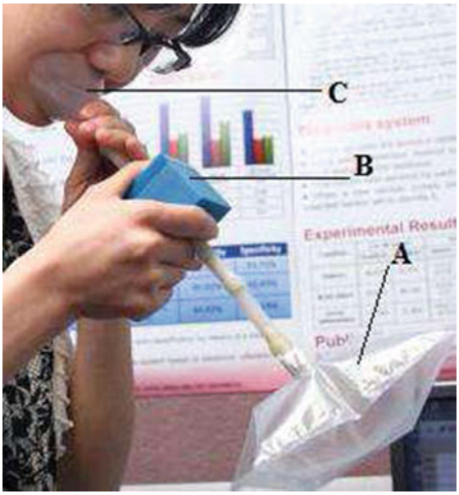
The twelve sensors are set in a stainless steel chamber. In the presence of breath gas samples, each sensor's conductivity increases depending on the concentration of acetone in the breath. An electrical circuit converts the change in conductivity to a voltage signal which corresponds to the gas concentration. Then, the signal measurement circuit converts the output voltage into an analog electrical signal. The analog signal is subsequently conditioned by signal filtering and amplification. Finally, the signal is sampled and transmitted through a USB interface to a computer for future analysis.

13.2.2 Breath Collection

The existence of breath acetone in diabetes occurs because plasma acetone is separated from blood and enters into the alveolar air via the alveolar pulmonary membrane. To guarantee what we collected is alveolar air, we used two breath collection bags during collection. One bag is a 150 ml normal plastic bag and the other is a 600 ml specially designed breath collection bag (Tedlar). The subjects were instructed to first take a deep breath first, then exhaled into and inflated the first 150 ml bag. This sample from the upper air passages was discarded because it may be contaminated (D'Amico et al. 2007). Finally, they exhaled into the second bag to produce a test sample.

Figure 13.1 shows how the subject's breath is collected using the breath collection bag (A), the air-tight box (B), which is filled with disposable hygroscopic material to absorb the water vapor from the breath, and a disposable mouthpiece (C). The hygroscopic material used is silica gel because it is stable and only reacts with a small number of compounds, such as fluoride, strong bases, and oxidizers. Our previous

Fig. 13.1 Exhaled air is collected with a gas sampling bag



experiments have shown that there has no obvious effect on breath acetone monitoring by using silica gel as a hygroscopic material. The disposable mouthpiece is equipped with an anti-siphon valve to prevent inhalation of collected gas and silica gel.

The breath gas is then introduced into the sensor chamber of the breath analysis device through the breath gas collection bag, which connects to the device inlet tubing via a plug-in connector.

13.2.3 Data Sampling and Preprocessing

The detailed data sampling and preprocessing have been introduced in our previous work (Guo et al. 2010). Generally, in the sampling procedure, program controlled system guarantees all samples are sampled under the same criterion. In the data preprocessing, the original signals are preprocessed by de-noising, baseline manipulation, and normalization. De-noising is to remove the noise from the original signals. Baseline manipulation is implemented for drift compensation, contrast enhancement, and scaling. Normalization is used to compensate for sample-to-sample variations caused by analyte concentration and pressure of oxygen (PO2).

13.3 Breath Sample Classification and Decision Making

The purpose of our investigation is to determine if the physical condition of diabetics is well controlled by monitoring patients' breath. This is a problem of ordinal classification, which arises frequently in medicine and information retrieval. In ordinal classification, the training samples are labeled by a set of ranks, for example, the labels of the samples in the categories {ill, sub-healthy, healthy} exhibit an order among the different categories. In our case, since the samples are grouped into "well controlled", "somewhat controlled", "poorly controlled", and "not controlled"', it is clear that there is an order among these labels: well controlled > somewhat controlled > poorly controlled > not controlled. Therefore, we use an ordinal regression technique to represent a coarse classification of the blood glucose levels represented by the labels of well controlled, somewhat controlled, poorly controlled, and not controlled. Patients can be informed the blood glucose levels by observing which category their samples fall into. We solve the problem by using SVOR technique. In this section, we first introduce the SVOR algorithm in detail, then present our method to get the probabilistic output of SVOR.

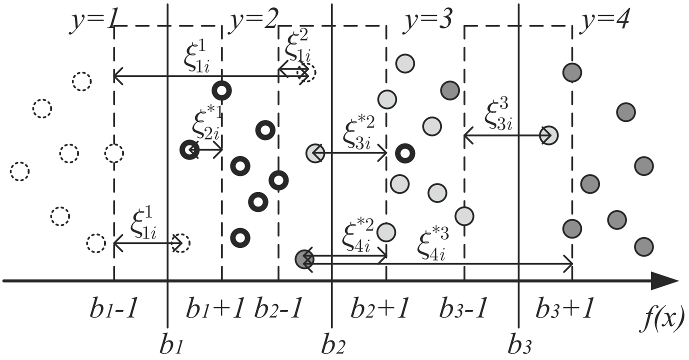


Fig. 13.2 An illustration of the idea of support vector ordinal regression. There are $r = 4$ categories. Each point stands for a sample $\mathbf{x}_i^j$. $y = j(j = 1, \ldots, r)$ indicates $\mathbf{x}_i^j$ should belong to the jth category. $b_1, \ldots, b_{r-1}$ are the thresholds between the r ordered categories. ξ_{ki}^j and ξ_{ki}^{*j} are slack variables

13.3.1 Support Vector Ordinal Regression

The ordinal regression can be illustrated in Fig. 13.2. There are $r = 4$ categories. In the figure, each point stands for a sample $\mathbf{x}_i^j$. $y = j(j = 1, \ldots, r)$ indicates $\mathbf{x}_i^j$ should belong to the jth category. The number of samples in the jth category is n_j. $b_1, \ldots, b_{r-1}$ are the thresholds between the r ordered categories. ξ_{ki}^j and ξ_{ki}^{*j} are slack variables to represent the prediction errors, where k indicates the sample $\mathbf{x}_i^j$ should belong to the kth category but actually it is not. For the r ordered categories, it generalizes the support vector formulation for ordinal regression by finding $r-1$ thresholds that divide the real line into r consecutive intervals. The method was first proposed by Shashua and Levin (2003). To guarantee the thresholds are accurately ordered, Chu and Keerthi (2005) improved the method from only considering the errors from the samples of adjacent categories to considering the errors from all samples in these categories and showed the performance of the improved method outperforms the previous one.

In our setting of ranking learning, there are $r = 4$ categories, so there are $r-1 = 3$ thresholds to consider. Each threshold has a margin, defined by the closest pair of two adjacent categories. $\mathbf{w}$ and b are scaled so that the distances from all boundary points to the corresponding thresholds are 1. Hence, the width of each margin is 2, from $b_j - 1$ to $b_j + 1, j = 1, \ldots, r - 1$. Same as SVR, we use the regression function $f(\mathbf{x}) = \left\langle \mathbf{w}, \phi(\mathbf{x}_i^j) \right\rangle$ to predict the target of sample $\mathbf{x}_i^j$ (Smola and Schölkopf 2004).

As shown in Fig. 13.2, samples out of the margin are deemed to fall into the correct categories, while samples in the margins might be misclassified. We can calculate the errors as follows when comparing the function values with the lower margin $(b_j - 1)$ and upper margin $(b_j + 1)$, respectively:

$$\xi_{ki}^{j} = \left\langle \mathbf{w}, \phi(\mathbf{x}_i^j) \right\rangle - (b_j - 1), \quad j = k = 1, \ldots, r-1, \tag{13.1}$$

$$\xi_{ki}^{*j} = (b_j + 1) - \left\langle \mathbf{w}, \phi(\mathbf{x}_i^j) \right\rangle, \ j = 1, \ldots, r-1 \text{ when } k = j+1, \tag{13.2}$$

where j in ξ_{ki}^{j} implies that the slack variable is associated with the lower categories of b_j, $*\,j$ in ξ_{ki}^{*j} indicates that the slack variable is associated with the upper categories of b_j, and the subscript ki denotes that the error is associated with the ith input sample in the kth category.

The purpose is to make the errors as small as possible. By taking all errors associated with all $r-1$ thresholds into account, the optimal problem can be defined as follows (Chu and Keerthi 2005):

$$\min_{\mathbf{w},b,\xi,\xi^*} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{j=1}^{r-1}\left(\sum_{k=1}^{j}\sum_{i=1}^{n_j}\xi_{ki}^{j} + \sum_{k=j+1}^{r}\sum_{i=1}^{n_j}\xi_{ki}^{*j}\right) \tag{13.3}$$

subject to

$$\begin{aligned} &\left\langle \mathbf{w}, \phi(\mathbf{x}_i^j) \right\rangle - b_j \le -1 + \xi_{ki}^{j}, \quad \xi_{ki}^{j} \ge 0, \\ &\qquad\qquad (k = 1, \ldots, j; i = 1, \ldots, n_j); \\ &\left\langle \mathbf{w}, \phi(\mathbf{x}_i^j) \right\rangle - b_{j-1} \le 1 + \xi_{ki}^{*j}, \quad \xi_{ki}^{*j} \ge 0. \\ &\qquad\qquad (k = j+1, \ldots, r; i = 1, \ldots, n_j). \end{aligned} \tag{13.4}$$

Reference Chu and Keerthi (2005) set a series of Lagrangian multipliers for the inequalities in Eq. 13.4 and obtained the dual problem. Solving the dual problem it is easy to get the discrimination function:

$$f(\mathbf{x}) = \langle \mathbf{w} \cdot \phi(\mathbf{x}) \rangle = \sum_{k,i}\left(\sum_{j=1}^{k-1}\alpha_{ki}^{*j} - \sum_{j=k}^{r-1}\alpha_{ki}^{j}\right)\kappa(\mathbf{x}_i^k, \mathbf{x}), \tag{13.5}$$

where α_{ki}^{*j} and α_{ki}^{j} are the Lagrangian multipliers and $\kappa(\mathbf{x}_i^k, \mathbf{x})$ is the kernel function. In the experiment, the Gaussian kernel $\kappa(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2}$ is used to train the classifier.

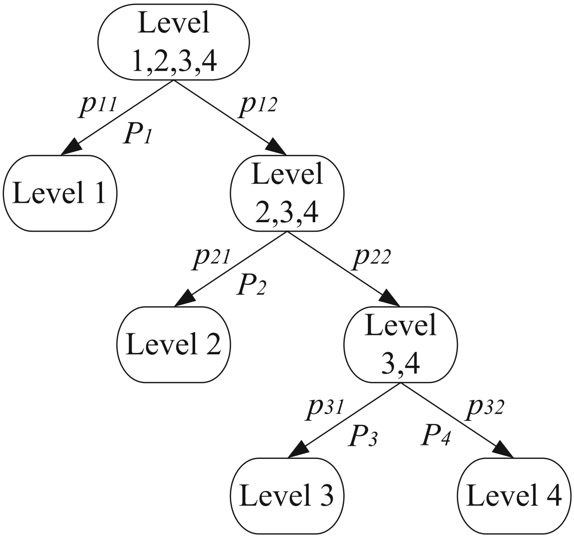
13.3.2 Probability-Based Classification

As Ref. Chu and Keerthi (2005) proposed, the predictive ordinal decision function is given by $\arg\min_i\{i : f(\mathbf{x}) < b_i\}$. It determines which category an input sample belongs to. However, the judgement is too absolute for medical application. It is like a binary output (0/1), which provides a judgement for the sample that is definitely healthy or diseased. However, in the case that one sample has the 49% probability to be diseased and 51% probability to be healthy, binary output cannot work well or often makes mistakes. So, we can just provide the probability that a sample belongs to one kinds of categories, and doctors can decide the condition of the samples according to the probability. The probability outputs enable us to make a flexible judgement about the real condition of the diseases, which is more suitable for medical applications.

In Sect. 13.3.1, the algorithm of SVOR is based on two assumptions: (1) Sample from the jth category should have a function value that is less than the lower margin $b_j - 1$, otherwise $\left\langle \mathbf{w}, \phi(\mathbf{x}_i^j) \right\rangle - (b_j - 1)$ is the error; (2) Sample in the $(j+1)$-th category should have a function value that is greater than the upper margin $b_j + 1$, otherwise $(b_j + 1) - \left\langle \mathbf{w}, \phi(\mathbf{x}_i^j) \right\rangle$ is the error. Each b_j has a working margin, from $b_j - 1$ to $b_j + 1$, as shown in Fig. 13.2. If the value of $f(\mathbf{x}_i)$ falls out of the margin, we can definitely decide which category the sample $\mathbf{x}_i$ belongs to. While, if the value of $f(\mathbf{x}_i)$ falls in the range, we may use probability to decide the category of $\mathbf{x}_i$.

We define a sequence to calculate the probability, as shown in Fig. 13.3. When a sample comes, we first get the corresponding $f(\mathbf{x}_i)$, meanwhile, regard Level 1 as one group and Level 2, 3, and 4 as another group. Then we calculate the probability the input sample belongs to Level 1 (p_{11}) and the probability that the input sample belongs to Level 2, 3, and 4 (p_{12}). In the next step, we regard Level 2 as one group and Level 3 and 4 as another group, and calculate the probability that the input sample belongs to the two groups, respectively, marked as p_{21} and p_{22}. Finally, we regard Level 3 as one group and Level 4 as another group and calculate the probability

Fig. 13.3 The framework of the probability calculation based on SVOR



the input sample belongs to the two groups, respectively, marked as p_{31} and p_{32}. In Fig. 13.3, $P_k (k = 1, \ldots, r)$ is the probability that the input sample belongs to each level. We can obtain P_k according to Fig. 13.3, where p_{j1} can be calculated by the following equations:

$$p_{j1} = \frac{1 - (f(\mathbf{x}_i) - b_j)}{2}, \tag{13.6}$$

$$p_{j2} = \frac{1 + (f(\mathbf{x}_i) - b_j)}{2}, \tag{13.7}$$

where $j = 1, \ldots, r-1$, b_j indicates the jth threshold to classify the ordinal categories, and $f(\mathbf{x}_i)$ is the function value of the ith test sample, defined by Eq. 13.5.

13.4 Experiments

In this section, some experiments based on our data set are presented. Comparative experiments between the proposed approach and other two common methods, Support Vector Machine (SVM) and Sparse Representation-based Classification (SRC), are also introduced. Finally, we present the probabilistic outputs of SVOR.

13.4.1 Breath Samples

According to the introduction of Sect. 13.2 about the data collection processing, breath samples and overnight fasting blood glucose values were sampled from 192 diabetics simultaneously. In all of the subjects, 110 of them were outpatients and 82 were inpatients. 123 of them were diagnosed type 2 diabetes and the rest were type 1 diabetes and other types. Table 13.1 lists the subjects' blood glucose levels and their corresponding number. The blood glucose levels are defined according to Chinese diabetes control criterion since all of samples we collected are from Chinese (Shang 2003). Totally, there are four levels. Fasting blood glucose <5.8 mmol/L implies the condition of patient is well controlled. Fasting blood glucose in 5.83–6.60 mmol/L shows the disease is somewhat controlled. When fasting blood glucose lies in 6.66–8.25 mmol/L, patient's condition is poorly controlled, while fasting blood glucose >8.31 mmol/L indicates that the condition is not controlled and doctors should adopt more effective treatment to control the condition.

Figure 13.4 shows the responses of the twelve different sensors (S1–S12) to the samples of four diabetics over the 90 s sampling period. Figure 13.4a–d are the responses from four diabetics' breath samples, their blood glucose levels increase gradually from Fig. 13.4a–d. In each of these figures, the horizontal axis stands for the sampling time (0–90 s) and the vertical axis shows the amplitude of the sensor output in volts. As aforementioned, Sensors 1–6, and 11 are specially sensitive

Table 13.1 The blood glucose levels and the corresponding number

Level stage	Name	Blood glucose value mmol/L (mg/dL)	Number	Female/Male
Level 1	Well controlled	<6.05 (110)	67	31/36
Level 2	Somewhat controlled	6.10-7.15 (111–130)	41	15/26
Level 3	Poorly controlled	7.21-8.80 (131–160)	39	16/23
Level 4	Not controlled	>8.86 (161)	45	20/25
Total diabetics		4.45-23.15 (81–421)	192	82/110

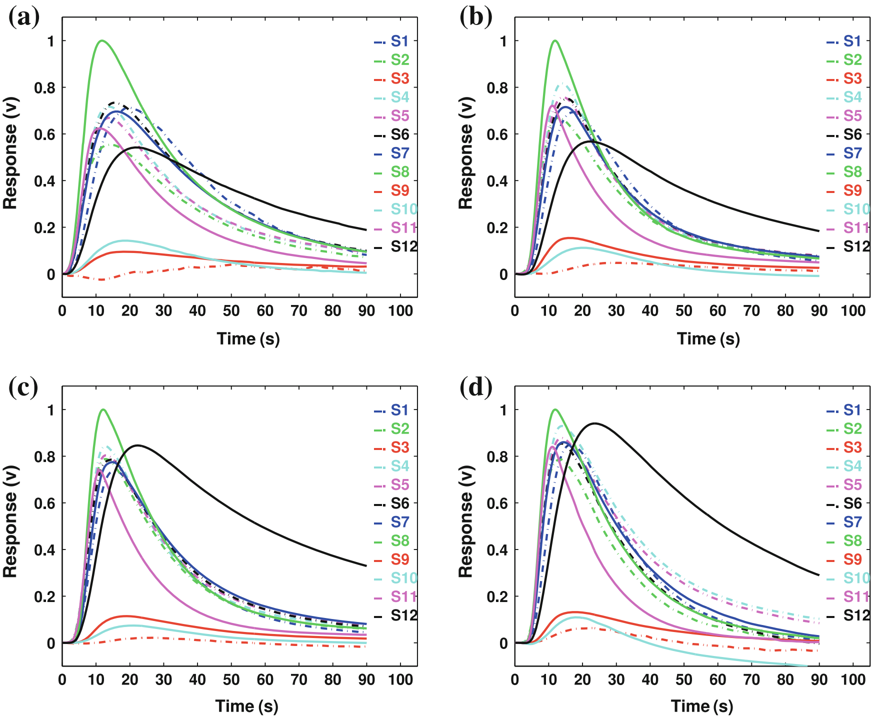


Fig. 13.4 Responses from four diabetics with different simultaneous blood glucose levels: **a** Level 1, **b** Level 2, **c** Level 3, and **d** Level 4. The *horizontal* axis stands for the sampling time (0–90 s) and the *vertical* axis shows the amplitude of the sensor output in volts

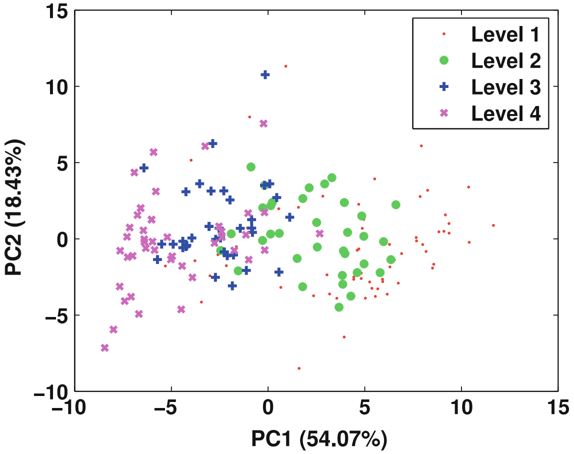


Fig. 13.5 PCA two-dimensional plot of the sensor signals corresponding to four levels: **a** samples from Level 1 (·), **b** samples from Level 2 (∗), **c** samples from Level 3 (+), and **d** samples from Level 4 (×)

to acetone. It has been reported that the transient responses of metal oxide semiconducting sensor are related to the type of gas and the amplitude of the responses is associated with the concentration of gas (Liess 2002). These figures display that the amplitudes of the sensors' responses increase with the blood glucose levels, from Fig. 13.4a–d. It is worth mentioning that Sensor 12 gives a very significant response, though it is not used for VOCs detection. In China, the special diet recommended for diabetics features large amounts of fermentable dietary fiber, which leads to colonic fermentation of indigestible carbohydrates (Brighenti et al. 2006). One product of colonic fermentation is hydrogen (Le Marchand et al. 2006). It is absorbed into the bloodstream and excreted through the breath. Therefore, the breath air of diabetics we have collected would include hydrogen. From the figures, we can see that it is possible to measure the blood glucose levels of diabetics by analyzing their breath gas samples.

