MAXIMUM LIKELIHOOD DETECTION

The task involved in pattern detection or Recognition is that of making a decision about the unknown, yet constant, nature of an observation. In this context, an observation could be a single scalar or a multidimensional vector, and the nature of such observations is related to their classification according to some criteria specific to the application. For instance, in a face detection scenario, the observations are images, and the overall goal of a system is to select those containing human faces.

The maximum likelihood principle states that in a given object classification scenario, one should pick the object class for which the observation in question is most likely to happen. For instance, if we knew that in some place most summer days are sunny and most winter days are cloudy and we are asked to guess the season based solely on the fact that one of its days is sunny, our best guess should be that it is summer.

For the purpose of object detection, we use as much of any available information as we can about the underlying pattern structure of the observations. In most cases, all available information comes in the form of examples whose classification is known beforehand. We refer to them as the training set. Although the basic idea of the maximum likelihood principle is simple, the estimation of the probability distributions of the observations from the training set could be rather complex. Therefore, two different approaches have been taken to deal with this, parametric versus nonparametric probability estimators.

In this article we deal mainly with object detection in the context of computer vision and image understanding. However, maximum likelihood detection and many other approaches in pattern recognition have much wider scope and applicability in a number of different scenarios. In the following sections we describe a visual object detection setup, the aforementioned approaches for maximum likelihood detection, and an automatic face detection system based on nonparametric probability models.

VISUAL OBJECT DETECTION

Most object detection techniques by themselves are not invariant to rotation, scale, illumination changes, object pose, and so on. To overcome this limitation, the training examples are normalized in illumination, scale, rotation, and position before they are used in the learning procedure. The result of this learning procedure is a pattern recognition module capable of detecting the objects in question within a limited range of variation in scale, rotation, and illumination.

Let us assume that a test image is given and that we are to detect objects on it. In the detection procedure, a collection of rescaled and rotated images is computed from the test image according to the desired range of detection capability. Then, each subwindow within these images is normalized for illumination and tested with the aforementioned pattern recognition module to decide whether the desired object is in this subwindow. As a result, a new collection of images is obtained. In these, the pixel value of each position is the result of the pattern recognition module for the corresponding subwindow position. For example, each pixel value could be proportional to the likelihood that this subwindow contains the object. Further analysis of these images is carried out to produce a robust list of candidates of the object being detected. Figure 1 illustrates the case in which faces of different sizes are detected using a pattern recognition module that computes the likelihood that there is a face in a subwindow of size 17×14 pixels.

The overall performance of the detection system depends on the choice of scale factors, rotation angles, illumination normalization algorithms, and the size of the detection subwindow. The narrower these ranges are set, the more consistent the patterns that are fed to the detection module in the learning procedure. However, a larger search space is also required in the detection procedure to cover a similar range of detection capability.

Different techniques can be used in the recognition module (1,2). The approaches of particular interest here are those based on the Bayes decision rule (3). These approaches take each subwindow and try to estimate the probability that it belongs to each of the object classes in question. Then, using the value of the probability as a confidence level, the class with highest probability is selected to describe the object in the subwindow. These approaches are known as probabilistic reasoning techniques and include both maximum likelihood and maximum a posteriori detection setups.

BAYES DECISION RULE

Assuming that the nature of the observations is well known and therefore that the conditional probability densities of the observations for each object class are given, then the Bayes Decision Rule yields the minimum error (4). This error, known as the Bayes error, is a measure of the class separability.

Let $\omega_1, \omega_2, \ldots, \omega_L$ be the object classes and O be the observation variable. Then, the a posteriori probability function of ω_i given O, is obtained using the Bayes Formula as

$$P(\omega_i \mid O) = \frac{p(O \mid \omega_i)P(\omega_i)}{p(O)}$$

where $p(O|\omega_i)$ is the conditional probability density of the observation and $p(\omega_i)$ is the a priori probability for the ith object class and $p(O)$ is the probability of the observation.

