Department of Signal and Image Processing
46, rue Barrault
Paris 75634 Cédex 13
France

A CONTRIBUTION TO MULTI-MODAL IDENTITY VERIFICATION USING DECISION FUSION

by

Patrick Verlinde

Dissertation submitted to obtain the degree of

Docteur de l'Ecole Nationale Supérieure des Télécommunications
Spécialité: Signal et Images

Composition of the thesis committee:

Jean-Paul Haton (LORIA) - *President*
Gérard Chollet (ENST) - *Director*
Marc Acheroy (RMA) - *Reporter*
Isabelle Bloch (ENST) - *Reporter*
Paul Delogne (UCL) - *Examiner*
Josef Kittler (UOS) - *Examiner*
Claude Vloeberghs (RMA) - *Examiner*

September 17th 1999

Dedicated to my wife and my twin daughters

Jasmien, Renate, and Charlotte.

Acknowledgments

In the first place I wish to thank my thesis director dr. Gérard Chollet from CNRS/URA 820 (FR), for his driving force, for his critical and very useful advises, for having involved my research in every suitable project he could find, and for the huge amounts of information he provided me with.

I also would like to thank prof. dr. ir. Jean-Paul Haton from LORIA (FR) for his useful advises he gave me and for having honored me by accepting to be the president of my thesis committee.

Special thanks go to prof. dr. ir. Marc Acheroy, head of the electrical engineering department of the RMA and director of the Signal and Image Centre (SIC) for believing in me, for having supported and motivated me all the time, for his continuous flow of advises, and, last but not least, for having accepted to be a reporter for this thesis.

I also want to express my sincere gratitude towards prof. dr. ir. Isabelle Bloch from ENST/TSI (FR) not only for helping me in a very friendly way to effectively control my "uncertainties", but also for having accepted to be a reporter for my work.

I am very proud to have prof. dr. ir. Josef Kittler from University of Surrey (UK) in my thesis committee, and I would like to thank him especially for his helping comments with respect to the statistical aspects of this study.

Thank you prof. dr. ir. Paul Delogne from UCL/TELE (BE), for your many "personalized" comments which I'm sure have improved the contents as well as the readability of this work, and for having accepted to be a member of my thesis committee.

I wish to thank prof. dr. ir. Claude Vloeberghs, head of the department of electrical engineering and telecommunications of the Royal Military Academy (RMA), for having granted me the time and the means to finish this thesis, and for having accepted to be a member of my thesis committee.

Thanks to ir. Charles Beumier from RMA/SIC (BE) for his help in the field of machine vision in general and in the framework of the M2VTS project in particular, to dr. ir. Stéphane Pigeon from RMA/SIC (BE) for his critical remarks in the general field of fusion and for his help in the design of the experimental protocols and for writing the software for the NIST99 evaluation and for the evaluation of the mixture of experts.

Thanks to dr. ir. Jan Černocký from VUT (CZ) for his hospitality during the NIST99 evaluations and for his help in generating the data for the experiments involving the mixture of experts.

Thanks to dr. ir. Gilbert Maître formerly from IDIAP/vision group (CH) for his very positive help in the design of experimental protocols and in the field of machine vision, to dr. ir. Eddy Mayoraz from IDIAP/machine learning group (CH) for his friendly guidance and for his help in formalizing the paradigm of the multi-linear classifier, to ir. Frédéric Gobry from IDIAP/machine learning group (CH) for his help in writing the code for the multi-linear classifier, and to dr. ir. Dominique (Doms) Genoud from IDIAP/speech group (CH) for his help in the field of speaker verification.

Thanks also to ir. Guillaume (Guig) Gravier from ENST/TSI (FR) for his friendship and for all his help in the fields of speaker verification and information technology.

I also would like to thank Bruno, Chris, Dirk, Florence, Idrissa, Lionel, Marc, Michel, Monica, Nada, Pascal, Vincianne, Wim, Xavier, Yann, Youssef, and all my other colleagues from RMA/ELTE (BE) and from ENST/TSI (FR) for the wonderful working atmosphere they all have contributed to.

I am grateful for the contributions of the following students: Benny Tops, and Dang Van Thuong.

Finally I would like to thank Renate for her love, support, and so much more ...

Contents

Abbreviations

ATM	*Automatic Teller Machine*
AVBPA	*Audio- and Video-based Biometric Person Authentication*
BDT	*Binary Decision Tree*
DET	*Detection Error Tradeoff*
EER	*Equal Error Rate (FAR=FRR)*
FA	*False Acceptance*
FAR	*False Acceptance Rate*
FE	*Frontal Expert*
FR	*False Rejection*
FRR	*False Rejection Rate*
GMM	*Gaussian Mixture Model*
HMM	*Hidden Markov Model*
k-NN	*k-Nearest Neighbor*
LC	*Linear Classifier*
LDA	*Linear Discriminant Analysis*
LR	*naive bayes classifier using a Logistic Regression model*
M2VTS	*Multi Modal Verification for Teleservices and Security applications*
MAJ	*Majority voting*
MAP	*Maximum A posteriori Probability*
MCP	*Maximum Conditional Probability*
ML	*Maximum Likelihood*
MLP	*Multi-Layer Perceptron*
NBG	*Naive Bayes classifier using Gaussian distributions*
NIST	*National Institute for Standards and Technology (USA)*
NN	*Nearest Neighbor*
NSA	*National Security Agency (USA)*
PE	*Profile Expert*
PIN	*Personal Identification Number*
PLC	*Piece-wise Linear Classifier*
QC	*Quadratic Classifier*
ROC	*Receiver Operating Characteristic*
TD	*Temporal Decomposition*
TER	*Total Error Rate*
VE	*Vocal Expert*
VQ	*Vector Quantization*

Chapter 1

Introduction

1.1 Introduction

The first chapter starts by introducing the subject of the thesis. To avoid confusion, this introduction is followed by an explanation of the differences and/or similarities between terms that are often encountered in the literature related to the field of automatic identity "determination", which are authentication, recognition, identification, and verification. These definitions are followed by a presentation of the structure of the thesis and this chapter is ended by clearly stating the *original* contributions of this thesis.

1.2 Subject of the thesis

This thesis deals with the automatic *verification* of the identity of a *cooperative* person under test, by combining the results of analyses of his or her face, profile and voice. This specific application which is used throughout this work, has been defined in the framework of the M2VTS (Multi-Modal Verification for Tele-services and Security applications) project of the European Union ACTS program [1]. The exact definition of verification and the differences with other, often encountered terms, such as identification, authentication or recognition, will be explained hereafter. The key idea in this thesis is to analyze the possibilities of using *data fusion techniques* to combine the results obtained by different biometric (face, profile and voice) experts that each have analyzed the identity claim of the person under test. In this work we are explicitly avoiding issues such as ethics, responsibility or privacy. The interested reader can find an introduction to these delicate topics in [185, 186].

The automatic verification of a person is more and more becoming an important tool in several applications such as controlled access to restricted (physical and virtual) environments. Just think about secure tele-shopping, accessing the safe room of your bank, tele-banking, accessing the services of interactive dialogue systems [175], or withdrawing money from automatic teller machines (ATM).

A number of different readily available techniques, such as passwords, magnetic stripe cards and Personal Identification Numbers (PIN) are already widely used in this context, but the only thing they really verify is, in the best case, a combination of a certain *possession* (for instance the possession of the correct magnetic stripe card) and of a certain *knowledge*, through the correct restitution of a character and/or digit combination. As is well known, these intrinsically simple (access) control mechanisms can very easily lead to abuses, induced for instance by the loss or theft of the magnetic stripe card and the corresponding PIN. Therefore a new kind of methods is emerging, based on so called *biometric* characteristics or measures, such as voice, face (including profile), eye (iris-pattern, retina-scan), fingerprint, palm-print, hand-shape or some other (preferably) unique and measurable physiological or behavioral characteristic information of the person to be verified.

In this work, a biometric measure will also be called a *modality*. This means that an identity verification system which uses several biometric measures or modalities (for instance a visual and a vocal biometric modality) is a *multi-modal* identity verification system.

Another term which will be used very often in this work is an *expert*. In this thesis, an expert is each algorithm or method using characteristic features coming from a particular modality to verify the identity of a person under test. In this sense, one single biometric measure or modality can lead to the use of more than one expert (the visual modality can for instance lead to the use of two experts: a profile and a frontal face expert). This means that a mono-modal identity verification system can still be a multi-expert system.

Biometric measures in general, and non-invasive/user-friendly (vocal, visual) biometric measures in particular, are very attractive because they have the huge advantage that one can not lose or forget them, and they are really personal (one cannot pass them to someone else), since they are based on a physical appearance measure. We can start using these user-friendly biometric measures now, thanks to the progress made in the field of automatic speech analysis and artificial vision. In general these new applications use a *classical* technique (password, or magnetic stripe card) to claim a certain identity which is then verified using one or more biometric measures.

If one uses only a single (user-friendly) biometric measure, the results obtained may be found to be not good enough. This is due to the fact that these user-friendly biometric measures tend to *vary with time* for one and the same person and to make it even worse, the importance of this variation is itself very variable from one person to another. This especially is true for the vocal (speech) modality, which shows an important *intra-speaker variability.* One possible solution to try to cope with the problem of this *intra-person* variability is *to use more than one biometric measure.* In this new *multi-modal* context, it is thus becoming important to be able to combine (or *fuse*) the outcomes of different modalities or experts. There is currently a significant international interest in this topic. The organization of already two international conferences on the specific subject of *Audio- and Video-based Biometric Person Authentication (AVBPA)* is probably the best proof of this [16, 38].

Combining the outcomes of different experts can be done by using classical data fusion techniques [2, 46, 70, 71, 101, 170, 172, 181], but the major drawback of the bulk of all these methods is their rather high degree of complexity, which is expressed - amongst else - by the fact that these methods tend to incorporate a lot of parameters that have to be estimated. If this estimation is not done using enough training data (*i.e.* if the estimation is not done properly), this places a serious constraint on the ability of the system to correctly generalize [9, 121]. But actually a major difficulty of this particular estimation problem is the scarcity of multi-modal training data. Indeed, to keep the automatic verification system user-friendly, the enrollment of a (new) client should not take too much time, and as a direct consequence from this, the amount of client training data tends to be limited. To try to deal with this lack of training data, one possibility is to develop *simple* classifiers (*i.e.* for instance classifiers that use only few parameters), so that their parameters can be estimated using only limited amounts of training data. The price to be paid when using simple methods is a decrease in system performance, as compared to what one could get with an optimal method.

1.3 Identity determination concepts

Automatic systems for *recognizing* a person or for *authenticating* his identity (which is equivalent), all have a database of N so-called authorized persons or *clients*. Authentication or recognition is the general term, which covers on one hand *identification* and on the other hand *verification*. These two processes are quite different as the following more detailed description will show.

Identification in the strict sense of the word supposes a *closed world* context. This means that we are sure that the person under test is a client. The only thing we need to find out is which client of the database of authorized persons matches "the best" the person under test. There is no criterion (such as a threshold for instance) to define how good the match has to be, to be acceptable. Identification is thus a 1-out-of-N matching process, and it is clear that the performances decrease with N.

Verification in the strict sense of the word operates in an *open world* context. This means that we are no longer sure that the person under test is a client. In this case, the person under test *claims* a certain identity, which of course has to be the identity of an authorized person. If the person under test is no member of the database of authorized persons, he is a so-called *impostor*. Verification is thus a 1-out-of-1 matching process, where it is important that the mismatch between the reference model from the database and the measured characteristics of the person under test stays below a certain threshold. The verification performances are independent of N.

Sometimes people do refer to *identification* in the large sense of the word as the (sequential) process of identification followed by a verification of the identified identity. Sometimes this double process is also called *identification in an open world context*.

In this thesis we will only consider *verification* problems. This means that the decision problem we are confronted with is a typical *binary* hypothesis test. Indeed, the decision we have to take is either to *accept* or to *reject* the identity claim of the person under test.

1.4 Structure of the thesis

This thesis has been divided into two parts. In the first part, general issues related to automatic multi-modal identity verification systems, such as a discussion on biometric modalities (including the characterization of automatic identity verification systems), the presentation of our experimental set-up (including the presentation and the analysis of our experts) and a general overview of data fusion related concepts, are treated. In the second part, the fusion of the different experts in a multi-modal identity verification system is implemented on the decision level, using parametric and non-parametric methods. These different methods are then compared with each other and a structured hierarchical approach for gradually upgrading the performances of automatic biometric verification systems is presented. At the end of these two parts, we are concluding this thesis by summarizing our contributions to the field and by looking at possible extensions of the work done.

To be more specific, the first part is organized as follows. In chapter 2 we deal with biometric modalities and we start by listing some theoretical and practical requirements that biometrics in general should conform to. This is followed by a section which presents a tentative classification of the most commonly found biometrics into two classes: the so-called physiological and behavioral biometrics. In the following section the general structure of an automatic mono-modal biometric verification system is presented, while in the next section some general arguments for using multi-modal biometric verification systems are developed. The following section is meant to introduce and define the classical performance characteristics used in the field of automatic identity verification, and the final section is giving an overview of the state of the art in multi-modal biometric identity verification systems. Chapter 3 gives details about the experimental set-up. It starts by presenting the M2VTS databases used in this work. After this, the experimental protocol is described. Finally, the three different biometric experts we have been using throughout this work are briefly introduced and their individual performances are highlighted and statistically analyzed. Chapter 4 introduces some elementary data fusion concepts such as the different data fusion levels and architectures, and shows how it is possible, by making some well-funded choices, to transform a general *data fusion* problem into a particular *classification* problem.

The second part of this work deals more particularly with the parallel combination or fusion of the partial (soft) decisions of the different experts. Chapter 5 explains why we have chosen to experiment with parametric as well as with non-parametric methods. Chapter 6 deals with parametric techniques, but to show the usefulness of these parametric methods first of all a trivial but original method is presented: the *monotone multi-linear (or piece-wise linear) classifier*. Unfortunately the performances of this classifier are not very good, mainly due to the fact that (valuable) information with respect to the probability density functions of the different populations is thrown away. Therefore in a fairly early stage of this work it has been decided to stop developing this simple method and to fall back instead on the less original, but more fundamental statistical decision theory, by using so-called *parametric* techniques. In this parametric class, classifiers based on the general Bayesian decision theory (Maximum A-posteriori Probability and Maximum Likelihood) and on a simplified version of it (the Naive Bayesian classifier, which has been applied in the case of simple Gaussians and in the case of a logistic regression model), have been studied. Furthermore experiments have also been done using Linear and Quadratic classifiers. Neural networks form a special case of the parametric family, since the number of parameters to be estimated can be very large. Therefore neural networks are sometimes classified as semi-parametric classifiers. Still we will present neural networks in the chapter on parametric techniques, by means of its most popular representative: the Multi-Layer perceptron. Chapter 7 deals with non-parametric techniques. This chapter starts by presenting a very simple family of non-parametric techniques. These *voting* techniques are sometimes referred to as k-out-of-n voting techniques, where k relates to the number of experts that have to decide that the person under test is a client, before the global voting classifier accepts the person under test as a client. After the voting methods, another simple but very popular technique, the k Nearest Neighbor (k-NN) technique, is presented with a number of variants. These variants include a distance weighted and a vector quantized version of the classical k-NN rule. This chapter ends by presenting the category of (binary) decision trees, by means of an implementation of the C4.5 algorithm, which is probably the most popular method in its kind. Chapter 8 deals with the comparison between the different parametric and non-parametric methods that have been presented in the second part of the thesis. Chapter 9 presents a multi-level decision fusion strategy that allows to gradually improve the performances of an automatic biometric identity verification system, while limiting the initial investments.

