Improving Biometric Authentication using Score-Averaging and Error-Correction Output-Coding
DEA INFORMATIQUE Internship Report

Norman POH Hoon Thian

poh@dpt-info.u-strasbg.fr
[1]LSIIT Laboratory
(Laboratoire des Sciences de l'Image,
de l'Informatique et de la Télédétection)
in collaboration with
[2]IDIAP Institute
(Institut Dalle Molle d'Intelligence Artificielle Perceptive)

First Supervisor : [1]Prof. Jerzy Korczak
Field Supervisor : [2]Dr. Samy Bengio

[1]LSIIT, Pôle API, boulevard Sébastien Brant, F67400 ILLKIRCH Cedex, France.
[2]IDIAP, Rue du Simplon 4, Case Postale 592, CH-1920 Martigny, Switzerland.

July 2002

Abstract

In this study, two techniques that can improve the authentication process are examined: (i) averaging scores obtained from multiple biometric modalities of multiple samples and (ii) improving classification by class-relabelling. In the first technique, by using the average operator, both the theoretical and empirical results show that integrating as many samples and as many biometric sources as possible can improve the overall reliability of the system. This strategy is called the multi-sample multi-source approach. The second technique is inspired by the error-correction output-coding that is widely used in information theory. Both techniques were tested on real-life databases using state-of-the-art machine-learning algorithms.

I would like to thank both of my supervisors Dr. Samy Bengio and Prof. Jerzy Korczak for making this collaboration possible between the LSIIT laboratory of Université Louis Pasteur and the IDIAP institute.

I would also like to thank Sébastien Marcel and Fabien Cardinaux from the Vision Group of IDIAP who have helped to prepare the face features needed to carry out the experiments.

For Johnny Mariéthoz and Norbert Crettol, I owe them great assistance in making my life easier as a beginner to familiarise myself with the Linux/Unix environment and the software.

For the official paperwork, lodgement and the logistic issue, I would like to thank Nadine Rousseau for being ever-ready and efficient in dealing with the administration.

I wish to thank the following persons for fruitful discussions: Mohamed Faouzi Benzeghiba, Guillaume Lathoud and Andrew Morris.

For Darren Moore, I wish to thank him for proof-reading this document and providing brilliant suggestions to improve the clarity of presentation.

Finally, the base classifiers in this project are realised using the Torch software under the GNU licence. Therefore, I would like to acknowledge the main contributors of the Torch system[1]*: Ronan Collobert, Samy Bengio and Johnny Mariéthoz.*

[1] Torch is available for download at www.torch.ch

About the hosting institute

Introduction

Created in 1991 by the Dalle Molle Foundation for the Quality of Life, the Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP[2]), located in Martigny (Valais, Switzerland), is a not-for-profit research institute affiliated with the Swiss Federal Institute of Technology in Lausanne (EPFL) and the University of Geneva.

In 2001, IDIAP numbered an average of 40-45 collaborators, including permanent scientific staff, postdoctoral fellows, PhD students (around 18), system and development engineers, and short-term to medium-term visitors.

The activities carried out at IDIAP can be described as follows: research and development activities, participation in European and national research projects, collaborations with organizations and companies, and teaching and training activities. IDIAP's mission therefore consists in:

- Carrying out fundamental and applied research activities aiming at long and medium term industrial transfer.
- Teaching and training activities.

Research activities

The focus of IDIAP's activities is on the development of advanced (multimodal) natural input and output interfaces to a computer through speech and vision, as well on new ways to access multimedia documents.

The field of multimodal interaction covers a wide range of critical activities and applications, including recognition and interpretation of spoken, written and gestural language, particularly when used to interface with multimedia information systems. Other key subthemes include the biometric protection of information access (through speaker and/or face recognition and verification), and the structuring, retrieval and presentation of multimedia information.

The resulting multimodal interfaces are expected to represent a new direction for computing, providing people (including non-specialists) with access to complex information systems (e.g., incorporating multimedia content). Ultimately, these multimodal interfaces should flexibly accommodate a wide range of users, tasks, and environments for which any single mode may not suffice. The ideal interface should primarily be able to deal with more comprehensive and realistic forms of data, including mixed data types (i.e., data from different input modalities such as image and audio).

Although all the IDIAP research and development activities are structured in three groups (speech processing, computer vision, and machine learning) these activities can also be summarised as follow:

- **Spoken language input**: Covering speech signal processing and multilingual robust speech recognition. **Research issues** include: improved robustness, portability across new appli-

[2] http://www.idiap.ch

cations, language modeling, automatic adaptation (of acoustic and language models), confidence measures, out-of-vocabulary words, spontaneous speech, prosody, modeling dynamics.

- **Written language input**: Including document image analysis; OCR (printed and handwritten, off-line recognition); handwriting as computer interface (on-line recognition). **Research issues** include: analysis of documents with complex layout, recognition of degraded printed text, recognition of running handwriting.

- **Visual input**: Shape tracking (including lips tracking, face tracking); gesture recognition; facial expressions; images (e.g., sketches, signatures, photos) used as input. **Research issues** include: robustness of the algorithms; combination of colour, motion, texture, and shape in the analysis; more accurate model-based analysis; computational complexity.

- **Input (spoken, written, visual) analysis and understanding**, involving parsing and syntactic and semantic analysis and modeling. **Research issues** include: specification and formalism of unimodal and multimodal syntactical and semantic constraints, using these constraints into unimodal and multimodal input signal processing, merging modalities through multimodal "grammars".

- **Protecting information access**, involving: speaker verification, signature recognition, face recognition; bio-metric (multimodal) user authentication. **Research issues** include: increasing robustness of user authentication techniques, multimodal user authentication (mixture of experts, confidence-based weighting of the different media, etc).

- **Modality integration**, involving, e.g.: Speech and gestures, facial movement and speech recognition, facial movement and speech synthesis, and interface agents. **Research issues** include: merging of different (media) data streams, possibly non-synchronous and with different data rate, fusion of the different modalities (e.g., based on signal-to-noise ratio or confidence level estimation).

- **Mathematical methods**, including: Statistical modeling and statistical pattern classification, signal processing techniques, connectionist techniques, expert fusion, support vector machines.

More information about IDIAP and its recent activities can be referred to [14].

Contents

List of Figures

List of Tables

Chapter 1

State of the Art in Biometric Authentication

1.1 Introduction

Biometric authentication is the problem of verifying an identity claim using a person's behavioural and physiological characteristics. Biometric authentication is becoming an important alternative to traditional authentication methods such as keys ("something one has", i.e., by possession) or PIN numbers ("something one knows", i.e., by knowledge) because it is essentially "who one is", i.e., by biometric information. Therefore, it is not susceptible to misplacement, forgetfulness or reproduction. Examples of biometric sources [1] are fingerprint, face, voice, hand-geometry and retina scans.

However, biometric-based security systems (devices, algorithms, architectures) still have room for improvement, particularly in their accuracy, tolerance to various noisy environments and scalability as the number of individuals increases. The focus of this study is on minimising the noise inherent in biometric samples by using various methods, particularly score-averaging and Error-Correcting Output Coding.

Biometric data is often noisy because of deformable templates, corruption by environmental noise, variability over time and occlusion by the user's accessories. The higher the noise, the less reliable the biometric system becomes.

Advancements in biometrics show two emerging solutions: combining several biometric sources [13, 29, 30] and combining several samples of a single biometric source [17]. Combining several biometric sources can further be divided into a loosely coupled solution and a tightly coupled solution. A loosely coupled solution assumes very little or no interaction among the inputs. It integrates biometric data output of a relatively autonomous agent. An example of a loosely coupled system is the integration of audio and visual biometric data in an asynchronous manner. On the other hand, a tightly coupled solution assumes a strong interaction among the input measurements. It integrates biometric data at the sensor or representation level. A possible example of a tightly coupled system is the integration of audio and visual biometric data in a synchronous manner. Until now, to the author's knowledge, there is no literature taking such approach. In our opinion, combining several samples of a single biometric source can be considered a very tightly coupled solution because taking several life-scans of the same source of biometric data implies that the samples must be strongly correlated. Another category of solutions is to combine a biometric system with a non-biometric system.

Combining several biometric sources offers the advantage of relaxing the assumption of universality (the fact that each user should possess the biometric information), collectability (the extent to which the biometric information is measurable and adequately represented for the matching pur-

[1] The term "source" is used in this report to signify different types of biometrics. Another common term in literature is "modality".

pose), acceptability (the fact that each user agrees to have his/her biometric information scanned) and integrity (the degree of trustworthiness of the biometric system) of a target population in a given application.

Several studies have shown that a multi-source biometric system can improve the incompleteness of any single-model biometric system [29, 30]. In particular, Hong *et al.* [13] have proven both theoretically and empirically that integrating multiple biometric sources at score level and decision level can improve the overall system accuracy. In their work, the "AND" and "OR" decision rules are used. In our previous work [27], the "AND" rule was used to combine the decisions of face and voice biometric sources. Kittler *et al*[17]. have shown that combining several samples of a single biometric source can also improve the accuracy of the overall system. In a more recent work, by using a clipped assumption on the Gaussian score distribution, Kuncheva [18] has studied several combination strategies.

The aim of this study is to propose all possible means of improving the accuracy of biometric authentication system using different combination strategies. Inspired by the work of Kittler *et al.* [17] and Hong *et al.* [13], their two seemingly separate approaches, namely multiple biometric sources and multiple samples, are combined. It is called a multi-sample multi-source approach. Furthermore, approach that uses multiple classifers (also called "ensemble methods") is also examined. One of the ensemble methods called Error-Correction Output-Code (ECOC) is studied in details. By using multiple samples, multiple sources and multiple classifiers, with existing algorithms, it is hoped that the performance of biometric authentication system can be improved.

1.2 A generic biometric integration model

The foundamental problem of biometric authentication is a problem of binary classification given a biometric source. One can model such a classification as a function f in $y = f(\mathbf{x})$, where $\mathbf{x}$ is a given biometric source and y is a similarity measurement (will be defined later). If y is more than pre-defined threshold, the biometric source is accepted as belonging to the claimed identity. Otherwise, it is rejected. In this way, biometric authentication is a task of deciding if a biometric source belongs to a claimed identity or not. Improving a biometric authentication system in this sense is to minimise the errors (to be defined later) committed by the system. The lower the error, the better the system performance.

In this section, a biometric-independent framework (see Figure 1.1) is proposed. This framework Based on this framework, the process involving sensors, extractors, classifiers and supervisors is shown to exhibit a serial chain process.

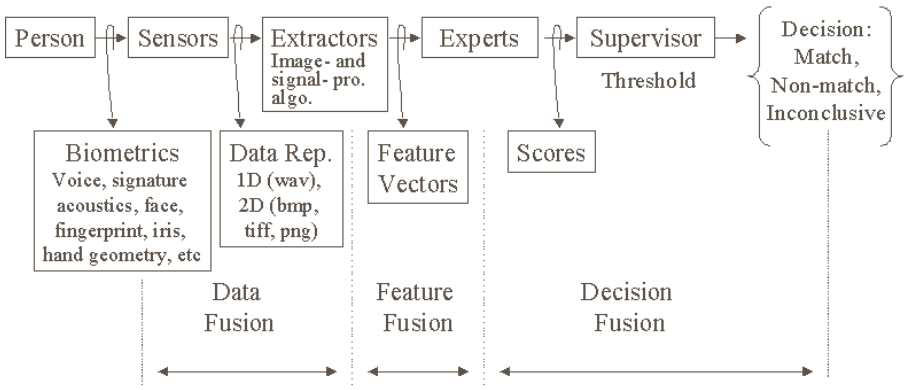


Figure 1.1: A generic biometric taxonomy and fusion scheme

In a biometric-independent framework (see Figure 1.1), a user's biometric data is captured using sensors. Examples of sensors are Charged Couple Device (CCD) cameras, Infrared-Red

(IR) cameras, fingerprint scanners and microphones. Each sensor has its own standard data representation. A set of operations, often founded on signal- and image-processing algorithms, constitute the building blocks of extractors. Extractors have two functions: to detect and to extract user-discriminant information. Each extractor produces its own type of vectors or feature vectors, also called templates in a more generic setting. Experts or classifiers are used to categorise these produced vectors. Classifiers are a set of pattern-matching algorithms, which may be learning-based (e.g. Multi-Layer Perceptron, Support Vector Machine, etc) or template-based (dynamic time warping, Euclidean distance, normalised correlation, etc). Classifiers' role is to map a vector to an associated identity. They do so with a certain degree of confidence commonly called a score or a confidence measure. It could be a scalar value or a vector when more information is supplied. A score could be interpreted as the estimated a posteriori probability that a given feature belongs to the claimed class label. When there are several classifiers, a supervisor merges different scores to obtain the final decision. If the final decision is a match, then the system accepts the identity claim. If the decision is a non-match, then the system rejects the identity claim. Finally, if the decision is inconclusive, a fall-back procedure should be activated. A detailed description of biometric framework and it's commonly used algorithms can be found in [28].

1.3 Quality measurement of a biometric authentication system

This section defines the probability of failure of a single-source biometric authentication system. In general, the core matching module of biometric system can be regarded as a function f that receives a vector feature $\mathbf{x}$ and outputs a score y: $y = f(\mathbf{x})$. The function f could be linear (e.g. Fisher discriminant analysis, Support Vector Machine with linear kernel) or non-linear (Multi-layer Perceptron, Support Vector Machines with non-linear kernel). y represents a measurement. It could be a score $\in [-1, +1]$, a confidence score (*a posteriori* probability) $\in [0, 1]$ or a distance metric $\in R^+$. As an example, the output value of an MLP having a single output neuron using a sigmoid activation function can be considered a posteriori probability.

An authentication system has to deal with two kinds of events: either the person claiming a given identity is the one who he claims to be (in which case, he is called a *client*), or he is not (in which case, he is called an *impostor*). Let the first event be w_2 and the second event be w_1. The decision taken by the system is governed by the score y. The probability that the system predicts a client given a score y is labeled as $p(w_2|y)$. In the similar way, the probability that the system predicts an impostor given a score y is labeled as $p(w_1|y)$.

The system may generally take two decisions: either *accept* the *client* or *reject* him and decide he is an *impostor*. Thus, the system may make two types of errors: *false acceptances* (FA), when the system accepts an *impostor*, and *false rejections* (FR), when the system rejects a *client*. In order to be independent on the specific dataset distribution, the performance of the system is often measured in terms of these two different errors, defined as follows:

$$\mathrm{FAR} = \frac{\text{number of FAs}}{\text{number of impostor accesses}} , \tag{1.1}$$

$$\mathrm{FRR} = \frac{\text{number of FRs}}{\text{number of client accesses}} . \tag{1.2}$$

A unique measure often used combines these two ratios into the so-called *Half Total Error Rate* (HTER) as follows:

$$\mathrm{HTER} = \frac{\mathrm{FAR} + \mathrm{FRR}}{2} . \tag{1.3}$$

Most verification systems output a score for each access. Selecting a threshold over which scores are considered genuine clients instead of impostors can greatly modify the relative performance of FAR and FRR. A typical threshold chosen is the one that reaches the *Equal Error Rate* (EER) where FAR=FRR on a separate validation set.

$p(w_2|y)$ defined earlier is in fact a probability distribution function (*pdf*) of client scores and $p(w_1|y)$ is the *pdf* of impostor scores. In practice, $p(w_2|y)$ and $p(w_1|y)$ can be estimated by using a histogram of client scores and impostor scores respectively.

The shaded area in Figure 1.2 then shows the mistakes (both false acceptance and false rejection errors) committed by the system at the threshold s. The bounded box in Figure 1.2 shows what the valid values of $p(w_1|y)$ and $p(w_2|y)$ which could have been otherwise normal if not bounded by the constraint that probability $\in [0, 1]$. This assumption was reported in the work of Kuncheva[18].

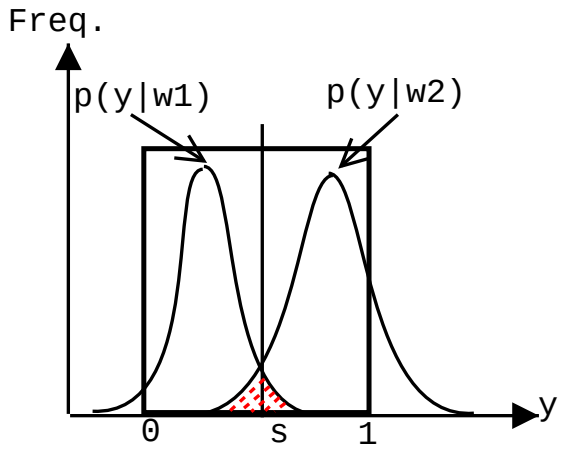


Figure 1.2: A schematic diagram of genuine and impostor distribution score

The probability that a system commits error given a threshold s, which is denoted as E_s, can be calculated using:

$$E_s = \int (P(\text{false rejection}) + P(\text{false acceptance}))\, dy \tag{1.4}$$

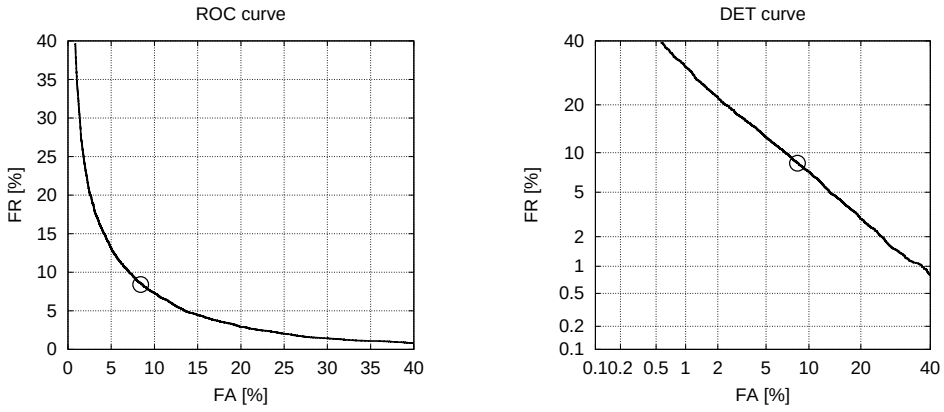
$$= \left(1 - \int_{-\infty}^{s} p(w_1|y)dy\right) + \left(\int_{s}^{+\infty} p(w_2|y)dy\right)$$

$$= FRR_s + FAR_s. \tag{1.5}$$

One seeks to minimise E_s such that $E_{min} = min_s(E_s)$. The optimum threshold s is at the point called Equal Error and E_{min} is called Equal Error Rate (EER). The assumption here is that the false acceptance and false rejection errors are considered equal. One can also define a cost function that give different weights to the false acceptance and the false rejection error. It is obvious that if the two distributions completely overlap each other, $E = 1$ and if they do not overlap at all, $E = 0$.