The Bayes decision rule states that we should pick the object class ω_i^* with maximum probability, given the observation. If we pick

$$\omega_i = \arg\max_{\omega_i}\{p(O \mid \omega_i)P(\omega_i)\}$$

for the observation in question, we obtain the maximum a posteriori (MAP) decision rule. However, in most cases, the a priori probabilities for the classes $p(\omega_i)$ are unknown, and therefore, for practical purposes, they are set equal. Then, the obtained rule

$$\omega_i^* = \arg\max_{\omega_i}\{p(O \mid \omega_i)\}$$

is known as the maximum likelihood (ML) decision rule.

PROBABILITY MODELS

In reality, the most serious limitation of the Bayes decision rule is the difficulty of estimating the probability distributions needed for its application. In most cases, the information or knowledge about the object classes is available in the form of examples, that is, a set of observations that have been classified beforehand, usually called the training set, are given, and the goal of the learning procedure is to find a discriminant function capable of classifying these observations and also others not available for training.

In this general approach, the training set is used to estimate the probability functions for each object class. The goal of the learning technique is determining the best set of parameters for the probability estimators. As is usual in data fitting problems, estimating the probability from a set of examples faces a number of issues, such as the completeness of the training set, the generalization properties of the models, the optimization criteria, etc.

Probability distributions are usually modeled with parametric functions, for instance, Gaussian mixture densities. Another approach, based on the assumption that the observations are of a discrete nature, is to model the probability functions using the statistical averages. We call the former parametric probability models and the latter nonparametric probability models. Once the probability functions are ob-

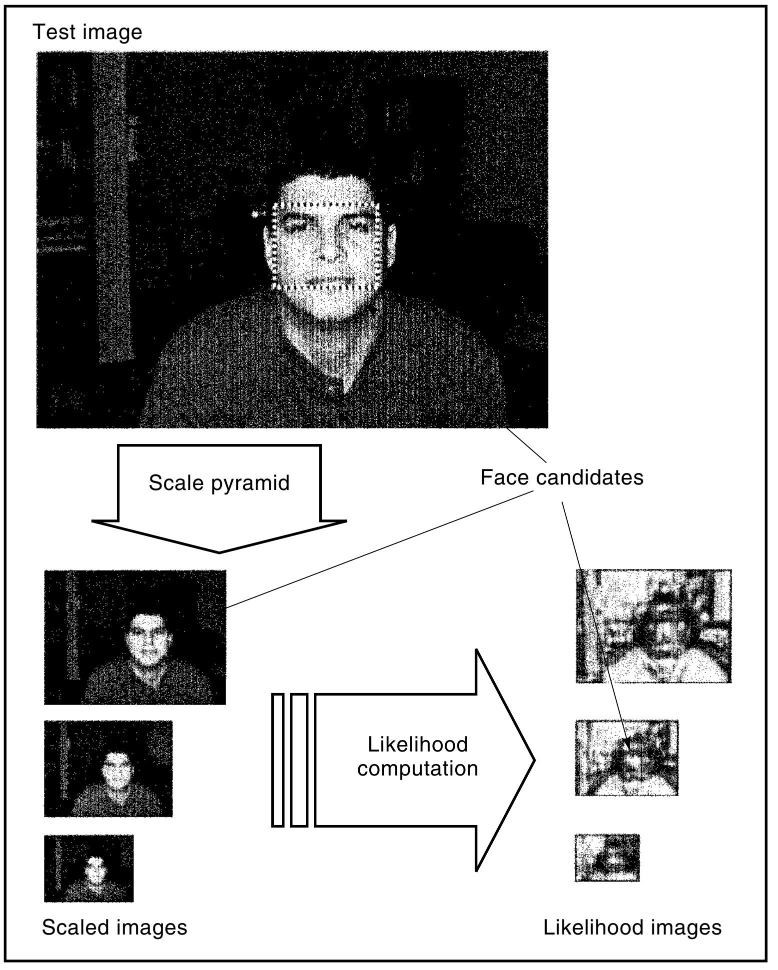

Figure 1. Scheme of a multiscale, maximum likelihood face detection setup. Each subwindow of the scaled version of the test image is tested, and the likelihood that it contains a face is displayed in the likelihood images from which the face candidates are obtained.

tained, they are used in a ML or MAP setup for object detection. In the following section we briefly describe an example of a parametric ML face detection system. However, in the rest of this article we concentrate our effort on a nonparametric ML face detection system.