Chapter 10 finally concludes this thesis, formulates some recommendations for developing automatic multi-modal biometric identity verification systems and identifies possibilities for future work in the same application field.

1.5 Original contributions of this thesis

The original contributions of this thesis are the following ones:

1. the formulation (in the framework of a multi-modal biometric identity verification system) of the fusion of the partial (soft) decisions of d experts in parallel as a particular classification problem in the d-dimensional space [179];

2. the systematic and detailed statistical analysis of the different experts that have been used;

3. the development of a simple decision fusion method, based on a *monotone* multi-linear classifier [179, 180];

4. the analysis of the applicability, the characteristics and the performance of the logistic regression method in a Bayesian framework [177];

5. the development of a Vector Quantization version of the classical k-Nearest Neighbor algorithm [173];

6. the systematic comparison of a large number of parametric as well as non-parametric techniques to solve the particular classification problem [174];

7. the introduction of either the non-parametric Cochran's Q test for binary responses, or the non-parametric Page test for ordered alternatives, to measure the statistical significance of the differences in performance of several (*i.e.* more than two) fusion modules at the same time;

8. the formulation of a multi-level fusion strategy which allows to gradually improve the performances of an automatic (biometric) identity verification system [176, 178];

9. the formulation of the mixture of experts paradigm in the framework of mono-modal multi-expert data fusion, applied to a segmental approach to text-independent speaker verification [171];

10. the introduction of the use of multi-modal identity verification in Interactive Dialogue Systems [175].

Part I

General issues related to automatic biometric multi-modal identity verification systems

Chapter 2

Biometric verification systems

2.1 Introduction

This chapter starts by defining the *ideal* theoretical and practical requirements for any biometric. This is followed by a section which presents a tentative classification (according to [120]) of the most commonly found biometrics into two classes: the so-called physiological and behavioral biometrics. In the following section the general structure of an automatic mono-modal biometric verification system is presented, while in the next section some general arguments for using multi-modal biometric verification systems are developed. The following section presents then the main characteristics of identity verification systems. In the final section, an overview of the state of the art of multi-modal biometric person verification systems is given.

2.2 Requirements for biometrics

Automatic biometric systems have to identify an individual or to verify his or her identity[1] using measurements of the (living) human body. According to [88, 89], in theory any human characteristic can be used to make an identity verification, as long as it satisfies the following desirable (ideal) requirements:

[1]As already mentioned in chapter 1, we will consider in this work only verification systems.

universality this means that every person should have the characteristic;

uniqueness this indicates that no two persons should be the same in terms of the characteristic;

permanence this means that the characteristic does not vary with time;

collectability this indicates that the characteristic can be measured quantitatively.

In practice, there are some other important requirements:

performance this specifies not only the achievable verification accuracy, but also the resource requirements to achieve an acceptable verification accuracy;

robustness this refers to the influence of the working or environmental factors (channel, noise, distortions, ...) that affect the verification accuracy;

acceptability this indicates to what extent people are willing to accept the biometric verification system;

circumvention this refers to how easy it is to fool the system by fraudulent techniques (make sure that the individual *owns* the data, and that he is not transforming it; this could also include a so-called *liveliness* test).

As mentioned before, these requirements should be regarded as ideal. In other words, the better a biometric satisfies these requirements, the better it will perform. In practice however, there is no single biometric which fulfills all these ideal requirements perfectly. This observation is one of the main reasons why combining several biometric modalities in multi-modal systems is gaining field.

2.3 Classification of biometrics

A range of mono-modal biometric systems is in development or on the market, because no one biometric meets all the needs.The tradeoffs in developing these systems involve cost, reliability, discomfort in using a device, and the amount of data needed. Fingerprints, for instance, have a long track record of reliability (*i.e.* they make very few classification errors), but the hardware for capturing fingerprints was until now rather expensive, and the amount of data that needs to be stored to describe a fingerprint (the template) tended to be rather large. In contrast, the hardware for capturing the voice is cheap (relying on low-cost microphones or on an already existing telephone), but it varies when emotions and states of health change. According to [120], biometrics encompasses both physiological and behavioral characteristics. This is illustrated for a number of frequently used biometrics in Figure 2.1.

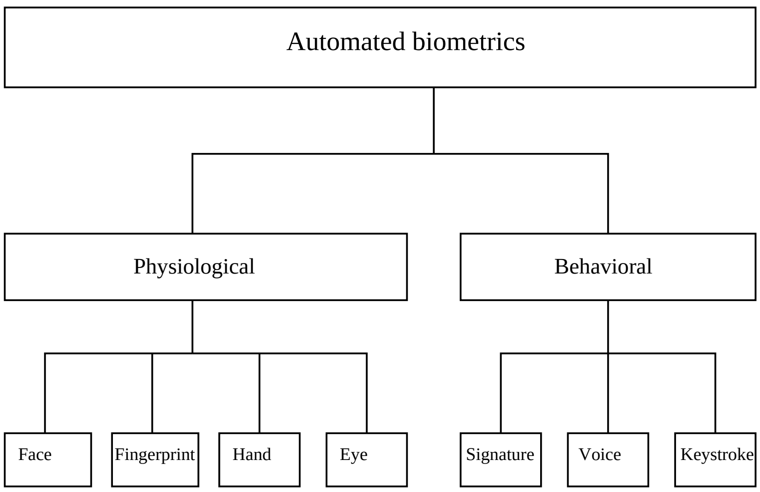

Figure 2.1: Classification of a number of biometrics in physiological and behavioral characteristics.

A physiological characteristic is a relatively stable physical feature such as a fingerprint [89, 130, 153], hand geometry [190], palm-print [188], infrared facial and hand vein thermograms [141], iris pattern [184], retina pattern [74], or facial feature [11, 12, 34, 39, 102, 116, 183, 189]. Indeed, all these characteristics are basically unalterable without trauma to the individual. A behavioral trait on the other hand, has some physiological basis, but also reflects a person's psychological (emotional) condition. The most common behavioral trait used in automated biometric verification systems is the human voice [3, 10, 20, 22, 31, 35, 36, 52, 60, 62, 63, 64, 65, 66, 69, 72, 73, 76, 81, 80, 105, 111, 112, 131, 132, 133, 134, 151, 154, 160]. Other behavioral traits are gait [126], keystroke dynamics [127], and (dynamic) signature analysis [124, 125]. One of the main problems with behavioral characteristics is that they tend to change over time. Therefore biometric

systems that rely on behavioral characteristics should ideally update their enrolled reference template(s) on a regular basis. This could be done either in an automatic manner, each time a reference is used successfully (*i.e.* the system decides that an access claim is an authentic client claim), or in a supervised manner, by re-enrolling each client periodically. The former method has the advantage to be user-friendly, but has the drawback that one updates the client references with a template from an impostor in the case that the system commits a False Acceptance. The latter approach has the advantage to update the client references always with client templates, but has the drawback that it is not very user-friendly, since the clients need to do additional training sessions.
The differences between physiological and behavioral methods are important. On one hand, the degree of intra-person variability is smaller in a physiological than in a behavioral characteristic. On the other hand, machines that measure physiological characteristics tend to be larger and more expensive, and may seem more threatening or *invasive* to users (this is for instance the case for retina scanners). Because of these differences, no one biometric will serve all needs.

2.4 General structure of a mono-modal biometric system

Automated mono-modal biometric verification systems usually work according to the following principles. In a typical functional system a sensor, adapted to the specific biometric, generates measurement data. From these data, features that may be used for verification are extracted, using image and/or signal processing techniques. In general, each biometric has its own feature set. Pattern matching techniques compare the features coming from the person under test with those stored in the database under the claimed identity, to provide likely matches. Last but not least, decision theory including statistics provides a mechanism for answering the question "Is the person under test who he or she claims to be?" and for evaluating biometric technology [77, 78, 158]. Automatic mono-modal biometric verification systems are usually built arranging two main modules in series: (1) a module which compares the measured features from the person under test with a reference client model and gives a scalar number[2] as output, followed by

[2]This scalar number will be called a *score* and it states how well the claimed identity has been verified

(2) a decision module realized by a thresholding operation. This threshold can be a function of the claimed identity.
The architecture of an automatic mono-modal biometric verification system is represented in figure 2.2.

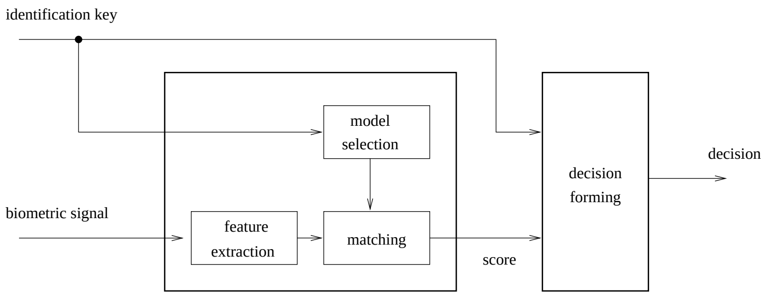

Figure 2.2: Typical mono-modal biometric verification system architecture.

2.5 The need for multi-modal biometric systems

There can be several reasons why one would prefer multi-modal biometric verification systems over mono-modal ones. Generally, the criterion to choose between mono- and multi-modal systems will be *system performance*. The end-user typically desires a guarantee that the classification errors (FAR and FRR) will be limited by maximal values that will depend on the application. And although there exist mono-modal biometric verification techniques that do offer very small classification errors, the main problem with this category of biometrics is that they are either too *expensive* to be used in a general purpose context (for instance identity verification in the case of credit card payments over the Internet using a PC) or perceived by the user as too *invasive*. So very often one is confronted with the obligation of using inexpensive hardware and non-invasive user-friendly biometrics. Two of the most popular biometrics that can conform to these constraints are faces and voices. However, the drawback of using inexpensive hardware (cheap black and white CCD-cameras and low-cost microphones) to obtain the raw data measurements of these biometrics, has as a direct consequence that the measurements generally will be corrupted with noise, or distorted. This obviously leads to a degradation of system performance. Other problems linked with these popular user-friendly biometrics are that the visual modality is rather sensitive to lighting conditions and that the vocal modality tends to vary with time (since it is a behavioral biometric). This makes the use of a mono-modal biometric verification system based solely either on the facial or on the vocal modality a very big challenge, especially since it is usually not possible to update the database references of the authorized users on a regular basis.

One possible solution to cope with this problem is to use not one single mono-modal biometric system, but to use several of them in parallel to form a so-called multi-modal biometric verification system. It can be felt intuitively that such a strategy can be helpful, if one considers *complementary* biometrics. This complementarity can be achieved with respect to the different requirements as they were presented in section 2.2. A possible example of complementary biometrics with respect to the *permanence* requirement would be the combined use of a physiological (face: more invariant in time) and a behavioral (voice: less invariant) biometric. The main and very general idea of using multi-modal biometric verification systems instead of mono-modal ones is thus the ability to use more (complementary) information with respect to the person under test in the former approach, than in the latter approach. In chapter 9, a more detailed step-by-step analysis of a multi-level strategy to gradually improve the performances of an automated biometric system is presented.

A possible and straightforward way of building a multi-modal verification system from d such mono-modal systems is to input the d scores provided in parallel into a fusion module, which combines the d scores and passes the fused score on to the decision forming module. This module then has to take the decision *accept* or *reject*, based on a threshold. Just as in the case of the mono-modal system, this threshold can be a function of the claimed identity. However, two alternatives remain for the fusion module: a global (i.e. the same for all persons) or a personal (i.e. tailored to the specific characteristics of each authorized person) approach. For the sake of simplicity and because the personal approach needs more training data (since in this case the fusion module needs to be optimized for *each* client), we have opted in this work for a global fusion module.

Figure 2.3 shows the typical architecture of a general multi-expert verification system, including the possible use of personalized fusion or decision forming. The formal presentation of this general data fusion problem will be given in chapter 4.

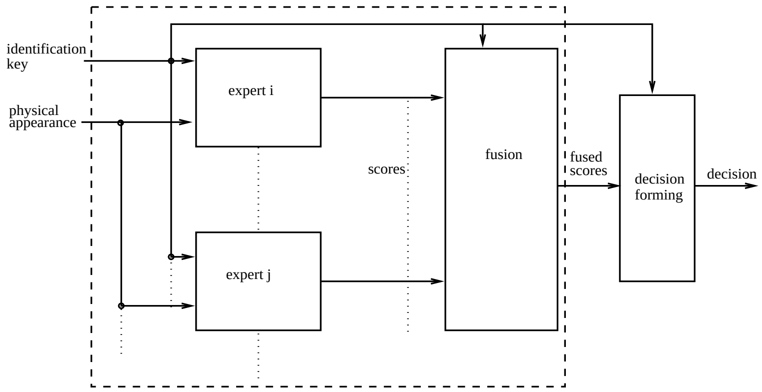

Figure 2.3: Multi-expert architecture.

2.6 Characterization of a verification system

In this work, we will consider the verification of the identity of a person as a typical two-class problem: either the person is the one (in this case he is called a *client*), or is not the one (in that case he is called an *impostor*) he claims to be. This means that we are going to work with a binary {accept, reject} decision scheme.
When dealing with binary hypothesis testing, it is trivial to understand that the decision module can make two kinds of errors. Applied to this problem of the verification of the identity of a person, these two errors are called:

- False Rejection (FR): *i.e.* when an actual *client* is rejected as being an *impostor*;
- False Acceptance (FA): *i.e.* when an actual *impostor* is accepted as being a *client*.