To present the data distribution of these samples intuitively, we gave the two-dimensional plot of Principal Component Analysis (PCA), as Fig. 13.5 shows. Samples marked with · are from Level 1, samples from Level 2 are represented by ∗, Level 3 samples are +, and samples from Level 4 are marked with ×. The two dimensions explains 72.5% of the variation in the data, 54.07% for PC1 and 18.43% for PC2. The figure demonstrates that the samples from the four categories distribute ordinally even though some samples overlap.

13.4.2 *Feature Extraction*

In the experiment, we adopted PCA to extract the characteristic features directly from the original samples. We calculated the eigenvectors and eigenvalues of the training set and sorted the eigenvectors, i.e., principal components of PCA, by descendant eigenvalues, then projected both test data and training data onto the PCA subspace spanned by selected principal components. The criteria for principal component selection is $r_\lambda = \sum_{k=1}^{p} \lambda_i \big/ \sum_{k=1}^{n} \lambda_i$, where r_i is the eigenvalue, p is the amount of selected principal components, and n is the total number of eigenvalues. We selected $p = 60$ eigenvectors as features to count over 92% variability in the data set. It lost some information, but with lower dimensions, the classification could be more robust since the size of our samples is limited.

13.4.3 Support Vector Ordinal Regression

Using the feature set extracted by PCA directly, we conducted experiments by using the proposed SVOR. We also employed two other popular classification methods, Support Vector Machine (SVM) and Sparse Representation-based Classification (SRC) to make comparisons. The source code of the three classification methods are listed in Ref. Chu (2005), Chang and Lin (2001) and Koh et al. (2007).

SVOR trained the data sets using a Gaussian kernel $\kappa(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2}$ and a regularization factor value of $C = 100$. In the experiment of SVM, we employed the same Gaussian kernel to train the classifier. Both of the two learnt classifiers were used for the test sample classification. We also employed SRC to conduct the classification. The basic idea of SRC is to search for the most compact representation of an input signal in terms of linear combination of atoms in an overcomplete dictionary (Huang and Aviyente 2007). We represented a test sample as the linear combination of a set of training samples. In the linear combination, ideally, the training samples that are from the same category as the test sample have non zero coefficients, while those from the different category as the test sample have zero coefficients. Accordingly, we can assign a test sample to the category whose training samples hold higher linear combination coefficients. We obtained the coefficients by solving an optimization problem with the constraint of l_1 norm minimization.

In the verification stage, we used the leave-one-out cross-validation to obtain the classification results. In this process, first selected one sample as the test sample and the rest as the training samples to compute whether the test sample was classified correctly. The process was reiterated until every sample was tested once. It is worth noting that SVOR and SVM may not work well if only using one test sample in the verification stage. To solve this problem, we built a test set which includes two samples, one is real test sample, another is simulative sample, and just picked up the classification result of the real test sample.

Either using SVOR or using SVM and SRC to conduct the classification, the degree of misclassification could be different when they classified an unknown sample into Level 1–Level 4, respectively. For example, an unknown sample should belong to the category of Level 1, if it was misclassified into the category of Level 3 by one classifier while misclassified into the category of Level 2 by another classifier, it is obviously the first classifier is worse than the second one. In this chapter, we define a function to emphasize the performance of the classifiers SVOR, SRC, and SVM:

$$Acc = 1 - \sum\nolimits_{j=1}^{4} |i - j|\, R, \tag{13.8}$$

where i stands for the category the samples should belong to and j represents the category the samples were classified into by each classifier actually. R shows the rate of all samples are classified into every level.

13.5 Results and Discussion

In this section, we will present the comparative classification results from SVOR, SVM, and SRC, respectively and the results of probability based classification.

13.5.1 Support Vector Ordinal Regression

The results are listed in Table 13.2. The accuracy of each classifier was calculated by Eq. 13.8. For instance, the performance of SVOR to classify the samples in Level 1 was calculated by $Acc = 1-(0\times48/67+1\times17/67+2\times2/67+3\times0/67) = 68.66\%$. From the table, we can see that the results provided by SVOR obtained the best results. Comparing with other two classification methods, the results of SVOR were more reasonable, because the samples intensively fell into the correct group and its adjacent categories. For example, in the classification of Level 2, SRC classified two samples into Level 1 category, five samples into Level 3 category, and one sample

Table 13.2 The classification results obtained by various classifiers

Number		Classifier	Test outcome of classifiers				Accuracy (%)
			Level 1	Level 2	Level 3	Level 4	
Level 1	67	SVM	47	18	2	0	67.16
		SRC	47	17	3	0	65.67
		SVOR	**48**	**17**	**2**	**0**	**68.66**
Level 2	41	SVM	2	24	13	2	53.66
		SRC	6	24	10	1	56.10
		SVOR	**4**	**26**	**10**	**1**	**60.98**
Level 3	39	SVM	3	12	24	0	53.85
		SRC	2	11	24	2	56.41
		SVOR	**1**	**9**	**26**	**3**	**64.10**
Level 4	45	SVM	1	8	10	26	35.56
		SRC	2	3	13	27	44.44
		SVOR	**0**	**4**	**13**	**28**	**53.33**

into Level 4 category, incorrectly. SVM has the similar problem as SRC. SVOR classified 2 samples into Level 1 and 6 samples into Level 4, incorrectly. Misidentifying a somewhat controlled (Level 2) sample into the not controlled (Level 4) category is a much severer mistake than misidentifying it into the poorly controlled (Level 3) category. As a result, from the medical point of view, SVOR performs better than SRC and SVM when solving the ordinal problem.

Frankly speaking, the current classification accuracy is not as good as the blood test, but the result is promising because it does provide a possibility of noninvasive blood glucose measurement. There are two main reasons for the low accuracy: (1) Breath air, on the one hand, is a direct reflection of organs' conditions; on the other hand, it is susceptible to oral odor, especially when patients are smokers, or have bad breath. So, the critical problem of breath analysis of diabetics is how to eliminate the negative impact from the oral odor. One possible solution is to select sensors with high selectivity, i.e., they are only sensitive to acetone. (2) The current database, because of the limitation of sampling condition, is not large enough. While the current prediction algorithm is a kind of statistical pattern recognition method, which relies on a large number of samples to achieve a stable prediction result. Therefore, improving the breath analysis device, enlarging the database, and developing effective algorithm that specially suits the prediction of small samples will be our future work.

13.5.2 Probability-Based Classification

Table 13.2 presents the classification result of SVOR. If we map the outputs of SVOR into probabilities, as introduced in Sect. 13.3.2, we may discover the classification information in detail. Users, either patients or doctors could use the probabilities of classification as reference values when they make a prediction using our system.

Figure 13.6 demonstrates the probabilities of the classification of each sample. In each figure, the horizontal axis stands for the number of samples in each level and the vertical axis shows the probabilities that the samples belong to the four levels. Figure 13.6a–d represent the detailed classification information of samples from the category of Level 1–Level 4, respectively. In each figure, there are four lines demonstrating the probabilities that the samples belongs to the four categories. The probability that the sample belongs to Level 1 is marked with □, the probability that the sample belongs to Level 2 is marked with $\circ$, same as the first two levels, Level 3 is marked with $\diamond$ and Level 4 is marked with $*$.

To decide the categories of these samples, the decision rule used in our experiment is: sample i is decided to belong to the category which holds the maximum probability. As Fig. 13.6a demonstrates, most of samples have the maximum probabilities, even 100%, to belong to Level 1. But there are still some samples with quite high probabilities to belong to other three levels. It implies that these samples are not completely well controlled. Patients and doctors should be warned when the case happens.

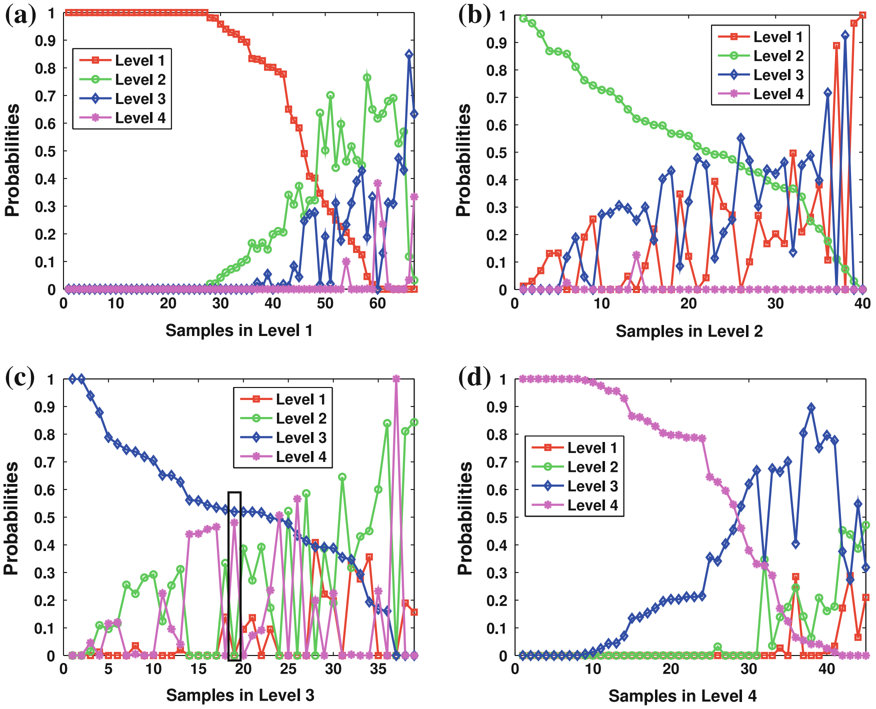


Fig. 13.6 Probability-based classification to classify the samples from four blood glucose levels: **a** Samples from Level 1, **b** Samples from Level 2, **c** Samples from Level 3, and **d** Samples from Level 4. The *horizontal* axis stands for the number of samples in each level and the *vertical* axis shows the probabilities that the samples belong to the four levels. The four lines in each figure represent four levels

The probabilities can be regarded as additional information of Table 13.2. In the application reported in this chapter, due to the limitation of current device and algorithm, it is difficult to provide an exact value of blood glucose. Our work focuses on the prediction of the condition of diabetic patients, i.e., well controlled, somewhat controlled, poorly controlled, and not controlled. Each condition includes a rough range of blood glucose values instead of an exact value. Samples near the boundary may have the same probabilities to belong to two conditions. Take the sample shown in Fig. 13.6c as an example, the sample in black rectangle has the probability of 52.98% belonging to Level 3 and the probability of 47.02% belonging to Level 4. Even though we predict the sample to belong to Level 3, its condition is severer than the average condition of Level 3 since it also has a quite high probability to belong to Level 4. Probability-based classification is especially useful in this case. We can combine the probabilistic classification results to the experience of the doctors to determine the exact status of the condition. It is more reasonable to make such judgment than only consider the absolute classification result provided by SVOR.

13.6 Summary

This chapter investigated the possibility of breath signals analysis as an approach for blood glucose monitoring. We collected and analyzed the breath samples of diabetic patients by using our self-designed breath analysis system and divided the samples into four categories marked with “well controlled”, “somewhat controlled”, “poorly controlled”, and “not controlled”, respectively, according to their simultaneous blood glucose values. Then we attempted to predict the condition of an input diabetic sample by using an ordinal regression technique, SVOR. Since the output of SVOR is an absolute value to indicate the sample’s condition, it does not involve the probability of a prediction. Therefore, we discovered a method to map the output of SVOR to probabilities to decide which levels the input sample belongs to. This approach enabled us to make a flexible judgment about the real condition of the disease, which is more suitable for medical applications than general classifiers. Frankly speaking, the current prediction accuracies are not quite high comparing with blood test, the results are promising because it does provide a possibility of noninvasive blood glucose measurement and further investigation on this topic will be continued.

References

Baraldi E, Carraro S (2006) Exhaled NO and breath condensate. Paediatr Respir Rev 7:20–22

Blatt R, Bonarini A, Calabro E, Della Torre M, Matteucci M, Pastorino U (2007) Lung cancer identification by an electronic nose based on an array of MOS sensors. Int Jt Conf Neural Netw 2007:1423–1428

Brighenti F, Benini L, Del Rio D, Casiraghi C, Pellegrini N, Scazzina F, Jenkins D, Vantini I (2006) Colonic fermentation of indigestible carbohydrates contributes to the second-meal effect. Am J Clin Nutr 83(4):817–822

Chang C, Lin C (2001) LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm

Chu W (2005) Source code for support vector ordinal regression. http://www.gatsby.ucl.ac.uk/~chuwei/svor.htm

Chu W, Keerthi S (2005) New approaches to support vector ordinal regression. In: Proceedings of the 22nd international conference on machine learning. ACM, pp 145–152

D’Amico A, Di Natale C, Paolesse R, Macagnano A, Martinelli E, Pennazza G, Santonico M, Bernabei M, Roscioni C, Galluccio G et al (2007) Olfactory systems for medical applications. Sens Actuators: B Chem 130(1):458–465

D’Amico A, Pennazza G, Santonico M, Martinelli E, Roscioni C, Galluccio G, Paolesse R, Di Natale C (2009) An investigation on electronic nose diagnosis of lung cancer. Lung Cancer 68:170–176

Davies S, Spanel P, Smith D (1997) Quantitative analysis of ammonia on the breath of patients in end-stage renal failure. Kidney Int 52(1):223–228

Dragonieri S, Schot R, Mertens B, Le Cessie S, Gauw S, Spanevello A, Resta O, Willard N, Vink T, Rabe K et al (2007) An electronic nose in the discrimination of patients with asthma and controls. J Allergy Clin Immunol 120(4):856–862

Dragonieri S, Annema J, Schot R, van der Schee M, Spanevello A, Carratú P, Resta O, Rabe K, Sterk P (2009) An electronic nose in the discrimination of patients with non-small cell lung cancer and COPD. Lung Cancer 64(2):166–170

Fleischer M, Simon E, Rumpel E, Ulmer H, Harbeck M, Wandel M, Fietzek C, Weimar U, Meixner H (2002) Detection of volatile compounds correlated to human diseases through breath analysis with chemical sensors. Sens Actuators B: Chem 83(1–3):245–249

Ferrante do Amaral C, Wolf B (2008) Current development in non-invasive glucose monitoring. Med Eng Phys 30(5):541–549

Galassetti P, Novak B, Nemet D, Rose-Gottron C, Cooper D, Meinardi S, Newcomb R, Zaldivar F, Blake D (2005) Breath ethanol and acetone as indicators of serum glucose levels: an initial report. Diabetes Technol Ther 7(1):115–123

Guo D, Zhang D, Li N, Zhang L, Yang J (2010) A novel breath analysis system based on electronic olfaction. IEEE Trans Biomed Eng 57(11):2753–2763

Horvath I, Loukides S, Wodehouse T, Kharitonov S, Cole P, Barnes P (1998) Increased levels of exhaled carbon monoxide in bronchiectasis: a new marker of oxidative stress. Br Med J 53(10):867–870

Huang K, Aviyente S (2007) Sparse representation for signal classification. Adv Neural Inf Process Syst 19:609–616

Kharitonov S, Wells A, O'connor B, Cole P, Hansell D, Logan-Sinclair R, Barnes P (1995) Elevated levels of exhaled nitric oxide in bronchiectasis. Am J Respir Crit Care Med 151(6):1889–1893

Koh K, Kim S, Boyd S (2007) l1 ls: a matlab solver for large-scale $l1$-regularized least squares problems

Laffel L (1999) Ketone bodies: a review of physiology, pathophysiology and application of monitoring to diabetes. Diabetes/Metab Res Rev 15(6):412–426

Le Marchand L, Wilkens L, Harwood P, Cooney R (2006) Breath hydrogen and methane in populations at different risk for colon cancer. Int J Cancer 55(6):887–890

Levey S, Balchum O, Medrano V, Jung R (1964) Studies of metabolic products in expired air. II. Acetone. J Lab Clin Med 63:574–584

Liess M (2002) Electric-field-induced migration of chemisorbed gas molecules on a sensitive film-a new chemical sensor. Thin Solid Films 410(1–2):183–187

Lin Y, Guo H, Chang Y, Kao M, Wang H, Hong R (2001) Application of the electronic nose for uremia diagnosis. Sens Actuators B: Chem 76(1–3):177–180

Maziak W, Loukides S, Culpitt S, Sullivan P, Kharitonov S, Barnes P (1998) Exhaled nitric oxide in chronic obstructive pulmonary disease. Am J Respir Crit Care Med 157(3):998–1002

Mohamed E, Linder R, Perriello G, Di Daniele N, Pöppl S, De Lorenzo A, (2002) Predicting type 2 diabetes using an electronic nose-based artificial neural network analysis. Diabetes nutr Metab 15(4):215–221

Rock F, Barsan N, Weimar U (2008) Electronic nose: current status and future trends. Chem Rev 108(2):705–725

Rooth G, Ostenson S (1966) Acetone in alveolar air, and the control of diabetes. Lancet 2(7473):1102–1105

Sehnert S, Jiang L, Burdick J, Risby T (2002) Breath biomarkers for detection of human liver diseases: preliminary study. Biomarkers 7(2):174–187

Shang D (2003) New concept of practical diabetes prevention

Shashua A, Levin A (2003) Ranking with large margin principle: two approaches. In: Advances in neural information processing systems, pp 961–968

Shih C, Lin Y, Lee K, Chien P, Drake P (2010) Real time electronic nose based pathogen detection for respiratory intensive care patients. Sens Actuators B: Chem 153–157

Smola A, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14(3):199–222

Sulway M, Malins J (1970) Acetone in diabetic ketoacidosis. Lancet 296(7676):736–740

Tassopoulos C, Barnett D, Russell Fraser T (1969) Breath-acetone and blood-sugar measurements in diabetes. Lancet 293(7609):1282–1286

Tura A, Maran A, Pacini G (2007) Non-invasive glucose monitoring: assessment of technologies and devices according to quantitative criteria. Diabetes Res Clin Pract 77(1):16–40

Wang C, Mbi A, Shepherd M (2010) A study on breath acetone in diabetic patients using a cavity ringdown breath analyzer: exploring correlations of breath acetone with blood glucose and glycohemoglobin A1C. IEEE Sens J 10(1):54–63

Wang P, Tan Y, Xie H, Shen F (1997) A novel method for diabetes diagnosis based on electronic nose. Biosens Bioelectron 12(9–10):1031–1036

Zhang Q, Wang P, Li J, Gao X (2000) Diagnosis of diabetes by image detection of breath using gas-sensitive laps. Biosens Bioelectron 15(5–6):249–256

Chapter 14
A Breath Analysis System for Diabetes Screening and Blood Glucose Level Prediction

Abstract It has been reported that concentrations of several biomarkers in diabetics' breath show significant difference from those in healthy people's breath. Concentrations of some biomarkers are also correlated with the blood glucose levels (BGLs) of diabetics. Therefore, it is possible to screen for diabetes and predict BGLs by analyzing one's breath. In this chapter, we describe the design of a novel optimized breath analysis system for this purpose. The system uses carefully selected chemical sensors to detect biomarkers in breath. Common interferential factors, including humidity and the ratio of alveolar air in breath, are compensated or handled in the algorithm. Considering the inter-subject variance of the components in breath, we design a feature augmentation strategy to learn subject-specific prediction models to improve the accuracy of BGL prediction. 295 breath samples from healthy subjects and 279 samples from diabetic subjects were collected to evaluate the performance of the system. The sensitivity and specificity of diabetes screening are 91.51% and 90.77%, respectively. The mean relative absolute error for BGL prediction is 20.6%. Experiments show that the system is effective and that the strategies adopted in the system can improve its accuracy. The system potentially provides a noninvasive and convenient method for diabetes screening and BGL monitoring as an adjunct to the standard criteria.