From Equation 1.5, it can be noted that the terms FRR_s and FAR_s correspond to Equations 1.2 and 1.1 but as a function of the decision threshold s. In another words, by moving the threshold s in Equation 1.4 from zero to one, one gets different values of FARs and FRRs. When pairs of FAR and FRR are plotted on a graph, one obtains *Receiver Operating Characteristic* (ROC) curve, which represents the FAR as a function of the FRR [37]. More recently, other researchers proposed the DET curve [23], a non-linear transformation of the ROC curve in order to make results easier to compare. The non-linearity is in fact a normal deviate, coming from the hypothesis that the scores of client accesses and impostor accesses follow a Gaussian distribution. If this hypothesis is true, the DET curve should be a line. Figure 1.3 shows examples of typical ROC and DET curves.

Note that in biometric applications, there are three categories of scores: genuine, "inter-template" (other clients) and impostors (also called "background database" [40]). There are also



(a) ROC curve (b) DET curve

Figure 1.3: Typical example of a ROC curve and its corresponding DET curve. The circle represents the threshold at equal error rate (EER), i.e. when FA=FR.

informed and uninformed impostors. In real-life hacking, impostors are informed, i.e., they possess a certain amount of information about the identity to be faked. Among these three major categories of scores, the genuine user scores are often the smallest data set.

1.4 Classification of biometric system models

Having studied the nature of chain process of biometric system and the error involved in such a system, one is led to study available types of biometric system models. In our opinion, biometric systems can be classified according to the number of samples per access and the number of biometric sources. The term "source" is used here to signify a particular class of biometric source such as face, voice and so on. This is to distinguish it from the term "model" to be introduced later to signify different architectures. A "sample" is a life-scan or shot of a biometric source. Using these two definitions, four biometric "models" are proposed. Figure 1.4(i) shows the typical serial process consisting of sensor, extractor and classifier. It is called a single-sample single-source (SSSS) biometric model. Figures 1.4(ii)-(iv) show a multi-sample single-source (MSSS), a single-sample multi-source (SSMS) and a multi-sample multi-source (MSMS) biometric model respectively. With this categorisation, Kittler *et al.*'s work [17] falls into MSSS model because several face samples are used during authentication. Hong *et al.*'s work [13] falls into SSMS model because they used a face sample and a fingerprint sample during authentication. Ross *et al.*'s work [30] that used face, fingerprint and hand geometry also falls into SSMS model.

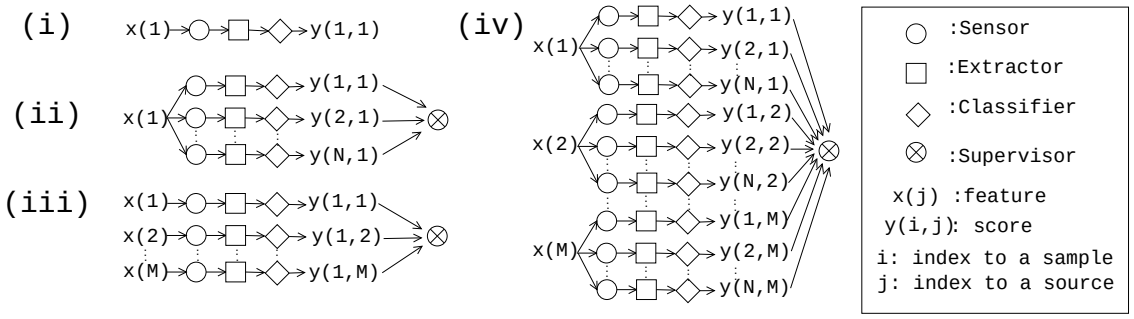


Figure 1.4: (i) Single-sample single-source, (ii) multi-sample single-source, (iii) single-sample multi-source and (iv) multi-sample multi-source biometric model.

In Figure 1.4, x(j) is the j-th biometric source (or modality) and j=1,. . .,M. y(i,j) is the score obtained from i-th sample of the j-th biometric source and i=1,. . .,N. A MSSS model differs from a SSSS model in that a MSSS model uses several samples (therefore several serial processes) originating from the same biometric source. Note that a SSSS model does not require any supervisor. On the other hand, a SSMS model differs from a SSSS model in that a SSMS model uses several independent biometric sources. A MSSS model is more tightly coupled than a SSMS model. This means that if a biometric source is corrupted by the same noise (for instance, a cut on finger, a soar throat, a sun-burnt face), a MSSS model will probably fail to verify the identity of the person. On the other hand, multi-source biometric models (SSMS and MSMS) will probably be more robust against this kind of noise because their sources are not corrupted by the same noise.

1.5 A Survey on improving biometric system

1.5.1 Principals of improving a biometric system

Section 1.2 described a generic biometric authentication system as a concatenation process of sensors, extractors, classifiers and optionally a supervisor (in the case of a multi-source biometric system, see Figure 1.5(a)). If there is a way to measure the reliability [2] of these components (denote as R_i), then it can be shown that the overall reliability of the system, R_s, is the product of all the reliability of it's components[28], i.e., $R_s = \prod_i R_i$. It can be deduced that $\forall_i (R_s < R_i)$, which means that the reliability of the overall system will be worse. This is because each component adds errors to the system. Unfortunately, there is no way one can measure the reliability of the system. In a separate study done by Jain and Pankanti [15], they used the terms *information limited behaviour*, *representation limited behaviour* and *invariance limited behaviour* to describe the errors of the three components, which is equivalent to $E_i = 1 - R_i$, for each E_i describing "limitation" (easier to be understood as error) of sensors, extractors and classifiers, respectively. However, one knows that if one can improve any one of the components in this serial process, one can improve the biometric system. This is the principal idea towards improvement of any biometric system.

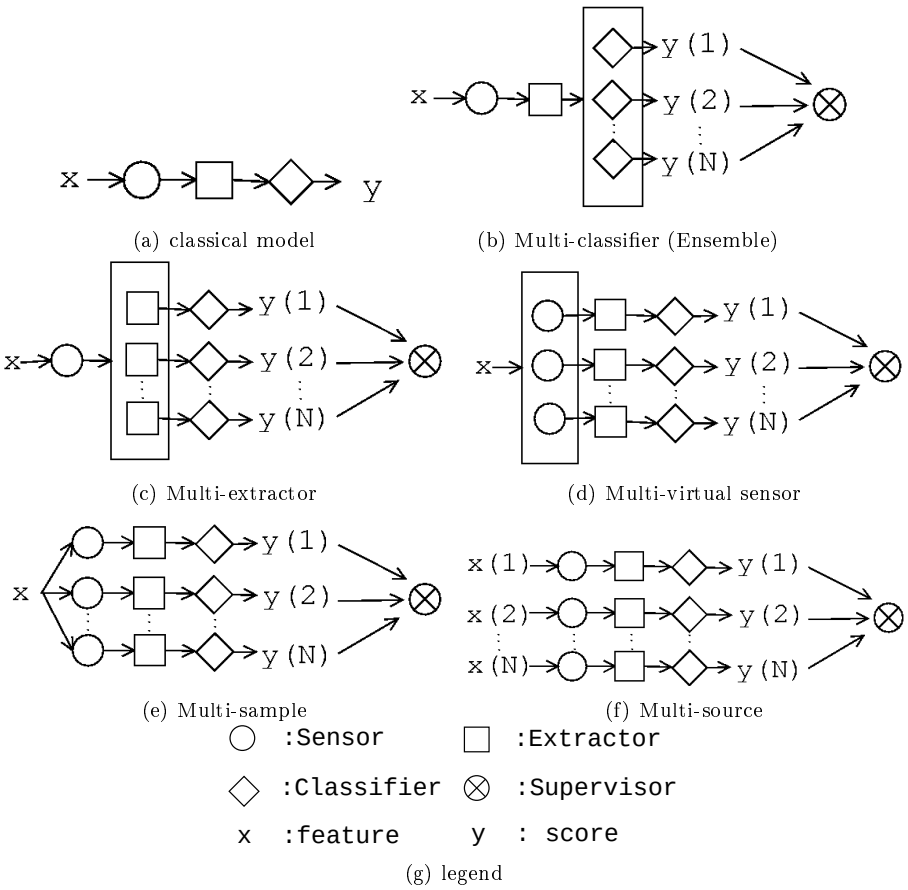
Figure 1.5 shows several possible models to improve a biometric authentication system. Figure 1.5(a) is a classical model. One can also improve the system by using multiple classifiers (also called ensemble method; see Figure 1.5(b)), multiple extractors (see Figure 1.5(c)), multiple "virtual" sensors (see Figure 1.5(d)), multiple samples (e), and multiple biometric sources, (f)). Multi-sample and multi-source model have been briefly explained in Section 1.4 and will be explored in details in Chapter 3. Exploiting such parallel structure is an important means towards a more robust system.

1.5.2 Ensemble methods

Classifier performance can be further improved by employing ensemble methods. Ensemble methods are learning algorithms that construct a set of classifiers and then classify new data points by taking a (weighted) vote of each classifier prediction. The main idea is that ensembles are often much more accurate than the individual classifiers that make them up, provided that the individual classifiers are accurate and diverse. If the base-line classifiers are understood as a set of level-0 classifiers, an ensemble method can be seen as their level-1 classifier, also called a stacked generaliser [16]. The intermediate results from the level-0 classifiers can be used directly in the final decision making of accepting or rejecting an identity claim.

Dietterich [8] groups ensemble methods into: (i) Bayesian voting, (ii) manipulation of the training examples (e.g. bagging, cross-validated committee and boosting), (iii) manipulation of the input features (e.g. sub-tasking), (iv) manipulation of the output targets (e.g. ECOC) and (v) injection of randomness. Biometric features are very susceptible to noise and different deformation. Therefore, these techniques are important considerations in our framework. Opitz and Maclin [20]

[2] Reliability is defined as the probability that the component works correctly



(a) classical model (b) Multi-classifier (Ensemble)

(c) Multi-extractor (d) Multi-virtual sensor

(e) Multi-sample (f) Multi-source

(g) legend

Figure 1.5: Different biometric models

make a thorough empirical study based on simulated and real data to test bagging and boosting methods. Their findings suggest that unstable classifiers like the ANN can be very useful in classifying difficult data. Kittler *et al* [16] have convincingly shown that a modified version of ECOC called multi-seed ECOC improves face recognition on the XM2VTS database.

Four particularly developed methods that deserve consideration in the generic biometric framework are listed here:

- AdaBoost.
  AdaBoost is a particular boosting algorithm that receives much attention [10]. The main idea of the AdaBoost is to assign a weight to each example of the training set. At the beginning of the training, all weights are equal. In each iteration, the weak learner returns a hypothesis, and the weights of all examples classified wrongly by that hypothesis are increased. This forces the weak learner to focus on the difficult examples of the training set. The final hypothesis is a combination of the hypotheses of all rounds, namely a weighted majority vote, where hypotheses with lower classification error have higher weight. Among the ensemble methods, AdaBoost can perform at least as good as bagging. However, AdaBoost has to be used with care because it can suffer from over-fitting when the data is too noisy [8]. A detailed explanation is presented in [10].

- Bagging.
  Dietterich [8] claimed that bagging is more tolerant to noise than AdaBoost because bagging does not over-fit the training samples. On each cycle of learning, bagging presents its classifiers with a random subset training set but training samples can be repeated. The subset

is called a *bootstrap replicate*.
Often a simple majority-voting rule can be applied to obtain the final result. Breiman [2] claims that bagging is effective on unstable classifiers like C4.5 and the ANN, where a little change in the training set results in a large change in the prediction.

- Simple artificial neural networks ensemble.
  A simple ANN ensemble is made up of a set of ANNs trained using the full training data but initialised with different weights. In [20], it is shown that this ensemble technique is effective, often producing results as good as bagging. This technique is attractive because of its simplicity in implementation.

- Error correction output-coding (ECOC).
  The ECOC provides a method to solve an K-class problem by reducing the problem to several P binary classification problems, and $P > K$. It is a concept derived from the information theory. Dietterich and Bakiri [9] suggest that classification can be modeled as a transmission channel. Bit data can often be corrupted (or miss-classified in our context). A class is represented by a code word (a binary string). Any pair of code words has a large Hamming distance. This can be generated using the most popular BCH codes [26]. When a set of code words is arranged by row, a code word matrix is formed. Each column bit in the codeword matrix is to be learnt by a classifier. Therefore, for a code word of P bits, P classifiers are needed. During identification, the P classifiers produce a vector score of P elements. Classical distance functions like Euclidean Distance (also called the second order Minkowski distance) or city-block distance (also called the first order Minkowski or Manhanttan distance) can be used to match against each code word (or class label) in the code word matrix [8]. The winning class label is the code word that produces the minimum distance. During verification, however, a threshold value has to be defined. An identity claim is authorised if the distance is smaller than the threshold and vice-versa. The ECOC has the advantage of being independent to classifiers. Heterogeneous type of classifiers can work together. By dividing the problem into several independent sub-classifiers, it is now possible to optimise the performance of each sub-classifier, because each individual classifier is now responsible to solve only parts of the problem.

1.5.3 Multi-extractor

In the strategy of multi-extractor, given a raw biometric data, several features are extracted. For example, one can extract the following information from speeach features: Linear Predictive Coding Coefficients or Wavelet coefficients. In face verification, One can extract projected pricipal components, linear discriminant components or independent components. Each features is often classified by an associated classifier. Since these features are different, it is expected the trained classifier should commit different errors. On the often false assumption that features are not correlated, the classifiers are therefore not correlated. In the hope that each classifier operating on different feature space makes different errors, the combined classifier will be able to correct the error.

For the problem of face verification, Marcel and Bengio [22] have shown that instead of using just face images, one can use normalised face colour histogram as an additional feature to the existing normalised face image to train client specific [3] classifiers. This yields an improved classification result.

1.5.4 Multi-virtual sensors

This is a significantly different approach that directly solves the problem of lack of training examples of an identity by generating multiple "virtual" examples from a single example. This approach

[3] also called one-per-class or one-versus-all approach

has been used for training in Support Vector Machines (SVMs) [34] and has also been widely used in Neural Networks[4].

In the context of face verification, Marcel and Bengio [22] have applied *relevant random transformation* on face images before the feature extraction stage. These transformations are random translation, rotation and scaling. The generated examples are used to train client-specific MLP classifiers. This approach can improve the classification result because it avoids the phenomenon of over-fitting. An experiment using this approach can be found in Appendix B. Unfortunately, it is not known how other biometric modalities such as voice features could be transformed to synthesize virtual examples.

A similar related proposal is to test a new access with virtual examples. From a single biometric access with one sample, one will have only a score. However, by applying N relevant random transformations directly on the sample of a given biometric modality, one will obtain N confidence scores. By using a supervisor that merges scores of classifiers, e.g. by averaging, one may obtain an improved result. This can be at least theoretically justified by the multi-sample single-source approach explained in Section 3.2, except that now the samples are not real samples but synthesized samples.

1.5.5 Multi-sample

In Section 1.2, it is understood that a serial process of sensor-extractor-classifier results in a certain rate of accumulated error (i.e., false rejection and false acceptance). To avoid this phenomenon of error accumulation, one way is to "distribute" the error through a parallel structure that is made up of several processes. This can be achieved through the use of multiple samples. Kittler *et al* [17] have demonstrated the efficiency of an integration strategy that fuses multiple snapshots of a single biometric property using a Bayesian framework. It is observed that as more and more samples are used, the classification error decreases until a point where it is "saturated", i.e., further increase of samples will not decrease the classification error further.

1.5.6 Multi-source

Several studies have shown that multi-source biometrics is superior to any single-modal biometrics. Brunelli and Falavigna [3] have proposed two independent biometric schemes by combining evidence from speaker verification and face recognition. Dieckmann *et al.* [7] have proposed an abstract level fusion scheme called "2-from-3-approach" which integrates facial features, lip motion and voice and is based on the principle that a human uses multiple clues to identify a person. Jain *et al* [12] have proposed a multi-source biometric system design that integrates face and fingerprints to make a personal identification. Ross *et al* [30] have used hand-scan, fingerprint and face-scan to improve the overall result.

1.5.7 Other approaches

Below are some heuristics that can improve a biometric system that do not fit into the proposed models above. Nevertheless, these techniques are very robust.

- Controlling biometric sample quality
  This approach directly improves a biometric system at the sensor level by assigning a quality index to each sample. For instance, in automatic fingerprint recognition, the quality index of each fingerprint can be used to select an optimal algorithm. Shen *et al* [36] have used Gabor wavelets to estimate the quality of sampled fingerprints. The estimated quality index can then be used either to guide the enhancement applied before feature extraction, or to decide upon another life-scan. Such methodology can easily be generalized to other biometric modules to estimate the noise model in the hope of achieving better recognition rate.

[4] In the Frequently Asked Questions of Neural Networks, learning with noise is called "jiterring".

- Combining biometrics with non-biometrics.
  Maes *et al* [21] have proposed to combine biometric data, e.g., voice-scan, with non-biometric data, e.g., password. Another suggestion of non-intrusive and highly measurable biometrics is the height and weight of a person. Although this is a simple approach and not unique, it can be highly accurate and single out a person from a possibly large population. Furthermore, height and weight sensors are relatively inexpensive. When applying such a measurement coupled with other biometric modalities, accuracy and reliability of the system can be improved. This can be a potential practical implementation. However, this idea has not been implemented yet.

The list is obviously not exhaustive. However, this survey can represent the state-of-the-art in biometric authentication at this time of writing. A more detailed survey can be found in [28].

1.5.8 Approaches used in this report

The aim of this study is to propose all possible means of improving the accuracy of biometric authentication system using different combination strategies. Inspired by the work of Kittler *et al.* [17] and Hong *et al.* [13], their two seemingly separate approaches, namely multi-source and multi-sample approaches, are combined. It is called a multi-sample multi-source approach. Furthermore, one of the ensemble methods, namely, Error-Correction Output-Code (ECOC) is studied in details. By using multiple samples, multiple sources and multiple classifiers, with existing algorithms, it is hoped that the performance of biometric authentication system can be improved.