The performances of a speaker verification system are usually given in terms of the global error rates computed during tests: the False Rejection Rate (FRR) and the False Acceptance Rate (FAR) [18]. These error rates are defined as follows:

$$\mathrm{FRR} = \frac{\text{number of FR}}{\text{number of client accesses}} \tag{2.1}$$

$$\text{FAR} = \frac{\text{number of FA}}{\text{number of impostor accesses}} \tag{2.2}$$

A perfect identity verification (FAR=0 and FRR=0) is in practice unachievable. However, as shown by the study of binary hypothesis testing [167], any of the two FAR, FRR can be reduced to an arbitrary small value by changing the decision threshold, with the drawback of increasing the other one. A unique measure can be obtained by combining these two errors into the Total Error Rate (TER) or its complimentary, the Total Success Rate (TSR):

$$\text{TER} = \frac{\text{number of FA} + \text{number of FR}}{\text{total number of accesses}} \tag{2.3}$$

$$\text{TSR} = 1 - \text{TER} \tag{2.4}$$

However, care should be taken when using one of these two unique measures. Indeed, from the definition just given it follows directly that these two unique numbers could be heavily biased by one or either type of errors (FAR or FRR), depending solely on the number of accesses that have been used in obtaining these respective errors. As a matter of fact, due to the proportional weighting as specified in the definition, the TER will always be closer to that type of error (FAR or FRR) which has been obtained using the largest number of accesses.
The overall performance of an identity verification system is however better characterized by it's so-called *Receiver Operating Characteristic (ROC)*, which represents the FAR as a function of the FRR [167]. The Detection Error Tradeoff (DET) curve is a convenient non-linear transformation of the ROC curve, which has become the standard method for comparing performances of speaker verification methods used in the annual NIST evaluation campaigns [142]. In a DET curve, the horizontal axis shows the normal deviate of the False Alarm probability in (%), which is a non-linear transformation of the horizontal False Acceptance axis of the classical ROC curve. The vertical axis of the DET curve represents normal deviate of the Miss probability (in %), which is a non-linear transformation of the False Rejection axis of the classical ROC curve. The use of the normal deviate scale moves the curves away from the lower left when performance is high, making comparisons between different systems easier. It can also be observed that, typically, the resulting curves are approximately straight lines, which do correspond to normal likelihood distributions, for at least a wide

portion of their range. Further details of this non-linear transformation are presented in [115]. Figures 2.4 and 2.5 give respectively an example of a typical ROC and a typical DET curve.

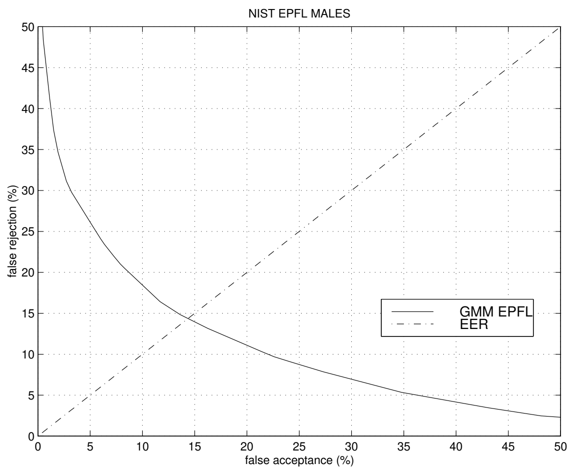

Figure 2.4: Typical example of a ROC curve.

Each point on a ROC or a DET characteristic corresponds with a particular decision threshold. The Equal Error Rate (EER: *i.e.* when FAR = FRR), is often used as the only performance measure of an identity verification method, although this measure gives just one point of the ROC and comparing different systems solely based on this single number can be very misleading [129].

High security access applications are concerned about break-ins and hence operate at a point on the ROC with small FAR. Forensic applications desire to catch a criminal even at the expense of examining a large number of false accepts and hence operate at small FRR/high FAR. Civilian applications attempt to operate at the operating points with both low FRR and low FAR. These concepts are shown in Figure 2.6, which was found in [88].

Unfortunately in practice, as will be shown further in the study of the fusion modules presented in this thesis, it is not always possible to explicitly identify a continuous decision threshold in a certain fusion module, which means that in that case it will a fortiori not be possible to vary the decision threshold to obtain a ROC or a DET curve. So in these specific cases only a single operating point on the ROC can be given. This is incidentally

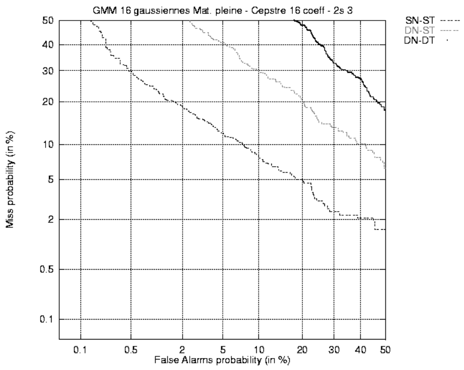

Figure 2.5: Typical example of a DET curve.

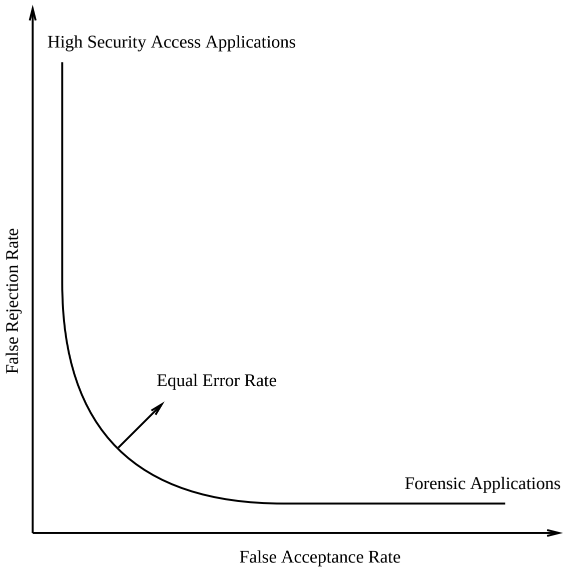

Figure 2.6: Typical examples of different operating points for different application types.

also the only correct way of determining the performance of an operational system, since in such systems the decision threshold has been *fixed*.
All verification results in this thesis will be given in terms of FRR, FAR, and TER. For each error the 95 % level confidence interval will be given between square brackets. The concept of *confidence intervals* refers to the inherent uncertainty in test results owing to small sample size. These intervals are *a posteriori* estimates of the uncertainty in the results on the test population. They do not include the uncertainties caused by errors (mislabeled data, for example) in the test process. The confidence intervals do not represent *a priori* estimates of performance in different applications or with different populations [182].
These confidence levels will be calculated assuming that the probability distribution for the number of errors is binomial. But since the binomial law can not be easily analyzed in an analytical way, the calculation of confidence intervals can not be done directly in an analytical way. Therefore we have used the Normal law as an approximation of the binomial law. This large sample approach is already statistically justified starting from 30 samples. Using this approximation, the 95% confidence interval of an error E based on N tests, is defined by the following lower (given by the minus sign) and upper (given by the plus sign) bounds:

$$E \pm 1.96\sqrt{\frac{E\,(1-E)}{N}}.$$

More detailed information about the calculation of confidence intervals can be found in [41, 44, 155].

2.7 State of the art

2.7.1 General overview

Some work on multi-modal biometric identity verification systems has already been reported in the literature. Hereafter, an overview is given of the most important contributions, with a brief description of the work performed.

1. As early as 1993, Chibelushi et Al. have proposed in [40] to integrate acoustic and visual speech (motion of visible articulators) for speaker recognition. The combination scheme used is a simple linear one. There is no mention of the database used and the result mentioned is an EER = 1.5%.

2. In 1995, Brunelli and Falavigna have proposed in [33] a person identification system based on acoustic and visual features. The voice modality is based on a text-independent vector quantization and it uses two types of information: static and dynamic acoustic features. The face modality implements a template matching technique on three distinct areas of the face (eyes, nose, and mouth). They use a database containing up to three sessions of 87 persons. One session was used for training, the others for testing, which did lead to a total number of 155 tests. The most performing fusion module is a neural network. The best results obtained on this particular database are: FAR = 0.5% and FRR = 1.5%.

3. In 1997, Dieckmann et Al. have proposed in [50] a decision level fusion scheme, based on a *2-out-of-3* majority voting. This approach integrates two biometric modalities (face and voice), which are analyzed by three different experts: (static) face, (dynamic) lip motion, and (dynamic) voice. The authors have tested their approach on a specific database of 15 persons, where the best verification results obtained were FAR = 0.3% and FRR = 0.2%.

4. In 1997, Duc et Al. did propose in [55] a simple averaging technique and compared it with the Bayesian integration scheme presented by Bigün et Al. in [13]. In this multi-modal system the authors use a frontal face identification expert based on Elastic Graph Matching, and a text-dependent speech expert based on person-dependent Hidden Markov Models (HMMs) for isolated digits. All experiments are performed on the M2VTS database, and the best results are obtained for the Bayesian fusion module: FAR = 0.54% and FRR = 0.00%.

5. In 1997, Jourlin et Al. have proposed in [93] an acoustic-labial speaker verification method. Their approach uses two classifiers. One is based on a lip tracker using visual features, and the other one is based on a text-dependent person-dependent HMM modeling of isolated digits using acoustic features. The fused score is computed as the weighted sum of the scores generated by the two experts. All experiments are performed on the M2VTS database, and the best results obtained for the weighted fusion module are: FAR = 0.5% and FRR = 2.8%.

6. In 1998, Kittler et Al. have proposed in [98] a multi-modal person verification system, using three experts: frontal face, face profile, and voice. The frontal face expert is based on template matching, the face

profile expert is using a chamfer matching algorithm, and the voice expert is based on the use of text-dependent person-dependent HMM models for isolated digits. All these experts give their soft decisions (scores between zero and one) to the fusion module. All experiments are performed on the M2VTS database, and the best combination results are obtained for a simple sum rule: EER = 0.7%.

7. In 1998, Hong and Jain have proposed in [82] a multi-modal personal identification system which integrates two different biometrics (face and fingerprints) that complement each other. The face verification is done using the eigenfaces approach, and the fingerprint expert is based on a so-called *elastic* matching algorithm. The fusion algorithm operates at the expert decision level, where it combines the scores from the different experts (under the statistically independence hypothesis), by simply multiplying them. The {accept, reject} decision is then taken by comparing the fused score to a threshold. The databases used in this work are the Michigan State University fingerprint database containing 1500 images from 150 persons, and a face database coming from the Olivetti Research Lab, the University of Bern, and the MIT Media Lab, which contains 1132 images from 86 persons. The results obtained for the fusion approach on this database are: FAR = 1.0% and FRR = 1.8%.

8. In 1998, Ben-Yacoub did propose in [7] a multi-modal data fusion approach for person authentication, based on Support Vector Machines (SVM). In his multi-modal system he uses the same experts and the same database as Duc et Al. in the work presented above. The best results which he obtained for the SVM fusion module are FAR = 0.07% and FRR = 0.00%.

9. In 1999, Pigeon did propose in [135] a multi-modal person authentication approach based on simple fusion algorithms. In this multimodal system the author uses a face identification expert based on template matching, a profile identification expert based on a chamfer matching algorithm, and a text-dependent speech expert based on person-dependent HMM models for isolated digits. All experiments are performed on the M2VTS database, and the best results are obtained for a fusion module based on a linear discriminant function: FAR = 0.07% and FRR = 0.78%.

10. In 1999, Choudhury et Al. did propose in [43] a multi-modal person

recognition system using unconstrained audio and video. The system does not need fully frontal face images or clean speech as input. The face expert is based on the eigenfaces approach, and the audio expert uses a text-independent HMM using Gaussian Mixture Models (GMMs). The combination of these two experts is performed using a Bayes net. The system was tested on a specific database containing 26 persons and the best results obtained using the best images and audio clips from an entire session are: FAR = 0.00% and FRR = 0.00%.

2.7.2 Results obtained on the M2VTS database

To facilitate the comparison with the work presented in this thesis, we have isolated from the previous state of the art the results which have been obtained on the same M2VTS database as the one we have been working on. These results are presented in Table 2.1 hereafter. Where available, the confidence interval is indicated between square brackets. Care should be taken however when comparing these results, since the experts used are not necessarily the same for all methods. The last line in this Table represents the best results obtained in this thesis, using a logistic regression model.

Table 2.1: State of the art of the verification results obtained on the M2VTS database.

Author(s)	Experts	FRR (%)	FAR (%)
Duc et Al.	frontal, vocal	0.00	0.54
Jourlin et Al.	lips, vocal	2.80	0.50
Kittler et Al.	frontal, profile, vocal	0.70 (EER)	0.70 (EER)
Ben-Yacoub	frontal, vocal	0.00	0.07
Pigeon	frontal, profile, vocal	0.78	0.07
Verlinde	frontal, profile, vocal	0.00	0.00

2.8 Comments

As already mentioned, each biometric technology has its strengths and limitations, and no single biometric is expected to effectively meet the needs of all applications. We have seen that voice is one of the most popular biometrics, thanks to its high acceptability and its user-friendliness [88]. Since voice is a behavioral biometric modality and since in a multi-modal approach it is wise to complement a behavioral modality with a physiological one, we wanted to add a physiological modality which also was highly acceptable. These considerations have led to choose the visual modality. In the framework of the M2VTS application, another important criterion for choosing the different biometrics was the availability of the hardware. With respect to the *tele-services*, the idea was to use so-called multi-media PC's, which are equipped with low-cost microphones and CCD-camera. These considerations reinforce each other and they explain why in the multi-modal system presented in this work, voice and vision were used as the two (complementary) biometric modalities. Analyzing the state of the art in automatic biometric multi-modal identity verification systems, it has been shown that on the M2VTS database, the best method presented in this thesis (based on the logistic regression model) is the overall best method.

Chapter 3

Experimental setup

3.1 Introduction

This chapter starts by presenting the M2VTS database used in this work. After this, the experimental protocol used for testing the individual experts and the fusion modules is described. Finally, the three different biometric experts (a frontal, a profile and a vocal one) we have been using throughout this work are briefly introduced and their individual performances are highlighted. This is followed by a thorough statistical analysis of the results given by these three different experts for both client and impostor accesses. In this analysis it is shown that the distribution of the scores per expert and per type of access (the so-called conditional probability density functions) do not satisfy the Normality hypothesis. Furthermore it is shown that the chosen experts do have good discriminatory power, and are complementary. The potential gain obtained by combining the results of these three different experts are shown by means of a simple linear classifier.

3.2 The M2VTS audio-visual person database

The M2VTS [1] multi-modal database comprises 37 different persons and provides 5 shots for each person. These shots were taken at intervals of at least one week. During each shot, people were asked (1) to count from "0" to "9" in French (which was the native language for most of the people) and (2) to rotate their head from 0 to -90 degrees, back to 0 and further to +90 degrees, and finally back again to 0 degrees. The most difficult shot to recognize is the 5^{th} shot. This shot mainly differs from the others because of face "variations" (head tilted, eyes closed, different hair style,

presence of a hat/scarf, ...), voice variations or shot imperfections (poor focus, different zoom factor, poor voice signal to noise ratio, ...). More details with respect to this database can be found in [136, 137, 135].
Taking into account the specificity of our problem (*i.e.* combining outputs of several experts) we are not going to use this 5^{th} shot, since we are not interested in developing individual powerful experts that work well even under these extreme conditions as presented by shot number 5.
To show the quality of the pictures contained in the small M2VTS database, Figures 3.1, 3.2, and 3.3 show respectively the frontal views of some persons, the rotation sequence and the 5 different shots for one and the same person [135].