Keywords Alveolar air • Blood glucose level • Diabetes screening • Humidity compensation • Inter-subject variance

14.1 Introduction

Diabetes has become a great threat to human health. The timely diagnosis and frequent monitoring are important for managing the disease. To diagnose or monitor diabetes, traditionally, one must draw blood samples to check if his blood glucose level (BGL) falls within the normal range. This method is accurate but painful, invasive, and inconvenient (Turner 2011). Therefore, noninvasive diabetes screening and BGL prediction is arousing more and more interest recently.

D. Zhang et al., *Breath Analysis for Medical Applications*,
DOI 10.1007/978-981-10-4322-2_14

Approaches including reverse iontophoresis, fluorescence technology, bioimpedance spectroscopy, and so on (Ramachandran et al. 2010; Vashist 2012) have been studied. These approaches are painless and convenient, but still suffer disadvantages such as lack of specificity, inaccuracy due to subject's movement and sweating, skin irritation, etc. (Ramachandran et al. 2010; Vashist 2012).

Breath analysis is a noninvasive approach for clinical applications. By analyzing the concentrations of the biomarkers in breath, we are able to detect disease, monitor disease progression, or monitor therapy (Risby et al. 2006). Lots of efforts have been devoted to studying the breath biomarkers of diabetes. Acetone (Deng et al. 2004; Righettoni et al. 2013; Ueta et al. 2009; Wang et al. 2010) as well as many other volatile organic compounds (VOCs) (Ghimenti et al. 2013) in breath are proved to either have abnormal concentrations in diabetics or correlate with BGL. Compared to other approaches, breath analysis is readily acceptable and easy to collect samples, which makes it an attractive way for noninvasive diabetes screening and BGL prediction (Minh et al. 2012; Turner 2011).

Gas chromatography, mass spectroscopy (GC/MS) and related techniques can be used to analyze components in breath. GC/MS-related methods have high accuracy but relatively high cost, low portability, and complex usage, which limit their applications in massive diabetes screening and household BGL monitoring. Another breath analysis method makes use of chemical sensor systems, also known as electronic noses (e-noses), which are generally cheaper, faster, more portable and easier to operate. With the development in sensor technology, their accuracy has been improving.

A few chemical sensor systems have been developed for either diabetes diagnosis (Guo et al. 2010; Wang et al. 1997; Yu et al. 2005; Zhang et al. 2000) or BGL prediction (Guo et al. 2012; Saraoğlu et al. 2013). However, there are still problems not solved in these systems. First, the chemical sensor array should be further optimized for the specific application. Second, in previous systems, common fluctuations in breath samples were not well taken into account, such as humidity and the ratio of alveolar air. More importantly, considering the inter-subject variance of the components in breath, subject-specific BGL prediction models should be built (Turner 2011). Furthermore, the number of samples in the experiments of previous studies is small.

In this chapter, a novel breath analysis system for both diabetes screening and BGL prediction is proposed. The sensors array is carefully selected with the help of two pilot devices to improve the accuracy. Some of the sensors are under temperature modulation, which is an effective technique to enrich the information content and enhance the selectivity of gas sensors. A temperature-humidity sensor and a carbon dioxide sensor are used to compensate for fluctuations in breath samples. In the algorithm of BGL prediction, subject-specific prediction models are built by incorporating subject identity information into the feature vector. The purpose of these optimization strategies is to enhance the accuracy and robustness of the system.

To evaluate the proposed system, a series of experiments were made. Experiments with simulated samples confirm the system's capability in predicting the concentration of acetone with the presence of interference in breath. In the experiments with real breath samples, a total of 295 healthy and 279 diabetes breath samples were collected. The sensitivity and specificity of diabetes screening is 91.51% and 90.77%, respectively. The mean relative absolute error for BGL prediction is 20.6%. The results prove the effectiveness of the system as well as the optimization strategies used in our system. The hemoglobin A1c (HbA1c) values of 62 diabetic subjects were also predicted.

The rest of this chapter is organized as follows. Section 14.2 is the overall description of the system. Section 14.3 introduces the optimization strategies in the sensor array and prediction algorithms. The details of the experiments with simulated samples and real breath samples are in Sects. 14.4 and 14.5, respectively. Section 14.6 summarizes the chapter.

14.2 System Description

14.2.1 Structure of the Device

The proposed system consists of two parts: a device for breath acquisition and a set of algorithms for data analysis. The framework of the device is presented in Fig. 14.1. The gas route is made up of a small vacuum pump and a gas chamber. Breath or fresh air is drawn from the outside and injected into the gas chamber, which is a metal container with the sensor array embedded in its shell. The sensor's signals are captured by a signal processing circuit, where they are magnified and filtered. Finally, a data acquisition card digitizes the processed signals and transmits them to a computer using a USB cable. On the other hand, the computer sends control signals to the data acquisition card to control both the on/off of the pump and the modulation voltage of the temperature modulated sensors. A fan is placed next to the gas chamber to take away the heat emitted by the MOS sensors. The whole device is powered by a 12 V power adapter. Some basic parameters of the device are listed in Table 14.1.

It is worth noting that instead of the common box-shaped gas chamber, we designed a column-shaped gas chamber, as shown in Fig. 14.2. The internal shape of the chamber is cylindrical and the external shape is hexagon. The sensors are embedded on the six facets of the chamber. This design has three advantages: its internal shape allows gases to flow smoothly; its symmetry ensures that the gas concentration in the head space of each sensor is similar; the size of the chamber is miniature.

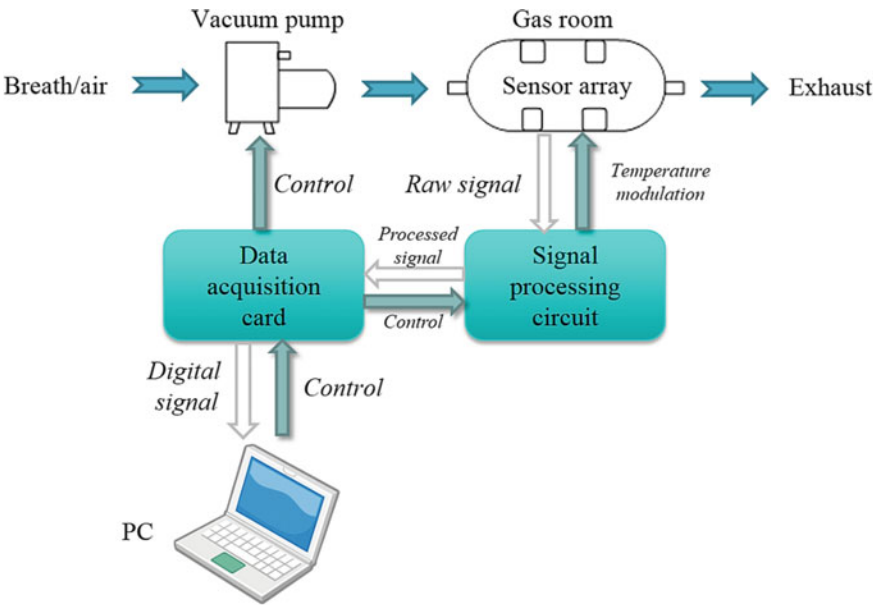


Fig. 14.1 Framework of the breath acquisition device

Table 14.1 Basic parameters of the proposed device

Device parameters	Specifications
Size	22 cm × 15 cm × 11 cm
Sampling duration	144 s
Sampling frequency	8 Hz
Injection flow rate	50 mL/s
Chamber volume	100 mL
Number of sensors	11

14.2.2 Sensor Array

We equip our device with 11 sensors, including 6 ordinary MOS sensors, 3 temperature modulated MOS sensors, a carbon dioxide sensor, and a temperature-humidity sensor. There are actually 12 input channels since the temperature-humidity sensor has 2 channels. Table 14.2 summarizes the model, manufacturer, and function of each sensor. The suffix "-TM" indicates that the sensor is a temperature modulated sensor. The sensors are specially selected for the purpose of diabetes diagnosis and BGL prediction. The selection scheme is introduced in Sect. 14.3.1 and Chap. 5.

Temperature modulated MOS sensors are believed to provide richer information and have better selectivity than MOS sensors operated in the ordinary way (Amini et al. 2012; Gutierrez-Osuna et al. 2003; Hosseini-Golgoo et al. 2011). The main

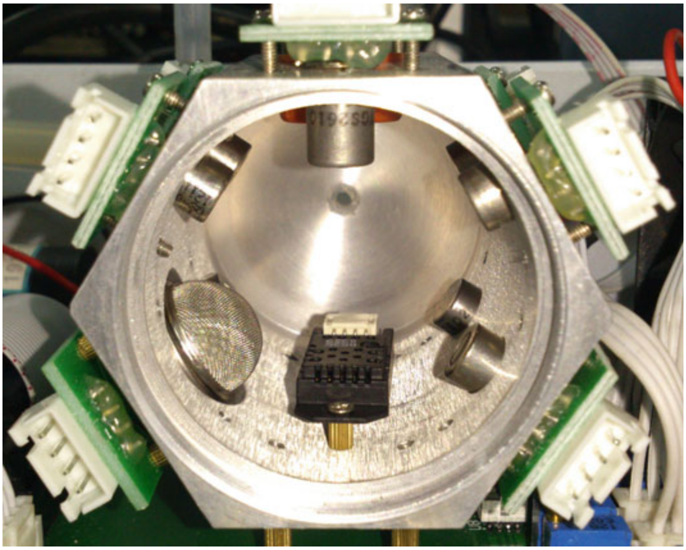


Fig. 14.2 A snapshot of the column-shaped gas chamber. Sensors are embedded on its wall. Gas enters the chamber from the inlet hole at one end. The outlet end was removed in the figure to show the inside of the chamber

Table 14.2 Summary of the sensor array

Channel	Model	Manufacturer	Function
1	TGS4161	Figaro Inc., Japan	Carbon dioxide
2	TGS822		VOCs, hydrogen, carbon monoxide, etc.
3	TGS826		
4	TGS2610-D00		
5	SP3S-AQ2	FIS Inc., Japan	
6	GSBT11	Ogam Inc., Korea	
7	WSP2111	Winsen Inc., China	
8	TGS2600-TM	Figaro Inc., Japan	
9	TGS2602-TM		
10	WSP2111-TM	Winsen Inc., China	
11	HTG3515CH	Humirel Inc., France	Temperature
12			Humidity

reason is that the response and gas sensitivity of MOS sensors are strongly temperature dependent (Hosseini-Golgoo et al. 2011). The reaction temperature, on the other side, is decided by the heating voltage. Traditionally, the heating voltage is fixed. But if we periodically modulate the heating voltage, we will be able to acquire the sensor's responses to the current analyte at different temperatures. With

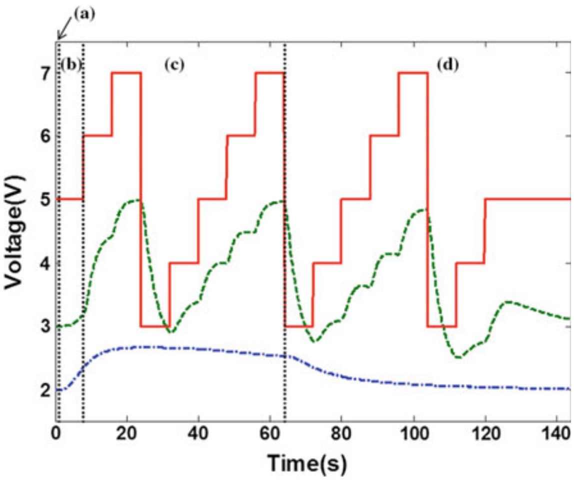


Fig. 14.3 Red solid line: heating voltage waveform for temperature modulation; Green dash line: a typical response of a temperature modulated sensor; Blue dash-dot line: a typical response of an ordinary sensor. The vertical dotted lines separate the 4 stages of the sampling procedure, **a** baseline stage; **b** injection stage; **c** reaction stage; **d** purge stage

more diagnostic information provided, the selectivity of the sensor can be enhanced.

In the proposed system, we use a staircase modulation voltage, which is similar to the methods in Gutierrez-Osuna et al. (2003) and Hosseini-Golgoo et al. (2011). The heating voltage changes between 3 V and 7 V in a staircase manner. It drives the temperature on the sensing material as well as the response of the sensor to oscillate in a similar way. Figure 14.3 illustrates the waveform of the heating voltage and compares typical responses of a temperature modulated sensor and an ordinary sensor. To our knowledge, this is the first time that the TM technique is used in breath analysis systems. The analysis results in Chaps. 5 and 10 confirm its efficacy.

14.2.3 Sampling Procedure

Similar to Guo et al. (2010), when collecting a breath sample, a subject is asked to take a deep breath and exhale into a 600 mL Tedlar® gas bag through a disposable mouthpiece. Then the filled gas bag is plugged onto the connector of the device. The computer software controls the device to complete the breath acquisition automatically. All breath samples are measured by the same process, which includes four stages as shown in Fig. 14.3:

(1) *Baseline stage (1 s):* The baseline values of the sensors are recorded for future data preprocessing.
(2) *Injection stage (7 s):* The pump is on. Breath is drawn from the gas bag to the gas chamber at a constant speed. The sensors' signals start to respond to the injected breath.

(3) *Reaction stage (56 s):* The pump is off. The sensors continue reacting with the components in breath. The responses of the MOS sensors without temperature modulation reach their maximum values.

(4) *Purge stage (80 s):* The pump is on again. Pure air is drawn into clean the gas chamber for 80 s. The sensors' responses gradually return to their baselines. After the responses remain stable in their baselines, the device is ready for the measurement of the next sample.

After the measurement process, we will get a digitized breath sample represented by 12 response curves (which will also be referred to as a "sample" hereinafter). Each response curve has 144 s*8 Hz = 1152 data points. The samples will be analyzed with the algorithms in the next section.

14.2.4 Data Analysis Methods

14.2.4.1 Signal Preprocessing

For each sensor, we compute its baseline value by averaging its response in the baseline stage. The value is then subtracted from the whole response curve. It is done to eliminate the interference of background noise of the sensors (Hierlemann et al. 2008). Humidity compensation is carried out by building a linear humidity response model for each sensor, which will be described in Sect. 14.3.2. Temperature compensation is not performed since it did not show big significance in our experiments.

14.2.4.2 Feature Extraction

The objective of this chapter is to verify the probability of the e-nose for diabetes screening and BGL prediction. Therefore, we only use simple and widely used data analysis algorithms. After signal preprocessing, the responses of the 10 chemical sensors (channels 1–10 in Table 14.2) are concatenated into a feature vector. The feature dimension (1152 * 10 = 11520) is very high, so principal component analysis (PCA) is used to extract low dimensional features. PCA projects high-dimensional data into a low-dimensional subspace while keeping most of the data variance. In the case of BGL prediction, considering the inter-subject variance between breath samples, we further add some categorical features to indicate the subject's identity. The detail of this feature is described in Sect. 14.3.3.

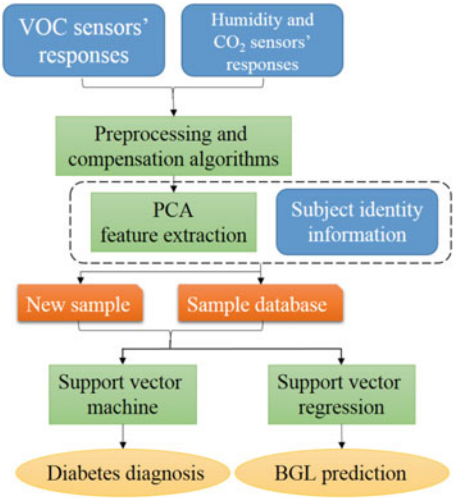

VOC sensors' responses
Humidity and CO_2 sensors' responses
Preprocessing and compensation algorithms
PCA feature extraction
Subject identity information
New sample
Sample database
Support vector machine
Support vector regression
Diabetes diagnosis
BGL prediction

Fig. 14.4 Framework of the data analysis algorithms. The rounded rectangles in *blue* are the features. The rectangles in *green* are the algorithms

14.2.4.3 Classification and Regression

Support vector machine (SVM) is among the most popular techniques for classification. It is a kernel-based method suitable for both linear and nonlinear problems. The main idea of the algorithm is to find a maximum margin hyperplane to separate the training samples. It has been proved to generalize well on test samples (Burges 1998). SVM has been adopted as the decision algorithm in many chemical sensor systems (Amini et al. 2012; Trincavelli et al. 2010). We will use it to discriminate between healthy and diabetes samples in the case of diabetes screening. The support vector regression (SVR) (Smola et al. 2004) algorithm is chosen to solve the BGL prediction problem, since it also has good generalization ability. The details of SVM and SVR can be found in Burges (1998) and Smola et al. (2004).