Parametric Probability Models

In modeling probability functions or distributions of multidimensional random variables, one encounters a difficult issue. There is a compromise between the complexity of the model and the procedure used to fit the model to the given data. Extremely complex models would have to be used to consider all of the underlying dependency among all of the variables and to fit the model well to the training data.

The Karhunen–Loeve transform (5) is often used to reduce the dimensionality of the data and to overcome the limitation imposed by the dependency among the variables. Rather than estimating the probability densities on the original space, the observations are projected to the eigenspace in which the energy is packed to a subset of the components and the components are uncorrelated. Then, in this new space, the probability of the observation is often estimated using a Gaussian distribution or a mixture of Gaussian densities with diagonal covariance matrices (6).

Moghaddam and Pentland reported an example of a maximum likelihood detection system using this approach (7). In their work, the parameter estimation is carried out using the expectation-maximization (EM) algorithm.

INFORMATION-BASED MAXIMUM DISCRIMINATION

The detection process described in this article is carried out as a classification using the Bayes decision rule. We mainly compute the likelihood ratio of an observation using the probability models obtained from the learning procedure and compare it to a fixed threshold to make the decision. We use statistical averages to construct nonparametric probability models, and the learning procedure is turned into an optimization whose goal is to find the best model for the given training data. From information theory we borrow the concept of Kullback relative information and use it as the optimization criteria that measure the class separability of two probability models.

Let the observed image subwindow be the vector $\boldsymbol{X} \in \boldsymbol{I}^N$, where $\boldsymbol{I}$ is a discrete set of pixel values. Let $P(\boldsymbol{X})$ be the probability of the observation $\boldsymbol{X}$ given that we know that it belongs to the class of objects we want to detect, and let $M(\boldsymbol{X})$ be the probability of the observation $\boldsymbol{X}$, given that we know that it belongs to other classes.

We use the likelihood ratio $L(\boldsymbol{X}) = P(\boldsymbol{X})/M(\boldsymbol{X})$ to decide whether the observation $\boldsymbol{X}$ belongs to the object class in question by comparing it to a threshold value. Setting this threshold to 1 leads to the Bayes decision rule. However, different values are used depending on the desired correct-answer-to-false-alarm ratio of the detection system.

Kullback Relative Information

Kullback relative information, also known as Kullback divergence or cross-entropy, measures the "distance" between two probability functions, and therefore it measures the discriminatory power of the likelihood ratio of these probability functions under the Bayes decision rule (8,9).

The divergence of the probability function P with respect to the probability function M is defined as

$$H_{P\|M} = \sum_{\boldsymbol{X}} P(\boldsymbol{X}) \ln \frac{P(\boldsymbol{X})}{M(\boldsymbol{X})}$$

Although it does not satisfy triangular inequality, this divergence is a nonnegative measure of the difference between the two probability functions that equals zero only when they are identical. In our context, we use the Kullback divergence as the optimization criteria in our learning procedure. Basically, we set up a family of probability models and find the model that maximizes the divergence for the given training data.

Modified Markov Model

Dealing with probability models that take full advantage of the dependency of all of the variables is limited by the dimensionality of the problem. On the other hand, assuming complete independence of the variables makes the model rather useless. In between these extremes, we use a modified Markov model. This family of models is well suited for modeling our random processes and also easy to handle mathematically.

We compute the probability of the modified kth order Markov model as

$$P(\boldsymbol{X}) = \prod_{i=1,\ldots,T} P\left(X_{S_i} \mid X_{S_{i-1}}, \ldots, X_{S_{i-k}}\right)$$

where $\boldsymbol{S} = \{S_1, \ldots, S_T\}$ is a list of indices (e.g., each S_i denotes the pixel location), and the Kullback divergence between the probability functions $P(\boldsymbol{X})$ and $M(\boldsymbol{X})$ of such random processes as

$$H_{P\|M}(\boldsymbol{S}) = \sum_{i=1,\ldots,T} H_{P\|M}(X_{S_i} \mid X_{S_{i-1}}, \ldots, X_{S_{i}-k})$$

Information-Based Learning

The key idea behind this learning technique is to restate the learning problem as an optimization in which the goal is to find the list $\boldsymbol{S}^* = \{S_1^*, \ldots, S_T^*\}$ that maximizes the Kullback divergence $H_{P\|M}(\boldsymbol{S})$ for a given training set. It is clear that the computational requirements of such an optimization problem are prohibitive. However, we make some simplifications to find a practical solution to this problem.