3.3 Experimental protocol

3.3.1 General issues

In the most general (but rich) case, three *different* data sets are needed for training, fine-tuning and testing the individual experts. The first data set is called the *training* set and is used by each expert to model the different persons. The second data set is called the *development* or *validation* set and is used to fine-tune the different experts, for instance by calculating the decision thresholds. The third data set is called the *test* set and it is used to test the performances of the obtained experts. For the fusion module, we can define in the most general case exactly the same data sets as in the case of the individual experts. This general concept of the use of the different data sets is illustrated in Figure 3.4. This does not necessarily mean that one always will need six completely separated data sets, since the fact that the test set for the individual experts is completely dissociated from the development of the experts, makes it suitable to be reused for the fusion module. Furthermore, not all types of experts, nor all fusion modules do include the modeling of the persons. This means that in the particular case of experts and fusion modules which do not use data to model persons and in the obvious case in which we do reuse the expert test set as a data set for the fusion module, one only needs *three* different data sets instead of *six* in the most general case. This is illustrated in Figure 3.5. In the intermediate case, where the experts do need separate training and development data, but the fusion module does not need any development data, one needs four different data sets, as illustrated in Figure 3.6.
If there is not enough data available to make this possible, the following errors will be introduced:

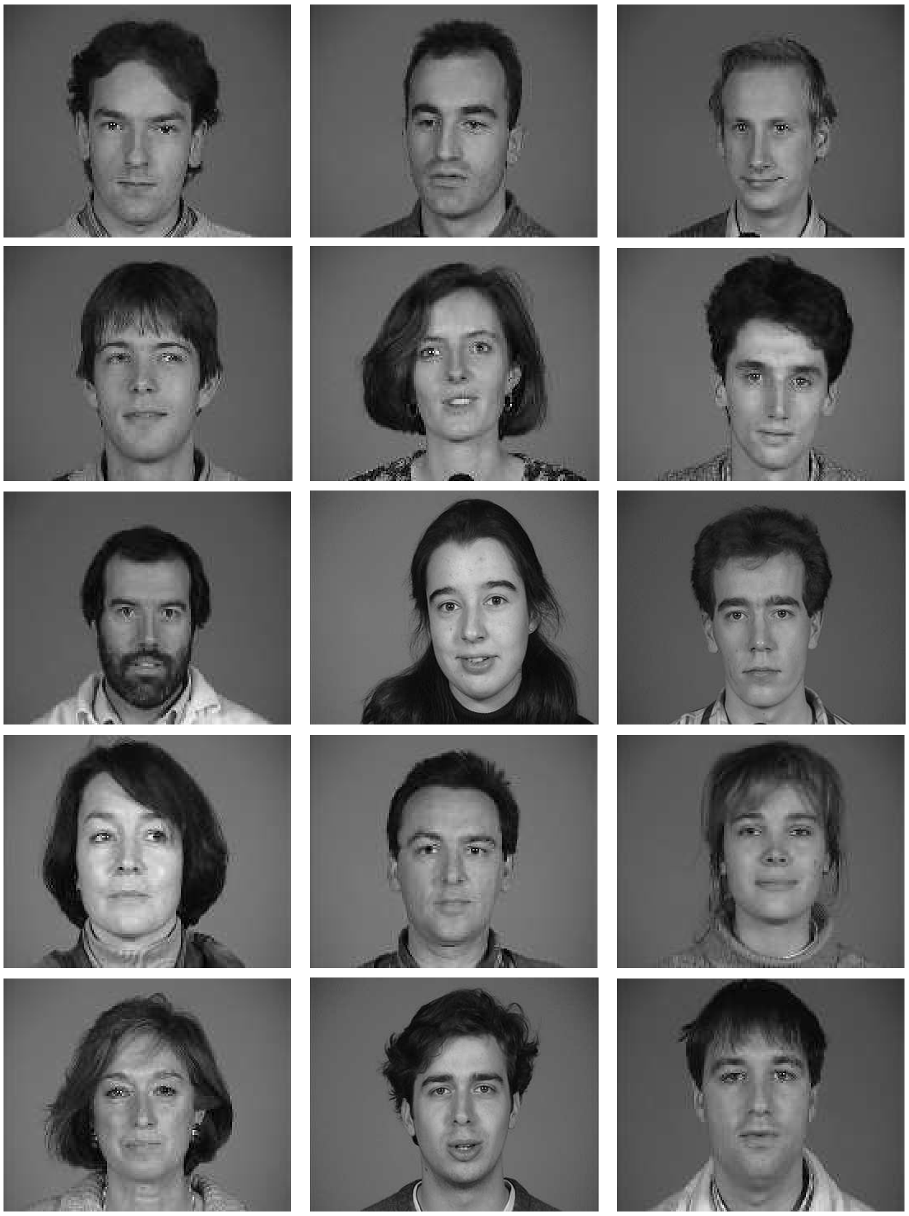

Figure 3.1: M2VTS database: some frontal views.

Figure 3.2: M2VTS database: views taken from a rotation sequence.

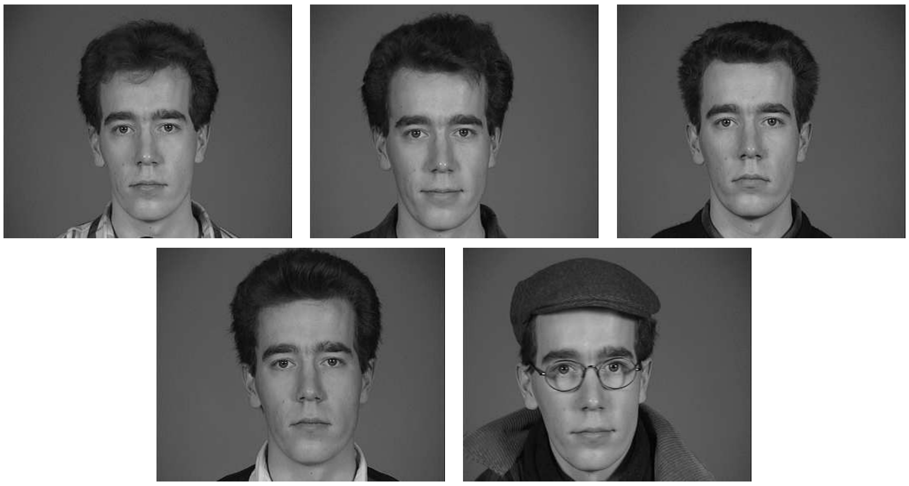

Figure 3.3: M2VTS database: frontal views of one person coming from the 5 different shots.

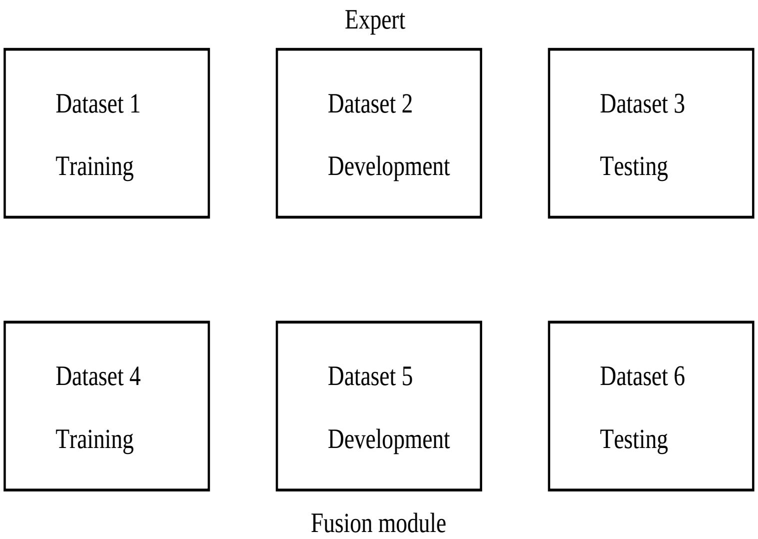

Figure 3.4: The most general case where 6 different datasets are used.

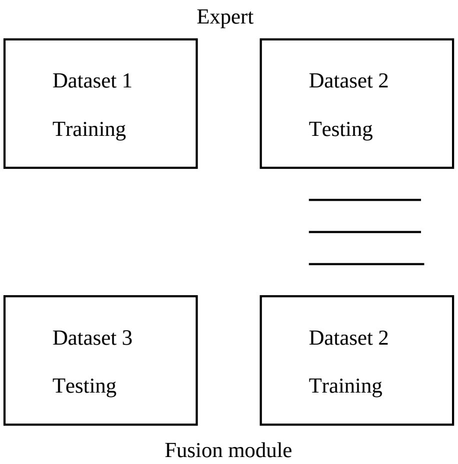

Figure 3.5: The case where only three different datasets are needed.

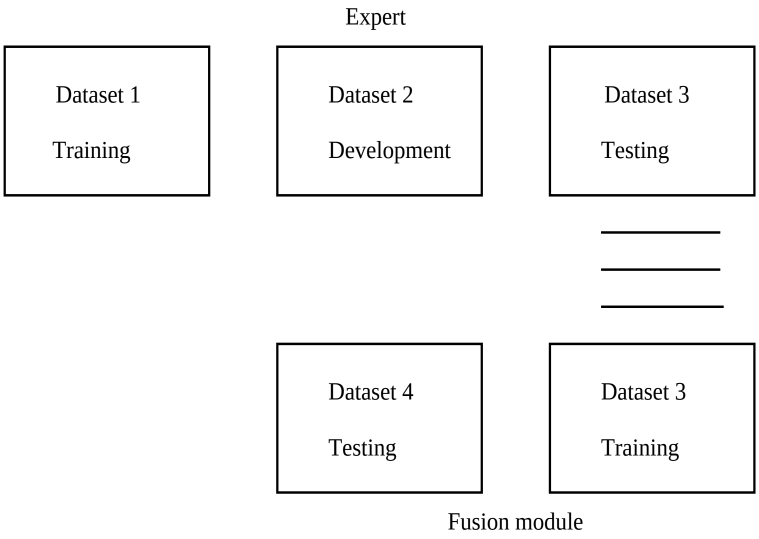

Figure 3.6: The intermediate case where four different datasets are needed.

- if the test data is the same as the training data, performances will be overestimated. This is true for both the individual experts and the fusion module. This is of course due to the fact that the experts and the fusion module will generate the best results for the same data they have been trained on.

- if the training data for the experts is the same as for the fusion module, the fusion module will be under performing. The reason for this is that the fusion module doesn't get enough information. Indeed, in the extreme case of experts that perform perfectly on their training data, the outcome of such an expert would be either 0 or 1, which leaves the fusion module with the arbitrary choice of setting the threshold somewhere in between.

3.3.2 Experimental protocol

For our experiments, we have opted for a very simple experimental protocol. In this protocol we use only the first four sessions of the M2VTS database in the following manner.

1. The first enrollment session has been used for training the individual

experts. This means that each access has been used to model the respective client, yielding 37 different client models.

2. Then the accesses from each person in the second enrollment session have been used to generate validation data in two different manners. Once to derive one single client access by matching the shot of a specific person with its own reference model, and once to generate 36 impostor access by matching it to the 36 models of the other persons of the database. This simple strategy thus leads to 37 client and $36 \times 37 = 1.332$ impostor accesses, which have been used for validating the performance of the individual experts and for calculating thresholds.

3. The third enrollment session has been used to test these experts, using the thresholds calculated on the validation data set. This same data set has also been used to train the fusion modules, which again leads to 37 client and 1.332 impostor reference points.

4. Finally, the fourth enrollment session has been used to test the fusion modules, yielding once more the same number of client and impostor claims.

The drawback of this simple protocol, is that the impostors are *known* at the expert and supervisor training time. In section 8.3.2, validation results will be presented using a protocol that does not suffer from the same drawback. This validation protocol is implemented using a so-called *leave-one-out* method [49].

3.4 Identity verification experts

3.4.1 Short presentation

All the experiments in this thesis have been performed using three different identity verification experts. Each one of these experts will be described briefly hereafter.

Profile image expert

The profile image verification expert is described in detail in [138] and its description hereafter has been inspired by the presentation of this expert in [98]. This particular profile image expert is based on a comparison of a candidate profile of the person under test with the template profile corresponding to the claimed identity. The candidate image profile is extracted from the profile images by means of color-based segmentation. The similarity of the two profiles is measured using the Chamfer distance computed sequentially [28]. The efficiency of the verification process is aided by pre-computing a distance map for each reference profile. The map stores the distance of each pixel in the profile image to the nearest point on the reference profile. As the candidate profile can be subject to translation, rotation and scaling, the objective of the matching stage is to compensate for such geometric transformations. The parameters of the compensating transformation are determined by minimizing the chamfer distance between the template and the transformed candidate profile. The optimization is carried out using a simplex algorithm which requires only the distance function evaluation and no derivatives. The convergence of the simplex algorithm to a local minimum is prevented by a careful initialization of the transformation parameters. The translation parameters are estimated by comparing the position of the nose tip in the two matched profiles. The scale factor is derived from the comparison of the profile heights and the rotation is initially set to zero. Once the optimal set of transformation parameters is determined, the user is accepted or rejected depending on the relationship of the minimal chamfer distance to a pre-specified threshold. The system can be trained very easily. It is sufficient to store one profile per client in the training set.

Frontal image expert

The frontal image verification expert is described in detail in [116] and the description hereafter was based on the presentation of this expert in [98]. This frontal image expert is based on robust correlation of a frontal face image of the person under test and the stored face template corresponding to the claimed identity. A search for the optimum correlation is performed in the space of all valid geometric and photometric transformations of the input image to obtain the best possible match with respect to the template. The geometric transformation includes translation, rotation and scaling, whereas the photometric transformation corrects for a change of the mean level of illumination. The search technique for the optimal transformation parameters is based on random exponential distributions. Accordingly, at each stage the transformation between the test and reference images is perturbed by a random vector drawn from an exponential distribution and the change is accepted if it leads to an improvement of a matching criterion. The score function adopted rewards a large overlap between the transformed face image and the template, and the similarity of the intensity distributions of the two images. The degree of similarity is measured with a robust kernel. This ensures that gross errors due to, for instance, hair style changes do not swamp the cumulative error between the matched images. In other words, the matching is benevolent, aiming to find as large areas of the face as possible, supporting a close agreement between the respective gray-level histograms of the two images. The gross errors will be reflected in a reduced overlap between the two images, which is taken into account in the overall matching criterion. The system is trained very easily by means of storing one template for each client. Each reference image is segmented to create a face mask which excludes the background and the torso as these are likely to change over time.

Vocal expert

The vocal identity verification expert is presented in detail in [22]. This text-independent speaker verification expert is based on a similarity measure between speakers, calculated on second order statistics [21].
In this algorithm a first covariance matrix X is generated from a *reference* sequence, consisting of M m-dimensional acoustical vectors, and pronounced by the person who's identity is claimed:

$$X = \frac{1}{M}\sum_{i=1}^{M} X_i X_i^T,$$

where X_i^T is X_i *transposed.*
A second covariance matrix Y is then generated in the same way from a sequence, consisting of M m-dimensional acoustical vectors, and pronounced by the person under test.
Then a similarity measure between these two speakers is performed, based on the *sphericity measure* $\mu_{AH}(X,Y)$:

$$\mu_{AH}(X,Y) = \log\frac{A}{H},$$

$$A(\lambda_1,\lambda_2,\ldots,\lambda_m) = \frac{1}{m}\sum_{i=1}^{m}\lambda_i = m^{-1} tr\left(YX^{-1}\right),$$

$$H(\lambda_1,\lambda_2,\ldots,\lambda_m) = m\left(\sum_{i=1}^{m}\frac{1}{\lambda_i}\right)^{-1} = m\left(tr\left(XY^{-1}\right)\right)^{-1}.$$

It can be shown that this sphericity measure is always non-negative and it is equal to zero only in the case that the two covariance matrices X and Y are the same. The verification process consists then of comparing the obtained sphericity measure with a decision threshold, calculated on a validation database.
One of the great advantages of this algorithm is that no explicit extraction of the m eigenvalues λ_i is necessary, since the sphericity measure only needs the calculation of the trace $tr(\cdot)$ of the matrix product YX^{-1} or XY^{-1}.