The entire framework of the data analysis algorithms is displayed in Fig. 14.4.

14.3 System Optimization

In order to enhance the system's accuracy and robustness for diabetes screening and BGL prediction, several optimization strategies are proposed, including sensor selection, compensation of influential factors and development of subject-specific prediction models.

14.3.1 Sensor Selection

The sensor array is the key part of a chemical sensor system. The sensors should be able to detect the breath biomarkers of diabetes, among which acetone is the most studied one. The concentration of breath acetone of diabetics is higher than that of healthy people (Deng et al. 2004; Ueta et al. 2009; Wang et al. 2010). Furthermore, Wang et al. (2010) split 30 diabetic subjects into four groups and found a linear correlation between the mean concentration of breath acetone and the mean BGL of each group. Turner et al. (2009) observed that the breath acetone declined linearly with BGL during hypoglycaemic clamps for each volunteer. However, the baseline values of breath acetone concentrations varied from subject to subject (Turner et al. 2009). Some other factors may influence the relationship between breath acetone and BGL as well, such as diet and age (Španěl et al. 2011). Besides acetone, compounds such as ethanol (Galassetti et al. 2005), carbon monoxide (Paredi et al. 1999), alkanes (Phillips et al. 2004), and methyl nitrate (Novak et al. 2007) in breath have also been proved to either have abnormal concentrations in diabetics or correlate with BGL. Some researchers have attempted to combine the concentrations of multiple VOCs and achieved good results in diabetes diagnosis and BGL prediction (Greiter et al. 2010; Lee et al. 2009).

To detect acetone, one way is to use specially designed acetone sensors (Wang et al. 1997). On the other hand, an array of carefully selected cross-sensitive VOC sensors can also have good performance in detecting one or more kinds of gases (Di Natale et al. 2005; Hierlemann et al. 2008), when pattern recognition algorithms are applied to discriminate different gas "patterns." As discussed in the last paragraph, there are drawbacks of detecting acetone alone and merits of detecting multiple biomarkers. So we first chose a set of candidate sensors to build pilot e-nose devices, then collected breath samples to evaluate them and select the best sensor combination.

Commercially available sensors are used as candidate sensors because they are easier to acquire, robust, and have good diversity. Some of the candidate sensors can detect and quantify VOCs as low as 0.05 parts-per-million (ppm) (Wolfrum et al. 2006), indicating that their precision is satisfactory. Two pilot devices were made with two batches of breath samples collected for sensor selection (Yan et al. 2012, 2014). Nine sensors were eventually selected to be employed in the final device, i.e., the sensors in channels 2–10 in Table 14.2. Sensors were selected using exhaustive searching experiments, see Chap. 5.

14.3.2 Compensation for Influential Factors

In breath analysis systems, influential factors such as humidity and the proportion of alveolar air in breath samples affect the responses of sensors. Compensation for these factors is important but often neglected in existing literature. In this section, the influence of these factors will be studied and the methods for compensation will be introduced.

14.3.2.1 Humidity

Humidity compensation is important in breath analysis systems, because human breath contains water vapor, and many VOC sensors are sensitive to humidity. An experiment was made to study the influence of humidity to the sensors. Acetone samples in five concentrations at four humidity levels were provided to the sensors. Results show that if the concentration of acetone is fixed, the maximum value of each sensor rises approximately linearly as the humidity rises. The water vapor in acetone samples has an additive effect to the response of the sensors. Thus, a "humidity coefficient" for each sensor can easily be estimated by linear regression, describing the increase of the sensor's maximum response when humidity increases 1%RH (Kashwan et al. 2005). The humidity compensation model for each VOC sensor is shown in Eq. 14.1.

$$\hat{R}_n^b(t) = R_n^b(t)\left(1 - s_n \frac{\Delta RH^b}{\max\left(R_n^b(t)\right)}\right) \tag{14.1}$$

In Eq. 14.1, $R_n^b(t)$ is the baseline-removed response curve of the nth sensor in the bth breath sample; s_n is the humidity coefficient of the nth sensor; ΔRH^b is the difference of humidity between the bth breath sample and the environment; $\hat{R}_n^b(t)$ is the compensated response curve. The proportion of magnitude which is considered to be brought by the water vapor in breath is subtracted. Experiment results in Sect. 14.5.3 show that the compensation improves the accuracy of the system.

Temperature in the gas chamber was also measured. Because the temperature was relatively stable among samples, temperature compensation is not applied. Besides, the compensation of instrumental variation and time-varying drift is discussed in Chaps. 6–9 (Yan et al. 2015, 2016a, b, 2017).

14.3.2.2 Proportion of Alveolar Air

General breath samples consist of two parts: dead-space air from the upper airway and alveolar air from the lungs. VOCs are exchanged between blood and alveolar air. In the case of diabetes screening and BGL prediction, dead-space air is a contaminant and dilutes the concentrations of VOCs in breath samples (Cao et al. 2007; Guo et al. 2010). So the proportion of alveolar air in a breath sample is an influential factor. This proportion is decided by the phase of the breath. In end-tidal breath, alveolar air is prevailing; whereas breath drawn from the initial phase contains more dead-space air.

Some researcher (Guo et al. 2012) tried to collect the two parts of breath separately with two cascade gas bags. However, the estimation of the volume of dead-space air may be inaccurate. Moreover, some patients are unable to blow up the two cascade gas bags because of their illness. Another method is to estimate the proportion of alveolar air from the CO_2 concentration in breath samples (Cao et al. 2007). Higher CO_2 concentration is an indication of higher proportion of alveolar air. Thereby, we employ a CO_2 sensor in the proposed device. The responses of the CO_2 sensor and the VOC sensors are combined to extract PCA features, which allow the pattern recognition algorithms to learn a better prediction model taking the information of CO_2 concentration into account. The experiment results in Sect. 14.5.3 show that with the information from the CO_2 sensor, better accuracy is acquired.

14.3.3 Subject-Specific Prediction Model

Researchers have identified the inter-subject variance of the relationship between breath acetone and BGL (Ghimenti et al. 2013; Turner et al. 2009). As shown in Turner et al. (2009), although breath acetone is correlated with BGL for each subject, the baseline values of breath acetone vary among subjects. The author of Turner (2011) concluded that calibration of acetone with BGL for each individual is required. However, in previous breath analysis systems aiming at predicting BGL (Guo et al. 2012; Saraoğlu et al. 2013), the prediction models are not subject-specific.

To make prediction models subject-specific, an intuitive way is to build a model for each subject with samples from the same subject as training samples. But this method is not applicable when the number of samples from one subject is not enough for an accurate model. In this chapter, we propose to add p categorical features in each feature vector to indicate the subject's identity, where p is the number of subjects in total. Concretely, for each sample, the ith additional categorical feature will be 1 if the training sample is from the ith subject, or be 0 otherwise. If the regression model is linear, this feature augmentation strategy is equivalent to adding a subject-specific bias in the model, which can compensate for the constant part of the inter-subject variance. From the results in Sect. 14.5.3.3, we find this method can markedly improve the accuracy for BGL prediction.

14.4 Experiments with Simulated Samples

An experiment was made to test the system's ability to quantify the main breath biomarker of diabetes, i.e., acetone. According to (Deng et al. 2004), the concentration of breath acetone in healthy subjects is ranged from 0.22 to 0.80 ppm, while that in subjects with type 2 diabetes is from 1.76 to 3.73 ppm. For subjects with type 1 diabetes, breath acetone could be as high as 21 ppm (Turner et al. 2009). So we prepared acetone samples in 8 concentrations (0.1, 0.2, 0.5, 1, 2, 5, 10, 20 ppm), with two samples for each concentration. The 16 samples were measured by our device using the sampling procedure in Sect. 14.2.3. Then the concentration of acetone in each sample was predicted by leave-one-out cross-validation. The data analysis method was preprocessing + PCA + SVR as introduced in Sect. 14.2.4. The prediction is evaluated by its mean absolute error (MAE) defined in Eq. 14.2, where x_i and $\hat{x}_i$ are the true and predicted concentration of the ith sample, respectively; $n = 16$. In this experiment, the MAE is 0.16 ppm, which indicates that the system can predict the concentration of acetone with high accuracy.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |x_i - \hat{x}_i| \tag{14.2}$$

Although acetone is among the most abundant VOCs in breath (Turner 2011), there are many other VOCs in breath that may interfere the measurement of acetone. Thus, another experiment was made to test the system’s ability to measure acetone with the presence of interfering VOCs. Eight breath samples were collected from each of five healthy volunteers. Then an addition of acetone was mixed with these 40 breath samples. The eight samples of each volunteer were made to contain an additional acetone of 0, 0.2, 0.3, 0.7, 1.7, 3.3, 5.0, and 6.7 ppm, respectively. These mixed samples were used to simulate the existence of interfering VOCs and the variation of baseline acetone concentrations in real breath samples. Then the concentration of the additional acetone in each sample was predicted. The leave-one-out strategy and the preprocessing + PCA + SVR algorithm were applied. It is worth noting that the feature augmentation strategy described in Sect. 14.3.3 was added to build subject-specific prediction models. In this experiment, the MAE is 0.22 ppm, proving that the system is able to predict the concentration of acetone in the presence of interfering VOCs and the variation of baseline acetone concentrations.

14.5 Experiments with Breath Samples

14.5.1 Overview of the Breath Samples

A total of 295 healthy samples and 279 diabetes samples were collected from Guangdong Provincial Hospital of Traditional Chinese Medicine (Guangzhou, China). The health states of the healthy subjects were confirmed by physical examinations. The diabetes samples were from 87 inpatient volunteers. For each diabetic subject, several samples were collected at 2 h after meal in different days together with the simultaneous BGLs. The number of samples per subject ranges from 1 to 11. Some information about the diabetic subjects is listed in Table 14.3. Figure 14.5 shows the distribution of BGLs of the diabetic samples.

Table 14.3 Basic information of the diabetic subjects

Item	Value
Number	87
Male/Female	39/48
Age	39–91
Type 1/Type 2	1/86
Disease duration (years)	0.5–19
Blood glucose level (mmol/L)	4.4–23.1

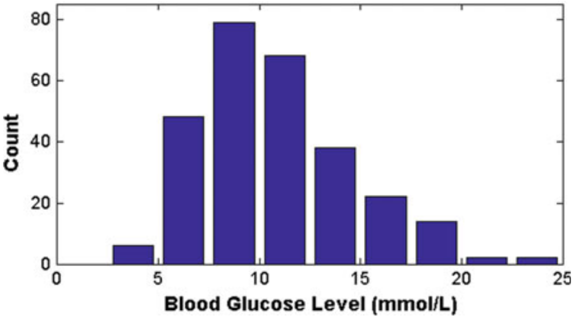


Fig. 14.5 The distribution of BGLs of the 279 diabetic samples

Hemoglobin A1c (HbA1c) is also an important parameter for the diagnosis and monitoring of diabetes. It serves as a marker for average BGL during the preceding 3–4 months with a higher weight over the latest 30 days (Rohlfing et al. 2002). Correlation between breath acetone and HbA1c of diabetic subjects has been reported (Ueta et al. 2009; Wang et al. 2010). In this study, 62 out of the 87 subjects had the HbA1c test within the last 13 days. Their HbA1c values range from 5.1% to 15.2%. An experiment was made to predict the HbA1c of the 62 subjects.

14.5.2 Data Analysis Procedure

14.5.2.1 Distinguishing Between Healthy and Diabetes Samples

Diabetes screening was achieved by distinguishing between healthy and diabetes samples. After a digitized breath sample was acquired, it underwent baseline removal, humidity compensation and PCA feature extraction. The ratio of variance in PCA was set to be 99.98%, extracting about 60 features. They were then scaled to have zero mean and unit variance. SVM (Chang et al. 2011) with a Gaussian kernel was used for classification. 140 healthy and 140 diabetes samples were randomly selected to train the SVM classifier. Another 139 healthy and 139 diabetes samples were used for testing. We ran this procedure 50 times and computed the average sensitivity and specificity.

14.5.2.2 Blood Glucose Level and HbA1c Prediction

In these two cases, only the diabetes samples were investigated. The data analysis procedures for BGL and HbA1c prediction are mostly the same. Baseline removal, humidity compensation and PCA feature extraction were applied to the samples. The optimized ratio of variance was set to be 99.1%, extracting about 12 features after PCA. The dimension is lower than that in diabetes screening so as to prevent the regression model from overfitting. The features were further scaled to have zero mean and unit variance. The leave-one-out cross-validation protocol was employed. The feature augmentation strategy described in Sect. 14.3.3 was added. However, when the HbA1c was predicted, only the first breath sample of each subject was used. There is no need to add the feature augmentation strategy since each subject had only one sample. SVR (Chang et al. 2011) with a linear kernel was adopted to do the prediction.

Three evaluation criteria were utilized to quantify the accuracy of the prediction. The mean absolute error (MAE) is the average deviation of the prediction from the true target. The mean relative absolute error (MRAE) measures the relative error by normalizing the absolute error with the true target. The correlation coefficient (r) measures the linear correlation between the true target and the predicted value. If we denote x_i as the true target (BGL or HbA1c) of the ith sample, $\hat{x}_i$ as the predicted value, n as the number of samples, $\bar{x}_i$ and $\bar{\hat{x}}_i$ as the mean of all the true and predicted values, the MRAE and r can be defined as follows:

$$MRAE = \left(\frac{1}{n} \sum_{i=1}^{n} \left| \frac{x_i - \hat{x}_i}{x_i} \right| \right) \times 100\% \tag{14.3}$$

$$r = \frac{\sum_{i=1}^{n} (x_i - \bar{x}_i)(\hat{x}_i - \bar{\hat{x}}_i)}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x}_i)^2} \sqrt{\sum_{i=1}^{n} (\hat{x}_i - \bar{\hat{x}}_i)^2}} \tag{14.4}$$

14.5.3 Results and Discussion

14.5.3.1 Distinguishing Between Healthy and Diabetes Samples

Figure 14.6 exhibits the average responses of the healthy samples and the diabetes samples. To observe their differences more clearly, we have made a comparison in Fig. 14.7. For most VOC sensors (S2-S10), the mean responses of the diabetes samples are larger than that of the healthy samples, showing that the concentration of VOCs in breath of the diabetics is higher than that of the healthy subjects.

The final sensitivity and specificity for diabetes screening are 91.51% and 90.77%, respectively. The breath analysis system can distinguish between healthy

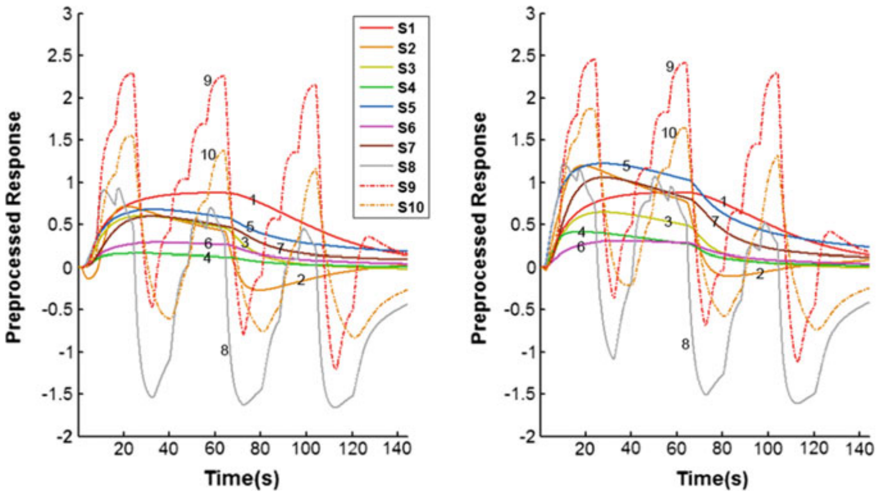
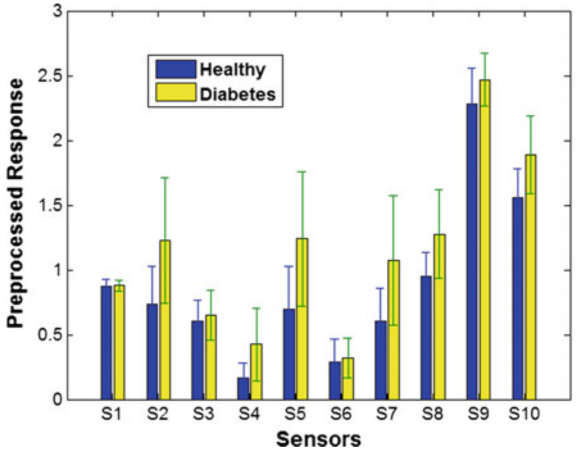


Fig. 14.6 Average responses of the two classes. *Left* healthy; *right* diabetes. S1 is a CO_2 sensor; S2-S7 are ordinary MOS sensors; S8-S10 are temperature modulated MOS sensors, so their responses are staircase-shaped

Fig. 14.7 Average responses of each sensor in the two classes. The coordinates on *x*-axis are the sensors' indices. S1 is the CO_2 sensor and S2-S10 are VOC sensors. The *y*-axis is the mean of the maximum value of the preprocessed response. *Error bars* represent the standard deviations



and diabetes samples with a promising accuracy. The accuracy is comparable to previous studies, especially given the fact that the database in this study is larger. The result shows that the system has the potential to be an assistive tool for diabetes screening.

14.5.3.2 Blood Glucose Level and HbA1c Prediction

In order to observe the difference between breath samples from subjects with different BGLs, we divide the diabetes samples in our database into four groups. The BGL thresholds are set to be 7.4, 9.7, and 13.2 mmol/L, so as to make the number of samples in each group close to each other. The mean responses of the VOC sensors in the four groups are shown in Fig. 14.8. The mean response is ascending from the first to the last group for most sensors except S6 and S9, which is probably because S6 and S9 have higher sensitivity to the interfering components than to acetone. It should also be noticed that the standard deviation in each group is large, which indicates that there are overlaps between groups. This result shows that the prediction task is challenging. The discoveries above are consistent to those in Wang et al. (2010).

BGLs of 279 samples from 87 diabetic subjects are predicted using the leave-one-out protocol. Correlation between the true and the predicted BGLs can be observed from the scatter diagram in Fig. 14.9. The final results are MAE = 2.09 mmol/L, MRAE = 20.6%, $r = 0.658$. The result is better than a recent study (Saraoğlu et al. 2013), in which a chemical sensor system was designed to predict the BGL of 30 samples with MRAE = 25.2%.