Comparing to our previous work done [27], this study covers a wider range of possibilities to improve a biometric system. In the previous work, the decisions of two biometric sources were combined via the "AND" operator. This approach requires two thresholds, i.e., one for each biometric source. In this study, only one threshold is needed because the scores of two or more biometric sources are averaged. The single threshold is used to make decision on the *averaged* score. Furthermore, in the previous work, only one classifier was used for each biometric source. In this study, ensemble method such as ECOC is used to improve the classification accuracy. These improvements are considered

1.6 Guide to the dissertation

Having explained the problem of biometric authentication, it's associated quality measurement and it's existing system models in Chapter 1, Chapter 2 gives an introduction to the problem of classification and classifiers considered in this study. Chapter 3 explains the technique of score-averaging, leading to a new approach called multi-sample multi-source approach[5]. Chapter 4 explains a second technique called Error-Correcting Output Coding (ECOC). Chapter 3 and 4 can be considered as the principal contributions of this research work. Chapter 5 discusses several potential research directions in the domain of biometrics with focus on multimedia information processing. This is followed by conclusions.

[5] This chapter will be published in the form of a conference paper in the coming NNSP'02 conference to be held in September 2002, Switzerland.

Chapter 2

Classifiers and the Classification Problem

2.1 Formalism of classification

Classification can be viewed as a task of mapping a feature vector to a class label. The way a feature vector is classified is assumed to be governed by an unknown concept F. A *learning algorithm* merely gives an estimate function $\hat{F}$ of this unknown concept F. The estimate function $\hat{F}$ is called a classifier. In other words, $F : \Omega \rightarrow \sigma$ takes the *input space* $\Omega \equiv \mathcal{R}^d$ and maps it into an *output space* $\sigma \equiv \{c_1, \ldots, c_K\}$ of discrete classes where: d is the number of elements which is often called a feature dimension and K is the number of classes. The learning phase is carried out using a training data set $\mathcal{X}$ of the form $< \mathbf{x}_i, y_i >$, $i \in \{1, \ldots, N\}$ where $\mathbf{x}_i$ is an instance, y_i is its class label and N is the number of instances in $\mathcal{X}$.

One can say that the function $F : \Omega \rightarrow \{c_1, \ldots, c_K\}$ defines a K-partition of the input space, with each partition containing the data of class c_K. Hence, the training data set $\mathcal{X} \equiv \bigcup_{k=1}^{K} \mathcal{X}_k$ is also partitioned into K class subsets.

The set $\mathcal{X}$ of training data contains a subset of the whole feature space $\mathcal{X} \subseteq \Omega$ and the created model $\hat{F}$ using the set $\mathcal{X}$ should generalise to instances of the feature space not contained in $\mathcal{X}$. $\hat{F}$ is typically called a classifier. The accuracy of this model is often measured on a independent *test* set $\mathcal{X}_{test}$, i.e., both the training and test sets are exclusive, i.e., $\mathcal{X}_{test} \cap \mathcal{X} \equiv \emptyset$.

2.2 Several chosen classifiers

2.2.1 Goal of study

In the following sections, several classifiers used and implemented during my DEA internship will be briefly discussed: Multi-Layer Perceptron (MLP), Support Vector Machines (SVMs) and Winnows. Each of these classifiers merits some discussion in this study due to several reasons:

- MLP is one of the most commonly studied and used classifiers. It is considered a universal classifier which can model any function F. Furthermore, it is readily adapted to multi-class problems. It is often used as a baseline classifier to be compared with other classifiers.

- SVMs are an example of large margin classifiers which have gained much attention in the machine learning community due to their well-understood statistical theory. They are being applied in text categorisation, image-recognition, hand-written digit recognition and bioinformatics [6, Chap. 8]. SVMs can therefore be applied to biometric authentication as long as features can be extracted and be represented in a fixed-size vector.

- Winnow is originally a linear threshold binary classifier (with linear connection for a two-class problem) with a guaranteed worst case mistake bound. It has fast convergence, making it efficient in learning when the feature dimension or the number of hypotheses (classes) are large. Winnow is a binary classifier. A sparse version of a collection of Winnows (one with sparse connection) turns Winnow into a multi-class classifier, known as SNoW (Sparse Networks of Winnows). SNoW has been applied in word prediction and image recognition [32, 31]. SNoW (or it's variations) is considered as a potential candidate to solve the problem of scalability in the context of biometric authentication.

In this report, only SVMs and MLPs are used. Winnow is discussed here to lay the foundation necessary in Section 5.3 as a potential future research direction. It is best to understand the discussion on SVM and MLP in the context of two-class classification problem. In the following section, these three classifiers are discussed. The decision-making using these classifiers will only be discussed in Section 2.3.1.

2.2.2 Multi-layer perceptron

A Multi-Layer perceptron (MLP) is a particular architecture of artificial neural networks [1], composed of layers of non-linear but differentiable parametric functions. For instance, the output $f(\mathbf{x})$ of a 1-hidden-layer MLP can be written mathematically as follows

$$f(\mathbf{x}) = b + \tanh\left(\mathbf{a} + \mathbf{x} \cdot \mathbf{V}\right) \cdot \mathbf{w}. \tag{2.1}$$

where the estimated output $f(\mathbf{x})$ is a function of the input vector $\mathbf{x}$, and the parameters $\{b, \mathbf{w}, \mathbf{a}, \mathbf{V}\}$. b and $\mathbf{a}$ are called bias parameters for the output and hidden units (or neurons)[1]. $\mathbf{w}$ and $\mathbf{V}$ are hidden-to-output and input-to-hidden weight connections. The matrix V has a dimension of the number of input feature $\mathbf{x}$ times the number of hidden neurons. The matrix w has a dimension of the number of hidden neurons times the number of output neurons, which is one in Equation 2.1. In this notation, the non-linear function tanh() returns a vector with size equals to the number of hidden units of the MLP and is used to control its capacity. The number of hidden units should be chosen carefully by technique such as cross-validation. These parameters can be estimated using optimisation algorithms to be discussed further. The process of estimating these parameters is called "training".

An MLP can be trained by gradient descent using the back-propagation algorithm [33] to optimize any differentiable criterion, such as the *mean squared error* (MSE):

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i))^2, \tag{2.2}$$

where N is the number of training examples and y_i is the target output of the i-th example.

The decision-making using $f(\mathbf{x})$ is discussed in Section 2.3.1.

2.2.3 Support vetor machines

Support vector machines (SVMs) were introduced by Vapnik [39]. The SVM algorithm constructs a separating hypersurface in the input space. It acts as follows:

- maps the input space into a higher dimensional feature space through some nonlinear mapping chosen *a priori* (kernel);
- constructs the maximal margin hyperplane (MMH) in this feature space The MMH maximizes the distance from the hyperplane to the closest vectors from each class.

[1] A "neuron" is mathematical model of a neuron cell found in brain. This mathematical model can be expressed in the function $f(\mathbf{x}\mathbf{V}')$, where $\mathbf{V}$ is a weight vector, $\mathbf{x}$ is an input vector and f is often a sigmoid function $\frac{1}{1-e^{\mathbf{x}\mathbf{w}'}}$. An introduction of Neural Networks can be found in [1]

The resulting function is of the form:

$$f(\mathbf{x}) = \sum_{i=1}^{N} y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{2.3}$$

where $\mathbf{x}$ is the input vector of an example to test, f is the estimated output of the model, $\mathbf{x}_i$ is the input vector of the i^{th} training example, N is the number of training examples, the α_i and b are the parameters of the model, y_i is the class label, and $K(\mathbf{x}, \mathbf{x}_i)$ is a kernel function that can have different forms, such as:

$$K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x}^t \, \mathbf{x}_i + 1)^d \tag{2.4}$$

which results in a Polynomial SVM with parameter d, or

$$K(\mathbf{x}, \mathbf{x}_i) = \exp\left(\frac{-\|\mathbf{x} - \mathbf{x}_i\|^2}{\sigma^2}\right) \tag{2.5}$$

which results in a Radial Basis Function (RBF) SVM with parameter σ. Either d or σ must be selected using methods such as cross-validation.

In order to train such SVMs, one needs to solve the following quadratic optimisation problem: find the parameter vector $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$ that maximises the objective function

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{2.6}$$

subject to the constraints

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{2.7}$$

and

$$0 \leq \alpha_i \leq C \ \forall i. \tag{2.8}$$

It is important to note that the training complexity of SVMs is quadratic on the number N of examples, which makes the use of SVMs for large datasets difficult. Note however that in the resulting solution(Equation 2.3), most α_i are equal to 0, and the examples with non-zero α_i are called *support vectors*. Intuitively, α_i can therefore be considered as the amount of contributions of a given training feature i to the classification function f. The higher α_i is, the more important it is in participating in the classification function.

2.2.4 Winnows

The Winnow classifier is essentially a linear classifier with binary features. Since, in general, biometric features are numerical, a conversion from numerical to binary feature space is necessary. Let $\mathcal{D}$ be this discretisation process of a feature $\mathbf{x} = x_1, \ldots, x_d$ such that: $\mathbf{x}$, $\mathcal{D} : \mathbb{R}^d \rightarrow \mathbb{B}^P$, where d is the dimension of an extracted biometric feature and P is the resultant dimension of the biometric feature in a *set of Boolean spaces* or a *discrete space*. A Boolean is defined as $\mathbb{B} \in \{true, false\}$ where a "true" indicates the presence of a feature and vice-versa for the "false". Note that $D \ll P$. An example of the discretisation function $\mathcal{D}$ will be given later.

Let $\mathbf{M} \subset \mathbb{R}^{K \times P}$ be a linear weight matrix where K is the number of classes and P is the number of dimension in the discrete space. The estimated classifier model, denoted as $f(\mathbf{x})$, when given a feature vector $\mathbf{x}$, can be written as:

$$f_k(\mathbf{x}) = \mathbf{M}_k \bullet \mathcal{D}(\mathbf{x}), \tag{2.9}$$

$$= \sum_{p=1}^{P} \mathbf{M}_{k,p} \mathcal{D}(\mathbf{x})_p, \tag{2.10}$$

where $\mathcal{D}(\mathbf{x})_p$ is the p-th element of vector $\mathcal{D}(\mathbf{x})$ and $p = 1, \ldots, P$.

Let the discretisation function be $\mathcal{D} = \mathcal{D}_1, \ldots, \mathcal{D}_d$, where d is the number of feature dimension. Each $\mathcal{D}_i$ takes a feature element $x_i \in \mathbb{R}$ from $\mathbf{x} = x_1, \ldots, x_d$. Thus, $\mathcal{D}$ is a function that maps from $\mathbb{R}^d$ to $\mathbb{B}^P$ such that $P = d \times N$, where N is the number of discretisation per feature. Then, one way to define $\mathcal{D}_i : \mathbb{R} \rightarrow \mathbb{B}^N$, for all $i = 1, \ldots, N$ is:

$$\mathcal{D}_i(x_i) = \begin{cases} true & \text{if } k_{j-1} \leq x_i \leq k_j \\ false & \text{otherwise, for } j = 1, \ldots, N \end{cases} \tag{2.11}$$

where, $k_0 = -\infty$, $k_{j-1} < k_j$ and $k_N = +\infty$. In other words, the series $k_1, \ldots, k_N$ establishes a bin within $k_{j-1} < x_i < k_j$ so that an element x_i of a feature vector $\mathbf{x}$ is mapped into a discrete space $\mathbb{B}^N$. The bin intervals k_j can be linear or non-linear. In fact, there exist several algorithms to realise the discretisation process that are data dependent, e.g. Linear Vector Quantisation (LVQ) and K-means.

To calculate the matrix $\mathbf{M}$, Algorithm 1 is used. The difference between Winnow and the standard Perceptron is that SNoW (Sparse Networks of Winnows) weights are updated only when an error is made. This approach is called *conservative*[2]. If the error is due to the rejection of a positive example (called false rejection), the weights linked to the target node are *promoted*. Alternatively, if the error is due to the acceptance of a negative example, the weights linked to the target node are *demoted*. The "rates" of promotion and demotion are controlled by pre-defined parameters α and β respectively. There exist a decision function (see Equation 2.12) that governs the acceptance and rejection of Winnow. This decision function requires the parameter θ. In the training of Winnow, this parameter is pre-defined. However, it needs not be the same as the value during testing. For instance, a more suitable value θ can be estimated *a posteriori* on a separate validation set. In short, to train a SNoW, one needs to determine the following parameters: promotion parameter α, demotion parameter β, threshold θ and discretisation function $\mathcal{D}$.

Algorithm 1 Train SNoW

```
Initialise: M = 0
for each example < x, c_k > do
  for each class c_k', c_k and k ≠ k' do
    if f_k(x) < θ then
      ∀_p(M_{k,p} = M_{k,p} + α) {false rejection so promote}
    else if f_k'(x) > θ then
      ∀_p(M_{k',p} = M_{k',p} − β) {false acceptance so demote}
    end if
  end for
end for
```

2.3 Discussion

2.3.1 Decision making

For the decision making of a two-class problem, one can use:

$$\hat{F} = \begin{cases} accept & \text{if } f(\mathbf{x}) > \theta \\ reject & \text{otherwise} \end{cases} \tag{2.12}$$

The threshold θ has to be chosen using a validation set or using cross-validation.

[2] The concept of conservativeness is governed by the magnitude of the mean squared error between the output value and the target value of output neurons. Algorithm 1 is explained here to illustrate the meaning of conservativeness.

For K-class problem, one usually uses K classifiers f_k, where $k = 1, \ldots, K$. The decision function is:

$$\hat{F} = \arg max_k f_k(\mathbf{x}). \tag{2.13}$$

2.3.2 Towards a multi-class problem

MLP can be easily adapted to a multi-class problem by using a network configuration of d input feature dimension, H number of hidden neurons, and K output neuron, denoted as d-H-K MLP. One can also build K MLPs each having the configuration d-H-1. The advantages and disadvantages are discussed in Section 4.3.2.

SVM is by its nature a binary classifier, i.e. it discriminates between a classification problem inlvoving only two classes. It can be extended to a multi-class problem via one-versus-all approach (where one class is treated as positive and the rest are treated as negative classes) or Error-Correction Output-Coding (ECOC) to be discussed in Chapter 4.

Winnow is by its nature a multi-class problem, i.e. it capable of distinguishing a problem involving many classes. It should be mentioned that the weights $\mathbf{M} \in \mathbb{R}^{K \times P}$ in Winnow is a *sparse* matrix, i.e., most weight connections have zero values. This is necessary because the binary feature dimension P or the number of classes K could be a very large value. This makes it suitable for very large problems such as in large vocabulary recognition problem [31].

However, Winnow is a linear classifier. Therefore it's capacity is limited compared to SVM and MLP. The capacity of a classifier is the degree of freedom of a classifier in estimating an unknown concept. The higher the degree of capacity, the higher the ability to model the concept. The capacity in MLP is controlled by the number of hidden units while the capacity in SVM is controlled by the kernel function.

The higher the capacity of a classifier, the easier it is to over-fit the unknown concept F. Therefore, it is important that a suitable degree of capacity is used. The only way to estimate the appropriate degree of capacity is by cross-validation or simple validation.

2.3.3 Training, evaluation and test sets

In practice, the entire data set $\mathcal{X} \equiv < \mathbf{x}_i, y_i >$ can be split into training ($\mathcal{X}_{train}$), validation ($\mathcal{X}_{validation}$) and test sets ($\mathcal{X}_{test}$). These sets are mutually exclusive, i.e., $\mathcal{X}_i \cap \mathcal{X}_j \equiv \emptyset, \forall_{i,j}$, where $i, j \in \{train, validation, test\}$. In general, the training set is used to train a given classifier directly while the validation set is used to choose the degree of capacity, other model-dependent parameters and the threshold θ of the decision function. All these parameters are called *hyper-parameters*. One can randomly choose $\mathcal{X} - \mathcal{X}_{test}$ to determine $\mathcal{X}_{train}$ and $\mathcal{X}_{validation}$ several times as long as the size of training set and validation set are consistent for each instance of random draw, i.e., $|\mathcal{X}_{train_i}| = |\mathcal{X}_{train_j}|, \forall_{i \neq j}$ where i and j are two instances of random draw. This applies to the validation set as well, i.e., $|\mathcal{X}_{validation_i}| = |\mathcal{X}_{validation_j}|, \forall_{i \neq j}$. If the size of training and validation sets are not equal in each random draw, the results of each random draw will not be comparable. For any instance i and j, it is necessary that the following condition be fullfilled: $\mathcal{X}_{validation_i} \cup \mathcal{X}_{train_i} = \mathcal{X} - \mathcal{X}_{test}$, for any instance i. Note that throughout all the instances, $\mathcal{X}_{test}$ remains a constant set. This is a common practice used in machine-learning community. It is also used in XM2VTS databases and is defined by the Lausanne Protocol [19], which has only two random draws and are called Lausanne Protocol One (LP1)and Two (LP2), respectively. Note that LP1 and LP2 have *different* size of training and validation sets. Therefore, the results of these two protocols are not comparable.

K-fold cross-validation is a special case of validation where the data of $\mathcal{X} - \mathcal{X}_{test}$ are partitioned into K subsets. One of these subsets will be used as an instance of $\mathcal{X}_{validation}$ while the rest of K-1 partitions are used as $\mathcal{X}_{train}$. In both cross-validation and K-fold cross-validation, $\mathcal{X}_{test}$ is an exclusive set that should be used for testing.

As a final remark, a point on the DET curve discussed in Section 1.3 is described as *a posteriori* when it is plotted using an optimised threshold on the EER criteria, i.e., FAR=FRR. If a threshold is pre-defined to make a decision, then that point on the DET curve is described as *a priori*. In

practice, one chooses a threshold that optimises EER on the evaluation set and this threshold is used to make decision on the test set. The threshold is said to be chosen *a posteriori* on the evaluation set and fixed *a priori* on the test set. This is the practice throught out this report whenever there exists three sets in the database (such is the case of XM2VTS database) and whenever there exists two subsets, only training and test sets are used (such are the cases of the vowel database and LSIIT database described in Appendix A). In this case, the first set is used to train the model and the second set is used to choose the hyper-parameters. The first set is consistantly called the training set while the second set is actually called the test set. Although the second set is more appropriate to be called the validation set but unfortunately, there is not the practice in the machine-learning community. In this report, the second set is called the *test set*.

2.4 Summary

In this section, several classifiers are discussed. These classifiers are discriminant functions f that solve the problem $y = f(\mathbf{x})$. In particular, Multi-Layer Perceptrons and Support Vector Machines are classification functions that are non-linear, and therefore are capable of estimating many functions. The "capacity" of these functions are dependent on the number of hidden units and the sigma value of Gaussian kernel for MLPs and SVMs respectively. These two classifiers are relatively inefficient to train comparing to Winnow. However, Winnow is a linear classifier, i.e., it cannot solve problems that do not have a linear solution. MLPs and SVMs will be used extensively as base-classifiers for problems in Chapter 3 and 4.