3.4.2 Performances

The performances achieved by the three mono-modal identity verification systems which have been used in these experiments are given in Table 3.1. The results have been obtained by adjusting the threshold at the EER on the validation set and applying this threshold as an a priori threshold on the test set. Observing the results for the profile an the frontal experts it can be seen that, although the optimization has been done according to the EER criterion, the FRR and the FAR are very different. This indicates that for these two experts, the training and validation sets are not very representative of the test set.

Table 3.1: Verification results for individual experts.

Expert	FRR (%) (37 tests)	FAR (%) (1.332 tests)	TER (%) (1.369 tests)
Profile	21.6 [11.4,37.2]	8.5 [7.1,10.1]	8.9 [7.5,10.5]
Frontal	21.6 [11.4,37.2]	8.3 [6.9, 9.9]	8.7 [7.3,10.3]
Vocal	5.4 [1.5,17.7]	3.6 [2.7, 4.7]	3.7 [2.8, 4.8]

3.4.3 Statistical analysis of the different experts

Introduction

A statistical analysis of the individual experts[1] is important to get an idea on one hand of their individual discriminatory power, and of their complementarity on the other hand.

[1]All the following statistical tests have been performed using the SPSS statistical software package [162].

The power of an expert to discriminate between clients and impostors will increase (for given variances) with the difference between the mean value of the scores obtained for client accesses and the mean value of the scores obtained for impostor accesses. The typical statistical test to see if there exist significant differences between the means (or more generally between the statistical moment of first order) of several populations is the so-called analysis of variance (ANOVA). In the general case, this analysis is implemented using an F-test. In the specific case of two populations, this ANOVA could also be performed using an independent samples t-test [123]. Another important characteristic of an expert is its variance (or more generally the statistical moment of second order). The equality of variances can be tested by a Levene test, which is also implemented using an F-test [114]. It is advantageous that the variance of an expert is the same for clients and for impostors, because this leads to simpler methods to combine the different experts (see chapter 6). Obviously we will need to perform t- and F-tests to analyze the means and the variances of the different experts. However, the t- and F-tests give only exact results if the populations have a Normal distribution. So before we can use t- or F-tests, we need to verify the Normality of the different populations. Thus this is the first statistical analysis that we need to perform. Since the ANOVA is only valid if the variances of the different populations per expert are equal, we have to check the equality of variances before performing the ANOVA. These remarks explain the forced order of the first three analyses that are presented below.
We can get an idea of the independence of the different experts (and thus of the amount of *extra* information that each expert brings in), by analyzing their correlation. And a linear discriminant analysis gives us a first idea of the combined discriminatory power of the experts.
Last but not least, the analysis of the extreme values gives us insight into the possible use of personalized approaches.

Analysis of Normality

The purpose of a Normality analysis is to check whether the observed data do or do not support the hypothesis H_0 that the underlying probability density function is Normal. There exist two types of tests to perform this analysis: objective (numerical) and subjective (graphical) tests. An important remark related to the verification of H_0 is that the assumption of Normality is much more difficult to verify when using small sample sizes. In a sample of small size, the probability of verifying that the data is coming from a Normal distribution is actually very small.

The best known representative of the objective/numerical type of tests is the so-called Kolmogorov-Smirnov (K-S) test for goodness of fit, applied to the Normal distribution [96]. The results obtained by this test on our data are presented in Table 3.2. This table shows the values obtained for the K-S statistic, the degrees of freedom (df) and the *significance* of this test at the 95% confidence level. This confidence level leads to a critical value for the significance of 0.05. If the significance is smaller than this critical value, then we *reject* the Normality hypothesis H_0. If on the other hand the significance is greater than the critical value, then we say that we do not have enough evidence to reject H_0, so in a binary decision concept we are forced to *accept* H_0 [123].

Table 3.2: Results for the Kolmogorov-Smirnov test for Normality.

Population	Statistic	df	Significance
Profile clients	.227	37	.000
Profile impostors	.195	1332	.000
Frontal clients	.133	37	.096
Frontal impostors	.052	1332	.000
Vocal clients	.087	37	.200
Vocal impostors	.060	1332	.000

To be able to analyze the results obtained by this K-S test, it is important to know that its severity increases with the sample size of the population. This means that the K-S test is not severe for small sample sizes (as is the case for the client populations), but very severe for large sample sizes (as is the case for the impostor populations). This means that if the results lead to an acceptance of H_0, and if the sample size is sufficiently large, then the Normality assumption is very good. But in the case of a rejection of H_0, this does not mean that we can not accept H_0 at all. In that case we need more information to be able to decide, and therefore we have to go on to the second type of normality tests: the subjective/graphical tests. In our case, the only two populations that are not being rejected by the K-S test as being Normal are the client distributions for the frontal and the vocal experts. But since the sample sizes coming from these two distributions are very small (37), this result has to be used with great care. For all the other populations there is enough evidence to reject the hypothesis H_0.
There exist several types of graphical representations, which can be used

as subjective tests. A first useful type of graphical representation is the so-called Normal Q-Q plot [162]. This kind of representation is shown in Figure 3.7. The idea is to judge if the plotted sample points do follow *sufficiently* (this is clearly subjective) the ideal line which is the representation of a perfect Normal distribution. Depending on this subjective opinion, we reject or not the Normality assumption.

A second representation is the detrended variant of the first one [162], which is shown in Figure 3.8. In this type of graphical representation, one needs to judge whether or not *enough* (another subjective measure) sample points are situated between 2.0 and -2.0 standard normal units. If this is the case we accept the Normality assumption. In the other case, we reject it.

Finally, a third representation can be obtained by plotting the histograms of the different populations and by comparing them with the ideal Normal distribution plots [162]. This kind of graphical representation is given in Figure 3.9. The idea is to check *how well* (again subjective) the actual histograms match the ideal Normal distribution plot.

Taking into account the results of the objective K-S test and having inspected these graphical representations, we conclude that the Normality assumptions in the strict sense of the word are *not* fulfilled for our populations. This is especially true for the profile expert, and in some lesser extent to the vocal expert. The frontal expert deviates the least from a Normal deviation. In practice this means a real drawback, because if the normality hypothesis is satisfied, this generally leads to substantial simplifications.

However, since the observed deviations of Normality are not too important, and taking into account that the classical t- and F-tests are robust with respect to deviations from Normality, we are nevertheless going to perform t- and F-tests, but with the important restriction that the results of these tests will have to be analyzed with the utmost care. In other words, we will accept the results obtained by t- and F-tests if and only if they have a significance level which is far away from the critical value (0.05 for the 95% confidence interval).

Analysis of variance

The results obtained by the Levene test are given in Table 3.3. The H_0 hypothesis is that there are no differences in the variances of the different populations. As we can see by the significance of 0.000 for the vocal and the profile experts and of 0.003 for the frontal expert, this H_0 hypothesis is strongly rejected for all experts. Since the rejection is so strong, we do accept the results of this Levene test and conclude that all experts have

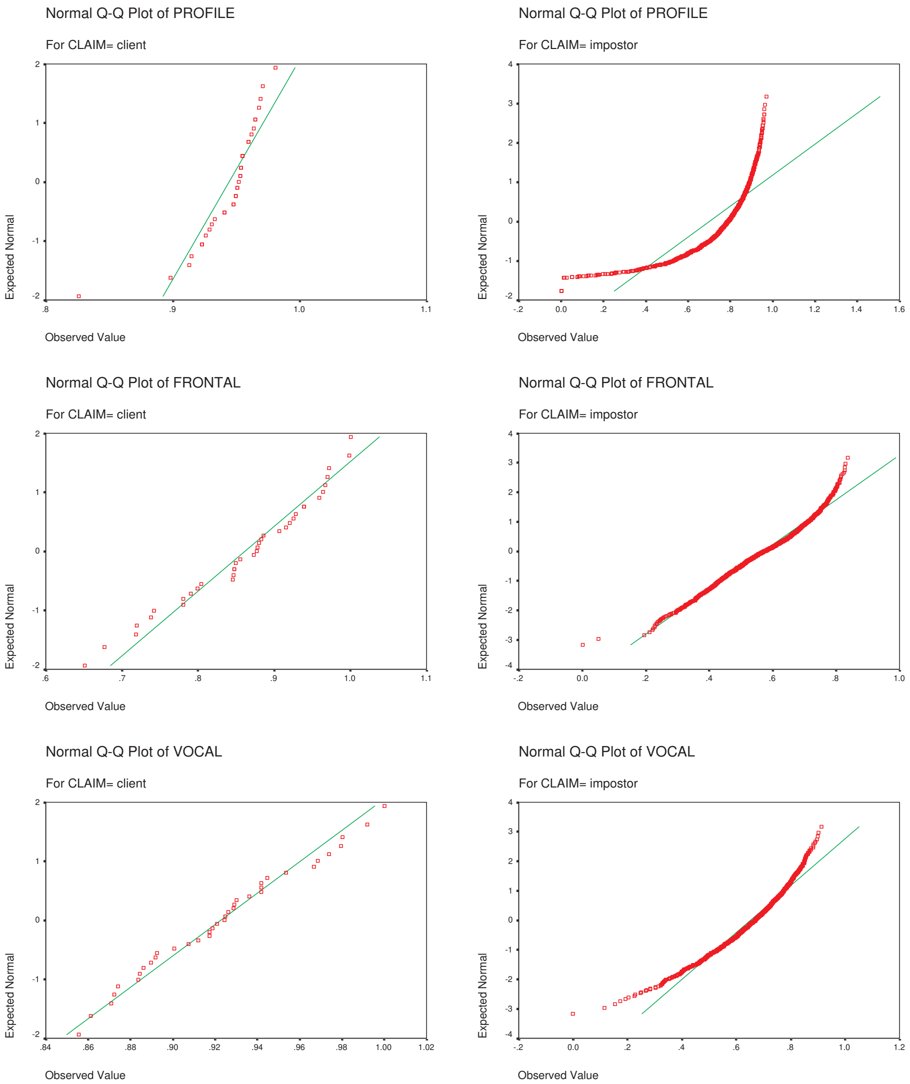

Figure 3.7: Normal Q-Q plots for clients and impostors, ranked per expert.

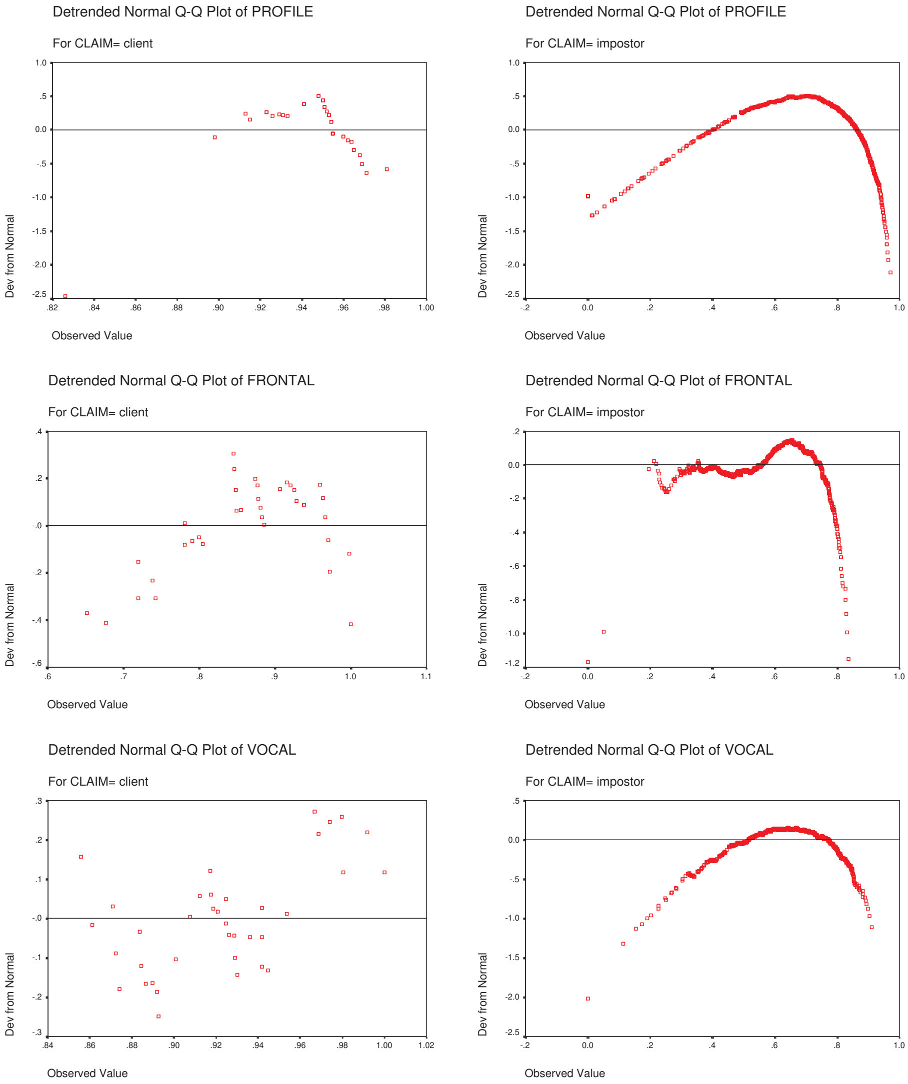

Figure 3.8: Detrended Normal Q-Q plots for clients and impostors, ranked per expert.