First, we requantize the observation vector as part of the image preprocessing step so that each pixel has only a few possible values, for instance, four gray levels $X_i = \{0, 1, 2, 3\}$ for $i = 1, \ldots, N$. Then, using a first-order Markov model, the divergence of the two probability functions for a given list of indices $\boldsymbol{S} = \{S_1, \ldots, S_T\}$ is obtained from

$$H_{P\|M}(\boldsymbol{S}) = \sum_{i=1,\ldots,T} H_{P\|M}\left(X_{S_i} \mid X_{S_{i-1}}\right)$$

where

$$H_{P\|M}(X_j \mid X_k) = \sum_{X_j, X_k} P(X_j, X_k) \ln \frac{P(X_j \mid X_k)}{M(X_j \mid X_k)}$$

is the divergence of each pair of pixels within the image subwindow, and is obtained from the training set using histogram counts and statistical averages.

Then, we treat our optimization as a minimum-weight spanning-tree problem in which the goal is to find the sequence of pairs of pixels that maximizes the sum of $H_{P\|M}(\boldsymbol{S})$. Finally, we use a modified version of Kruskal's algorithm to obtain suboptimal results (10).

Once a solution is obtained, it is used to precompute a three-dimensional lookup table with the log likelihood ratio for fast implementation of the detection test. Given an image subwindow $\boldsymbol{X} \in \boldsymbol{I}^N$, the computation of its log likelihood is carried out as $\log L(\boldsymbol{X}) = \sum_{i=1,\ldots,T} L'[i][X_{S_i}][X_{S_i-1}]$, where

$$L'[i][X_{S_i}][X_{S_{i-1}}] = \log \frac{P(X_{S_i} \mid X_{S_{i-1}})}{M(X_{S_i} \mid X_{S_{i-1}})}$$

It is worth noting that such an implementation results in very fast, highly parallelizable algorithms for visual pattern detection. This is particularly important when we consider that the likelihood ratio is computed for each of the image subwindows obtained from the tested image.

FACE AND FACIAL FEATURE DETECTION AND TRACKING

We tested the previously described learning technique in the context of face and facial feature detection. Examples of faces were obtained from a collection of "mug shots" from the FERET database (11) using the locations of the outer eye corners as a reference to normalize the face size and position within the image subwindows. As negative examples, we also used a collection of images of a wide variety of scenes with no frontal-view faces.

We used the likelihood model obtained with the training set in a ML detection setup to locate face candidates in the test images. Several scaled and rotated images are obtained from the input image and tested with this face detection module according to the desired range of detection capability. In addition to locating the face candidates, the system further tests those candidates with likelihood models for the right and left eyes so that the algorithm can accurately locate these facial features. A detailed description of this implementation, testing procedure, error criteria, performance description, etc. can be obtained from Refs. (12,13).

A real-time, automatic face and facial feature tracking system was implemented on an SGI-ONYX with 12 R10000 processors and a SIRIUS video acquisition board. Real-time video is grabbed from a camera to the computer memory for processing and sent back out to a monitor with additional labeling information, such as the position of the face and the facial