Chapter 3

Towards a Multi-sample Multi-source Biometric Model

3.1 Motivation

In this study, two approaches that can improve the authentication process are examined: (i) multi-sample approach and (ii) multi-source approach. By using N samples for a single request access, it will be shown that N chain processes of sensor-extractor-classifier are "created" and that noise can be reduced at most by a factor of N. In the same way, by using M biometric sources, the noise can be further reduced by a factor of M. These claims are based on the demonstration shown by Bishop[1] the context of "committee of classifiers". By using the average operator, both the theoretical and empirical results show that integrating as many samples and as many biometric sources as possible can improve the overall reliability of the system by a factor of $N \times M$ at most. This strategy is called multi-sample multi-source approach. This section is accepted for publication in the coming IEEE International Workshop on Neural Network in Signal Processing Conference 2002 (NNSP'02).

3.2 Multi-sample single-source approach

Kittler *et al.* [17] have shown that a MSSS model can be an effective way to boost a biometric system. In their experiment setting with face images, several face samples are used. The optimal final decision score is found by averaging classifier scores. This can be justified by modelling a score corrupted by noise.

Let y_i be the "observed" measure (i.e. a score, $y_i = f(\mathbf{x}^i)$ of a given sample i out of a total of N samples. There is a "true" measure $\hat{y}$ and it is corrupted by the noise η_i. This noise is assumed to be drawn from a random zero-mean additive distribution. The observed measure y_i can be written as:

$$y_i = \hat{y} + \eta_i. \tag{3.1}$$

The mean of y_i, denoted as $\bar{y}$ is:

$$\bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i. \tag{3.2}$$

With enough samples, the expected value of y_i, denoted as $\xi[y_i]$, which is calculated as the mean of y_i (Equation 3.2), approximates the "true" measure:

$$\xi[y_i] = \hat{y}. \tag{3.3}$$

The expected value of random noise η_i, i.e., $E(\eta_i)$ is always zero. The variance of the observed y can be written as:

$$\sigma_y^2 = \frac{1}{N}\sigma_\eta^2. \tag{3.4}$$

Therefore, it can be concluded that when N scores of a single biometric source are averaged, noise that occurs due to classification can be reduced by a factor of N. The effect of averaging in Equation 3.2 can best be observed using synthetically generated data in Figure 3.1. Assume that in the original problem, the genuine user scores have a normal distribution of 1.0 mean and 0.36 variance, denoted as $\mathcal{N}(1, 0.36)$, and that the impostor scores have a normal distribution of $\mathcal{N}(1, 0.81)$ (both graphs are plotted with '+'). If for each access, three confidence scores are available, according to Equation 3.4, the variance of the resulting distribution will be reduced by a factor of three. Both the resulting distributions are plotted with 'o'. Note the area where both the distributions cross before and after. This area corresponds to the shaded area of the schematic diagram in Figure 1.2. It is the zone where minimum amount of mistakes will be committed given that the threshold is optimal. The decrease in the area means an improvement in the recognition rate. Therefore, score averaging is a simple yet effective way to increase system accuracy.

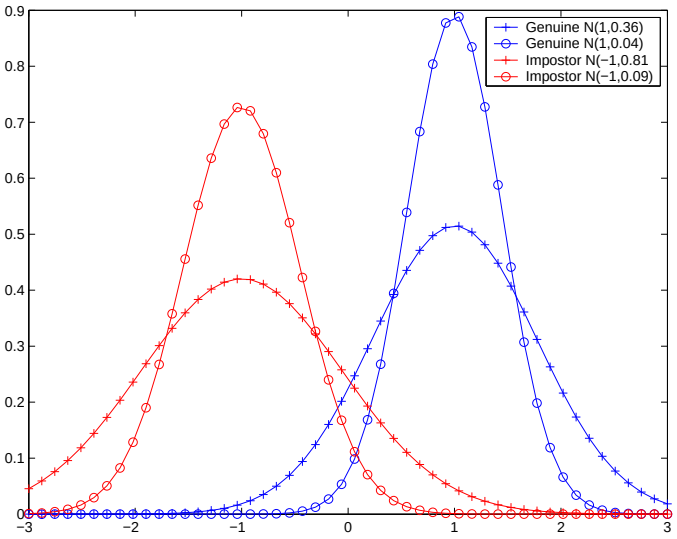


Figure 3.1: Effects of averaging to the scores distribution in a two-class problem

3.3 Error committed by a committee of classifiers

The proof in Section 3.2 has actually been shown in [1, Chap. 9]. The proof is presented here because it is used to derive both the single-sample multi-source and the multi-sample single-source approaches.

Starting from Equation 3.1, where y_k is taken as the score of classifier k (and not the score of each sample, y_i , as defined before), one can apply the similar assumption of the noise model for each classifier and rewrite it as:

$$y_k = \hat{y} + \eta_k. \tag{3.5}$$

Similarly, by using averaging as a means to join the scores of Q classifiers, one obtains the score of the committee, y_{COM}:

$$y_{COM} = \frac{1}{Q}\sum_{k=1}^{Q} y_k. \tag{3.6}$$

one can deduce that the average sum-of-squares error for each classifier k as:

$$E_k = \xi[(y_k - \hat{y})^2] = \xi[\eta_k^2]. \tag{3.7}$$

The average error committed by each classifier individually, denoted as E_{AV}, can then be calculated as:

$$E_{AV} = \frac{1}{Q}\sum_{k=1}^{Q} E_k. \tag{3.8}$$

Replacing Equation 3.7 into Equation 3.8, one can write E_{AV} as:

$$E_{AV} = \frac{1}{Q}\sum_{k=1}^{Q} \xi[\eta_k^2]. \tag{3.9}$$

By using the average operator in Equation 3.6, the error of the committee is:

$$\begin{aligned} E_{COM} &= \xi[(y_{COM} - \hat{y})^2] = \xi[\eta_k^2], \\ &= \xi[(\frac{1}{Q}\sum_{k=1}^{Q} y_k - \hat{y})^2], \\ &= \xi[(\frac{1}{Q}\sum_{k=1}^{Q} \eta_k)^2], \\ &= \frac{1}{Q^2}\sum_{k=1}^{Q} \xi[{\eta_k}^2], \end{aligned} \tag{3.10}$$

Assuming that the noise η_k are independent and uncorrelated, i.e., $\xi[\eta_k] = 0$ and $\forall_{p,q}(\xi[\eta_p\eta_q] = 0)$, Equation 3.10 can be written in the term of E_{AV} from Equation 3.9 as:

$$E_{COM} = \frac{1}{Q}E_{AV}. \tag{3.11}$$

In summary, it is shown that this simple form of committee of Q classifiers can reduce the classification error by a factor of Q, when the errors are independent and uncorrelated.

3.4 Single-sample multi-source approach

In this section, it is desirable to show that the error of the joint system is lower than the error of each of the sub-components when they operate independently. This can be represented as: $\forall_j(E_d \leq E_j)$, where E_j is the error of one of M sub-components j and E_d is the error of the joint system. Note that the index j is used to signify a biometric source so as to distinguish it from the index i that was used to signify one of N samples taken from a given biometric source.

It is desirable that the following relationship holds:

$$E_d = \prod_{j=1}^{M} E_j. \tag{3.12}$$

Equation 3.12 can be interpreted as such: in a system with M biometric sources, the whole system will always select the best sub-component. In other words, this is the optimal decision, called Oracle in [18]. Such a supervisor (one that merges the scores, as defined earlier) is the best result that one can get out of scores combination. From Equation 3.12, it is obvious that $\forall_j(E_d \leq E_j)$. Using normal and uniform distribution to model the probability of false rejection, Kuncheva [18] studied six supervisors: minimum, maximum, average, median, majority vote and Oracle. In practice, the Oracle supervisor does not exist because one does not know in advance

the true identity during verification. Therefore, it is omitted from this discussion. Among the five remaining classifiers mentioned, Kuncheva found that the average supervisor works the best when the error comes from the two distributions mentioned above.

By considering each sub-component of the multi-source system as an independent classifier, the proof discussed by Bishop [1], as expressed in Equation 3.11, can be used to show that the average supervisor also satisfies $E_d \leq mean_j(E_j)$, instead of $\forall_j(E_d \leq E_j)$ the "perfect" requirement established earlier. He has shown that a committee of average and weighted average classifiers could perform better than a single classifier. It is obvious that in a multi-source system, the error different biometric sourcs η_j are uncorrelated, i.e., $\forall_{j_1,j_2}(\xi[\eta_{j_1}\eta_{j_2}] = 0)$. It is assumed that the expectation of error of each biometric source is zero, i.e., $\xi[\eta_j] = 0$.

In the context of multi-source system, Equation 3.11 can be rewritten in the following inequality:

$$E_d \leq \frac{1}{\beta} mean_j(E_j), \tag{3.13}$$

which is valid when $1 \leq \beta \leq M$. Equation 3.11 shows that the upper bound of E_d occurs when $\beta = M$. The lower bound of E_d occurs when $\beta = 1$. This can be interpreted as follows: If the errors made by each sub-component are dependent, i.e., they make exactly the same error in the extreme case $(\forall_{j1,j2}(E_{j1} = E_{j2}))$, then $mean(E_j) = E_j = E_d$, which implies that $\beta = 1$.

Note that the difference between the context in [1] and the context here is that the independency of each sub-component is true and not an assumption. This is because each biometric source is really independent from other biometric sources.

3.5 Multi-sample multi-source approach

The goal of this section is to combine the two findings above using the strategy called the multi-sample multi-source (MSMS) approach. In such a system, it is assumed that there are M biometric sources and for each source, N samples are available. Scores made available to this system are denoted as $y_{i,j}$, where $i \in 1, \ldots, N$ and $j \in 1, \ldots, M$. For each $y_{i,j}$, there is an associated error: $E_{i,j}$.

We argue that the inequality 3.13 used in multi-source biometric system holds as well for multi-sample biometric system, with the weak assumption that the errors ($\forall_i(E_{i,j)}$ for a given j) are independent and have a zero mean. One can therefore write inequality 3.13 by changing the index from j to i, as follows:

$$E_d \leq \frac{1}{\alpha} mean_i(E_i), \tag{3.14}$$

where α is the factor of increase of performance and $1 \leq \alpha \leq M$. Empirical results by the work of Kittler [17] and the result at the end of this chapter also support this observation. The inequality 3.14 can be used to deduce the inequality of the MSSS model, as follows:

$$E_{mssss_j} \leq \frac{1}{\alpha} mean_i(E_{i,j}). \tag{3.15}$$

In the same way, the inequality $E_d \leq \frac{1}{\beta} mean_j(E_j)$ can be applied to the SSMS model as follows:

$$E_{ssms_i} \leq \frac{1}{\beta} mean_j(E_{i,j}). \tag{3.16}$$

An MSMS model is by its definition a multi-source version of the MSSS model. So, the inequality $E_d \leq \frac{1}{\beta} mean_j(E_j)$ that applies to multi-source holds for the MSMS model. Therefore, it is valid to write:

$$E_{msms} \leq \frac{1}{\beta} mean_j(E_{mssss_j}). \tag{3.17}$$

By replacing inequality 3.15 into inequality 3.17, one can write:

$$E_{msms} \leq \frac{1}{\beta} mean_j(\frac{1}{\alpha} mean_i(E_{i,j})) \tag{3.18}$$

$$E_{msms} \leq \frac{1}{M\beta}\sum_{j}^{M}(\frac{1}{N\alpha}\sum_{i}^{N}(E_{i,j}))$$

$$E_{msms} \leq \frac{1}{\alpha\beta NM}\sum_{j}^{M}\sum_{i}^{N}(E_{i,j})$$

$$E_{msms} \leq \frac{1}{\alpha\beta NM}\sum_{(i,j)}^{(N,M)}(E_{i,j})$$

$$E_{msms} \leq \frac{1}{\alpha\beta} mean_{(i,j)}^{(N,M)}(E_{i,j}). \tag{3.19}$$

Therefore, by the MSMS approach, the system error is reduced by a factor within $[1, MN]$ comparing to the average error when each sub-component works individually. The combined score via the MSMS approach using the mean operator is similar to the inequality 3.18. It is defined as follows:

$$y_{msms} = mean_j(mean_i(y_{i,j})) \tag{3.20}$$

$$= \frac{1}{M}\sum_{j}^{M}(\frac{1}{N}\sum_{i}^{N}(y_{i,j}))$$

$$= \frac{1}{NM}\sum_{(i,j)}^{(N,M)}(y_{i,j}). \tag{3.21}$$

3.6 Database, Experiments and Results

Briefly, there are 30 persons in the publicly available LSIIT database[1]. It has two types of biometric sources: face and voice. For each type of biometric source, 5 out of 10 samples of each person are used for training and the other 5 samples are used for testing. Both the training and test sets are mutually exclusive. Detailed description of the database can be found in Appendix A.2.

For each biometric model, a set of one-versus-all configuration Multi-Layer Perceptrons (MLPs) are used, where each MLP is associated with the biometric sample of a single client. In other words, the samples of a particular client are treated as positive examples while all other samples belonging to other clients are treated as negative examples. The MLP of the face features extracted using 144 Fisher basis vectors has a configuration of 144-25-1 (144 input neurons, 25 hidden neurons and 1 output neuron). The MLP of the voice features extracted using Morlet wavelets normalised into a fixed length vector of 64 elements has a configuration of 64-25-1. More details on feature pre-processing and extraction can be found in [29] The number of hidden neurons for both MLP architecture was chosen arbitrarily. Each MLP has an output neuron with a sigmoid activation function so that it can estimate the *a posteriori* probability $p(w_1|\mathbf{x})$ when given a feature $\mathbf{x}$.

For a database of 30 persons, 30 MLPs are needed to classify face features and another 30 MLPs are needed to distinguish voice features. During testing with the previously unseen 5 samples, the experiment is first carried out with 1 sample, then 2 samples, and so on up to 5 samples. Detailed explanation of the database, feature extraction and protocols can be found in Appendix A.2. Figure 3.2 and Figure 3.3 show the DET curves obtained for the experiments using only face and voice features. Table 3.1 shows the performance obtained for each of these experiments, using only Face features, only Voice features, or combining both. Each value represents the Half Total Error

[1] The LSIIT BAS database is available at http://hydria.u-strasbg.fr/~norman/BAS

Rate (HTER) defined in Equation 1.3. These values are computed by choosing a threshold such that FAR equals FRR: this corresponds to the Equal Error Rate (EER). Note that this threshold is thus chosen *a posteriori* on the test data.

Table 3.1: The HTERs of face and voice biometric features when FAR=FRR

No. of samples used	Face HTER	Voice HTER	Combined HTER
1	7.184	6.897	0.805
2	2.701	4.828	0.690
3	1.207	4.540	0.000
4	0.000	2.126	0.000
5	0.000	2.414	0.000

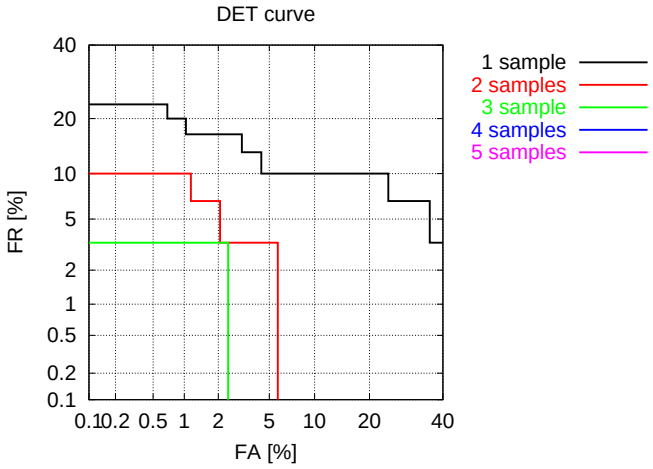


Figure 3.2: DET curves plotted using 1-5 face samples

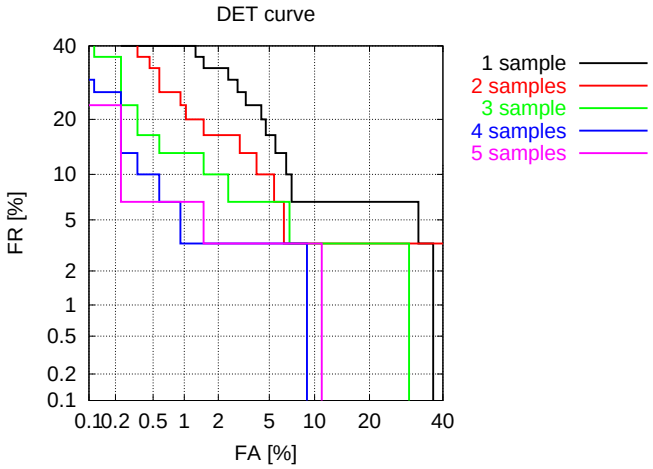


Figure 3.3: DET curves plotted using 1-5 voice samples

The experiment shows that the performance of the MSMS model increases as more samples of each biometric source become available. Using the current database, perfect performance is achieved after three or more samples of each of the face and voice biometric features are used. However, in reality, one does not have the luxury of such a large pool of data. In our opinion, to further validate this approach, more experiments should be done with different biometric features and larger databases that have several samples for each biometric source.

3.7 Summary

Biometric authentication can be viewed as a serial process involving a sensor, an extractor, a classifier and an optional supervisor. One way to increase the reliability of such systems is to utilise several such chain processes in parallel. To do so, one can use multiple samples and multiple biometric sources. In the first technique, by assuming noise at the score level, it is proven that by averaging classifier scores from N biometric samples, noise can be reduced by a factor between 1 and N. In the second technique which uses M biometric sources, by averaging classifier scores, it is proven that one can reduce the error by a factor between 1 and M. Since the correlation between samples of *different* biometric sources is typically lower than the correlation between samples of the *same* biometric source, the gain in combining different biometric sources is higher than the gain in combining different biometric samples of the same biometric source. By combining these two techniques, which is called the multi-sample multi-source approach, one obtains a reduction of error by a factor within $[1, NM]$. This leads to a very significant increase in verification rate. Unfortunately, the disadvantage of this approach is the higher cost of computation and longer access time due to the fact that more samples of different biometric sources are needed.

This hybrid approach was implemented using a set of neural network classifiers and tested on a face and voice biometric database of 30 persons. Using this small database, perfect verification was achieved. This result is certainly promising but most importantly, it shows that one can use multiple samples or multiple biometric sources to boost the reliability of the whole system. An interesting application using this approach is in the inconclusive situation, i.e., the final decision score is marginal for acceptance. In such a situation, the multi-sample multi-source approach can be taken immediately. Furthermore, this approach suggests that it is always beneficial to life-scan longer features (i.e., longer speech signal) and more frames of facial features to increase robustness without adding much cost to the existing system. Finally, the multi-source biometric system that is inherent in our approach may not be appropriate for all applications but only those where the level of required security is high and the higher cost of deployment can be absorbed.