The MAE, MRAE, and correlation coefficient of the HbA1c prediction are 1.86, 21.0%, and 0.56, respectively. The MAE and MRAE are lower than that in the BGL prediction experiment, which is possibly because HbA1cs are more stable and range in a smaller interval. The BGL prediction models are subject-specific and in fact more accurate, so the correlation coefficient of BGL prediction is higher.

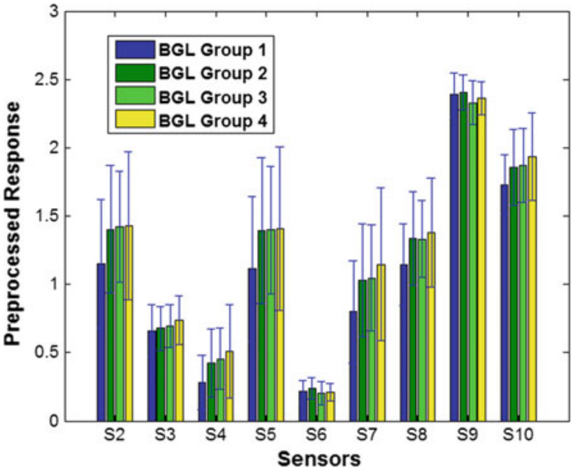


Fig. 14.8 Mean responses of the VOC sensors in different BGL groups. The diabetes samples are divided into four groups according to their BGLs. The coordinates on *x*-axis are the sensors' indices. The *y*-axis is the mean of the maximum value of the preprocessed response. *Error bars* represent the standard deviations

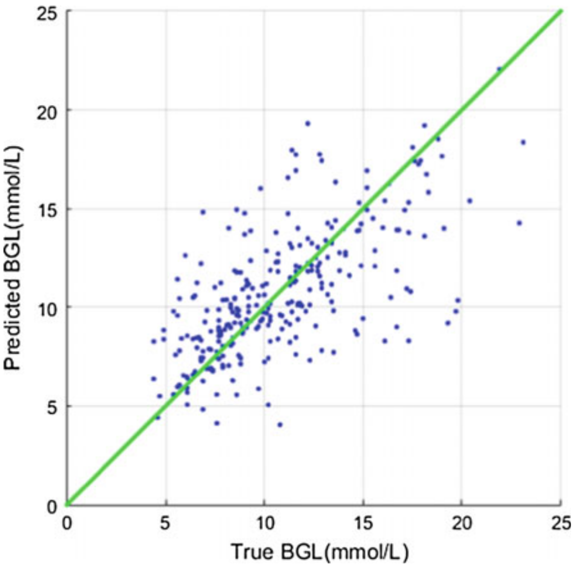


Fig. 14.9 Scatter diagram for BGL prediction. The *x*-axis is the true BGL. The *y*-axis is the predicted BGL. The MAE, MRAE, and correlation coefficient of the prediction are 2.09, 20.6%, and 0.658, respectively

14.5.3.3 Effectiveness of the Optimization Strategies

In this section, the effect of the optimization strategies proposed in this chapter is assessed. First, the accuracies acquired with or without the compensation algorithms in Sect. 14.3.2 are compared. The results for both diabetes screening and BGL prediction are demonstrated in Table 14.4. In the table, HC is short for humidity compensation. CO_2 stands for the information from the CO_2 sensor. For diabetes screening, the addition of HC and CO_2 improves the sensitivity and specificity. For BGL prediction, with the addition of HC and CO_2, the MAE and MRAE are reduced and the correlation coefficient is increased. Therefore, the proposed algorithms aiming at compensating fluctuations of humidity and the proportion of alveolar air are effective.

Table 14.5 shows that the BGL prediction accuracy is improved by the strategy of building subject-specific prediction models. We can infer that the influence of the inter-subject variance has been reduced. Figure 14.10 gives another hint on how much the subject identity information helps the prediction. The diabetic subjects are

Table 14.4 Comparison of the performance on diabetes screening and BGL prediction with different strategies

Method	Sensitivity (%)	Specificity (%)	MAE	MRAE (%)	*r*
No HC, no CO_2	90.23	88.87	2.31	20.7	0.599
No HC, add CO_2	90.81	89.13	2.11	23.2	0.658
Add HC, no CO_2	90.74	90.14	2.12	20.9	0.650
Add HC, add CO_2	91.51	90.77	2.09	20.6	0.658

Table 14.5 Comparison of the performance on BGL prediction with or without feature augmentation

Method	MAE	MRAE (%)	*r*
Without feature augmentation	2.82	28.2	0.310
With feature augmentation	2.09	20.6	0.658

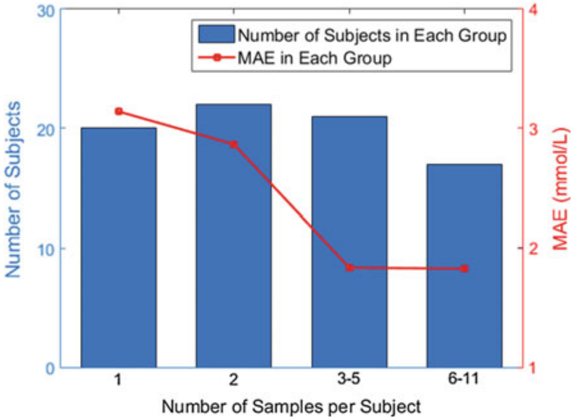


Fig. 14.10 The relationship between the number of samples collected from a subject and the prediction MAE. Group 1–4 contain subjects who have 1, 2, 3–5, and 6–11 samples, respectively. The *bars* show the number of subjects in each group (corresponding to the *y*-axis on the *left*). The *red line* shows the MAE of the samples in each group (corresponding to the *y*-axis on the *right*)

grouped according to the number of samples collected from them in the database. Groups are designed so that the number of subjects in each group is close to each other. Then we compute the MAE of BGL prediction in each group. We find that the MAE of the subjects with more than 3 samples are better than those with only 1–2 samples. This is probably because that with more training samples provided for each subject, the subject-specific prediction model can be more accurate. But this hypothesis still needs further validation using a larger database. To sum up, the subject identity information is important for the training of the prediction model; to improve the prediction accuracy, we may increase the number of training samples of each subject.

14.6 Summary

This chapter proposes a breath analysis system for diabetes screening and blood glucose level prediction. The system includes a breath acquisition device and a set of data analysis algorithms. The device has the advantage of being noninvasive, portable, and easy to operate.

To increase the accuracy and robustness of the system, targeted improvements were made on the sensor array, preprocessing, and feature extraction algorithms. The improvements can be roughly categorized into two aspects. In the aspect of medicine, some results in breath analysis studies were consulted. A CO_2 sensor was employed to compensate for the difference of proportion of alveolar air in breath samples. Subject-specific prediction models were built for BGL prediction to reduce the influence of the inter-subject variance. In the aspect of sensor technology, an optimal cross-sensitive VOC sensor array was selected. Temperature-modulated MOS sensors were adopted and proved to be useful. The humidity drift of the sensors was compensated. The effectiveness of these improvement strategies was confirmed by experiments. These strategies are expected applicable not only in the proposed system, but in other breath analysis systems as well.

Over 500 breath samples were collected to evaluate the performance of the system. We achieved a promising accuracy in diabetes screening. For BGL and HbA1c prediction, the mean relative absolute error is 20.6% and 21.0%, respectively. The BGL prediction result is better than previous breath analysis systems, but still not quite adequate for practical use. One of the error sources is the inter-subject variance of the components in breath samples. We have made attempts to reduce the influence of this variance by feature augmentation. With more training samples for each subject and more sophisticated prediction models, the error may be further diminished. Since our experiments were not conducted in strictly controlled environments, there is also intra-subject variance caused by factors such as diet, exercise, and insulin injection. The influence of these factors needs to be further studied to build a prediction model properly taking them into consideration (Španěl et al. 2011; Turner 2011). Larger database should be collected to validate the models.

References

Amini A, Bagheri MA, Montazer G (2012) Improving gas identification accuracy of a temperature-modulated gas sensor using an ensemble of classifiers. Sens Actuators B: Chem

Burges CJ (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Disc 2:121–167

Cao W, Duan Y (2007) Current status of methods and techniques for breath analysis. Crit Rev Anal Chem 37:3–13

Chang C-C, Lin C-J (2011) Libsvm: a library for support vector machines. ACM Trans Intell Syst Technol (TIST) 2:27

Deng C, Zhang J, Yu X et al (2004) Determination of acetone in human breath by gas chromatography-mass spectrometry and solid-phase microextraction with on-fiber derivatization. J Chromatogr B 810:269–275

Di Natale C, Paolesse R, D'arcangelo G et al (2005) Identification of schizophrenic patients by examination of body odor using gas chromatography-mass spectrometry and a cross-selective gas sensor array. Med Sci Monit: Int Med J Exp Clin Res 11:CR366

Galassetti PR, Novak B, Nemet D et al (2005) Breath ethanol and acetone as indicators of serum glucose levels: an initial report. Diabetes Technol Ther 7:115–123

Ghimenti S, Tabucchi S, Lomonaco T et al (2013) Monitoring breath during oral glucose tolerance tests. J Breath Res 7:017115

Greiter M, Keck L, Siegmund T et al (2010) Differences in exhaled gas profiles between patients with type 2 diabetes and healthy controls. Diabetes Technol Ther 12:455–463

Guo D, Zhang D, Li N et al (2010) A novel breath analysis system based on electronic olfaction. IEEE Trans Biomed Eng 57:2753–2763

Guo D, Zhang D, Zhang L et al (2012) Non-invasive blood glucose monitoring for diabetics by means of breath signal analysis. Sens Actuators B: Chem 173:106–113

Gutierrez-Osuna R, Gutierrez-Galvez A, Powar N (2003) Transient response analysis for temperature-modulated chemoresistors. Sens Actuators B: Chem 93:57–66

Hierlemann A, Gutierrez-Osuna R (2008) Higher-order chemical sensing. Chem Rev 108:563–613

Hosseini-Golgoo S, Hossein-Babaei F (2011) Assessing the diagnostic information in the response patterns of a temperature-modulated tin oxide gas sensor. Meas Sci Technol 22:035201

Kashwan K, Bhuyan M (2005) Robust electronic-nose system with temperature and humidity drift compensation for tea and spice flavour discrimination. In: 2005 Asian conference on sensors and the international conference on new techniques in pharmaceutical and biomedical research. IEEE, pp 154–158

Lee J, Ngo J, Blake D et al (2009) Improved predictive models for plasma glucose estimation from multi-linear regression analysis of exhaled volatile organic compounds. J Appl Physiol 107:155–160

Minh TDC, Blake DR, Galassetti PR (2012) The clinical potential of exhaled breath analysis for diabetes mellitus. Diabetes Res Clin Pract 97:195–205

Novak B, Blake D, Meinardi S et al (2007) Exhaled methyl nitrate as a noninvasive marker of hyperglycemia in type 1 diabetes. Proc Nat Acad Sci 104:15613–15618

Paredi P, Biernacki W, Invernizzi G et al (1999) Exhaled carbon monoxide levels elevated in diabetes and correlated with glucose concentration in blood: a new test for monitoring the disease? Chest 116:1007–1011

Phillips M, Cataneo RN, Cheema T et al (2004) Increased breath biomarkers of oxidative stress in diabetes mellitus. Clin Chim Acta 344:189–194

Ramachandran A, Moses A, Shetty S et al (2010) A new non-invasive technology to screen for dysglycaemia including diabetes. Diabetes Res Clin Pract 88:302–306

Righettoni M, Schmid A, Amann A et al (2013) Correlations between blood glucose and breath components from portable gas sensors and ptr-tof-ms. J Breath Res 7:037110

Risby TH, Solga S (2006) Current status of clinical breath analysis. Appl Phys B 85:421–426

Rohlfing CL, Wiedmeyer H-M, Little RR et al (2002) Defining the relationship between plasma glucose and HbA1c analysis of glucose profiles and HbA1c in the diabetes control and complications trial. Diabetes Care 25:275–278

Saraoğlu HM, Selvi AO, Ebeoğlu MA et al (2013) Electronic nose system based on quartz crystal microbalance sensor for blood glucose and HbA1c levels from exhaled breath odor. IEEE Sens J 13:4229–4235

Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14:199–222

Španěl P, Dryahina K, Rejšková A et al (2011) Breath acetone concentration; biological variability and the influence of diet. Physiol Meas 32:N23

Trincavelli M, Coradeschi S, Loutfi A et al (2010) Direct identification of bacteria in blood culture samples using an electronic nose. IEEE Trans Biomed Eng 57:2884–2890

Turner C (2011) Potential of breath and skin analysis for monitoring blood glucose concentration in diabetes. Expert Rev Mol Diagn 11:497–503

Turner C, Walton C, Hoashi S et al (2009) Breath acetone concentration decreases with blood glucose concentration in type i diabetes mellitus patients during hypoglycaemic clamps. J Breath Res 3:046004

Ueta I, Saito Y, Hosoe M et al (2009) Breath acetone analysis with miniaturized sample preparation device: in-needle preconcentration and subsequent determination by gas chromatography–mass spectroscopy. J Chromatogr B 877:2551–2556

Vashist SK (2012) Non-invasive glucose monitoring technology in diabetes management: a review. Anal Chim Acta 750:16–27

Wang C, Mbi A, Shepherd M (2010) A study on breath acetone in diabetic patients using a cavity ringdown breath analyzer: exploring correlations of breath acetone with blood glucose and glycohemoglobin a1c. IEEE Sens J 10:54–63

Wang P, Tan Y, Xie H et al (1997) A novel method for diabetes diagnosis based on electronic nose. Biosens Bioelectron 12:1031–1036

Wolfrum EJ, Meglen RM, Peterson D et al (2006) Metal oxide sensor arrays for the detection, differentiation, and quantification of volatile organic compounds at sub-parts-per-million concentration levels. Sens Actuators B: Chem 115:322–329

Yan K, Kou L, Zhang D (2017) Learning domain-invariant subspace using domain features and independence maximization. IEEE Trans Cybern

Yan K, Zhang D (2012) A novel breath analysis system for diabetes diagnosis. In: 2012 international conference on computerized healthcare. Hong Kong, China, pp 166–170

Yan K, Zhang D (2014) Sensor evaluation in a breath analysis system. In: 2014 international conference on medical biometrics (ICMB). IEEE, Shenzhen, pp 35–40

Yan K, Zhang D (2015) Improving the transfer ability of prediction models for electronic noses. Sens Actuators B: Chem 220:115–124

Yan K, Zhang D (2016a) Calibration transfer and drift compensation of e-noses via coupled task learning. Sens Actuators B: Chem 225:288–297

Yan K, Zhang D (2016b) Correcting instrumental variation and time-varying drift: a transfer learning approach with autoencoders. IEEE Trans Instrum Meas 65:2012–2022

Yu J-B, Byun H-G, So M-S et al (2005) Analysis of diabetic patient's breath with conducting polymer sensor array. Sens Actuators B: Chem 108:305–308

Zhang Q, Wang P, Li J et al (2000) Diagnosis of diabetes by image detection of breath using gas-sensitive laps. Biosens Bioelectron 15:249–256

Chapter 15
A Novel Medical E-Nose Signal Analysis System

Abstract Some components in human breath have been proven to be associated with certain diseases, such as diabetes and renal disease. The concentration of these components can also be linked to condition status, for example, blood glucose levels (BGLs). We called these components diseases biomarkers and seek ways to detect them in human breath by using a specially designed e-nose system plus advanced pattern recognition algorithms. In this chapter, a novel optimized medical e-nose system specially for disease diagnosis and BGL prediction is proposed. A large scaled breath dataset is collected by the proposed system. Experiments are conducted on the collected dataset and the experimental results have shown that the proposed system can well solve the problems of existed systems and the methods have effectively improved the classification accuracy.

Keywords E-nose · Chemical sensors · Breath analysis · Blood glucose level

15.1 Introduction

Electronic noses, or e-noses, are devices that "smell" or detect odor. An e-nose consists of a mechanism for chemical detection, such as an array of electronic sensors, and a mechanism for processing. Different sensors respond differently to odor samples and transmit the signal to the processing module. By analyzing the signals, the components or characteristics of the samples can be distinguished. E-noses are now attracting increasing interests from researchers in various areas because of the wide range of applications (Röck et al. 2008), including drunk driving testing, hazardous gas monitoring (Chou 1999), and air quality monitoring (Romain et al. 2010; Zampolli et al. 2004; Zhang et al. 2012). Table 15.1 lists some of the e-noses manufacturers and the application of their products. These commercial e-noses always provide some versatility in applications, such as coffee, wine, and fragrances identification for the sake of their marketing concerns. The versatility may in contrast improve the price and limit their performance since their

D. Zhang et al., *Breath Analysis for Medical Applications*,
DOI 10.1007/978-981-10-4322-2_15

Table 15.1 The developed products of e-noses

Manufacturer	Product	Application
The eNose Company, Netherland	AEONOSE (Aeonose 2017)	Medical
Airsense Analytics GmnH, Germany	PEN3 (PEN3 2017)	Food, wine, material, environment, medical
Alpha-Mos, France	HERACLES (HERACLES 2017)	Food, material, process management
Sensigent, USA	Cyranose 320 (Cyranose 2017)	medical, materials identification, food
Electronic Sensor Technology Inc., USA	Z-Nose (zNose 2017)	Investigation, food, environment, medical
Owlstone Inc., UK	LONESTAR (LONESTAR 2017)	Food, materials, industry

sensor selection must match broad applications. So, it is a better choice to design specific e-nose system for specific applications.

Medical application is another important area of e-noses. They have been used in medicine for the diagnosis of renal disease (Li et al. 2017; Lin et al. 2001), diabetes (Yu et al. 2005), lung cancer (Blatt et al. 2007; Van Hooren et al. 2016), arthritis (Brekelmans et al. 2016), and asthma (Dragonieri et al. 2007). Although gas chromatography (GC) has been proved to be effective in breath diagnosis (Nakhleh et al. 2016), using e-nose instead of GC (Phillips 1997) to analysis human breath, which are generally cheaper, faster, more portable, and easier to operate (Yan et al. 2014).