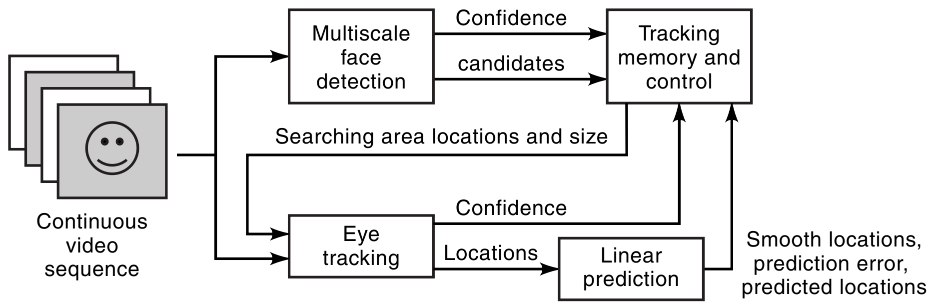

Figure 2. Block diagram of a face detection and eye tracking system. Both the initial face detection and the continuous eye tracking are implemented using maximum likelihood detection setups.

features. As illustrated in the block diagram in Fig. 2, the system handles continuous video sequences. Faces and their eyes are first detected in an upright frontal view using our ML setup. Then, the eyes are tracked accurately over the video sequence under face translation, rotation, and zooming by applying a similar ML detection setup. In the tracking setup, the predicted position of the eyes is used to limit the search for eye detection.

The system operates in two modes. In the detection mode, the system constantly carries out an exhaustive search to find faces and their outer eye corners. Set up to detect faces in a range of sizes (actually, the distance between the outer eye corners) between 100 and 400 pixels, this detection loop runs on 10 processors at about 3 frames per second. Once a face is successfully detected, that is, when the confidence level of the detection is above a fixed threshold, the system switches to the tracking mode.

In the tracking mode, the predicted positions of the outer eye corners are used to normalize the incoming video frames. A normalized image is obtained for each frame so that the tracked face lays in an upright position and with the appropriate size. Then, the locations of the eyes are continuously updated by applying the eye detector in these normalized images. The eye detection module is based on the likelihood models obtained with the aforementioned visual learning technique but at a much higher resolution than that used in face detection. As a result, there is no error accumulation or inaccuracy over long video sequences, and a wide range of rotation and zooming can be handled successfully. Whenever, the confidence level of the eye tracking falls below a predefined threshold, the system switches back to the detection mode, and this cycle starts over.

BIBLIOGRAPHY

1. R. Schalkoff, *Pattern Recognition: Statistical, Structural and Neural Approaches,* New York: Wiley, 1992.
2. K. Fukunaga, *Introduction to Statistical Pattern Recognition,* 2nd ed., New York: Academic Press, 1991.
3. H. Stark and J. Woods, *Probability, Random Processes, and Estimation Theory for Engineers,* Englewood Cliffs, NJ: Prentice-Hall, 1986.
4. L. Devroye, L. Gyërfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition,* New York: Springer-Verlag, 1996.
5. I. T. Jolliffe, *Principal Component Analysis,* New York: Springer-Verlag, 1986.
6. R. A. Redner and H. F. Walker, Mixture densities, maximum likelihood and the EM algorithm, *SIAM Rev.,* **26** (2): 195–239, 1984.
7. B. Moghaddam and A. Pentland, Probabilistic visual learning for object representation, *IEEE Trans. Pattern Anal. Mach. Intell.,* **19**: 696–710, 1997.
8. R. M. Gray, *Entropy and InformationTheory,* New York: Springer-Verlag, 1990.
9. J. N. Kapur and H. K. Kesavan, *The Generalized Maximum Entropy Principle,* Waterloo, Canada: Sandford Educational Press, 1987.
10. T. H. Cormen, C. E. Leirserson, and R. L. Rivest, *Introduction to Algorithms,* New York: McGraw-Hill, 1990.
11. P. J. Phillips et al., The FERET September 1996 database and evaluation procedure, *Proc. 1st Int. Conf. Audio Video-Based Biometric Person Authentication,* Crans-Montana, Switzerland, March 12–14, 1997.
12. A. Colmenarez and T.S. Huang, Maximum Likelihood Face Detection, *Int. Conf. Automatic Face Gesture Recognition,* Vermont, USA, 1996.
13. A. Colmenarez and T. S. Huang, Face detection with information-based maximum discrimination, *CVPR,* San Jose, Puerto Rico, 1997.

ANTONIO J. COLMENAREZ
THOMAS S. HUANG
University of Illinois at Urbana-Champaign