Chapter 4

Error Correction Output-Coding

4.1 Previous works

Error Correction Output-Coding (ECOC)[1] is a method proposed by Dietterich and Bakiri [9]. It is derived from the information theory called error-correcting codes. Error-correcting codes are used to correct K messages transmitted by P channels. If the channels are perfect, P channels can transmit 2^P messages. However, since channels could be noisy, messages may be altered during transmission via P channels. By using more channels than necessary to send K messages, errors due to some channels may be self-corrected. The principal idea of ECOC in classification problems is to assume that each classifier as a binary transmission channel. One has to figure out ways to partition a multi-class problem (say K classes) into P subsets of two-class problems, where in general $K \leq P$. This is the *decomposition* stage. When the feature is "transmitted" through these channels (or classifiers), a score vector of P elements is produced. This score vector is then submitted to a *reconstruction* scheme to finally obtain a decision. One way to realise the decomposition and the reconstruction phase is to use BCH coding [26]. In the thesis of Moreira [25] , the ECOC approach and its variant are studied. Independently, ECOC has also been used by Kittler *et al* [16] in the context of face verification and identification. These previous works have pathed way to this chapter, which is aimed at improving a biometric verification problem, hopefully to be tested across different biometric modalities.

4.2 Classifier diversification via ECOC

4.2.1 Preliminary

ECOC belongs to the family of *ensemble methods*. Ensemble methods are aimed at diversifying classifiers. Other ensemble methods are AdaBoost, bagging and simple neural-networks as discussed in Section 1.5.2. Classifier diversification has been identified as an important way to increase the accuracy of the overall system. Each of the diversified classifiers will tend to make different mistakes so that when the results are combined (or recomposed), these mistakes cancel out each other. The net effect is an improved classification result.

Binary classification is frequently carried out by using a real-valued function $f : \mathcal{X} \subseteq \mathbb{R}^d \rightarrow \mathbb{R}$ in the following way: the feature vector $\mathbf{x} = (x_1, \ldots, x_d)'$ from the feature space $\mathcal{X}$ is assigned to a positive class if $f(\mathbf{x}) \geq \theta$, and otherwise to the negative class. In the context of biometric authentication, the extracted feature $\mathbf{x}$ (output of an extractor) is a biometric feature and f_p be the p-th classifier from an ensemble of classifiers $p = 1, \ldots, P$.

[1] A brief introduction has already been given in Section 1.5.2.

4.2.2 The ECOC procedure

The ECOC procedure can be summarised in Algorithm 2.

Algorithm 2 Train and test ECOC

1. Build an ECOC matrix $\mathbf{Z} \in \{0,1\}^{K \times P}$
   where 0 indicates a negative class and 1 indicates a positive class.
2. During train phase: Train f_p, $\forall_p$ on each of the class labels $\mathbf{Z}_p$.
   where $\mathbf{Z}_p$ is a given column of the matrix $\mathbf{Z}$
3. During test phase: Execute

$$\hat{F} = \text{arg}min_k(\sum_{p=1}^{P} |\mathbf{Z}_{k,p} - f_p|) \tag{4.1}$$

Step 1 is the problem of designing an ECOC code matrix. In the one-versus-all (also called one-per-class) method, the matrix is an identity matrix. Building this matrix corresponds to decomposing a K-class problem into P binary-class problems. This is the *decomposition* phase. Step 2 is a training process of P binary classifiers. Step 3 is the execution of P binary classifiers. This is the *recomposition* phase. Note that "arg" in "argmin_k" is a function that returns the index or argument k of the minimum value. In this case, it returns the class label. This notation applies to "argmax_k" which returns a class label k having the maximum value.

4.2.3 ECOC and existing approaches

In the precedent section, a matrix $\mathbf{Z} \in \{0,1\}^{K \times P}$ is used. However, it could be any matrix such as $\mathbf{M} \in \mathbb{R}^{k \times p}$. Then the decision rule will be:

$$\hat{F} = \text{arg}min_k(\sum_{p=1}^{P} |\mathbf{M}_{k,p} - f_p|) \tag{4.2}$$

We call this the *minimum rule* distance criterion. It can be easily seen that $\mathbf{Z}_{k,p} \subset \mathbf{M}_{k,p}$. However, it is also possible to construct a decision rule such as follows:

$$\hat{F} = \text{arg}max_k(\mathbf{M} \bullet f_p) \tag{4.3}$$

$$= \text{arg}max_k(\sum_{p=1}^{P} \mathbf{M}_{k,p} \times f_p) \tag{4.4}$$

We call this the *maximum rule* distance criterion.

How one could solve the problem of $\hat{F}$? If one freezes f_p and tries to solve the matrix $\mathbf{M}_{k,p}$ numerically, then Equation 4.4 is a linear discriminant function. If one introduces a sigmoid function, denoted as sig, into Equation 4.4, such as the following:

$$\hat{F} = \text{arg}max_k \text{sig}(\sum_{p} \mathbf{M}_{k,p} \times f_p), \tag{4.5}$$

one gets a typical linear Perceptron. The ECOC approach fixes $\mathbf{M}_{k,p}$ and then constructs each classifier f_p later with the matrix $\mathbf{Z}_{k,p} \in \{0,1\} \bigcup \{*\}$ or $\mathbf{Z}_{k,p} \in \{-1,+1\} \bigcup \{*\}$, instead of using $\mathbf{M}_{k,p} \in \mathbb{R}$. $\{*\}$ denotes a "don't care" class. That means the feature vector $\mathbf{x}$ belonging to the class k in $\mathbf{M}_{k,p}$ is excluded from the training set of f_p. In the current implementation, $\mathbf{Z}_{k,p} \in \{0,1\}$ or $\mathbf{Z}_{k,p} \in \{-1,+1\}$. Section 4.2.4 will discuss more the difference between the minimum and maximum distance criteria together with the choice of $\mathbf{Z}_{k,p} \in \{0,1\}$ or $\mathbf{Z}_{k,p} \in \{-1,+1\}$ in specific classifier implementations, particularly that of SVMs and MLPs.

4.2.4 ECOC and discriminative classifiers

An MLP can have several possible implementation of neuron activation function: *sigmoid*, *tanh* [2] or linear.

1. A *sigmoid* neuron has an output value in $[0, 1]$. It can be used to estimate the *a posteriori* probability, which is the probability of an observed feature $\mathbf{x}$ belonging to the class c_k, i.e., $p(c_k|\mathbf{x})$. In this case, $\mathbf{Z}_{k,p} \in \{0, 1\}$ should be used to train all f_p.

2. If the output neuron has a *tanh* activation function, then the neuron output will have a value $\in [-1, +1]$. Consequently the corresponding matrix $\mathbf{Z}_{k,p} \in \{-1, +1\}$ should be used.

3. If the output neuron has a linear activation function, then the output value is $\in [-\infty, +\infty]$. However, the matrix can still remain in $\mathbf{Z}_{k,p} \in \{-1, +1\}$.

What is the difference between Item 1 and Item 2? What decision rules should be used? Which classifiers (MLPs or SVMs) should be used? These are open questions that this section tries to solve theoretically.

We first define two variables: $p^+(k)$ and $p^-(k)$ as follows:

$$p^+(k) \in \{m|m \in \mathbf{Z}_{k,p}, m = \text{positive class}\}, \tag{4.6}$$
$$p^-(k) \in \{m|m \in \mathbf{Z}_{k,p}, m = \text{negative class}\}. \tag{4.7}$$

In other words, $p^+(k)$ is the number of positive class for row (or class) k and $p^-(k)$ is the number of negative class for row k. $|\{p^+(k)\}|$ counts the number of positive binary classes and $|\{p^-(k)\}|$ counts the number of negative binary classes in class k. For the purpose of clarity, we drop the k so they can be written as p^+ and p^-.

By introducing p^+ and p^- into the minimum rule as shown in Equation 4.2, one obtains:

$$\begin{aligned}
\hat{F} &= \text{arg}min_k(\sum |\mathbf{Z}_{k,p} - f_p|), \\
&= \text{arg}min_k(\sum_{p^+}(p^+ - f_{p^+}) + \sum_{p^-}(f_{p^-} - p^-)), \\
&= \text{arg}min_k(\sum_{p^+} p^+ - \sum_{p^+} f_{p^+} + \sum_{p^-} f_{p^-} - \sum_{p^-} p^-), \\
&= \text{arg}min_k(|\{p^+\}| - \sum_{p^+} f_{p^+} + \sum_{p^-} f_{p^-} - |\{p^-\}|), \\
&= \text{arg}min_k(\sum_{p^-} f_{p^-} - \sum_{p^+} f_{p^+} + |\{p^+\}| - |\{p^-\}|), \\
&= \text{arg}max_k(\sum_{p^+} f_{p^+} - \sum_{p^-} f_{p^-} + |\{p^-\}| - |\{p^+\}|).
\end{aligned} \tag{4.8}$$

In the same way, by introducing p^+ and p^- into the maximum rule as shown in Equation 4.4, one obtains:

$$\begin{aligned}
\hat{F} &= \text{arg}max_k(\sum \mathbf{Z}_{k,p} \times f_p), \\
&= \text{arg}max_k(\sum_{p^+} p^+ f_{p^+} + \sum_{p^-} f_{p^-} p^-).
\end{aligned} \tag{4.9}$$

Now, there are two cases here: $p^+, p^- \in \{0, 1\}$ or $p^+, p^- \in \{-1, +1\}$:

[2] *sigmoid* is also called sigmoid logistic function and *tanh* is called tangent hyperbolic function.

Case 1: $p^+, p^- \in \{0, 1\}$
In making assumption that $p^+, p^- \in \{0, 1\}$, Equation 4.8 becomes:

$$\hat{F} = \text{arg}max_k(\sum_{p^+} f_{p^+} - \sum_{p^-} f_{p^-} - \sum_{p^+} 1), \tag{4.10}$$

since $|\{p^-\}| = 0$. Note that $\sum_{p^+} 1 = |\{p^+\}|$ because $p^+ = 1$. Inversely, $\sum_{p^-} = |\{p^-\}| = 0$ because $p^- = 0$

In making assumption that $p^+, p^- \in \{0, 1\}$, Equation 4.9 becomes:

$$\hat{F} = \text{arg}max_k(\sum_{p^+} f_{p^+}), \tag{4.11}$$

since $|\{p^-\}| = 0$.

Case 2: $p^+, p^- \in \{-1, +1\}$
In making assumption that $p^+, p^- \in [-1, +1]$, Equation 4.8 becomes:

$$\hat{F} = \text{arg}max_k(\sum_{p^+} f_{p^+} - \sum_{p^-} f_{p^-} + \sum_{p^+} 1 - \sum_{p^-} 1). \tag{4.12}$$

In making assumption that $p^+, p^- \in \{-1, 1\}$, Equation 4.9 becomes:

$$\hat{F} = \text{arg}max_k(\sum_{p^+} f_{p^+} - \sum_{p^-} f_{p^-}). \tag{4.13}$$

We can summarise the result in Table 4.1:

Table 4.1: Summary of comparison between methods and assumptions

Assumptions	min rule	max rule
$p^+, p^- \in \{0, 1\}$	$\text{arg}max_k(\sum_{p^+} f_{p^+} - \sum_{p^-} f_{p^-} - \sum_{p^+} 1)$	$\text{arg}max_k(\sum_{p^+} f_{p^+})$
$p^+, p^- \in \{-1, 1\}$	$\text{arg}max_k(\sum_{p^+} f_{p^+} - \sum_{p^-} f_{p^-} + \sum_{p^+} 1 - \sum_{p^-} 1)$	$\text{arg}max_k(\sum_{p^+} f_{p^+} - \sum_{p^-} f_{p^-})$

Table 4.1 shows that it if $\forall_k(\sum_{p^+} 1 = c)$, i.e., the matrix $\mathbf{Z}$ has the same number of elements of positive class (a value of 1) in each row (c is a constant), both the maximum and minimum distance criteria will not introduce any bias. As a result, Table 4.1 can be reduced to Table 4.2. This additional assumption also implies that $\forall_k(\sum_{p^-} 1 = c)$ for an arbitrary constant c.

Table 4.2: Summary of the decision function $\hat{F}$ using minimum and maximum criteria assuming that all rows have equal number of ones

Assumptions	min rule	max rule
$p^+, p^- \in [0, 1]$	$\text{arg}max_k(\sum_{p^+} f_{p^+} - \sum_{p^-} f_{p^-})$	$\text{arg}max_k(\sum_{p^+} f_{p^+})$
$p^+, p^- \in [-1, 1]$	$\text{arg}max_k(\sum_{p^+} f_{p^+} - \sum_{p^-} f_{p^-})$	$\text{arg}max_k(\sum_{p^+} f_{p^+} - \sum_{p^-} f_{p^-})$

Table 4.2 shows that when $\mathbf{Z}$ has the same number of positive classes in each row, then, the minimum rule is *exactly* equivalent to the maximum rule in the case $p^+, p^- \in [-1, 1]$ and this is not the case when $p^+, p^- \in [0, 1]$.

An intuitive matrix that satisfies $\forall_k(\sum_{p^-} 1 = c)$ is an identity matrix. This is the typical one-versus-all approach. Therefore, the one-versus-all approach is a particular example of an ECOC matrix as well. The design of a valid ECOC matrix will be discussed in the Section 4.2.5.

To combine ECOC with SVMs, one needs to take into account the fact that the output of an SVM f_p is a distance from the decision boundary (also called a margin). A positive distance ($f_p > 0$) means a positive response and vice-versa. Therefore, $f_p \in [-\infty, +\infty]$. In this case, the maximum rule should be used.

Table 4.3: Recommended output values, ECOC matrix and decision criteria when all rows in the ECOC matrix has equal number of ones.

Classifier	MLP-sigmoid	MLP-tanh	MLP-linear or SVM
Output value	$[0,1]$	$[-1,+1]$	$[-\infty,+\infty]$
ECOC matrix	$\mathbf{Z}_{k,p} \in \{0,1\}$	$\mathbf{Z}_{k,p} \in \{-1,+1\}$	$\mathbf{Z}_{k,p} \in \{-1,+1\}$
Decision criteria	min rule	min = max rule	max rule

4.2.5 Designing the ECOC matrix

The goal of the ECOC matrix is to discriminate one class from another. Since, each row of the ECOC matrix represents a class, an intuitive way to increase the discriminating power between classes is to separate them as far as possible using a distance criteria. In this section, the goal is to define criteria for a "good" ECOC matrix. These criteria are:

1. **Large row separation.** Each row $\mathbf{Z}_k$ in an ECOC matrix should be separated as far as possible from other rows according to Hamming distance criterion, i.e., $a \leq |\mathbf{Z}_k - \mathbf{Z}_{k'}| \leq b, \forall_{k \neq k'}$. a and b are the lower and upper covariance distance between any class c_k and $c_{k'}$. It is important that $a > 0$. Furthermore, it is desirable to maximise a and b using the minimum number of bits available. The lower the number of bits, the lower the number of classifiers needed and thus the more efficient the recognition can be. Accuracy and efficiency are the trade-off. If $a = b$, which implies that $|\mathbf{Z}_k - \mathbf{Z}_{k'}| = a = b$, then the matrix $\mathbf{Z}$ is called an equal distance matrix. This is desirable but unfortunately it is not always possible to minimise the bit-length and to maintain the property of equal distance.

2. **Large column separation.** Column separation is necessary to ensure that neither a column is complementary of the other nor it is similar to the others (correlated). Certain classifiers like C4.5 decision tree will produce the exact tree structure when a problem is divided into a two-class problems [9]. If two columns are highly correlated, the classifier produced may behave similarly and thus may make the same mistakes. This criterion *does not apply* to unstable classifiers like neural networks but *does apply* to SVMs. Indeed, repeating columns in ECOC using MLPs as the base classifiers may improve the result due to the effect of the simple neural networks ensembles [20] (See also Section 1.5.2).

There are three techniques to generate ECOC matrix. These methods can be found in [9]. They are:

1. **Exhaustive.** The exhaustive search is a simple algorithm that searches all possible combinations of binary configurations for a given number of classifiers. An example of 11-class exhaustive code is shown in Figure 4.1.

2. **ECOC.** It uses the ECOC table published by [26] as is.

3. **Random.** Given the number of rows and columns of a matrix, this algorithm randomly generates bits of ones and zeros with equal probability.

The three algorithms mentioned above produce a matrix that is either too large in columns (as is the case of the exhaustive method) or low in row separation which can be improved further (as is the case of random method). For the first case, sub-sampling of columns is necessary while for the second case, row separation can be further improved via a *hill-climbing* algorithm. More details can be found in [9].

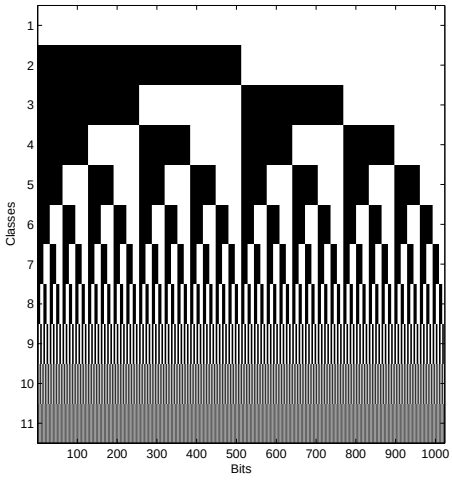


Figure 4.1: An example of 11-class matrix produced using the exhaustive method.

4.3 Empirical study

4.3.1 The vowel database

Firstly, the vowel database is used to carry out the empirical study because it has a benchmark performance (See Appendix A.3) so one has an idea of how accurate the absolute performance should be. The database is used to verify our ECOC algorithm to check for the program correctness and then to understand the genaralised behaviour of the ECOC approach compared with the one-versus-all approach. After this program verification phase, the ECOC program is tested on the XM2VTS database. This is discussed in Section 4.4.