However, most of the existing trials (Di Natale et al. 2014) on breath diagnosis only focus on very limited kinds of diseases. One possible reason might be the design of commercial e-noses for broad applications rather than for breath analysis specifically or the specific designed devices only detect limited components (Broza et al. 2014). Moreover, some of the fluctuations in breath samples were not well taken into account, such as humidity and the ratio of alveolar air. Furthermore, the numbers of samples in the experiments of previous studies are small. We thus designed a novel medical e-nose device for breath analysis with optimized structure and sensor arrays for the specific application in order to extend the applications in medicine. A breath analysis dataset was then collected by this e-nose. Experiments were organized on the collected dataset to evaluate the performance of the system in disease diagnosis and Blood Glucose Level (BGL) classification. By analyzing the e-nose signals of human breath, it is possible for us to recognize the difference in contribution of the biomarkers so that certain diseases can be detected (Risby et al. 2006). For example, acetone (Righettoni et al. 2013; Ueta et al. 2009; Wang et al. 2010) as well as many other volatile organic compounds (VOCs) (Ghimenti et al. 2013) in breath are already proved to either have abnormal concentrations in diabetics or correlate with BGL. Moreover, in order to reduce the influence of device

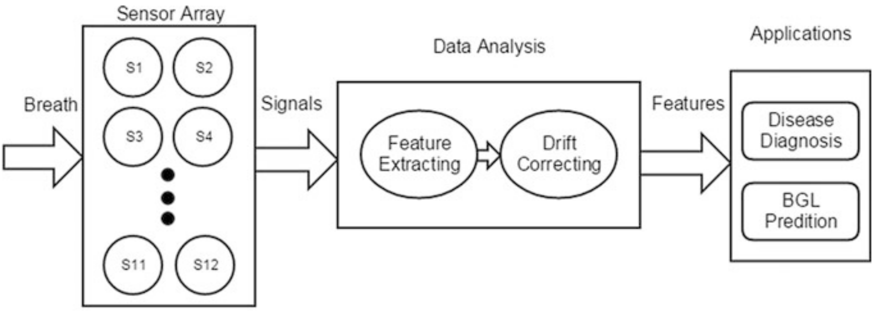


Fig. 15.1 Global working flow of the system

variety and time drift and make the system more robust, drift compensation methods will be introduced in the system before classification. Figure 15.1 gives a global working flow of the system.

15.2 Optimal System Design

15.2.1 Sensor Array Selection

Human breath is largely composed of oxygen, carbon dioxide, water vapor and nitric oxide, and the rest is less than 100 ppm (parts per million) of mixture with over 500 kinds of components, including carbon monoxide, methane, hydrogen, acetone and numerous volatile organize compounds (VOCs) (Cao et al. 2007; Yan et al. 2014). While the metabolic processes and partition from blood changed with some diseases, the types and concentrations of components in human breath will also change.

Nowadays, the concentration of some biomarkers in breath has been proved to be related with certain diseases. Selecting proper sensors that could respond to the components make it possible to analyze a person's breath odor and patients' health state. A few examples will further prove the points. The level of nitric oxide can be used as a diagnostic for asthma (Deykin et al. 2002). Patients with renal disease have higher concentration of ammonia (Davies et al. 1997). The concentration of VOCs, such as cyclododecatriene, benzoic acid, and benzene are much higher in lung cancer patients (Phillips et al. 2007). Table 15.2 lists the relationship between biomarkers and some typical diseases.

Among the breath biomarkers related with BGL, acetone is higher in concentration and easier for analysis. People with diabetes are insufficient of insulin secretion or cannot effectively use their own insulin. As a result, glucose is difficult to enter the cells, leading to the rising of the BGL. On the other hand, because the cells cannot get enough energy, the liver will increase lipolysis and produce more

Table 15.2 Breath biomarkers and related diseases

Diseases	Breath biomarkers
Diabetes (Risby et al. 2006)	Acetone
Renal disease (Davies et al. 1997)	Ammonia
Heart disease (Phillips et al. 2004)	Propane
Lung cancer (Phillips et al. 2007)	Benzene, 1, 1-oxybis-, 1, 1-biphenyl, 2, 2-diethyl, furan, 2, 5-dimethyl-, etc.
Breast cancer (Phillips et al. 2003)	Nonane, tridecane, 5-methyl, undecane, 3-methyl, etc.
Digestive system disease (Eisenmann et al. 2008)	Hydrogen

ketones. Other biomarkers in lower concentration including ethanol and methyl nitrite are also found in the breath of patients with diabetes (Turner 2011).

The characteristics of breath and the association of biomarker with diseases indicate the sensors in e-nose should be sensitive to the VOCs, carbon dioxide, humidity, and temperature. Thus, a sensor array with 11 sensors is optimized for the purpose of detecting one's breath. The sensor array includes six ordinary metal oxide semiconductor (MOS) sensors, three temperature modulated MOS sensors, a carbon dioxide sensor, and a temperature-humidity sensor. Specially, the temperature-humidity sensor has two input channels for temperature and humidity respectively. As a result, there are totally 12 input channels. The model, manufacturer and function of the sensors are listed in Table 15.3. The suffix "-TM" indicates that the sensor is a temperature modulated sensor.

Table 15.3 Summary of the sensor array

Channel	Model	Manufacturer	Function	Sensitivities (ppm)
1	TGS4161	Figaro Inc., Japan	CO_2	350–10000
2	TGS826	Figaro Inc., Japan	VOCs, NH_3	30–5000
3	QS01	FIS Inc., Japan	VOCs, H_2, CO	1–1000
4	TGS2610D	Figaro Inc., Japan	H_2, VOCs	500–10000
5	TGS822	Figaro Inc., Japan	VOCs, H_2, CO	50–5000
6	TGS2602-TM	Figaro Inc., Japan	VOCs, NH_3, H_2S	1–30
7	TGS2602	Figaro Inc., Japan	VOCs, NH_3, H_2S	1–30
8	TGS2600-TM	Figaro Inc., Japan	H_2, VOCs, CO	1–100
9	TGS2603	Figaro Inc., Japan	NH_3, H_2S	1–10
10	TGS2620-TM	Figaro Inc., Japan	VOCs, H_2	50–5000
11	HTG3515CH	Humirel Inc., France	Temperature	
12			Humidity	

15.2.2 Optimized System Structure

Figure 15.2 shows the frame of the proposed e-nose system, including mainly five modules: the gas route, the sensor arrays, the signal processing circuitry, the controlling circuitry, and the host computer.

The gas route of the device contains a vacuum pump and a gas chamber. Breath samples from outside was drawn and injected it into the gas chamber. With the purpose of finding a balance between portable and effectiveness, the gas chamber is designed to be a column-shaped metal container with the capacity of 100 ml. Sensors are embedded on the six facets, so that gases can flow smoothly the gas concentration in the head space of each sensor is similar, and the size of the chamber can be kept relatively small, as shown in Fig. 15.3.

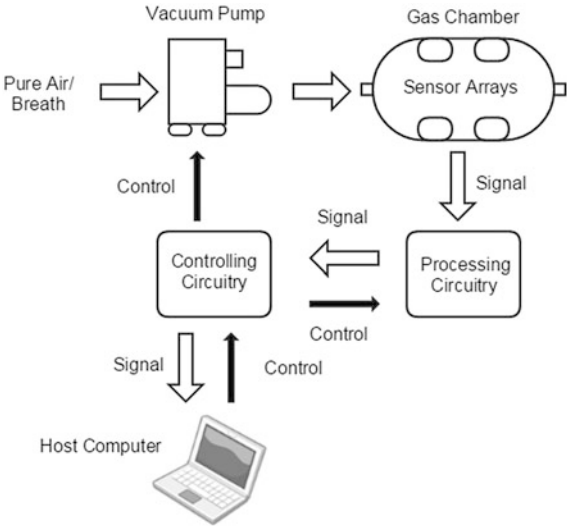
The resistances of the sensors change from R_0 to R_S when they are exposed to sampled gas. The output voltage is

$$V_{Out} = \frac{1}{2} V_{CC} \left(1 - \frac{R_S}{R_0} \right) \tag{15.1}$$

where V_{CC} is the transient voltage crossing the sensor and V_{Out} is the transient output voltages of the measurement circuits.

The origin sensors' signals are magnified by a signal processing circuit. The signal processing circuit also filters high-frequency noises. The controlling circuitry is used to control the pump and the processing circuitry, then digitize the processed signals and transmit them to a host computer for further processing. In order to remove the waste heat produced by the sensors, a fan is set next to the gas chamber. The fundamental parameters of the system are summarized in Table 15.4.

Fig. 15.2 The frame of the e-nose system with 5 modules: the gas route, the sensor arrays, the signal processing circuitry, the controlling circuitry, and the host computer



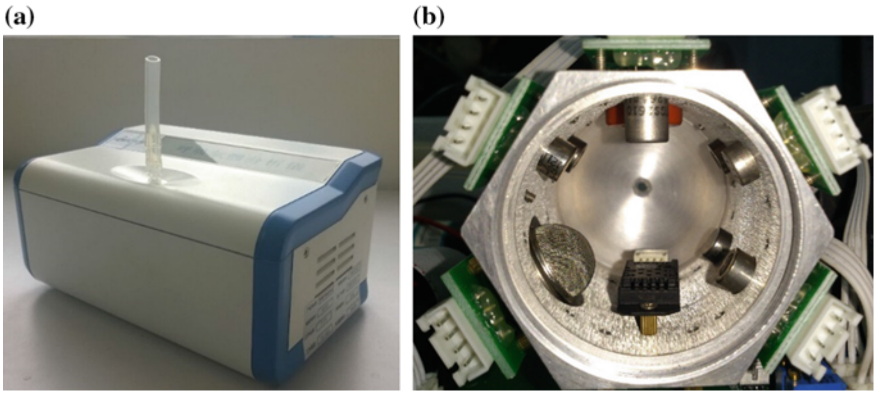


Fig. 15.3 Snapshot of the device **a** and gas chamber **b**. Sensors are embedded on its wall. Samples are injected to the chamber from the inlet hole at one end and pumped out through the outlet end

Table 15.4 Fundamental parameters of the system

Parameters	Specifications
Device size	22*15*11 cm
Working temperature	25 ± 10 °C
Gas chamber volume	600 mL
Injection rate	50 mL/s
Sampling frequency	8 Hz
Sampling time	144 s
Working voltage	5 V
Working voltage for temperature modulated sensors	3–7 V cycle
Resolution of the ADC	12 bit

15.2.3 Sampling Procedure

During working, sensors' temperature will graduate increase then remain constant. This phenomenon will result in a change in baseline response of the sensors. So, the device should be switched on for about 20 min till the baseline response shown on the host computer is stable. Besides, the devices would be calibrated every two weeks with 8 different kinds of standard gas samples to reduce the time drift. The standard gas samples include VOCs, H_2, CO_2 and NH_3 with two different concentrations respectively.

600-mL Tedlar gas bags supplied by Beijing Safelab Technology Ltd. are used to collect breath samples. Subjects are asked to take a deep breath and exhale into a gas bag through a disposable mouthpiece and an airtight box filled with disposable hygroscopic material to absorb the water vapor. The gas bags also allow those weak patients to give enough exhales with more than one exhale. Then the gas bag with the breath sample is plugged onto the preheated device connected to a computer. The measurement procedure is automatically controlled by the software in the computer. The measurement procedure is divided to four stages

(1) Baseline stage (0–1 s): The baseline values of the sensors are recorded for future data preprocessing.
(2) Injection stage (1–8 s): The pump is ON. Breath is drawn from the gas bag to the gas chamber at a constant speed. The sensors' signals start to respond to the injected breath.
(3) Reaction stage (9–64 s): The pump is OFF. The sensors continue reacting with the components in breath. The responses of the MOS sensors without temperature modulation (TM) reach their maximum values.
(4) Purge stage (64–144 s): The pump is ON again. Clean air is drawn into clean the gas chamber for 80 s. The sensors' responses gradually return to their baselines. After the responses remain stable in their baselines, the device is ready for the measurement of the next sample.

Figure 15.4 is the responses of the sensors in four stages. It can be seen that the responses keep stable in the baseline stage and start to change from injection stage. Since the injection speed is 50 mL/s, 350 mL of the sample gases will be pumped in. The first 250 mL will directly go through the chamber to remove air in the chamber and the rest 100 mL will stay in the chamber for reacting. Each sensor will reach its highest response value at least once within the reaction stage. Finally, during the purge stage, the sample gas will be cleaned by pure air and the responses return to baseline. The system does not require very high sampling frequency, which is chosen at 8 Hz. After the whole process, a digitized breath sample is represented by 12 response curves. Each response curve has 144 s × 8 Hz = 1152 data points. The samples will be then used for further analysis.

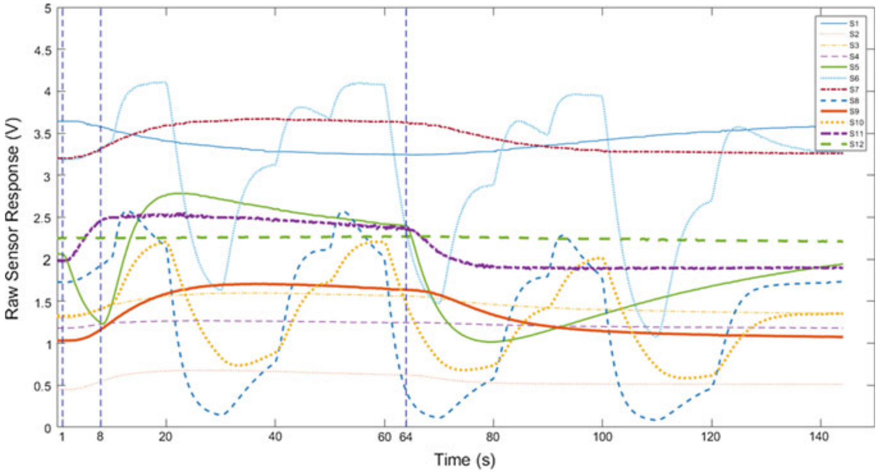


Fig. 15.4 The four stages of measurement procedure

15.3 Signal Analysis

15.3.1 Preprocessing

Before analyzing the data, original signals should be preprocessed so that to be transformed into standard samples. Four steps will be taken: faulty signal removing, de-noising, baseline manipulation, and normalization.

Faulty signal is a common problem in devices with sensors. In our system, causes of faulty signals are complicated, such as misoperation, bad connection, and device damage. In order to make the system more robust, these signals should be removed before analysis.

De-noising aims to remove the noise from the original signals by utilizing a low-pass filter to remove the noise since the signal is mainly interfered by high-frequency noise.

The purpose of baseline manipulation is to compensate baseline drift. The baseline value is the average response in the baseline stage of each sensor. The value is then subtracted from the whole response curve to eliminate the interference of background noise of the sensors (Marco et al. 2012). Assume that for each sensor transient of each sample, there are k dimensions, where $k = 1, \ldots, N_k$, and b dimensions in the baseline stage, where $b = 1, \ldots, N_b$. The response at time t_k is denoted as $R(t_k)$. The baseline response is $B(t_b)$. Then baseline manipulation can be computed as

$$R^B(t_k) = R(t_k) - \frac{1}{N_b} \sum_{t_b=1}^{N_b} B(t_b) \tag{15.2}$$

Normalization is used to compensate for sample-to-sample variations caused by analyte concentration. $R^B(t_k)$ is a sample after the baseline manipulation step, and the normalized response $R^{BN}(t_k)$ can be defined as

$$R^{BN}(t_k) = \frac{R^B(t_k)}{\max(R^B(t_k))} \tag{15.3}$$

15.3.2 Feature Extraction

To reduce the dimension of the origin features, principal component analysis (PCA) can be used. PCA projects high-dimensional data into a low-dimensional subspace while keeping most of the data variance.

Some low-dimensional geometric features can also be extracted from the origin response curves. Traditional features of gas sensors are their steady-state responses. When a gas sensor is used to sense a gas sample, its response will reach a steady state in a few minutes. The steady-state response has a close relationship with the

Table 15.5 Summary of the transient features

Feature		Description
Spatial	PCA	Reduce the dimension of the origin features with PCA method
	Magnitude	Down-sampled values of the curve's magnitude *M*
		The maximum magnitude
		Down-sampled values of the normalized magnitude *M*/max(*M*)
		Mean values of the magnitude
	Derivative	Down-sampled values of the curve's derivative *D*
		The maximum and minimum derivative
	2nd Derivative	The maximum and minimum 2nd derivative in both injection and purge stage
	Integral	The integral of the 5 intervals of the curve, the intervals are the same with the difference feature
	Slope	The slope of the 5 intervals of the curve, the intervals are the same with the difference feature
	Phase Feature	The phase feature is proposed in (Martinelli et al. 2003). First, the response is transformed to the phase space, which is spanned by its magnitude and derivative. Then, the phase features are defined as $\int_{M(t_i)}^{M(t_{i+1})} DdM$
Frequency	FFT	Fast fourier transformation
	Wavelet	Wavelet transformation

concentration of the measured gas. Therefore, the 9D feature vector contains the most important information needed for disease screening.

However, additional useful information is carried in the transient responses (Hierlemann et al. 2008). Transient responses are often related to the change of gas flow (injection/purge) or temperature (for TM sensors). The feature set includes magnitude, difference, derivative, second derivative, integral, slope and phase features, as well as features in frequency domain such like FFT and wavelet. The extracted features in both space domain and frequency domain are described in Table 15.5.

15.3.3 Drift Compensation

15.3.3.1 Sensor Drift

Drift is a comment problem and challenging task for chemical sensors, which may influent the robust of e-nose systems. The breath samples are collected by different devices in different time periods. On the one hand, because of the variations in of sensors and devices, the responses to the same signal source may not be different for different instruments. On the other, for a same device or same sensor, the stability also changes over time.

Additionally, studies by Phillips et al. (2000) and Klaassen et al. (2015) also proved that there is relationship with age and breath biomarkers. So, the influence of age should also be regard as a drift factor.

Widely used methods including algorithms based on variable standardization (Feudale et al. 2002; Zhang et al. 2011) and component correction PCA (CC-PCA) (Artursson et al. 2000) method. Moreover, Yan et al. proposed a Drift Correction Autoencoder (DCAE) (Yan et al. 2016).

Most of these methods require a set of predefined gas samples collected with each device and in each time period as transfer samples to provide mapping information between the source and the target domains. Nevertheless, collecting transfer samples may be a difficult job if there are not convenient predefined gases or thee operators are not professional e-nose users. For this situation, as an unsupervised domain adaption method is proposed to correct the drifts with unlabeled data, the optimized Maximum Independence Domain Adaptation (MIDA) method will be used in the system for drift compensation.

15.3.3.2 Optimized MIDA

Like transfer learning, domain adaption (DA) aims to solve the problem of transferring knowledge between domains with different distribution. Maximum Independence Domain Adaptation (MIDA) (Yan et al. 2017) was proposed to find this latent subspace in which the samples and their domain features are maximally independent in the sense of Hilbert-Schmidt independence criterion (HSIC) (Gretton et al. 2005).