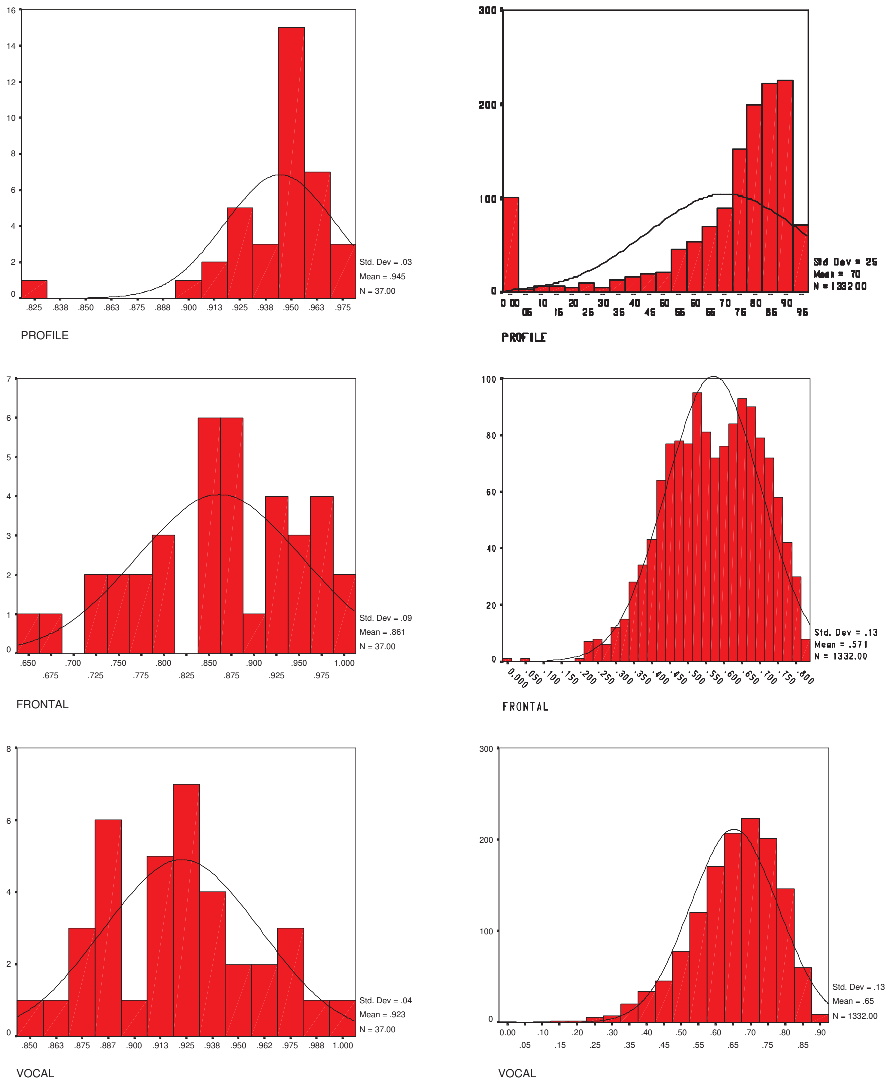

Figure 3.9: Histogram plots for clients and impostors, showing the Normal distribution, and ranked per expert.

significant different variances for both populations.

Table 3.3: Results for the Levene test for equality of variances.

Population	F-Statistic	Significance
Profile	33.125	.000
Frontal	9.062	.003
Vocal	28.284	.000

To confirm the results of this analysis, we can have a look at a *box-plot* representation[2] of the different experts, presented in Figure 3.10. From these representations it follows that for each expert the variance of the client population is indeed significantly smaller than the variance of the impostor population.
Since an ANOVA for analyzing the differences between the means of the populations can strictly speaking only be used with Normal distributions and equal variances, we do have a problem here. Fortunately, since we have only two different populations, we can also use an independent samples t-test, which can be calculated as well for equal variances (in this case the t-test is in an exact form) as for unequal variances (this time the t-test is in an approached form). This means that we will calculate the difference of means hereafter, using an independent samples t-test with *unequal* variances.

Analysis of means

The results of the independent samples t-test with unequal variances are given in Table 3.4. The H_0 hypothesis is that there are no differences between the means of the different populations. As we can see by the significance of 0.000, this H_0 hypothesis is strongly rejected for all experts. Since the rejection is so strong, we do accept the results of this t-test and

[2]The "box" in the box-plot is delimited at the bottom by the 25th and at the top by the 75th percentile. The height of the box thus gives an idea of the variance. The black line in the middle of the box represents the median (50th percentile), which is a robust estimation of the mean. The whiskers underneath and on top of the box respectively show the lowest and the highest values, with the exception of outliers (represented by a circle) and extreme values (represented by an asterisk). An *outlier* is defined by a value which is situated 1.5 times the thickness of the box outside the box, and an *extreme value* is defined by a value which is situated 3 times the thickness of the box outside the box [162].

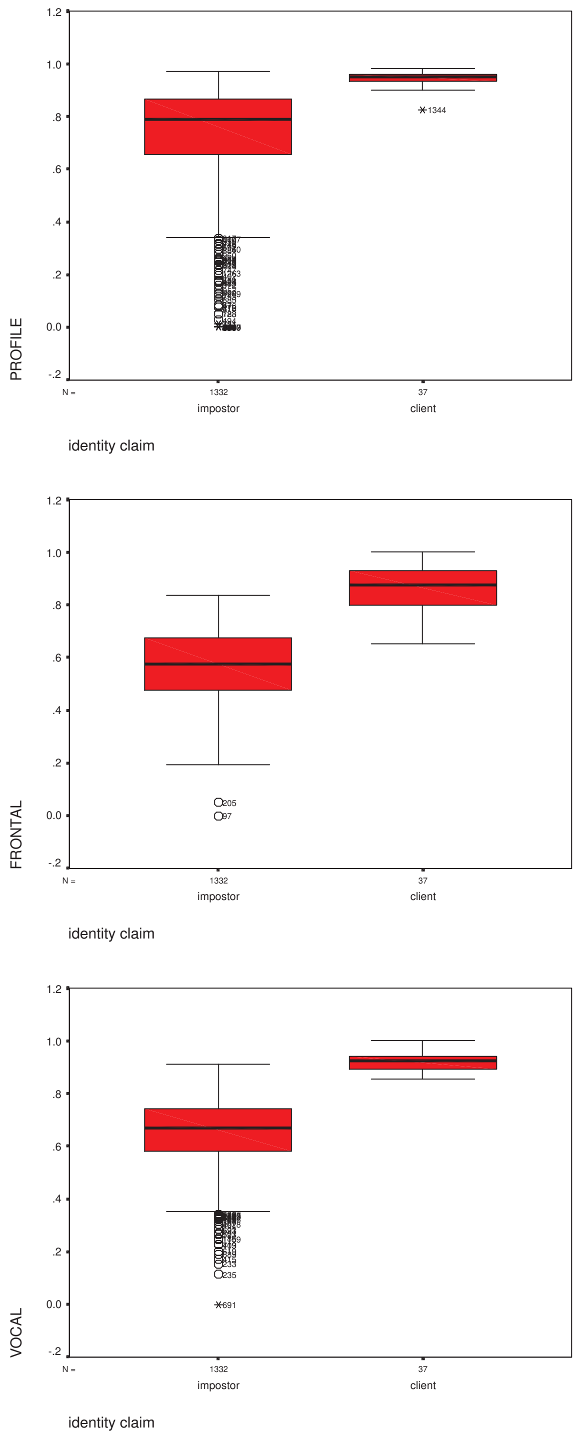

Figure 3.10: Box-plots giving for each expert an idea of the means and variances for the client and impostor scores.

conclude that all experts have significant different means for both populations.

Table 3.4: Results for the independent samples t-test with unequal variances for detecting differences in means.

Expert	t-Statistic	df	Significance
Profile	-29.398	372.665	0.000
Frontal	-18.855	40.275	0.000
Vocal	-38.198	61.642	0.000

To confirm the results of this analysis, we can again have a look at the box-plot representation of the different experts, presented in Figure 3.10. In these representations it can be seen that for each expert the median of the client population is indeed significantly higher than the median of the impostor population. And since the median is a robust estimation of the mean, the same conclusions are valid for the mean.

Analysis of correlation

Another important statistical analysis is the calculation of the correlation that exists between the different experts. A popular way of seeing the importance of this is to say that the more the errors the different experts make are de-correlated, the better our fusion could get since the amount of *new* information introduced by each expert will tend to be larger. The correlation matrix is represented in Table 3.5. As could be expected from the diversity of the experts we are using, the correlation is very low.

Table 3.5: Correlation matrix for our three experts.

Correlation	Profile	Frontal	Vocal
Profile	1.000	0.011	-0.043
Frontal	0.011	1.000	0.256
Vocal	-0.043	0.256	1.000

This correlation can also be visualized by taking the experts two-by-two

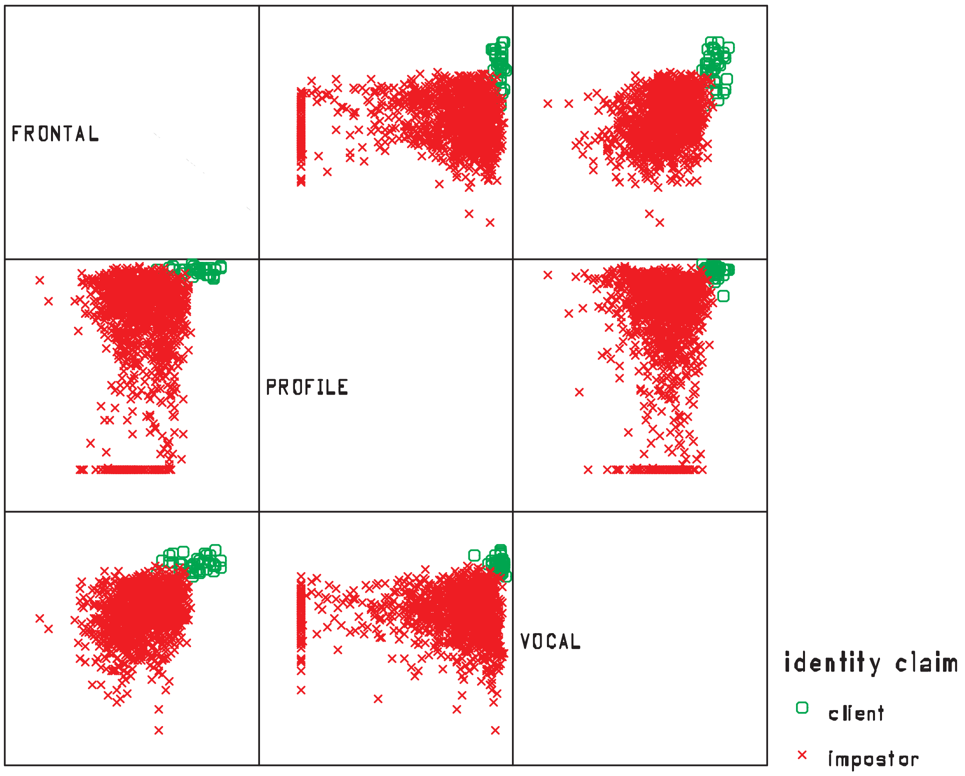

Figure 3.11: Visual representation of the correlation of the different experts taken two-by-two.

A direct conclusion from this correlation analysis is that a Principal Component Analysis (PCA) is not useful here, because of the low correlation between the different experts. Another reason is obviously the fact that we only have used three experts, so there is no real need for performing a data reduction by means of PCA.

Linear discriminant analysis

To examine the discriminatory power and the complementarity of the three experts at the same time, we have performed a linear discriminant analysis [114, 162]. The results of this linear discriminant analysis are shown in Table 3.6.

When we compare the results of the linear discriminant analysis with those obtained by the individual modalities, it is clear that combining the three experts does lead to far better performances, even if the combination is done using a simple linear classifier. This indicates that the different experts do

Table 3.6: Results of the linear discriminant analysis (LDA).

Method	FRR (%) (37 tests)	FAR (%) (1332 tests)	TER (%) (1369 tests)
LDA	0.0 [0.0,9.4]	5.4 [4.3,6.7]	5.3 [4.2,6.6]

have enough discriminatory power and are sufficiently complementary to make it worth while to investigate the combination problem in more detail.

Analysis of extreme values

Another important point in descriptive statistics is the analysis and the handling of extreme values. Normally speaking, extreme values should be discarded from the calculation of characteristic statistical measures such as means or variances. In our work however we will not do that, since these extreme values can contain interesting information as will be shown hereafter.
Indeed, in Figure 3.10 it can be seen that the profile expert presents a large number of extreme values (represented by asterisks) for impostor accesses. These extreme values can also be observed in Figure 3.11, where they form very specific alignments in the four sub-plots where the scores of the profile expert are plotted against one of the axis.
One of the possible explanations for this phenomenon, taking into account the experimental protocol, is that the profile of one the clients is very different from the profiles of all other clients. After analyzing the scores of the profile expert it turned out that, when claiming the identity of client number eight, 22 out of the 36 other clients obtained a score equal to zero! This phenomenon is represented under form of matrix scatter plots in Figure 3.12.
In order to understand this phenomenon, it is interesting to have a look at some typical profile images of client number eight. Such images are presented in Figure 3.13.
From these profile images, it is easy to see that the chin of this specific client is very pronounced, which makes it a very personal characteristic. Therefore few profiles of other clients of the database will present good results when being matched against this typical profile, which obviously leads to very good impostor rejection performances. This observation suggests that,

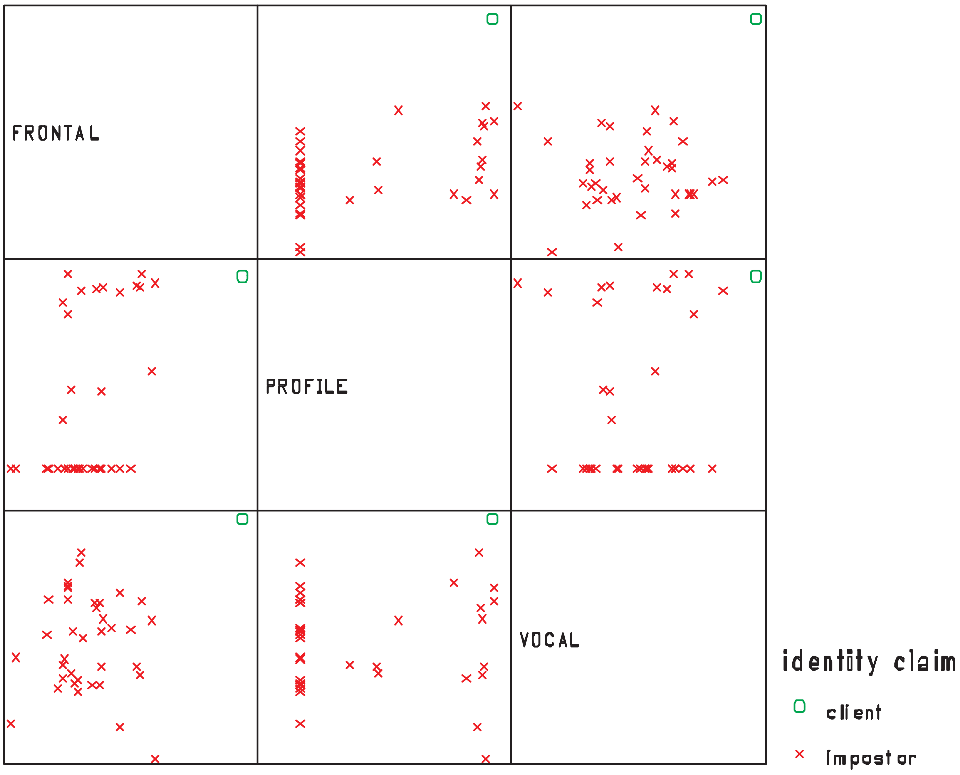

Figure 3.12: Matrix plots representing the scores for all 37 persons claiming the identity of client number eight (i.e. one client access and 36 impostor accesses).