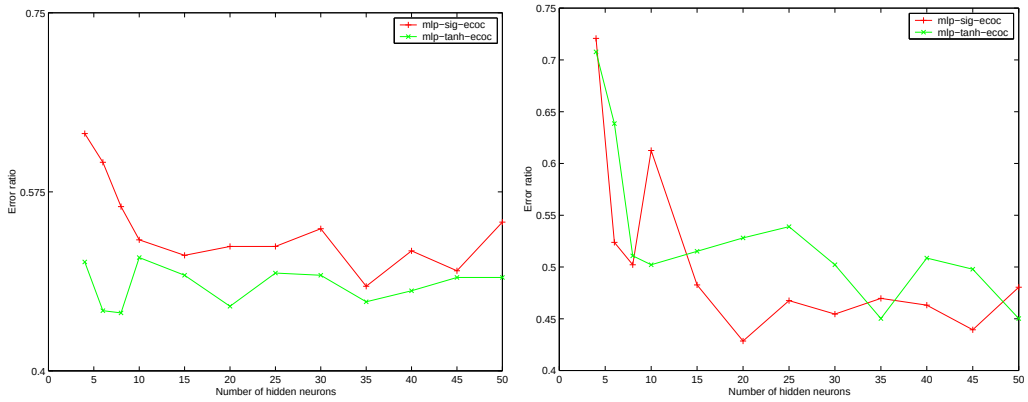
The vowel recognition problem is a speaker independent recognition problem of the eleven steady state vowels of British English using a specified training set of LPC-derived (Linear Predictive Code) log area ratios. The data set has 11 vowels that are spoken by 15 speakers. Each speaker makes 6 utterances of each vowel. Therefore, this makes 990 utterances (15 speakers $\times$ 11 vowels $\times$ 6 utterances). The LPC-derived feature 10 real numbers. The utterances of the first eight speakers are used for training while the utterances of the seven speakers are used *a posteriori* for testing the performance of a given model. More details can be found in Appendix A.3.

4.3.2 ECOC architecture

This section examines the choice between a single MLP with P output neurons (corresponding to P bit length of the ECOC matrix) and P MLPs with 1 output neuron. We call the first case a single-machine architecture and the second case a separate-machine architecture. In the single-machine architecture, the first layer of MLP connections serves as a non-linear projection that is shared by all MLPs. The second layer of network connections is then responsible to map into the P output space. In the separate-machine architecture, there is no sharing of weight connections as in the single-machine architecture. The advantage of the separate-machine architecture is that it allows parallel and independent training.

To test if one architecture is better than the other in terms of classification performance, we propose to use both the architectures with two types of activation functions, i.e., *tanh* and *sigmoid*, on the vowel database described above. Different numbers of hidden neurons are used. The results are plotted in Figure 4.2. The Y-axis is the error ratio of the validation set. Lower error ratio is desirable. It is observed that one cannot judge whether a particular architecture is better than the other except by experimentation. It should be noted that the single-machine architecture may suffer from what is called the "herd" phenomenon, i.e., the weights may be influenced by features

of different categories. This may cause convergence be more difficult than the separate-machine architecture. Unfortunately, the degree of difficulty of convergence between the two architectures is not studied here.



(a) separate-machine architecture (b) single-machine architecture

Figure 4.2: Generalisation of ECOC using different architecture on the vowel database.

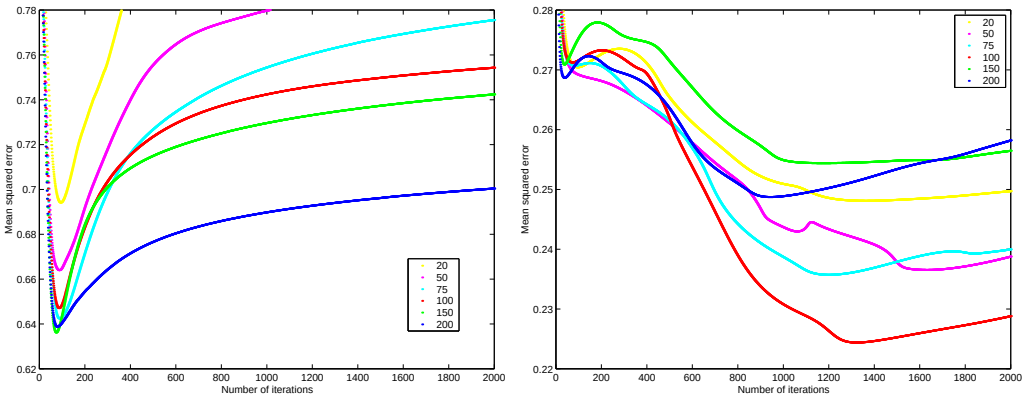
In the following experiments, single-machine architecture is used because of the efficiency in terms of memory and computation.

4.3.3 Generalisation using MLP-based ECOC

Training MLPs involves tuning many parameters, such as the early stopping criterion and the number of hidden neurons. In the vowel database, there are only two sets. The training set is used to change the MLP weights and the validation set is used to tune the parameters. There is no proper "test" set. For a good guess of the optimal number of hidden neurons, several MLPs are trained and tested on the second set (called validation set henceforth).

Figure 4.3 shows the mean squared error (MSE) on the validation set as a function of the number of iterations for both the ECOC approach and the one-versus-all approach. To compare these two approaches using the vowel database, all other factors are fixed, i.e. the learning rate (0.0001) and the number of hidden neurons (20, 50, 75, 100, 200). All MLPs have 10 input neurons and 1023 output neurons (corresponding to the bit-length of the ECOC matrix generated using the exhaustive approach). The matrix is shown in Figure 4.1. It can be noted that the MSE curve on the validation set of the ECOC at first monotonically decreases, then reaches its bottom and monotonically increases. Such regular monotonous pattern that decreases until reaching a minimum point before rebouncing is not observed in the one-versus-all approach. Excessive number of output neurons in the ECOC approach may explain why the MSE is smoother. When the MSE on the validation set increases, one should stop the training to avoid the phenomena of "over-fitting", i.e., the MLP model fits tightly to the training data and thus performs worse in terms of recognition rate on the test set.

Figure 4.4 shows the classification error between the ECOC and the one-versus-all approach as a function of the number of hidden neurons. It can be observed that ECOC converges faster than the one-versus-all approach and then the phenomena of "overfitting" in ECOC takes over. When this happens, the classification error increases. Although the one-versus-all approach has "bumpy slops" of MSE on the validation set, continuing the training can still improve the recognition rate until the point where the one-versus-all approach becomes better than the ECOC approach. Note that the average mean squared error of each output node (y-axis of Figure 4.4 has higher values than that of the one-versus-all approach. This shows that each sub-classifier in the ECOC approach in fact does not solve its sub-problem better than the classifiers in the one-versus-all approach. Despite this fact, the classification error of the ECOC approach is not always worse than the one-versus-all approach.



(a) ECOC (b) one-versus-all

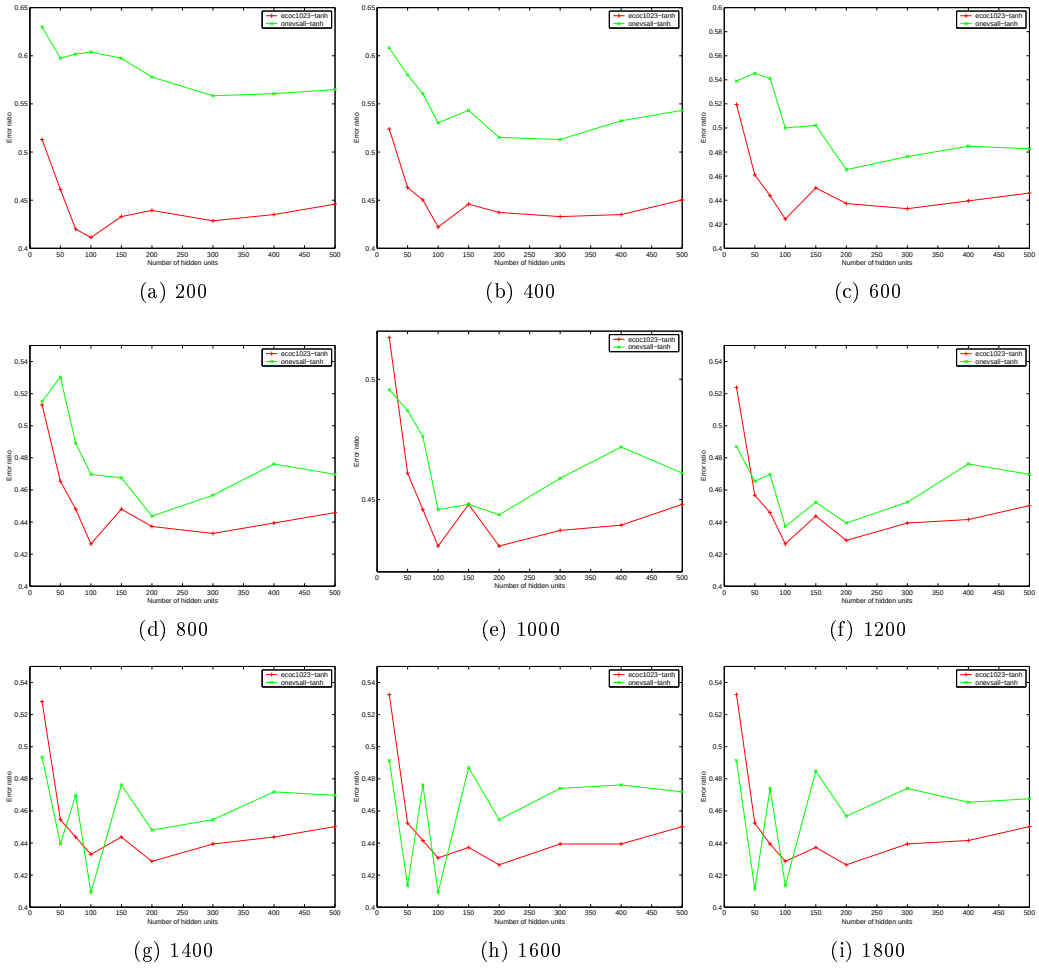
Figure 4.3: The MSE of ECOC and one-versus-all MLP-tanh over different iterations on the validation set of the vowel database

The ECOC approach corrects the bias due to parameterisation of MLPs. However, until now, there is no theoretical proof that the model constructed using the ECOC approach is *always better* than the one constructed using the one-versus-all approach. In other words, by using ECOC, one has a better chance to obtain an MLP model with good generalisation even though the parameters of its sub-classifiers may be sub-optimal.

4.3.4 Generalisation using SVM-based ECOC

Since SVM is a binary classifier, the only possible implementation of SVM with the ECOC approach is to use the separate-machine architecture. Both the one-versus-all and ECOC approaches are compared using the Gaussian kernel with different sigma values: 0.10, 0.20, 0.40, ,0.60, 0.80, 1.00, 1.20, 1.40, 1.50, 1.75, 2.00, 2.25 and 2.50.

The results are shown in Figure 4.5. As the sigma value increases, the Gaussian kernel-SVM will have a lesser capacity, i.e., having lesser flexibility on the decision boundary function. It can be seen that in the vowel database, the ECOC approach deteriorates when the sigma increases. The same sigma value is applied to all binary sub-classifiers in ECOC. Since each binary classifier has a more difficult task to learn in the ECOC approach, the ECOC approach deteriorates faster compared with the one-versus-all approach as sigma increases.



(a) 200 (b) 400 (c) 600

(d) 800 (e) 1000 (f) 1200

(g) 1400 (h) 1600 (i) 1800

Figure 4.4: Generation on validation set of the vowel database during different iterations

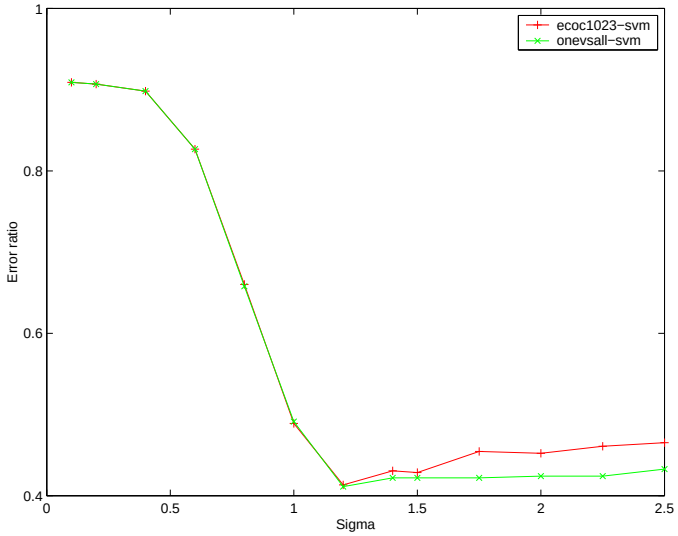


Figure 4.5: The Error ratio of SVM with different Gaussian kernels tested on the validation set of the vowel database

4.4 Experiments on XM2VTS database

4.4.1 About the database

The XM2VTS database [19] is divided into three sets: training set, evaluation set, and test set. The training set is used to build client models, while the evaluation set is used to compute the decision (by estimating thresholds for instance, or parameters of a fusion algorithm). Finally, the test set is used only to estimate the performance of different verification algorithms. These sets are determined by two Lausanne Protocols. In this experiment, only Lausanne Protocol I is used.

The database has 295 persons, 4 recording sessions, and for each recording sessions, there are two repetitive shots. The database was randomly divided into 200 clients, 25 evaluation impostors, and 70 test impostors. More details can be found in Appendix A.1.

4.4.2 ECOC with MLP classifiers

Hyper-parameter selection

The ECOC approach is compared against the baseline one-versus-all approach. For both configuration, the MLPs used have 156 inputs, corresponding to 156 principal components, *nhu* (a variable) hidden units and 511 output units, corresponding to 511 bit-length ECOC matrix generated using the ECOC method discussed in Section 4.2.5. *nhu* is a variable to be determined empirically using initially the following values: 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 60, 80 and 100. Out of these configurations, we hope that an optimised number of hidden units will be chosen for the given particular approach. When training these networks, two conditions are used as stopping criteria: minimum mean squared error = 0.0001 (for both approaches) or the maximum number of iteration = 100 for the one-versus-all approach and 50 for the ECOC approach. These measures are taken so as to avoid over-fitting and under-fitting so that the trained network models are optimised. At the end of the training, the classification error of the evaluation set is measured. These results are shown in Figure 4.6.

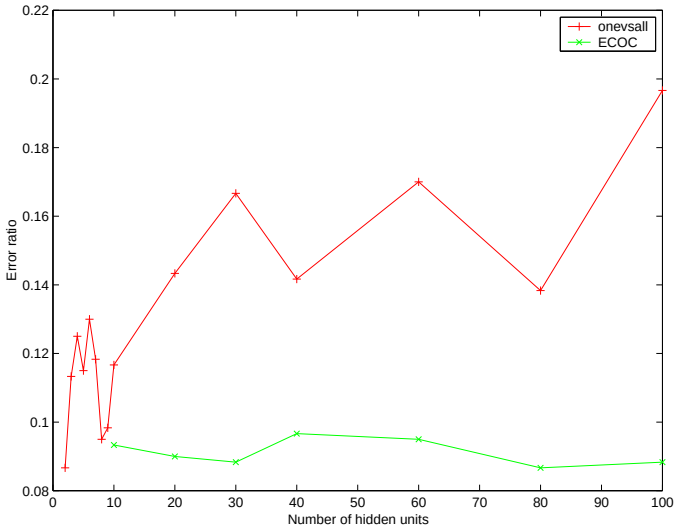


Figure 4.6: Hyper-parameter selection with validation set on the XM2VTS Lausanne Configuration I

Figure 4.6 shows that optimised hyper-parameter for the ECOC approach in this particular database has 30 and 80 hidden neurons and both have a classification error of 0.0883 and 0.0867 respectively on the validation (or evaluation) set. As for the one-versus-all approach, there are 3 possible solutions: 2, 8 and 9 hidden neurons, each has a classification error of 0.0867, 0.1133 and 0.0983 on the validation set.

A posteriori recognition result

Having selected the optimised hyper-parameter for the one-versus-all and ECOC approach, the a priori DET curves of these four trained network models are plotted based on the evaluation set. The optimised threshold parameter for each model is selected based on the Equal Error Rate criteria. Based on the threshold of each model, the a posteriori FAR, FRR and HTER of each model are calculated on the test set. Figure 4.7 shows the DET curves of the following MLP models on the LPI Evaluation and Test set: 156-2-200, 156-8-200 and 156-9-200 for the one-versus-all approach; and 156-30-511 and 156-80-511 for the ECOC approach. It is desirable that the DET curve be nearer to the origin so as to minimise the FAR and FRR at the same time. The a posteriori FAR, FRR and HTER of the five MLP models on the test sets are shown

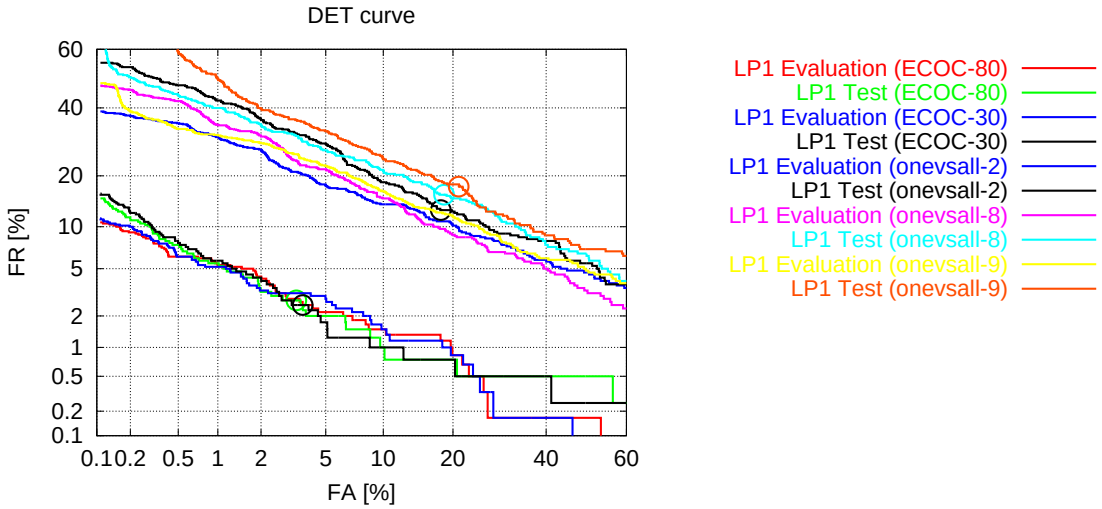


Figure 4.7: DET curves of several MLP models

in Table 4.4. It can be seen that the MLP-based ECOC approach achieves an HTER of 3.056% and 3.085 when using 80 and 30 hidden neurons respectively. The most optimised network model for the one-versus-all approach is 156-2-200 (denoted as onevsall-2 in Figure 4.7) and it achieves an HTER of 15.335%. This shows that the MLP-based ECOC approach performs better than the MLP-based one-versus-all approach in this particular database. Due to heavy computations, these experiments were not carried out several times to obtain the average HTER of the runs.