HSIC measures the dependence between two sample sets $X, Y \in \mathbf{R}^{d \times n}$

$$\mathrm{HSIC} = (\mathrm{n}-1)^{-2} tr\left(K_x H K_y H\right), H = I - n^{-1} 1_n 1_n^T \in \mathbf{R}^{n \times n} \tag{15.4}$$

We first define the domain features to describe the background information: the device label, the acquisition time, and the age. Suppose there are n_{dev} devices, the domain feature is then $\mathbf{d} \in \mathbf{R}^{3n_{dev}}$, and

$$d_q = \begin{cases} 1, & q = 3p - 2 \\ t, & q = 3p - 1 \\ age, & q = 3p \\ 0, & \text{otherwise.} \end{cases} \tag{15.5}$$

Suppose $\mathrm{X} \in \mathbf{R}^{m \times n}$ is the matrix of n samples containing both the training and the test samples. More importantly, we do not have to explicitly differentiate which domain a sample. A linear or nonlinear mapping function Φ can be used to map X to a new space. According to the kernel trick, the inner product of $\Phi(X)$ can be represented by the kernel matrix $K_x = \Phi(X)^{\mathrm{T}}\Phi(X)$. Then, a projection matrix $\widetilde{W}$ is applied to project $\Phi(X)$ to a subspace with dimension h, leading to the projected samples $Z = \Phi(X)\widetilde{W} \in \mathbf{R}^{h \times n}$. To express each projection direction as a linear combination of all samples in the space, $\widetilde{W} = \Phi(X)^T W$, $W \in \mathbf{R}^{n \times h}$ is the projection matrix to be actually learned. Thus, the projected samples are

$$Z = \Phi(X)\Phi(X)^T W = K_x W \tag{15.6}$$

The kernel matrix $K_z = K_x W W^T K_x$. Set the matrix of background feature as $D \in \mathbf{R}^{n \times m_d}$, m_d is the dimension of background feature. The linear kernel $K_d = DD^T$. Omitted the scaling factor in HSIC, the expression to be minimized is:

$$tr(K_z H K_d H) = tr(K_x W W^T K_x H K_d H) \tag{15.7}$$

In domain adaption problems, the other goal is to preserve important properties of data, such as the variance, by maximizing the trace of the covariance matrix of the project samples.

$$\begin{aligned} \operatorname{cov}(Z) &= \operatorname{cov}(K_x W) \\ &= \frac{1}{n}\left(K_x W - \frac{1}{n} 1_n 1_n^T K_x W\right)^T \left(K_x W - \frac{1}{n} 1_n 1_n^T K_x W\right) \\ &= W^T K_x H K_x W \end{aligned} \tag{15.8}$$

Thus, the learning problem then becomes

$$\begin{aligned} &\max\nolimits_W \; -tr\left(W^T K_x H K_d H K_x W\right) + \mu tr\left(W^T K_x H K_x W\right) \\ &\quad s.t.\; W^T W = I \end{aligned} \tag{15.9}$$

Using the Lagrangian multiplier method, we can find that W is the eigenvectors of $K_x(-HK_d H + \mu H)K_x$ corresponding to the h largest eigenvalues.

15.4 Experiments

15.4.1 Breath Dataset

To evaluate the performance of our device, a large-scaled breath dataset is collected. We cooperated with Guangzhou Hospital of Traditional Chinese Medicine and collected data from inpatient volunteers. Two devices with same model are used for data collection. Patients are asked to rinse their mouth with medical mouthwash and not to use fragrance. The devices are placed in a well-ventilated room without the interruption of medical alcohol or odor of traditional Chinese

Table 15.6 Number of samples in each class

Class	Number
Healthy	1291
Diabetes	491
Kidney disease	398
Cardiopathy	537
Lung disease	376
Breast disease	527
Gastritis	241

medicine. For each sample, we first collected the patient's breath and recorded the signals. The diagnosis was then given by an authoritative doctor as the classification labels. Moreover, some biochemical indicators are also collected, such as blood glucose, blood pressure, and blood lipid. Finally, in this dataset, there are totally over 10000 samples of 47 classes, including 1491 healthy samples and samples of 46 different kinds of diseases. In this chapter, a subset of healthy and six kinds of diseases were selected for experiments, including breast disease, cardiopathy, diabetes, lung disease, kidney disease and gastritis. Table 15.6 shows the number of each class used in selected subset.

All the samples are collected from hospitals in Guangzhou. However, since most of the healthy samples are provided by medically examined young people while disease samples are from elder patients, it is difficult to make age-matched subsets, which is a shortage of this dataset. Operations will be taken to reduce the infection of age.

15.4.2 Disease Diagnosis

To check the performance of the system, six binary-classification tasks were performed to detect samples with one of the diseases from the healthy ones.

For each class, the first 50 samples collected by the first device are chosen as the labeled training sets and the rest are test samples. Logistic regression method was adopted as the classifier after drift compensation optimized MIDA. Sequential Forward Selection (SFS) method is used to optimal the features. SFS method is a greedy strategy. In each iteration, one feature is selected from all features that could achieve a best classification accuracy together with the features already selected. Figure 15.5 shows the results of forward selection in different disease diagnosis tasks.

In Table 15.7, we conclude the best combination of features and sensors selected for each task. It can be find that Wavelet, MaxMag, Slope and Phase features contribute most in all the tasks.

MeanMag feature and Integral feature are also discriminating in detecting cardiopathy, lung disease, and gastritis. Derivative feature only shows its importance in tasks of breast disease and gastritis. While other features did not improve the

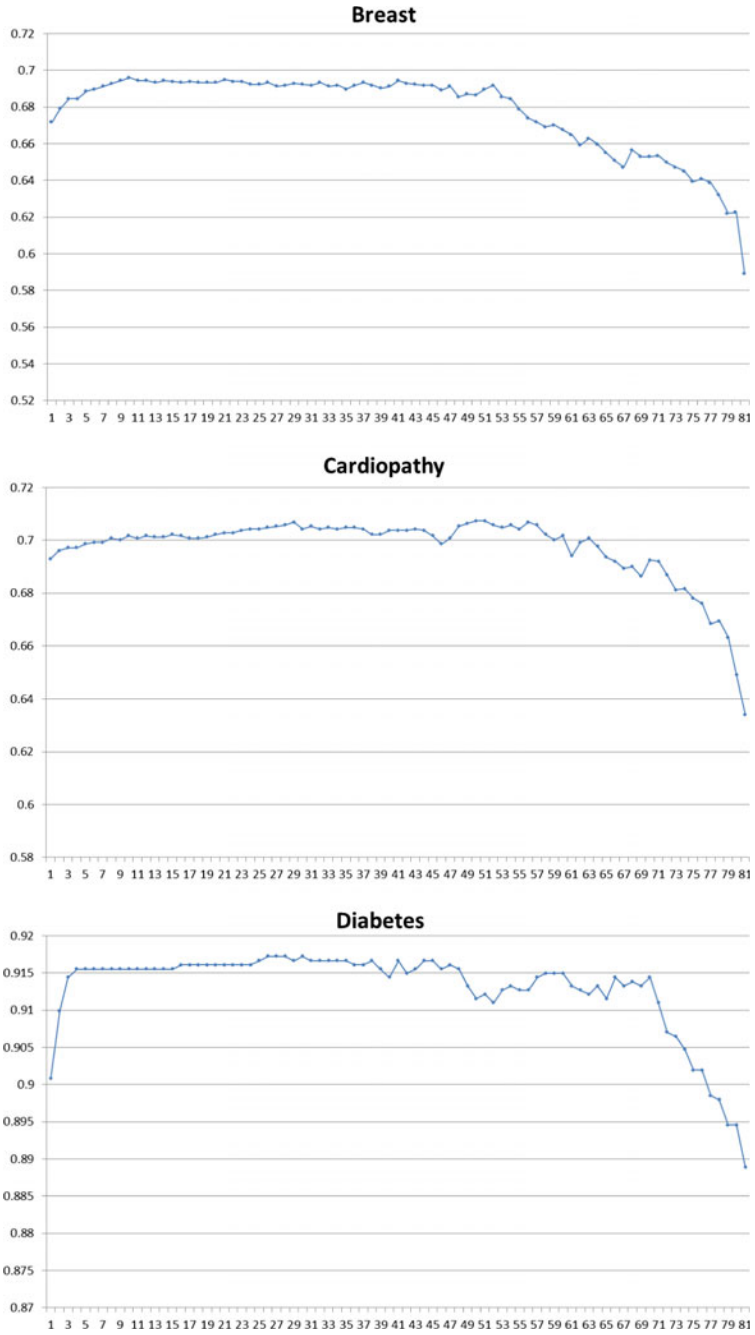


Fig. 15.5 Forward selection result of six binary-classification tasks. For each graph, the horizontal axis is the number of features used and the vertical axis is the classification result

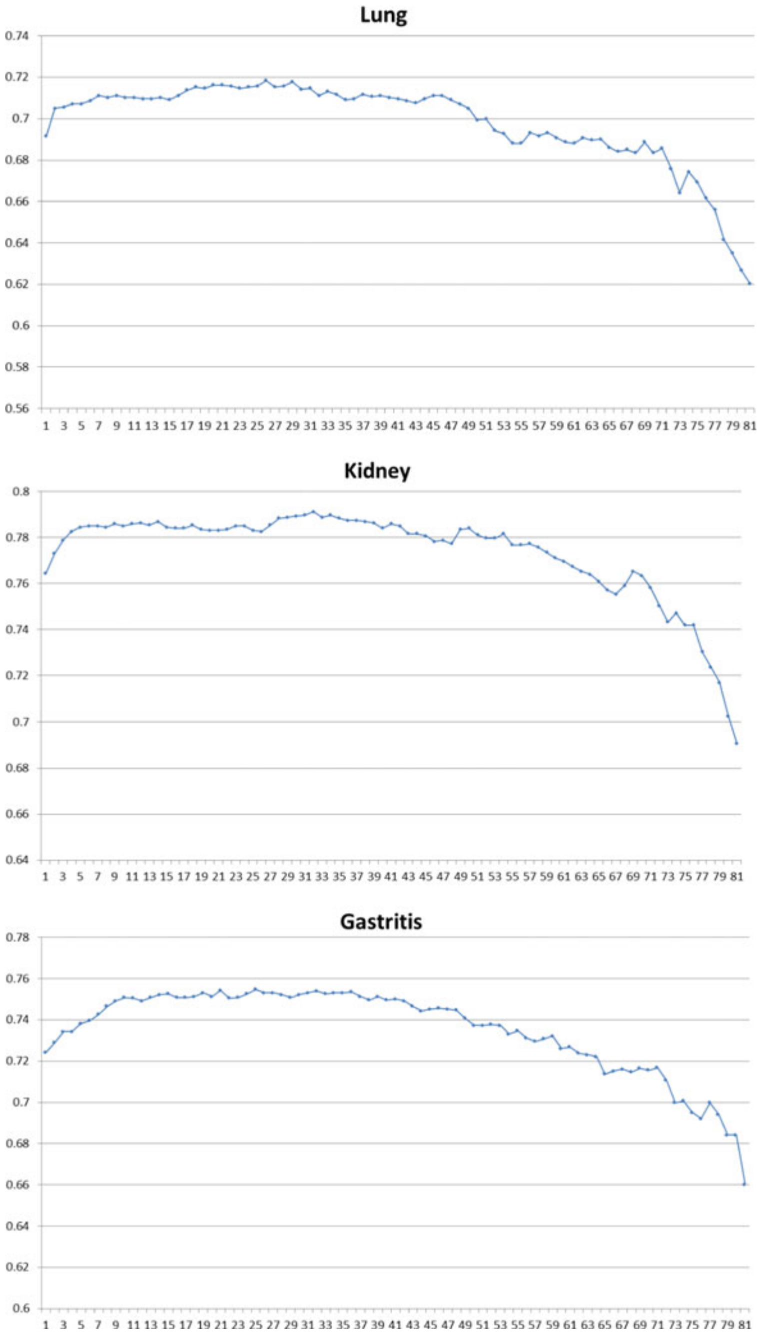


Fig. 15.5 (continued)

Table 15.7 Selected features and sensors and the sensitivity (SEN), specificity (SPE) and Accuracy (ACC) for each task

Task	Features and sensors		SEN	SPE	ACC
Diabetes	Wavelet of TGS2602	Wavelet of TGS2610D	0.8815	0.9495	0.9155
	Phase of TGS2602-TM	MaxMag of TGS826			
Kidney disease	Wavelet of TGS2602	Wavelet of TGS2600-TM	0.7002	0.8698	0.7850
	Slope of TGS2602-TM	Slope of TGS2620-TM			
	Phase of TGS826	Phase of TGS2610D			
Cardiopathy	Wavelet of TGS822	Integral of TGS826	0.7433	0.7125	0.7279
	MeanMag of TGS2603	Slope of TGS822			
	Integral of TGS2602	Slope of TGS826			
	MaxMag of TGS2603	Phase of TGS826			
	Phase of TGS2610D	Integral of TGS2610D			
Lung disease	Wavelet of QS01	MeanMag of QS01	0.7117	0.7209	0.7163
	Slope of TGS2603	Slope of QS01			
	Integral of TGS2602	Integral of TGS826			
	Phase of TGS2620-TM	Slope of TGS826			
	MaxMag of TGS2602	Integral of TGS2610D			
	MeanMag of TGS2610D	MeanMag of TGS2603			
	MaxMag of TGS2603	MeanMag of TGS2602-TM			
	Integral of QS01	MeanMag of TGS2602			
	MaxMag of TGS2602-TM	MaxMag of TGS826			
	Phase of TGS826	Phase of QS01			
	Phase of TGS2602				
Breast disease	Wavelet of TGS826	MaxMag of TGS822	0.6321	0.7599	0.6960
	MaxMag of TGS2602	Derivative of TGS2610D			
	Derivative of TGS2620-TM	MaxMag of TGS2603			
	Phase of TGS2600-TM	MeanMag of TGS2600-TM			
	MeanMag of TGS2602-TM	Phase of TGS2620-TM			
Gastritis	Wavelet of TGS822	Slope of TGS2600-TM	0.6436	0.8582	0.7509
	Integral of TGS2603	MaxMag of TGS2620-TM			
	Phase of TGS2602-TM	Integral of QS01			
	Wavelet of TGS2620-TM	Derivative of TGS2610D			
	MaxMag of TGS2602-TM	Slope of TGS2603			

performance of the system. On the other hand, the sensors those contribute most in each task meet the relationship between diseases and breath biomarkers listed in Tables 15.2 and 15.3.

15.4.3 BGL Classification

BGL prediction is another application of the system. In the collected dataset, the blood glucose levels are given by blood glucose meters in hospital. Since the blood glucose meters may already have a ±10% to ±25% of error, the final error rate will be further accumulated if we use these samples for regression. As a result, instead of regression experiments, we group the samples into different classes based on the BGL ranges and performed the BGL classification experiments on the datasets.

According to Chinese diabetes control criterion (Shang 2004), the dataset is divided into four parts based on BGL. The BGL range and sample number of each class are listed in Table 15.8. Because of the detection error of the meters, samples within the ±0.2 mmol/L of the thresholds are not selected as learning samples to improve the robust of the classification methods. We use a Random Forest (RF) method for the triplet-classification task.

The classification result and optimal features can be seen in Table 15.9. It can be seen that Magnitude features (MaxMag, MeanMag, and DownSampleMag), Slop features and Integral features are the most important features for BGL classification while the most useful sensors include TGS2602-TM, TGS2602, TGS826, TGS822, QS01, and TGS2610D.

Table 15.8 Number of samples in each class of BGL

Class	BGL (mmol/L)	Number
Normal	Lower than 6.1	1851
Impaired glucose tolerance	6.1–7.11	168
Hyperglycemia	Higher than 7.11	241

Table 15.9 Selected sensors and features for BGL classification

Features and sensors		Accuracy
MaxMag of TGS2602-TM	MaxMag of TGS2602	0.7778
MeanMag of TGS2602-TM	DownSample of TGS826	
DownSample of QS01	DownSample of TGS2610D	
DownSample of TGS822	Slope of TGS826	
Slope of QS01	Integral of QS01	

15.5 Conclusions

This chapter gave out a novel medical e-nose system that specified on disease diagnosis and BGL prediction. The scientific basis, structure, optimizing strategies, sensor arrays, sampling procedure, and signal preprocessing methods were introduced, as well as a large-scaled medical dataset collected by the system. In order to better correct the drifts, an optimized domain adaption method was adopted in the system. Experiments were taken on the new collected datasets to evaluate the performance of both the system and the methods.

The experimental results showed that better accuracy can be achieved by optimize the combination of features and sensors for different tasks. Wavelet, MaxMag, Slope and Phase features are the most significant in most of the disease diagnosis tasks while different sensors contribute differently based on the relationship of diseases and biomarkers. The BGL classification tests also produced a satisfactory output. However, it is still possible to further improve the performance and extent the applications. Multi-feature and multi-classification methods will be mainly investigated in future work. Neural networks will also be introduced to the system to further discover the relationship between the signals and human states.