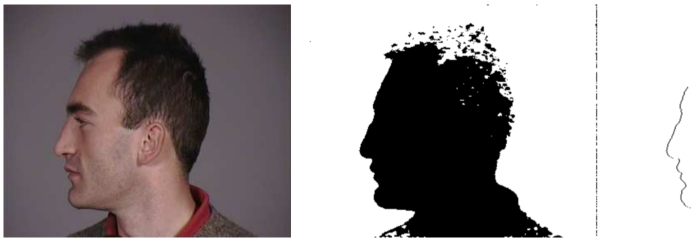

Figure 3.13: Typical profile images of client number eight.

in some specific cases, a *personalized approach* based for instance on specific characteristics of certain persons, might improve system performance substantially. This is an interesting observation, especially when seen in the light of the actual efforts to come to *robust* methods, in which extreme values, such as the ones we have been considering here, are very likely to be excluded!
In this work we did however not use such a personalized approach, since the chosen application does not provide enough training data. This will also become obvious in section 8.3.2, where it will be shown that the validation protocol which is presented there, does not support a personalized approach. Finally one can add that in some applications it is not always possible to use a personalized approach. This is for instance the case in the field of *mine detection*, where the specific characteristics of a certain type of (buried) mine can change dramatically as a function of a large number of parameters related to the burying conditions.

3.5 Comments

In this chapter we have analyzed the performances of three biometric experts (frontal, profile and voice) using the proposed protocol on the M2VTS multi-modal database. Furthermore, the behavior of these experts has been statistically analyzed. This analysis did lead to the following observations:

1. The normality hypotheses for underlying probability distributions for the different populations involved, are not satisfied. The deviations from normality are however not very large.

2. The three experts do show good discriminatory power.

3. The variances of the different populations are not the same.

4. The three experts are complementary.

5. There is evidence that combining the three experts improves the performances onto a level that is better than those of the best expert.

6. There is evidence that suggests that a personalized approach could (further) improve system performance.

Chapter 4

Data fusion concepts

4.1 Introduction

This chapter starts by giving a general overview of the different possible levels on which data can be fused. After having restricted data fusion to decision fusion for this application, several architectures for our fusion module are presented, from which the parallel architecture is chosen. Finally, the decision fusion problem using several biometric expert opinions in parallel, is transformed into a particular *classification* problem. The advantage of this formulation is that it allows to fall back immediately on a broad class of solution techniques coming from the field of Pattern Recognition, which will be experimented and commented in part two of this work [2, 19, 24, 25, 27, 29, 32, 45, 47, 48, 49, 56, 61, 70, 71, 75, 84, 97, 98, 99, 100, 101, 103, 109, 144, 145, 157, 164, 170, 172, 181, 187].

4.2 Taxonomy of data fusion levels

Data fusion in the broad sense can be performed at different hierarchical levels or processing stages. A very commonly encountered taxonomy of data fusion in the broad sense is given by the following three-stage hierarchy [46]:

Data fusion Data fusion in the strict sense is the process of combining directly the data streams of raw measurements coming out of the different sensors. These measurements could for instance be the grey values of the pixels generated by several cameras looking in different parts of the spectrum at the same scene;

Feature fusion Feature fusion is the process of combining features extracted from the raw measurements. Typical vocal features, related to the problem of speaker recognition, could for instance be the cepstral coefficients calculated for different frequency-bands in a multi-band approach [10]. Examples of visual features, linked with identity verification, could be the distance between the eyes and the mouth in a facial image or the distance between the eyes and the nose tip in a profile image.

Decision fusion Decision fusion is the process of combining partial soft (for instance a continuous score between 0 and 1) or hard (0 for a reject and 1 for an acceptance) decisions, given by the different experts. In this case the term *expert* is appropriate, since each single module uses expert knowledge to transform the information carried by the measured data into a decision.

This three-level hierarchy has become fairly accepted, although it stays a matter of subjective choice. One could add an additional dimensionality to the fusion process, representing *temporal* fusion. Temporal fusion can be defined as fusion of data acquired over a period of time. But since this kind of fusion can occur at any of the three levels discussed above, temporal fusion can be viewed as orthogonal to the presented three-level categorization. One could also add *spatial* or *spectral* fusion, which are terms that appear in the literature, but these two processes are essentially examples of data or feature fusion rather than new categories in themselves. In this work, we will limit ourselves to the use and the discussion of *decision* fusion techniques. We will consider that all experts output their local decisions by generating scores in the interval [0,1]. These scores are a measure of their respective belief of the acceptability of the identity claim: the higher the score, the higher the belief that the identity claim is genuine. This way of doing, has the great advantage to separate the design of the specialized experts (which is obviously very application dependent), from the fusion problem. This allows for developing generic decision fusion rules, which are application independent. The only thing we need to suppose is that there exist *good* experts for the application we are studying. Another reason we did choose for a decision fusion rather than for a feature fusion approach, is that this choice decreases the dimensionality of the problem. This reduction in dimensionality is benificial, since it comes along with a reduction in the number of training examples needed for training the different possible fusion modules.

4.3 Decision fusion architectures

Combining the partial decisions from the d different experts in a decision fusion strategy without considering the temporal fusion aspect, could be done using one of the two following basic architectures [46]:

Serial suite As shown in Figure 4.1, a *serial* expert architecture consists of a set of d experts whose decisions are combined in series or tandem. This architecture is for instance well-suited to deal with situations where the different experts do not use a binary {accept, reject}, but rather a ternary {accept, reject, undecided} decision scheme. If in the latter case, the current expert can not decide, he hands the information he has on to the next expert in the sequence. For this serial scenario to be effective, the next expert in line obviously needs to be designed as a real *expert* in dealing with the cases that can not be solved by the previous expert. This architecture is thus particularly well-suited to combine the decisions from experts which have varying ranges of effectiveness and to model sequential decision refining from one sensor to the next. This is not the case in our problem.

Parallel suite As shown in Figure 4.2, a *parallel* expert architecture consists of a set of d experts that are interrogated in parallel. The decisions derived from these experts are combined in parallel by the fusion module. This architecture is particularly well-suited to combine the decisions or scores from experts that are capable of operating simultaneously and independently of one another. This is the case in our problem.

Next to the two fairly simple architectures presented above, one can also imagine more complicated combinations of these two basic schemes, such as parallel-serial or serial-parallel architectures. These combinations are more complex than the previous two and fall outside the scope of this work. Another possible extension of architectures presented so far, is the introduction of some kind of generalized feed-back mechanism. In this case, the idea is to postpone the decision until for instance a new set of measurements has been taken. The basic idea behind this technique can be illustrated by the following example: if a vocal expert is undecided in a ternary decision scheme, the automatic verification system could prompt the user under test to more speech instances, until the vocal expert has enough information to make his decision. This extension also falls outside the scope of this thesis.

Our choice between one of either basic architectures was not only based upon the descriptions presented above, but also on the results of the important research presented in [166]. In this paper, Viswanathan et Al. have compared the serial and the parallel distributed decision fusion mechanisms. Their conclusions are:

1. For certain noise distributions, the parallel structure is not superior to the serial scheme. For additive white Gaussian noise (AWGN) and two sensors for instance, it can be shown that the serial fusion scheme performs better than the parallel one. However, with *three or more* sensors, the performance is essentially the same.
2. As a drawback, any serial network is vulnerable to link failures.
3. Considering the complexity of the serial scheme and the results from the(limited) comparative study, the choice seems to favor the parallel fusion for the distributed decision fusion problem.

The results of this study are a confirmation of the conclusions of the research presented in [149].
Taking into account the descriptions of the basic architectures and the results of the two studies mentioned above, we do opt for a parallel decision fusion scheme in the case of our application.

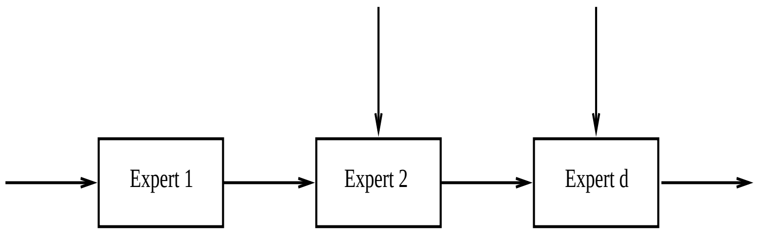

Figure 4.1: A typical serial multi-expert decision fusion architecture.

4.4 Parallel decision fusion as a particular classification problem

In a verification system with d experts in parallel, the decision fusion module using a binary decision scheme has to realize a mapping from the unitary hypercube of $\mathbb{R}^d$ into the set $\{\text{rejected}, \text{accepted}\}$. A classifier having

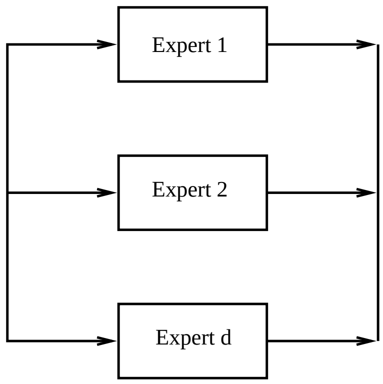

Figure 4.2: A typical parallel multi-expert decision fusion architecture.

a d-dimensional input vector and two classes $\{\text{rejected}\}, \{\text{accepted}\}$ is characterized by such a mapping. The *multi-expert* fusion module can therefore be considered as a *multi-dimensional* classifier. This particular classification case will be our standard fusion approach, since it allows to fall back immediately onto techniques available in the vast field of Pattern Recognition.

4.5 Comments

In this chapter we have presented different aspects of data fusion techniques. For all these aspects, we had to make motivated choices to come to the data fusion solution which suits best our application. These choices are commented hereafter:

1. A first choice that we had to make was that of the level at which the fusion was going to take place. We have opted for the *decision* fusion level because of two main reasons:

   (a) This way of doing, has the great advantage to separate the design of the specialized experts which is obviously very application dependent, from the fusion problem. This allows for developing generic decision fusion rules, which are application independent.

   (b) This choice decreases the dimensionality of the problem.

2. A second choice was the one involving the architecture of the fusion module. We have opted for a *parallel* decision fusion structure for the following reasons:

    (a) The parallel architecture is particularly well-suited to combine decisions or scores from experts that are capable of operating simultaneously and independently of one another.

    (b) In a situation where at least three sensors are combined (as is the case in our application), it has been shown that the performances in noisy conditions of the parallel architecture are not worse than those of a serial structure.

    (c) The parallel structure is less vulnerable than the serial one.

    (d) The parallel structure is less complex than the serial one.

3. Finally we decided to implement the parallel decision fusion strategy as a particular classification problem, to be capable of reusing directly the methods of the field of Pattern Recognition.

Part II

Combining the different experts in automatic biometric multi-modal identity verification systems

Chapter 5

Introduction to part two

5.1 Goal

The goal of this chapter is to justify the use of both parametric and non-parametric methods as paradigms in this second part of the thesis. A first justification can immediately be found in the fact that these two types of methods represent in fact the two possible approaches to statistical inference [169]:

1. the particular (parametric) inference, which aims to create simple statistical methods of inference that can be used for solving real-life problems, and

2. the general (non-parametric) inference, which aims to find one single induction method for any problem of statistical inference.

A more detailed study of advantages and drawbacks of these two approaches is given in the next section.

5.2 Parametric or non-parametric methods?

In the well-studied area of decision fusion, the basic problem is to combine the decisions made by a number of distributed experts [147]. A typical fusion rule in this case is in the form of a Bayesian rule [37, 163] or a Neyman-Pearson test [53, 165]. Such rule can be derived both in the case of independent and correlated individual decisions. In either case, some knowledge of the underlying probability densities is needed for an accurate implementation of the test, under a parametric form. Furthermore, these expressions for the probability densities must be in a convenient form to ensure reasonable computational speeds. If both these conditions are fulfilled together, then parametric methods, such as the ones presented in chapter 6, are the best choice. If on the other hand either the underlying probability distributions are not known, or the Bayesian test is too difficult to implement, then non-parametric methods based on the availability of training samples, such as the ones presented in chapter 7, might be used. The empirical data set, which is finite, can only result in an approximate implementation of the optimal fusion rule. The degree of approximation between a fusion rule that can be obtained if the underlying probabilities are known and its empirical implementation based on a finite sample, depends on the sample size. In this context it is worth mentioning Vapnik's result that it is easier - in an information theoretic sense - to estimate a classifier directly from data than estimating a distribution [169]. Furthermore Rao has shown that, under some smoothness conditions, the optimal fusion rules derived for known distributions can be implemented with an arbitrary level of confidence, given a sufficiently large training sample [146]. These observations do justify the use of the non-parametric methods from chapter 7, which do not estimate distributions, but are just sample based.

Are there then any justifications for the use of parametric methods? Generally, parametric methods are preceded by a model verification step where the assumed distributions are tested, using for example, Kolmogorov-Smirnov type tests. The results from this type of test, assuming Normal distributions, have been presented in section 3.4.3. And, although the Normality hypothesis is - sensu stricto - not fulfilled, we have shown that the deviations from this Normality hypothesis are not too important, so that it is at least interesting to see how parametric methods, assuming Normal distributions, do perform. This has led to the Bayesian approach, explained in detail in the next chapter. The main *advantage* of the Bayesian approach is that it leads to the optimal classifier, in the sense that it implements the lowest Bayes risk [37, 163]. There are however a number of problems with this approach. The most important problem is that the probability density functions (pdfs) have to be estimated correctly. This usually implies the selection of the structure (class of functions) for the approximator and the optimization of the free parameters to best fit the pdf. This optimization is performed on a training set. According to Occam's razor principle (which pleads for preferring the simplest hypothesis that fits the data [121]), the plasticity of the approximator has to be chosen carefully. For highly plastic approximators, quite general pdfs may be approached, but an important (often impossible to obtain) number of samples is needed for performing the training. Furthermore, the training set should be representative (which in general does not correspond to the equal a priori probability hypothesis) and over-training has to be avoided to reach good generalization [23]. On the other hand, by using an approximator with limited plasticity (few parameters, regularization techniques, etc.), fewer examples are needed but more a priori knowledge is implicitly encoded by limiting the possible solutions. This also means that poor prior knowledge will lead to bad results. In practice, the best compromise should be searched, but the true decision rules can most of the time not be implemented and the theoretical minimal Bayes risk remains an unachievable lower bound. This has as a consequence that in the field of pattern recognition and related disciplines, it is common practice to see that other, non-Bayesian, methods are being used. However, sometimes it is possible to justify some of those approaches in the light of the general Bayesian approach, which has the advantage of expliciting the underlying conditions/constraints. This will be done in chapter 6, where we will suppose that the probability distributions involved are (1) simple Gaussian distributions, or (2) members of the exponential family with equal dispersion parameters (the logistic regression model).

5.3 Comments

According to these considerations, it is absolutely worthwhile to investigate both parametric and non-parametric techniques as paradigms for solving our particular classification problem.