4.4.3 ECOC with SVM classifiers

Hyper-parameter selection

In this section, the SVM is used as ECOC sub-classifier. All the SVMs with Gaussian kernel have the same sigma values. To find the optimal sigma value of the XM2VTS test set (using Lausanne Protocol I or LPI) for the Gaussian kernel, the following "rules of thumbs" are used:

1. Start with a "small" sigma value. Train the SVMs.
2. Increase the sigma value. Train the SVMs. The classification error is expected to decrease, until a global minimum of classification error is reached.
3. Increase the sigma value further, until the classification error increases again.
4. Choose the first few "optimal" sigma values. A sigma value is considered optimal when it has among the lowest classification error on the validation set. If there exists several sigma

values with the same classification error, small sigma values are given priority because the smaller a sigma value, the higher it's capacity and vice-versa.

With the above procedure, the following sigma values are applied to the SVM-based ECOC on the validation set: 5, 8, 10, 12, 15, 20, 25, 30, 35, 40 and 45 for the ECOC approach. Similarly the following sigma values are applied on the SVM-based one-versus-all approach: 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 75, 80, 85, 90, 100, 110, 120, 130 and 140. Figure 4.8 shows the error ratio on the evaluation set of LPI. It shows that the probable optimal sigma value is 15 for the ECOC approach. To probable optimal sigma values for the one-versus-all approach are: 15, 20 and 25. Although not shown here in this report, the following sigma values are also used in the one-versus-all approach: 40, 70 and 80. They give very poor a posteriori HTER on the evaluation set, i.e, 36.82%, 37.52% and 37.33% respectively.

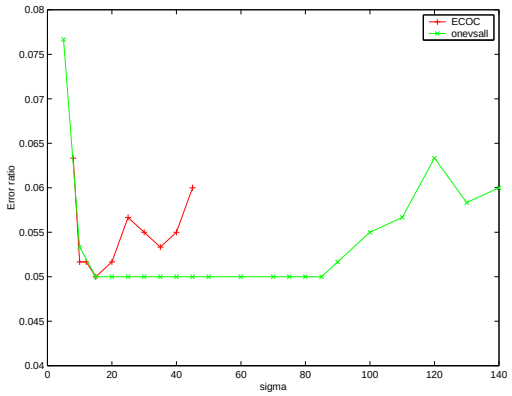


Figure 4.8: Hyper-parameter selection of ECOC and one-versus-all with SVMs tested on the evaluation set of the XM2VTS using LPI

A posteriori recognition result

The selected optimal SVM-based ECOC with Gaussian kernel (sigma value = 15) is tested on the test set of LPI. The result is shown in Figure 4.9. The circles show the point where a threshold is optimised with the EER criterion based on the evalaution (or validation) set. This point corresponds to *a posteriori* false acceptance on the X-axis and false rejection on the Y-axis on the test set. These values are shown in Table 4.4 so they can be compared with the MLP-based ECOC method.

It can be seen that Gaussian kernel SVM-based ECOC is better than MLP-based ECOC. SVM is theoretically optimal in constructing a decision boundary given that the model hyper-parameter, i.e., the kernel has "adequate" capacity.

It can be observed that for both SVM and MLP base classifiers, the ECOC approach performs better than the one-versus-all approach. Furthermore, SVM-based ECOC approach performs slightly better than MLP-based ECOC approach.

4.5 Summary

In this chapter, we have studied several commonly used ECOC matrix schemes and distance criteria, typically the minimum (Hamming distance) or maximum rule. We have shown that the number of ones in each row of a valid ECOC matrix could introduce bias and it is desirable to have equal number of ones across each row although this is not a strict requirement. The main bias in the ECOC approach is introduced by inequivalent Hamming distance between each row

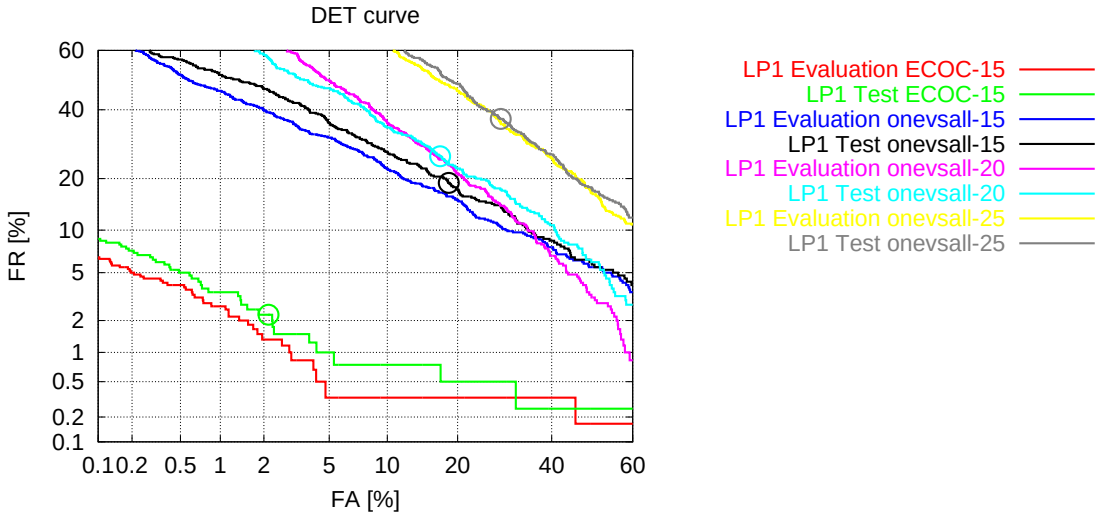


Figure 4.9: DET curve of optimised SVM model

Table 4.4: Performace of 156 PCA face features of XM2VTS database on LP1 configuration

Models	FA[%]	FR[%]	HTER[%]
MLP 156-30-511 (ECOC)	3.669	2.500	3.085
MLP 156-80-511 (ECOC)	3.361	2.750	3.056
MLP 156-2-200 (onevsall)	17.921	12.750	15.335
MLP 156-8-200 (onevsall)	18.452	15.750	17.101
MLP 156-9-200 (onevsall)	21.026	17.500	19.263
SVM Gauss(15) (ECOC)	2.144	2.250	2.197
SVM Gauss(15) (onevsall)	18.483	19.000	18.742
SVM Gauss(20) (onevsall)	17.084	25.750	21.417
SVM Gauss(25) (onevsall)	28.376	37.000	32.688

(representing each class). Such bias does not exist if one uses the exhaustive method to generate the ECOC matrix. However, due to large bit-length of the generated matrix ($P = 2^{K-1} - 1$ bit-length for a K-class problem), one may have to reduce the bit-length. Reducing bit-length will inevitably introduce difference in Hamming distance. Such a Hamming distance among each class can be distributed normally. Thus, reducing the bit-length not will reduce the number of classifiers needed (which results in increased efficiency) but will also introduce bias (which results in consistent bias of inter-distance between two classes, i.e., some pairs of classes could be consistently mis-classified than the others.)

It is found analytically that it is desirable to have the same number of ones in each row of ECOC matrix. Otherwise, this will introduce bias in the distance criteria using Hamming distance (also called minimum rule in this report).

The empirical results show that the ECOC approach does not guarantee to be superior than the one-versus-all approach. However, one knows that ECOC is an effective alternative to diversify classifiers in ensemble methods and can be easily implemented with any base-classifiers, typically stable or non-stable classifiers.

In the case of ECOC with MLPs, it is shown that the parameterisation of MLPs does not affect much the recognition quality as compared to that of one-versus-all. This shows that the ECOC approach can be robust against unoptimised hyper-parameters, i.e., the early stop criteria and the number of hidden neurons.

Chapter 5

Conclusions

5.1 Results of face verification in XM2VTS database using LP1

Table 5.1 shows all the results obtained using LP1 (Lausanne Protocol One) configuration of the face XM2VTS database. It is taken from the following references[24, 22] combined with the results obtained from this report. The following institutes are involved: Aristotle U. of Thessaloniki (indicated as AUT in Table 5.1), University of Sydney (Sydney), University of Surrey (Surrey), IDIAP and LSIIT.

It can be observed that:

- Linear Discriminant Components yields better performance face verification than Principal Components (labeled PCA). (Ref. rows 1 and 3)
- Combining face histogram features with normalised face features yields better performance than using normalised face features alone. (Ref. rows 2 and 6)
- Down-sampling image may not yields better performance. This may be due to lost of information during such process. (Ref. rows 2 and 5)
- SVMs can be more accurate than MLPs. (Ref. rows 3 and 7)
- Local thresholding, i.e., a technique that adapts threshold to specific users could be an easy but effective way to improve the accuracy of a given client specific model. (Ref. rows 4 and 8)
- Jittering produces well generalized classifiers.
- ECOC can improve classification results comparing to one-versus-all approach.

5.2 Results and contributions

A generic biometric authentication system requires the following interdependent sub-components to work: sensors, extractors, classifiers and an optional supervisor. The interdependency among these components could introduce errors and information loss. Therefore, one way to improve a biometric authentication system is to improve any of the sub-components. By taking these independent process as a serial process transmitting a confidence score, it is obvious that creating several such processes could improve the system. Motivated by this idea of "parallelism", we have proposed to create multiple such serial processes through the use of multiple samples and multiple biometrics.

Table 5.1: Performace of XM2VTS face database on LP1 configuration using different features and classifiers. (*) indicates experiments done in this report. N.A. indicates not available.

No.	Institutes	Feature	Classifiers	FA[%]	FR[%]	HTER[%]
1	Surrey	Linear Discriminant Components	MLP (ECOC)	N.A.	N.A.	0.80
2	IDIAP	Normalised face (30×40) + RGB histograms + random-shiftings	MLP one-versus-all	1.50	2.75	2.13
3	IDIAP/ LSIIT*	PCA 156 components	SVM Gauss(15) (ECOC)	2.14	2.25	2.20
4	Surrey	Normalised face image	Linear Discriminant (Local Threshold)	2.30	2.50	2.40
5	IDIAP/ LSIIT*	Normalised face (15×20) + RGB histograms + random-shiftings	MLP one-versus-all	2.89	2.00	2.44
6	IDIAP	Normalised face (30×40) + random-shiftings	MLP one-versus-all	2.36	3.25	2.81
7	IDIAP/ LSIIT*	PCA 156 components	MLP 156-80-511 (ECOC)	3.36	2.75	3.06
8	Surrey	Normalised face image	Linear Discriminant (Global Threshold)	6.50	5.30	5.90
9	AUT	Multi-scale dilation and erosion	Elastic graph matching	8.20	6.00	7.10
10	IDIAP	Gabor features	Ridge and deformable matching	8.10	8.50	8.30
11	Sydney	Fractal codes	Fractal Neighbour Distance (Template matching)	13.60	12.30	12.95

It is proved theoretically and shown empirically that by using N samples of M biometric modalities, one could improve the biometric system by a factor between 1 and $N \times M$ when the supervisor (one that merges classifier scores) is an average operator. This approach is called multi-sample multi-source approach.

To improve the classifiers, the Error-Correction Output-Coding (ECOC) has been chosen among existing ensemble methods, such as bagging and boosting. In this study, several choices of ECOC matrix and discriminant classifiers are considered, i.e., Multi-Layer Percetron and Support Vector Machines. We have observed that the number of ones in each row of an ECOC matrix can introduce bias during decision making. As a result of this theoretical demonstration, we propose that ECOC matrix with -1, +1 elements should be used with MLP with tanh output and SVM. For both of these cases, the ideal distance criteria should be the inner vector product between a row of ECOC matrix and the vector scores of sub-classifiers. The bigger this inner product, the more similar it is between the row of ECOC matrix and the vector scores of sub-classifiers.

Although ECOC has been reported as among the best method across different types of classifiers (Neural Networks, Support Vector Machines, Decision Tree), we have not been able to prove that it can always achieve better performance than the one-versus-all approach. However, it is observed through experiments that the ECOC approach can generalise better than the one-versus-all approach. This thus can provide a protection against sub-optimal model parameters in MLP. The experiment result using 156 principal components of the face feature in the XM2VTS database shows that one can achieve the HTER of 2.75% and 2.50% respectively using MLP-based ECOC and SVM-based ECOC.

We hope that better image- and signal-processing algorithms will emerge in the future so that better extractors could be constructed and applied into the problem of biometric authentication.

5.3 Future research direction in biometric authentication

5.3.1 Problems in biometric authentication

In our opinion, before the biometric authentication system can be delivered to real-life applications, the following areas should be solved:

1. Scalability
   What happens when a biometric authentication system contains a very big population of users (say ten-thousand) ? What is the effect of addition of a new user to the system? Would the whole system need to be retrained? One possible solution is to use a classifier that is capable of handling a very large number of classes. SNoW discussed in Chapter 2 has been identified as one possible solution to the problem of scalability.

2. Reliability
   How one can decide if a system falls below it's optimal performance? When this happens, what kind of fall back procedure can be used? One way to solve the problem of reliability is to add more sensors. Another way is to measure biometric source that can be easily measured such as heights and weights.

3. Stability
   What happens if a feature changes overtime? A biometric system has to be retrained or updated, if possible automatically. A system can use each approved access for further training so that the gradual change of feature overtime can be adapted.

5.3.2 Adaptive discriminative models

One way to adapt the "changes" either due to change over time or adding a new user is to adapt existing model to a newly modified problem. The technique of adaptation is not new in generative model, especially in speech verification. One can adapt a speech model trained using extremely a large data, called a "world model" to a particular user, called a "client model". This technique is called called *client adaptation*. An example of client adaptation technique is called maximum a posteriori. It is used to adapt a trained Gaussian mixture modeling (GMM) after a world model to a specific client by reestimating the mean values of the client GMM.

However, such a technique has not been considered in discriminative model (e.g. MLPs and SVMs). One particular example is to adapt a face detection model into a client-specific face verification model. The hypothesis here is that a face detection model may provide or at least project from a given face feature space to a an "easier" space for classification.

5.3.3 Error Correction Output-Code

Below are some of the potential directions related to ECOC:

1. Sparse version or real-valued ECOC
   ECOC has to function with all the sub-classifiers during an authentication. When a class is added to the problem, all sub-classifiers have to be retrained. Although the accuracy is increased, but even so for the training complexity. One way to increase performance without using all sub-classifiers is to introduce a "don't-care" flag into the ECOC matrix. One other method is to introduce continuous codes into the matrix. All these are open research questions. Some of these questions are tackled by [4].

2. ECOC with other base classifiers
   Winnow is identified as a possible alternative sub-classifier to be combined with ECOC because of its conservativeness and low computation. More ie explained in the next section.

   Motivated by ECOC in discriminative context, to our opinion, one can also enhance GMM with correction capability.

5.3.4 Sparse boolean classifiers

To our opinion, sparse boolean classifiers such as Winnows [32] or MIRA [5] are potentially efficient classifiers to tackle the problem of scalability. They are mistake bound models with feature selection. In both methods, discretization is an important and necessary step so features are translated from numerical to boolean space.

Appendix A

Databases used in experiments

A.1 XM2VTS database

A.1.1 Database description

The XM2VTS database contains synchronized image and speech data as well as sequences with views of rotating heads. The database contains four recording sessions of 295 subjects taken at one month intervals. On each session, two recordings were made, each consisting of a speech shot and head rotation shot (Figure A.1). The speech shot consisted of frontal face and speech recordings of each subject during the pronunciation of a sentence. During the rotating head shot, the subject was asked to rotate his/her head from center to left to right to center; then to rotate his/her head up and then down then back to center. If the subject wore glasses he/she was then asked to remove them for a few seconds.

The database was acquired using a Sony VX1000E digital cam-corder and a DHR1000UX digital VCR. Video was captured at a color sampling resolution of 4:2:0 and 16 bit audio at a frequency of 32 kHz. The video data was compressed at a fixed ratio of 5:1 in the proprietary DV format. In total the database contains approximately 4 TBytes (4000 Gbytes) of data.

When capturing the database the camera settings were kept constant across all four sessions. The head was illuminated from both left and right sides with diffusion gel sheets being used to keep this illumination as uniform as possible. A blue background was used to allow the head to be easily segmented out using a technique such as chromakey. A high-quality clip-on microphone was used to record the speech. One speech shot consisted of three sentences:

1. "0 1 2 3 4 5 6 7 8 9"
2. "5 0 6 9 2 8 1 3 7 4"
3. "Joe took fathers green shoe bench out"

The use of digits was chosen as this corresponds to a typical application scenario of speaker verification. The three sentences were the same for all speakers to allow the simulation of impostor accesses by all subjects. The second digit utterance was chosen to compensate for prosodic and co-articulation effects. The third item was supposed to represent a phonetically balanced sentence.

A.1.2 Experimental protocol

A protocol has been defined [19] to evaluate the performance of vision- and speech-based person authentication systems on the XM2VTS database. The use of a common protocol should allow the comparison of different methods.

The database was divided into three sets: training set, evaluation set, and test set (see Figure A.2). The training set was used to build client models, while the evaluation set was used to

Figure A.1: Face images of the same persons during different sessions and shots.

compute the decision (by estimating thresholds for instance, or parameters of a fusion algorithm). Finally, the test set was used only to estimate the performance of different verification algorithms.

The protocol was based on 295 subjects, 4 recording sessions, and two shots (repetitions) per recording sessions. Only the first two digit sequences were used for each shot. The database was randomly divided into 200 clients, 25 evaluation impostors, and 70 test impostors (See [19] for the subjects' IDs of the three groups). Two different evaluation configurations were defined. They differ in the distribution of client training and client evaluation data as can be seen in Figure A.2. Both the client training and client evaluation data were drawn from the same recording sessions for Configuration I which might lead to optimistic performances on the evaluation set. For Configuration II on the other hand, the client evaluation and client test sets are drawn from different recording sessions which might lead to more realistic results.