References

Artursson T, Eklöv T, Lundström I et al (2000) Drift correction for gas sensors using multivariate methods. J Chemom 14:711–723

Blatt R, Bonarini A, Calabro E et al. (2007) Lung cancer identification by an electronic nose based on an array of mos sensors. In: 2007 International Joint Conference on Neural Networks, IJCNN 2007. IEEE, pp 1423–1428

Brekelmans MP, Fens N, Brinkman P et al (2016) Smelling the diagnosis: The electronic nose as diagnostic tool in inflammatory arthritis: a case-reference study. PloS one 11:e0151715

Broza YY, Zuri L, Haick H (2014) Combined volatolomics for monitoring of human body chemistry. Sci Rep 4:4611

Cao W, Duan Y (2007) Current status of methods and techniques for breath analysis. Crit Rev Anal Chem 37:3–13

Chou J (1999) Hazardous gas monitors: a practical guide to selection, operation, and applications. McGraw-Hill Professional Publishing

Davies S, Spanel P, Smith D (1997) Quantitative analysis of ammonia on the breath of patients in end-stage renal failure. Kidney Int 52:223–228

Deykin A, Massaro AF, Drazen JM et al (2002) Exhaled nitric oxide as a diagnostic test for asthma: Online versus offline techniques and effect of flow rate. Am J Respir Crit Care Med 165:1597–1601

Di Natale C, Paolesse R, Martinelli E et al (2014) Solid-state gas sensors for breath analysis: a review. Anal Chim Acta 824:1–17

Dragonieri S, Schot R, Mertens BJ et al (2007) An electronic nose in the discrimination of patients with asthma and controls. J Allergy Clin Immunol 120:856–862

Eisenmann A, Amann A, Said M et al (2008) Implementation and interpretation of hydrogen breath tests. J Breath Res 2:046002

Feudale RN, Woody NA, Tan H et al (2002) Transfer of multivariate calibration models: a review. Chemom Intell Lab Syst 64:181–192

Ghimenti S, Tabucchi S, Lomonaco T et al (2013) Monitoring breath during oral glucose tolerance tests. J Breath Res 7:017115

Gretton A, Bousquet O, Smola A et al. (2005) Measuring statistical dependence with hilbert-schmidt norms. In: Algorithmic learning theory. Springer, p 63–77

Hierlemann A, Gutierrez-Osuna R (2008) Higher-order chemical sensing. Chem Rev 108:563–613

Klaassen EM, Van De Kant KD, Jöbsis Q et al (2015) Exhaled biomarkers and gene expression at preschool age improve asthma prediction at 6 years of age. Am J Respir Crit Care Med 191:201–207

Li J, Zhang D, Li Y et al (2017) Joint similar and specific learning for diabetes mellitus and impaired glucose regulation detection. Inf Sci 384:191–204

Lin Y-J, Guo H-R, Chang Y-H et al (2001) Application of the electronic nose for uremia diagnosis. Sens Actuators B: Chem 76:177–180

Marco S, Gutiérrez-Gálvez A (2012) Signal and data processing for machine olfaction and chemical sensing: a review. Sens J IEEE 12:3189–3214

Martinelli E, Falconi C, D'amico A et al (2003) Feature extraction of chemical sensors in phase space. Sens Actuators B: Chem 95:132–139

Nakhleh MK, Amal H, Jeries R et al (2016) Diagnosis and classification of 17 diseases from 1404 subjects via pattern analysis of exhaled molecules. ACS nano

Phillips M (1997) Method for the collection and assay of volatile organic compounds in breath. Anal Biochem 247:272–278

Phillips M, Altorki N, Austin JH et al (2007) Prediction of lung cancer using volatile biomarkers in breath1. Cancer Biomark 3:95–109

Phillips M, Boehmer JP, Cataneo RN et al (2004) Heart allograft rejection: Detection with breath alkanes in low levels (the hardball study). J Heart Lung Transpl 23:701–708

Phillips M, Cataneo RN, Ditkoff BA et al (2003) Volatile markers of breast cancer in the breath. Breast J 9:184–191

Phillips M, Cataneo RN, Greenberg J et al (2000) Effect of age on the breath methylated alkane contour, a display of apparent new markers of oxidative stress. J Lab Clin Med 136:243–249

Righettoni M, Schmid A, Amann A et al (2013) Correlations between blood glucose and breath components from portable gas sensors and ptr-tof-ms. J Breath Res 7:037110

Risby TH, Solga S (2006) Current status of clinical breath analysis. Appl Phys B 85:421–426

Röck F, Barsan N, Weimar U (2008) Electronic nose: current status and future trends. Chem Rev 108:705–725

Romain A-C, Nicolas J (2010) Long term stability of metal oxide-based gas sensors for e-nose environmental applications: an overview. Sens Actuators B: Chem 146:502–506

Turner C (2011) Potential of breath and skin analysis for monitoring blood glucose concentration in diabetes. Expert Rev Mol Diagn 11:497–503

Ueta I, Saito Y, Hosoe M et al (2009) Breath acetone analysis with miniaturized sample preparation device: In-needle preconcentration and subsequent determination by gas chromatography–mass spectroscopy. J Chromatogr B 877:2551–2556

Van Hooren MR, Leunis N, Brandsma DS et al (2016) Differentiating head and neck carcinoma from lung carcinoma with an electronic nose: a proof of concept study. Eur Arch Otorhinolaryngol 273:3897–3903

Wang C, Mbi A, Shepherd M (2010) A study on breath acetone in diabetic patients using a cavity ringdown breath analyzer: exploring correlations of breath acetone with blood glucose and glycohemoglobin a1c. Sens J IEEE 10:54–63

Yan K, Kou L, Zhang D (2017) Learning domain-invariant subspace using domain features and independence maximization. IEEE Trans Cybern

Yan K, Zhang D (2016) Correcting instrumental variation and time-varying drift: a transfer learning approach with autoencoders. IEEE Trans Instrum Meas 65:2012–2022

Yan K, Zhang D, Wu D et al (2014) Design of a breath analysis system for diabetes screening and blood glucose level prediction. IEEE Trans Biomed Eng 61:2787–2795

Yu J-B, Byun H-G, So M-S et al (2005) Analysis of diabetic patient's breath with conducting polymer sensor array. Sens Actuators B: Chem 108:305–308

Zampolli S, Elmi I, Ahmed F et al (2004) An electronic nose based on solid state sensor arrays for low-cost indoor air quality monitoring applications. Sens Actuators B: Chem 101:39–46

Zhang L, Tian F, Kadri C et al (2011) On-line sensor calibration transfer among electronic nose instruments for monitoring volatile organic chemicals in indoor air quality. Sens Actuators B: Chem 160:899–909

Zhang L, Tian F, Nie H et al (2012) Classification of multiple indoor air contaminants by an electronic nose and a hybrid support vector machine. Sens Actuators B: Chem 174:114–125

Aeonose (2017) Aeonose and Aeolus bring tail wind. Available online: http://www.enose.nl/products/aeonose/ (2017). Access on 30 Jan 2017

PEN3 (2017) Portable Electronic Nose | AIRSENSE analytics. Available online: http://www.airsense.com/en/products/portable-electronic-nose/ (2017). Accessed on 30 Jan 2017

HERACLES (2017) HERACLES Electronic Nose, instrument for sensory analysis. http://www.alpha-mos.com/analytical-instruments/heracles-electronic-nose.php (2017). Accessed on 30 Jan 2017

Cyranose (2017) Cyranose Electronic Nose. Available online: http://www.sensigent.com/products/cyranose.html (2017). Accessed on 30 Jan 2017

zNose (2017) COMPUTER INTEGRATED zNose® Model 4600. Available online: http://www.estcal.com/product/computer-integrated-znoser (2017). Accessed on 30 Jan 2017

LONESTAR (2017) Lonestar Gas Analyzer. Available online: http://www.owlstonenanotech.com/lonestar (2017). Accessed on 30 Jan 2017

Shang D (2004) New concept of practical diabetes prevention. Anhui Science & Technology Publishing House

Chapter 16
Book Review and Future Work

Some components in human breath have been proven to be associated with certain diseases and the concentration of these components is linked to disease status. Breath signal contains important information about health status. Recently, breath signal diagnosis has attracted increasing research interest. Many kinds of breath signal acquisition systems and breath signal processing methods have been reported. However, there are still a lot of challenging works to be done, for example, how to acquire breath signal in a fast, accurate, and informative way, how to preprocess the breath signal to rule out the outliers and increase the quality of the signal, and how to extract efficient features and find our proper classifiers for breath diagnosis. This book focuses on these challenging issues. We first introduce novel breath signal acquisition systems equipped with multiple breath sensors. In order to collect samples effectively, we developed a sample acquisition system with sensor fusion technology. To detect the drift of breath signals, in this book, we provided optimized preprocessing frameworks with corresponding learning classification and regression models. To represent breath signals completely, this book discovered different types of breath signal features, such as spatial feature, frequency feature, deep learning feature, etc. Moreover, this book also provided many effective algorithms for breath signal classification and recognition, such as curve-fitting models and sparse representation classification.

In this chapter, we first recapitulate the contents of this book in Sect. 16.l. Then, Sect. 16.2 discusses the future of breath analysis for medical applications.

16.1 Introduction

The first chapter of this book sets the scene for this book. It first introduces the background that stimulates this research work. The motivation for the focus of the work is then explained, highlighting the importance of the breath analysis used in disease diagnosis, of the development of breath analysis device, and of the design

D. Zhang et al., *Breath Analysis for Medical Applications*,
DOI 10.1007/978-981-10-4322-2_16

of specific pattern recognition algorithm for breath analysis. This is followed by a statement of the objective of the research, a brief summary of the work, and a general outline of the overall structure of the present study.

Chapter 2 discusses some of the key issues in breath analysis and reviews some previous research work in the areas which are particularly relevant to the present study. Following a brief introductory overview of the field, the chapter first presents the development of breath analysis. Traditional approaches like GC which have been used to analyze the compounds of breath and identify several diseases are then described. This is followed by a detailed introduction of current major approaches, e-noses, for breath analysis. The final section gives a short summary of the chapter.

Chapter 3 proposes a novel system that is special for breath analysis. We selected chemical sensors that are sensitive to the biomarkers and compositions in human breath, developed the system, and introduced the odor signal preprocessing and classification method. To evaluate the system performance, we captured breath samples from healthy persons and patients known to be afflicted with diabetes, renal disease, and airway inflammation respectively and conducted experiments on medical treatment evaluation and disease identification. The results show that the system is not only able to distinguish between breath samples from subjects suffering from various diseases or conditions (diabetes, renal disease, and airway inflammation) and breath samples from healthy subjects, but in the case of renal failure is also helpful in evaluating the efficacy of hemodialysis (treatment for renal failure).

In Chap. 4, we propose a linear discriminant analysis (LDA)—based sensor selection technique (LDASS) which chooses an optimal configuration of sensors for a particular application from a whole set of available sensors. The proposed method finds the direction *w* via the LDA such that when data are projected onto this direction, the samples from two classes are as separate as possible. It is found that after projection, the difference of means of the two distinct sample classes can be expressed as the linear combination of the responses of all the sensors in the system, and *w* can be regarded as the weight vectors for these sensors which indicate the contribution weight of each sensor. Accordingly, it is possible to determine which sensor has a greater contribution in classifying the two classes. A series of experiments on different databases show that the proposed method outperforms other sensor selection techniques, such as the sequential forward selection (SFS) and genetic algorithm (GA) in recognition accuracy and processing time. This technique is not only applicable for breath analysis, but also useful in the general applications of e-noses.

In Chap. 5, we focus on the evaluation of sensor performance instead of particular sensor selection techniques. First, a breath acquisition system for diabetes diagnosis with 16 sensors is described. Based on this system, several methods are proposed to evaluate the importance, unique discriminant information and redundancy of each sensor. They are based on the results of exhaustive sensor selection. These methods are made convenient to observe and draw intuitive conclusions. They are applied to the breath acquisition system and some useful discoveries about the sensors in the system are made accordingly.

In Chap. 6, the transfer ability of prediction models is improved in two simple yet effective steps. First, windowed piecewise direct standardization (WPDS) is used to standardize the slave device, i.e., to transform the variables from the slave device to match the master one. Then, data from the master device are used to develop prediction models with a novel strategy named standardization error-based model improvement (SEMI). Finally, the standardized slave data can be predicted by the models with a better accuracy. The proposed WPDS is a generalization of the widely used PDS algorithm. To evaluate the algorithms, three e-noses specialized for breath analysis are adopted to collect a dataset, which contains pure chemicals and breath samples. Experiments show that WPDS outperforms previous methods in the sense of standardization error and prediction accuracy; SEMI consistently enhances the accuracy of the master model applied to standardized slave data.

In Chap. 7, we introduce transfer-sample-based multitask learning (TMTL) to simultaneously address two problems in e-nose signals: instrumental variation and time-varying drift. Data collected with each device or in each time period define a domain. Transfer samples measured in every domain are used to share knowledge across domains. TMTL reduces the influence of drift in the target domains by aligning the transfer samples at the model level. Two paradigms, parallel and serial transfer, are designed to reflect different relationships between domains, which are dependent on the cause of drift. A dynamic model strategy is proposed to predict samples with known acquisition time and to handle noise in transfer samples. Classification and regression experiments on three real-world datasets confirm the efficacy of the proposed methods. They achieve good accuracies compared with traditional feature-level drift correction algorithms and typical labeled-sample-based MTL methods, with few transfer samples needed. TMTL is a practical algorithm framework which can greatly enhance the robustness of sensor systems with complex drift.

In Chap. 8, we propose drift correction autoencoder (DCAE) to deal with instrumental variation and time-varying drift of e-noses. DCAE learns to model and correct these influential factors explicitly with the help of transfer samples. It generates drift-corrected and discriminative representation of the original data, which can then be applied to various prediction algorithms. Experimental results show that DCAE outperforms typical drift correction algorithms and autoencoder-based transfer learning methods. In particular, it is better than TMTL in the last chapter in datasets with complex drift, at the cost of longer training time and more hyper-parameters.

Chapter 9 proposes Maximum Independence Domain Adaptation (MIDA) for unsupervised drift correction. MIDA borrows the definition of domain features in the previous chapters and learns features which have maximal independence with them, so as to reduce the inter-domain discrepancy in distributions. A feature augmentation strategy is designed so that the learned subspace is background-specific. Semi-supervised MIDA (SMIDA) extends MIDA by exploiting the label information. The proposed algorithms are flexible and fast. The effectiveness of our approaches is verified by experiments on synthetic datasets and three real-world ones on sensors and measurement.

In Chap. 10, we study the classical support vector machine recursive feature elimination (SVM-RFE) algorithm and improve it by incorporating a correlation bias reduction (CBR) strategy into the feature elimination procedure. Experiments are conducted on a synthetic dataset and two breath analysis datasets. Large and comprehensive sets of transient features are extracted from the sensor responses. The classification accuracy with feature selection proves the efficacy of the proposed SVM-RFE + CBR. It outperforms the original SVM-RFE and other typical algorithms. An ensemble method is further studied to improve the stability of the proposed method. By statistically analyzing the features' rankings, some knowledge is obtained, which can guide future design of e-noses and feature extraction algorithms.

Chapter 11 proposes a Sparse Representation-based Classification (SRC) method for breath sample identification. The sparse representation expresses an input signal as the linear combination of a small number of the training signals, which are from the same category as the input signal. The selection of a proper set of training signals in representation, therefore, gives us useful cues for classification. Two experiments were conducted to evaluate the proposed method. The first one was to distinguish diabetes samples from healthy ones. The second one aimed to classify these diseased samples into different groups, each standing for one blood glucose level. To illustrate the robustness of this method, two different feature sets, namely, geometry features and principle components were employed. Experimental results show that the proposed SRC outperforms other common methods, such as Support Vector Machine (SVM) and *K*-Nearest Neighbor (KNN), irrespective of the features selected.

In Chap. 12, we introduce a breath analysis system to measure acetone in human breath, and therefore to evaluate the blood glucose levels of diabetics. The system structure, breath collection method, and signal preprocessing method are introduced. To enhance the system performance, we use a novel classification approach, i.e., Sparse Representation-based Classification (SRC), to classify diabetics' breath samples into different blood glucose levels. Experimental results show that coupling with SRC, the system is able to classify these levels with satisfactory accuracy.

Chapter 13 investigates the potential of breath signals analysis as a way for blood glucose monitoring. We employ a specially designed chemical sensor system to collect and analyze breath samples of diabetic patients. Blood glucose values provided by blood test are collected simultaneously to evaluate the prediction results. To obtain an effective classification results, we apply a novel regression technique, SVOR, to classify the diabetes samples into four ordinal groups marked with "well controlled", "somewhat controlled", "poorly controlled", and "not controlled", respectively. The experimental results show that the accuracy to classify the diabetes samples can be up to 68.66%. The current prediction correct rates are not quite high, but the results are promising because it provides a possibility of noninvasive blood glucose measurement and monitoring.

In Chap. 14, we describe the design of a novel optimized breath analysis system for this purpose. The system uses carefully selected chemical sensors to detect biomarkers in breath. Common interferential factors, including humidity and the ratio of alveolar air in breath, are compensated or handled in the algorithm. Considering the inter-subject variance of the components in breath, we design a feature augmentation strategy to learn subject-specific prediction models to improve the accuracy of BGL prediction. 295 breath samples from healthy subjects and 279 samples from diabetic subjects were collected to evaluate the performance of the system. The sensitivity and specificity of diabetes screening are 91.51% and 90.77%, respectively. The mean relative absolute error for BGL prediction is 20.6%. Experiments show that the system is effective and that the strategies adopted in the system can improve its accuracy. The system potentially provides a noninvasive and convenient method for diabetes screening and BGL monitoring as an adjunct to the standard criteria.

Chapter 15 introduced a new medical e-nose signal analysis system. A novel optimized medical e-nose system specially for disease diagnosis and BGL prediction is proposed. A large-scaled breath dataset is collected by the proposed system. Experiments are conducted on the collected dataset and the experimental results have shown that the proposed system can well solve the problems of existed systems and the methods have effectively improved the classification accuracy.

16.2 Future Work

Even though the current work for breath analysis has reached great achievement, some limitations of this research should be stated. First, the results of breath analysis are not stable enough to meet the clinic requirements. A major cause of this problem is that breath samples are easily volatile and they are susceptible to the life habits and dietary habits of the subjects, such as smoking and drinking. Second, the accuracy of breath analysis highly relies on the device. A device with highly accurate sensors is of benefit to the disease diagnosis. Based on the current technique, however, the performance of chemical sensors is limited. Third, not all kinds of diseases can be detected by breath analysis. Breath analysis only can detect several kinds of diseases. Therefore, this technique cannot take the place of blood test completely. Additionally, the relationship between breath and some diseases are not clear currently. Consequently, these disadvantages limit the development of breath analysis. To achieve a better result, a possible solution is to improve the breath analysis system and the pattern recognition methods.

The book has studied several issues in breath analysis and its medical applications. A few directions could extend current work and improve the accuracy and robustness of the system:

(1) Improvement of the breath analysis system: the chemical sensors are critical for breath analysis, as the accuracy of system relies on the development of sensor technology. The system in Chap. 3 employs the chemical sensors from the same company (FIGARO Engineering Inc.), where the sensors' working principles, performance parameters, and sensitivities are similar. Because of

this, the responses of sensors are not significantly discriminating, such that the diagnosis is not very remarkable for some diseases. As a result, discovering the sensors with various working principle and performance are valuable to enhancing the accuracy of data collection, which leads to the work in Chaps. 5 and 14.

(2) Extension of the database: the database with a larger number of samples can provide stronger statistical evidence in disease diagnosis and to provide a satisfactory diagnosis result. In our database, the samples of lung cancer and gastroenteritis are limited, which is one of the reasons that the diagnosis accuracies of them are not very satisfactory. Therefore, to increase the size of the database and to enhance its varieties are the main tasks of future work.

(3) Discovering the technique of health condition monitoring: currently, more and more sub-healthy condition and chronic illnesses are not likely to be detected by the formal sector health services until they result in complications or death. A new approach to health assessment and monitoring of health is urgently needed. The health condition monitoring technique can serve as a domestic tool for daily monitoring in this situation. When a subject is in sub-healthy or diseased condition, alarm is emitted automatically by the breath analysis system. This will be one of our future works.

(4) Discovering more medical treatment evaluation: our current study has introduced the hemodialysis treatment evaluation. In fact, much more interesting and useful treatment evaluation should be discovered, such as airway inflammation, asthma, and COPD. The common blood check is not effective in detecting whether these diseases are cured or not. However, the biomarkers of these diseases are easily detected by breath analysis. This would be a promising research direction.

Index

D. Zhang et al., *Breath Analysis for Medical Applications*,
DOI 10.1007/978-981-10-4322-2