The following number of subjects were used:

- Clients: 200
- Impostors - Evaluation: 25
- Impostors - Test: 70

This led to the following statistics (see also Figure A.2 for the partitions):

- 1. client training examples: Conf. I: 3 per client, Conf. II: 4 per client
- 2. evaluation - clients: Conf. I - 600, Conf. II - 400
- 3. evalution - impostors: 40'000 (25 * 8 * 200)
- 4. test client accesses: 400 (200 * 2)
- 5. test impostor accesses: 112'000 (70 * 8 * 200)

A.2 LSIIT database

A.2.1 Database description

The LSIIT database contains face images and voice passwords of thirty clients. It is used to simulate an biometric authentication scenario of a multi-modal biometric system in a secured

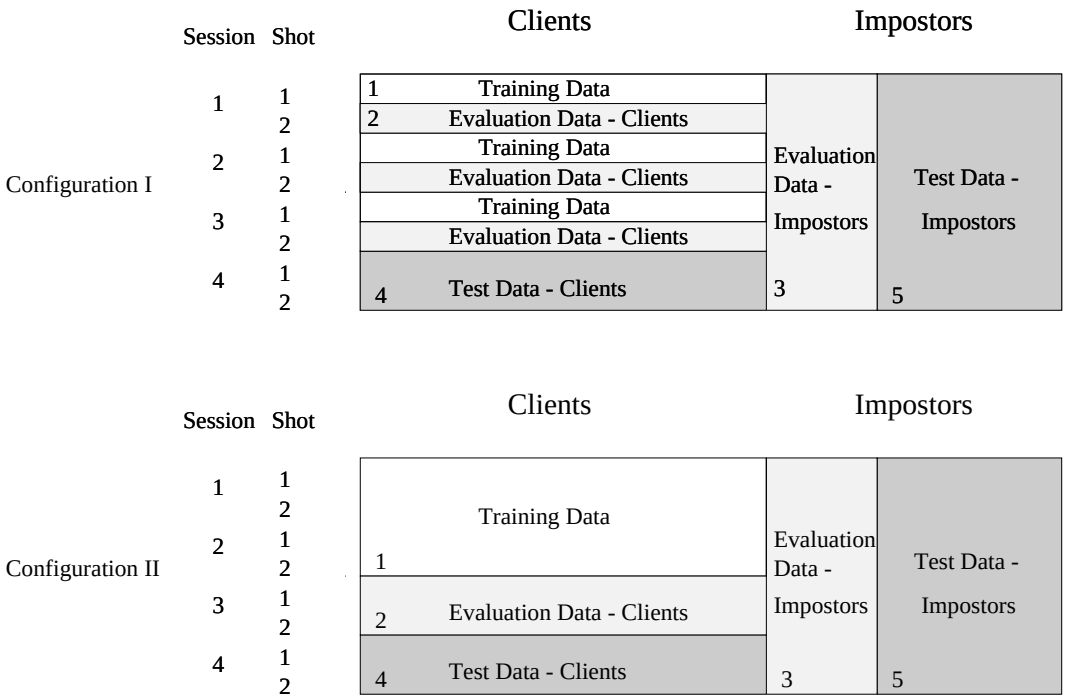


Figure A.2: Diagram showing the partitioning of the XM2VTS database according to the protocol Configurations I (top) and II (bottom).

workplace with a small population of users. It is made publicly available at http://hydria.u-strasbg.fr/∼norman/BAS.

A generic PC web cam, i.e., Creative WebCam Bluster II, is used for sampling a 320 × 240 RGB image. Within thihs area of viewing, a face image is cropped out to the dimension of 150 × 225. The cropped out image is saved in Windows' 24-bit bitmap (BMP) format. When taking the photo, the person is requested to move his face into the area of interest where the cropped face image is expected. The recorded image contains an upright frontal image.

Under Windows' system, each voice password is sampled for a duration of three seconds at 8K Hz on a mono-channel microhpone. The data is saved in a wave (WAV) file format with approximately 24K bytes. The password of each client could be any short word such as his or her name. The voice-scan is taken in the laboratory environment to model a typical indoor environment. No effort is made to make the problem more challenging or particularly easier.

A.2.2 Feature Extraction

Face features

The face image is extracted using Principle Component Analysis (PCA). From the three hundred images available, the PCA covariance matrix is calculated. It is based on the difference between each image with the average image. In PCA, the Eigenvectors of the covariance matrix are sorted in decreasing order of it corresponding Eigenvalues. 144 components are selected, roughly corresponding to 95 percents of accumulated normalised Eigenvalues. The images are then projected linearly into the PCA space to obtain a feature dimension of 144. Detailed description on Eigenface feature extraction can be found in [38].

Voice features

The front end of the speech module aims to extract the user dependent information. It includes three important steps: speech acquisition, detection and extraction. In general, the users vocal password is sampled via a microphone at 8 kHz over a period of 3 seconds. In the second step, the presence of speech is then detected and then extracted using Morlet wavelets. A more detailed description of this work can be found in [27].

In our experiments, a wavelet transform on a speech signal of 3 seconds gives 8 analysable scales. By using signal-to-noise analysis on the wavelet coefficients scale, we were able to determine that wavelets of scale-1, 2, 3 and 4 are more significant than other scales. Each of these scales is then truncated, normalised and then sampled before being merged to form a vector of 64 values. This is shown in Figure A.3. Through this sampling process, some important data could be lost. Such data reduction is necessary to make sure that the final vector is small enough to train the neural network [27].

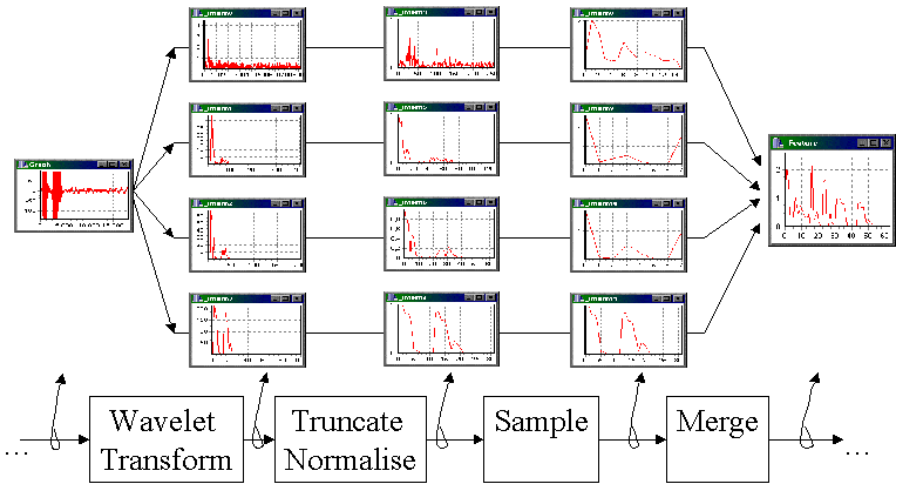


Figure A.3: A generic biometric taxonomy and fusion scheme

A.2.3 Experimental protocol

For each of the face and voice biometric modalities, five out of ten samples of each person are used for training and the other five samples are used for testing. These samples are randomly shuffered. Both the training and test sets are mutually exclusive. During testing with the previously unseen 5 samples, the experiment is first carried out with 1 sample, then 2 samples, and so on up to 5 samples.

A.3 Vowel Recognition databse

A.3.1 Database description and methodology to experimentation

The vowel recognition problem is a speaker independent recognition problem of the eleven steady state vowels of British English using a specified training set of LPC (Linear Predictive Coding) derived log area ratios. The data was prepared by David Deterding, Mahesan Niranjan and Tony Robinson in the late 1980's.

The data set has eleven vowels that are spoken by fifteen speakers. Each speaker utters six times the eleven vowels. Therefore, this makes 990 utterances (15 speakers $\times$ 11 vowels $\times$ 6 utterances). For each utterance, there are ten real numbers. The utterances of the first eight

speakers are used for training while the utterances of the seven speakers are used *a posteriori* for testing the performance of a given model.

To measure the performance, the number of test vowels classified correctly, i.e. the number of occurrences when distance of the correct output to the actual output was the smallest of the set of distances from the actual output to all possible target outputs, are recorded for each model.

A.3.2 Benchmark results

Some of the results of using different type of classifiers are shown on Table A.1. These results were obtained by Tony Robinson.

Table A.1: Performance of different classifiers on the Vowel Recognition database

Classifiers	No. of hidden units	Percentage correct
Single-layer perceptron	-	33
Multi-layer perceptron	88	51
Multi-layer perceptron	22	45
Multi-layer perceptron	11	44
Modified Kanerva Model	528	50
Modified Kanerva Model	88	43
Radial Basis Function	528	53
Radial Basis Function	88	48
Gaussian node network	528	55
Gaussian node network	88	53
Gaussian node network	22	54
Gaussian node network	11	47
Square node network	88	55
Square node network	22	51
Square node network	11	50
Nearest neighbour	-	56

Notes:

- Each of these numbers is based on a single trial with random starting weights.
- The numbers given above are final performance figures after about 3000 trials, not the peak performance obtained during the run.

Appendix B

Learning with noise

B.1 Jittering

Learning with noise, also called "jittering". Jitter is artificial noise deliberately added to the inputs during training. Training with jitter is a form of smoothing related to kernel regression. It is also closely related to regularization methods such as weight decay and ridge regression.

Training with jitter works because the functions that neural networks (NNs) try to learn are usually smooth. NNs can learn functions with discontinuities, but the functions must be piecewise continuous in a finite number of regions if NNs is restricted to a finite number of hidden units.

In other words, if one has two cases with similar inputs, the desired outputs will usually be similar. Therefore, one can take any training case and generate new training cases by adding small amounts of jitter to the inputs. As long as the amount of jitter is sufficiently small, it can be assumed that the desired output will not change enough to be of any consequence, so the same target value can be used. The more training cases, the better the result. However, too much jitter will obviously produce garbage, while too little jitter will have little effect [11].

Consider any point in the input space, not necessarily one of the original training cases. That point could possibly arise as a jittered input as a result of jittering any of several of the original neighboring training cases. The average target value at the given input point will be a weighted average of the target values of the original training cases. For an infinite number of jittered cases, the weights will be proportional to the probability densities of the jitter distribution, located at the original training cases and evaluated at the given input point. Thus the average target values given an infinite number of jittered cases will, by definition, be the Nadaraya-Watson kernel regression estimator using the jitter density as the kernel. Hence, training with jitter is an approximation to training with the kernel regression estimator as target. Choosing the amount (variance) of jitter is equivalent to choosing the bandwidth of the kernel regression estimator [35].

B.2 Random transformation as a means to jittering

In the context of biometric authentication, one often does not have enough training data. A user will intially give a very few biometric samples (typically three in the XM2VTS database for LP1 configuration and four for LP2 configuration, see Appendix A.1). Therefore, for a classifier arranged in one-versus-all configuration, only very few positive examples are available. Instead of applying jittering directly, one way is to apply *relevent transformation* on the training data set.

In the face verification of XM2VTS database, these transformations are random shifting, translation, rotation and scaling. These transofrmations are adequate because in reality, the face detection algorithm will detect the face position from an image with a certain amount of accuracy.

B.3 Experiments and Results

By using the XM2VTS dababase, random transformations are applied on the training set of face images. Both the LP1 and LP2 protocols of XM2VTS database are used. The experiment is applied on training data set which has 396 features, corresponding to 300 pixels of normalised face features and 96 values from the RGB histogram values. Each MLP has the following configurations:396 input neurons, 50 hidden neurons and 1 output neuron (396-50-1). Each MLP is trained with 500 iterations and convergence error of 0.0001. These parameters are chosen arbitrarily. The DET curves of LP1 and LP2 are shown in Figure B.1.

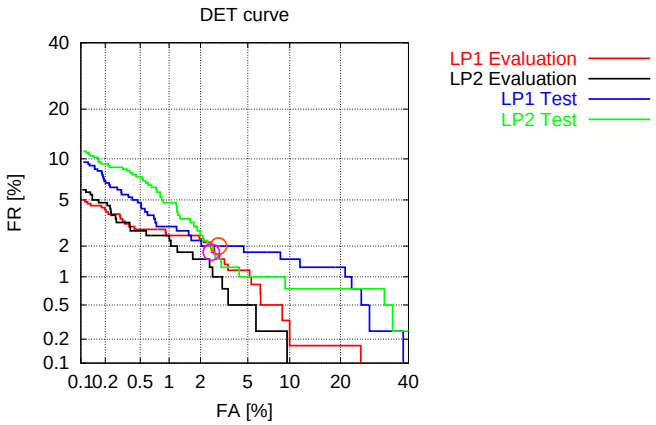


Figure B.1: DET curves of learning with noise

Table B.1 shows the corresponding False Acceptance (FA), False Rejection (FR) and HTER the DET curves in Figure B.1 at the point when FAR=FRR. The threshold in evaluation sets are fixed *a priori* on the evaluation set. This means that the threshold is fixed at the optimum point EER (Equal error rate). Once this threshold is found, this same threshold is applied *a posteriori* on the test set. The resultant HTER is used to compare different algorithms.

Table B.1: Performace of 396 face features of XM2VTS database on LP1 and LP2 configurations

Data sets	Threshold	FA[%]	FR[%]	HTER[%]
Evaluation LP1	*a posteriori*	2.167	2.167	2.167
Test LP1	*a priori*	2.887	2.000	2.444
Evaluation LP2	*a posteriori*	1.666	1.750	1.708
Test LP2	*a priori*	2.496	1.750	2.123

Bibliography

[1] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1999.

[2] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[3] R. Brunelli and D. Falavigna. Personal identification using multiple cues. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, volume 17(10), pages 955–966, 1995.

[4] K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. In *Computational Learing Theory*, pages 35–46, 2000.

[5] K. Crammer and Y. Singer. Ultraconservative online algorithms for multiclass problems. In *14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001, Amsterdam, The Netherlands, July 2001, Proceedings*, volume 2111, pages 99–115. Springer, Berlin, 2001.

[6] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines (and other kernel-based learning methods)*. Cambridge University Press, 2000.

[7] U. Dieckmann, P. Plankensteiner, and T. Wagner. Sesam: A biometric person identification system using sensor fusion. In *In Pattern Recognition Letters*, volume 18(9), pages 827–833, 1997.

[8] T.G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15, 2000.

[9] T.G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[10] Y. Freund and R. Schapire. A short introduction to boosting. In *J. Japan. Soc. for Artificial Intelligence*, pages 771–780, 1999.

[11] L. Holmström and P. Koistinen. Using additive noise in back-propagation training. In *IEEE Transactions on Neural Networks*, volume 3(1), pages 24–38, 1992.

[12] L. Hong and A.K. Jain. Multi-model biometrics. In *Biometrics: Person Identification in Networked Society*, 1999.

[13] L. Hong, A.K. Jain, and S. Pankanti. Can multibiometrics improve performance? Technical report msu-cse-99-39, MSU-CSE, 12 1999.

[14] IDIAP. Activity report 2001. IDIAP-COM 01, IDIAP, 2002.

[15] A.K. Jain and S. Pankanti. Biometrics systems: Anatomy of performance. Technical Report MSU-CSE-00-20, Department of Computer Science, Michigan State University, East Lansing, Michigan, September 2000.

[16] J. Kittler, R. Ghaderi, T. Windeatt, and J. Matas. Face identification and verification via ecoc. In *The 3rd International Conference on Audio-Visual Biometric Person Authentication (AVBPA)*, pages 1–13, 2001.

[17] J. Kittler, G. Matas, K. Jonsson, and M. Sanchez. Combining evidence in personal identity verification systems, 1997.

[18] L.I. Kuncheva. A theoretical study on six classifier fusion strategies. In *IEEE Transaction on Pattern Analysis and Machine Intelligence*, volume 24(2), pages 281–286, February 2002.

[19] J. Lüttin. Evaluation protocol for the XM2FDB database (lausanne protocol). Technical Report COM-05, IDIAP, 1998.

[20] R. Maclin and D. Opitz. An empirical evaluation of bagging and boosting. In *AAAI/IAAI*, pages 546–551, 1997.

[21] S.H. Maes and H.S.M. Beigi. Open sesame! speech, password or key to secure your door? In *Asian Conference on Computer Vision, Hong Kong*, pages 8–11, January 1998.

[22] S. Marcel and S. Bengio. Improving face verification using skin color information. IDIAP-RR 44, IDIAP, 2001. Submitted: ICPR02, Quebec City, Canada, 2002.

[23] A. Martin, G. Doddington, T. Kamm, M. Ordowsk, and M. Przybocki. The DET curve in assessment of detection task performance. In *Proceedings of Eurospeech'97, Rhodes, Greece*, pages 1895–1898, 1997.

[24] J. Matas, M. Hamouz, K. Jonsson, J. Kittler, Y. Li, C. Kotropoulos, A. Tefas, I. Pitas, T. Tan, H. Yan, F. Smeraldi, J. Begun, N. Capdevielle, W. Gerstner, S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz. Comparison of face verification results on the xm2vts database. In *Proceedings of the 15th ICPR*, volume 4, pages 858–863, 2000.

[25] L. M. Moreira. *The use of Boolean Concepts in General Classification Contexts*. Department of Mathematics of the EPFL, Lausanne, Switzerland, 200.

[26] W.W. Peterson and E.J. Weldon. *Error-Correcting Codes*. The MIT Press, Cambridge, Massachussetts, 2 edition, 1972.

[27] N. Poh. Biometric authentication system. Master's thesis, USM, Penang, August 2001.

[28] N. Poh and J. Korczak. Biometric authentication: A taxonomy framework. Technical report, LSIIT, December 2001.

[29] N. Poh and J. Korczak. Hybrid biometric authentication system using face and voice features. In *The 3rd International Conference on AVBPA*, pages 348–353, 2001.

[30] A. Ross, A. Jain, and J-Z. Qian. Information fusion in biometrics. In *The 3rd International Conference on Audio-Visual Biometric Person Authentication (AVBPA)*, pages 354–359, 2001.

[31] D. Roth. Learning in natural language. In *International Joint Conferences on Artificial Intelligence*, pages 898–904, 1999.

[32] D. Roth. Learning in natural language. In *The IEEE Int. Conf. of Computer Vision and Pattern Recognition*, 2000.

[33] D.E. Rumelhart, G.E. Hinton, and R.J. William. *Learning internal representations by error propagation*, volume 1. MIT Press, Cambridge, MA., 1986.

[34] B. Scholkopf and J.A. Smola. *Learning with Kernels*. MIT Press, 2002.

[35] D.W. Scott. *Multivariate Density Estimation*. Wiley, 1992.

[36] L.L. Shen, A. Kot, and W.M. Koo. Quality measures of fingerprint images. In *The 3rd International Conference on Audio-Visual Biometric Person Authentication (AVBPA)*, pages 266–271, 2001.

[37] H.L. Van Trees. *Detection, Estimation and Modulation Theory, vol. 1.* Wiley, New York, 1968.

[38] M. Turk and A. Pentland. Eigenfaces for recognition. In *J. of Cognitive Neuroscienc*, volume 3(1), pages 71–86, 1991.

[39] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, 1995.

[40] J.L. Wayman. *BIOMETRICS: Person Identification in Networked Society*, chapter 7. Kluwer Publishers, 1999.