Chapter 6

Parametric methods

6.1 Introduction

In this chapter, a trivial but original method is presented first of all: the *monotone multi-linear (or piece-wise linear) classifier* [179, 180]. The main purpose of this method is to be *simple*. Unfortunately the performances of this simple classifier are only good if some user-defined parameters (α and Δ) are correctly set. However, this parameter setting problem is a very delicate one, since it is application dependent and since it needs enough training data. This lack of robustness is mainly due to the fact that (valuable) information with respect to the probability density functions of the different populations is thrown away. Therefore in a fairly early stage of this work it has been decided to stop developing this simple method and to fall back instead on the less original, but more fundamental statistical decision theory, by using so-called *parametric* techniques. In this parametric class, classifiers based on the general Bayesian decision theory (Maximum a-posteriori Probability and Maximum Likelihood) and on a simplified version of it (the Naive Bayesian classifier, which has been applied in the case of simple Gaussians and in the case of a logistic regression model), have been studied [177]. Furthermore experiments have also been done using Linear and Quadratic classifiers. Neural networks form a special case of the parametric family, since the number of parameters to be estimated can be very large. Therefore neural networks are sometimes classified as semi-parametric parameters. Still we will present neural networks in this chapter on parametric techniques, by means of its most popular representative: the Multi-Layer perceptron. All of the aforementioned methods are presented in more detail in the following sections.

6.2 A simple classifier: the multi-linear classifier

To solve the decision fusion problem explained in the previous chapter, the first fusion module that we have studied, was developed on a very simple basis. The main idea behind this first classifier was to approach the *boundary (supposed to be monotonic)* which separates the two populations by a number of monotone hyper-planes. We called this classifier monotone *multi-linear classifier* or *piece-wise linear classifier* in reference to the use of several hyper-planes, each one building a monotone linear classifier. In this particular classifier, the score given by each expert is assumed to be a *monotone* measure of identity correctness. Formally this property can be stated as: given the two scores $s_1 \leq s_2$, if *accept* is the best decision for s_1, then *accept* is the best decision for s_2, and if *reject* is the best decision for s_2, then *reject* is the best decision for s_1. A detailed description of the development and the characteristics of this monotone multi-linear classifier can be found in appendix A. A short summary presenting the results and including the main conclusions is given hereafter.

6.2.1 Decision fusion as a particular classification problem

As was explained in section 4.4, this multi-expert fusion module can be designed as a multi-dimensional classifier, however with some *paradigm* specific constraints (see also Figure 6.1):

Monotonicity The monotonicity hypothesis of the scores as formulated above, induces a monotonicity constraint for the separation border between the two populations, and thus also for our multi-linear classifier. Formally this constraint can be stated as follows: given the two sets of scores $(s_1^1, s_2^1, \ldots, s_d^1)$ and $(s_1^2, s_2^2, \ldots, s_d^2)$ such that $\forall i$: $s_i^1 \leq s_i^2$, if the decision for $(s_1^1, s_2^1, \ldots, s_d^1)$ is *accept*, then the decision for $(s_1^2, s_2^2, \ldots, s_d^2)$ is *accept*, and if the decision for $(s_1^2, s_2^2, \ldots, s_d^2)$ is *reject*, then the decision for $(s_1^1, s_2^1, \ldots, s_d^1)$ is *reject*.

Scarcity of training data In an operational verification system, large amounts of impostor accesses can be simulated with the recordings of other persons. In most applications, client accesses however are scarce since clients would not accept performing long training sessions. This scarcity of client training data has led us to approximate the possibly *complex* separation boundary between the two populations, by a set of *simple* linear segments.

Tunable FAR/FRR trade-off As described in section 2.6, any of the two errors, FAR and FRR, can be reduced as close to zero as desired, with the drawback of increasing the other one. In certain applications security is preferred (FAR small), in others client comfort (FRR small). A user-definable parameter to tune the FAR/FRR trade-off is therefore desired in the development of a classifier. In this multi-linear classifier, this trade-off role will be played by a parameter α which will be defined hereafter.

In the next section we present a classifier (fusion module) designed to take into account the constraints mentioned above.

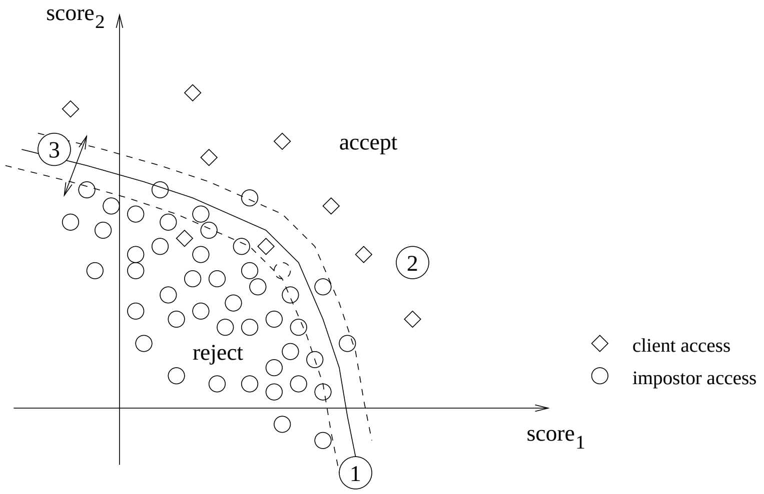

Figure 6.1: Particular classification problem: (1) monotonicity, (2) scarcity of client accesses for training, (3) tunable FAR/FRR trade-off.

6.2.2 Principle

The classifier developed for this fusion module realizes a mapping from the unitary hypercube of $\mathbb{R}^d$, d being the number of experts, into $\{0,1\}$ or $\{\text{rejected}, \text{accepted}\}$. The principle of a multi-linear classifier is to use hyper-planes in $\mathbb{R}^d$, chosen so that *each* pair consisting of a client example C_1 and an impostor example C_2 is *sufficiently* discriminated. The classifier training consists of a supervised phase in which the different hyper-planes are determined in order to optimally separate pairs of points of either class.

The regions generated by these hyper-planes are labeled with the class identifier (accept, reject). At testing, each data point from the test set is simply given the class label of the region it is belonging to.

6.2.3 Training

Given examples of the two classes, the goal is to find hyper-planes separating optimally all pairs of points of either class and to label the generated regions with the corresponding class identifier. The training of the multi-linear classifier consists of three steps:

First step Reduction of training samples using the monotonicity hypothesis;

Second step Determination of the set of hyper-planes that realizes the optimal separation;

Third step Class attribution to the generated regions using the Logical Analysis of Data (LAD) method..

As an initial step, the training data can be cut down using the monotonicity constraint. As a result of this data reduction, only the data points situated along the separation surface of the two classes are maintained. The aim of the training phase is to determine a set of S hyper-planes maximizing the global discrimination between the client and impostor examples. As discussed in [117], when more than one separator is involved, it is more natural to consider the discrimination between the set of pairs of client/impostor points instead of reasoning on the discrimination between the two sets of client and impostor points.
Thus, our goal is to find hyper-planes maximizing the global pairwise discrimination, which we call Δ. This parameter can be set by the user, and it will have an influence on the number of hyper-planes that will be generated. A reference value for Δ is given by Δ_0 defined as half of the minimal Euclidean distance between a pair of client/impostor points. This reference value is nothing else than the discrimination one would obtain using a single hyper-plane cutting orthogonally and at the middle, the segment which links the minimal distant pair of client/impostor points. Clearly, the bigger the required Δ is, the more hyper-planes will be necessary to achieve the discrimination.
Another user-definable parameter is α, which dictates the bias these hyper-planes show towards one of either classes. This bias is achieved by weighting

differently the distances between a particular hyper-plane and data points coming from one or the other class. The default value of α is 1, which is the value that introduces no bias at all. Values of α greater than one introduce a bias towards the client prototypes and values of α smaller than one give a bias towards the impostor prototypes.

Figure 6.2 shows the set of hyper-planes resulting from the application of the method cited above on a synthetic data set for $\alpha = 1$ and $\Delta = \Delta_0$.

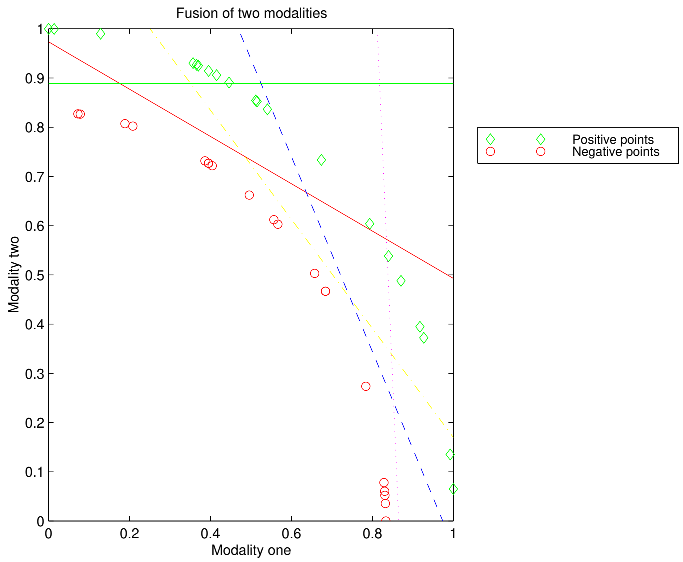

Figure 6.2: Set of hyper-planes generated for $\alpha = 1$ and $\Delta = \Delta_0$.

In practice, the set of hyper-planes is determined in two phases. First, an incremental procedure introduces them one by one. At each iteration, the hyper-planes previously introduced are fixed and the new one is adjusted in order to separate the pairs with the poorest discrimination. This phase terminates when every pair is sufficiently discriminated. Second, a global post-optimization tends to increase the global discrimination of each pair without adding new hyper-planes. The resulting set of S hyper-planes after training induces a partition of the d dimensional space. Each region of this partition is then coded by a word of S bits, indicating its membership to each hyper-plane. Afterwards the label of one of either classes is attributed to each region, using the Logical Analysis of Data (LAD) [30].

Table 6.1: Verification results for the individual experts.

Expert	Mean value		
	FRR (%)	FAR (%)	TSR (%)
Vocal	29.5	0.0	85.6
Profile	11.1	30.6	79.2
Frontal	5.0	7.8	93.6

6.2.4 Testing

During testing the membership of each data point w.r.t. the set of hyper-planes is calculated and each data point receives simply the class label of the region of the hyper-space it is lying in.

6.2.5 Results

The tests have been carried out using the M2VTS database [137] and the same test protocol as the one specified in [55]. This protocol is the very first one that has been developed in the M2VTS project team and it is not the same as the one we have used in the remainder of this work. However, due to the bad results of this simple method, this is of no practical importance since we are not going to use the results of this method for comparison or any other further purposes. In this specific protocol, we have used three shots for training purposes (one shot has been left out for test purposes), each shot containing 36 persons (one person has been left out for impostor tests). Each test set contains 36 client and 36 impostor accesses. Several tests have been performed for different values for α and Δ. For each setting of α and Δ five different *experiments* have been carried out using different shots and persons. The obtained verification results are the means over these five experiments and are expressed in terms of FRR, FAR and TSR (in %).

Table 6.1 shows the verification results for the individual experts where the decision has been taken with a threshold fixed to achieve EER on training data. The verification results after fusion are given in Tables 6.2, 6.3 and 6.4.

The influence of α is shown in Table 6.2. For values of α greater than one we observe an increase of the FRR, as could be expected. By using

Table 6.2: Verification results after fusion as a function of α only.

α	Δ	Mean value		
		FRR (%)	FAR (%)	TSR (%)
0.80	$1.00 * \Delta_0$	22.2	0.0	88.9
0.85	$1.00 * \Delta_0$	18.9	0.0	90.5
0.90	$1.00 * \Delta_0$	13.9	0.0	93.0
0.95	$1.00 * \Delta_0$	27.8	0.0	86.0
1.00	$1.00 * \Delta_0$	28.4	0.0	85.8
1.10	$1.00 * \Delta_0$	30.0	0.0	85.0

Table 6.3: Verification results after fusion as a function of Δ only.

α	Δ	Mean value		
		FRR (%)	FAR (%)	TSR (%)
1.00	$2.00 * \Delta_0$	35.4	0.0	82.6
1.00	$1.00 * \Delta_0$	28.4	0.0	85.8
1.00	$0.67 * \Delta_0$	20.5	0.0	89.7
1.00	$0.50 * \Delta_0$	18.9	0.0	90.5
1.00	$0.40 * \Delta_0$	32.8	0.0	83.6
1.00	$0.25 * \Delta_0$	30.0	0.0	85.0

values of α smaller than one we first see as expected a decrease of the FRR, but when α reaches the value of 0.85, the FRR starts increasing again. Normally when the hyper-planes lean more and more towards the impostor prototypes, we would expect a further decrease of the FRR and a gradual increase of the FAR. The fact that this doesn't happen can be explained by the following observations:

- a change in α not only modifies the position, but also the number of hyper-planes;
- due to the use of the monotonicity propriety for reducing the number of data points in the training phase, the information related to the probability densities of both populations is lost. Also, some of the

Table 6.4: Verification results after fusion as a function of both α and Δ.

α	Δ	Mean value		
		FRR (%)	FAR (%)	TSR (%)
0.85	$0.67 * \Delta_0$	7.3	0.0	96.3
0.85	$0.50 * \Delta_0$	10.6	0.0	94.7
0.90	$0.67 * \Delta_0$	6.7	0.0	96.6
0.95	$0.67 * \Delta_0$	5.6	0.0	97.2
0.95	$0.50 * \Delta_0$	10.6	0.0	94.7

remaining data points might not be representative of the rest of the population (there could be outliers);

- this method is very sensitive to the problem of over-training, as it is based only on the border between the two populations. This means that the generalization capability of this method on unseen data is very bad;
- the impostor and/or client prototypes determined during the training are not really representative for the client and/or impostor accesses made during testing;
- the monotonicity hypothesis is not really valid.

The influence of Δ is shown in Table 6.3. We could expect an increase in performance when the number of hyper-planes becomes larger, since the LAD method seems to get more information for labeling the different sections of the partition of the hyper-space. But as the number of hyper-planes increases, the number of sections in that partition also increases and since in this method we are using only very few training data, the population of these sections will be getting sparse very rapidly. This phenomenon is in our opinion at the basis of what we observe in Table 6.3. When Δ increases, the FRR also increases and when Δ decreases the FRR decreases until a certain point ($0.50 * \Delta_0$) from where on the FRR starts increasing again. This last phenomenon can be explained by the fact that when Δ gets to small, the corresponding hyper-plane(s) don't have to be "very good" (the stop criterion for each iteration is reached sooner), which obviously will lead to more errors.

The combined influence of α and Δ is shown in Table 6.4. These results indicate that this method can have good performances. Indeed, the best results of the multi-linear classifier outperform those of the best individual expert. The main problem with this is the fact that these "best results" are obtained for specific values of the parameters α and Δ and unfortunately there is no easy way of knowing *a priori* which values to give to these parameters for optimizing the classifier in a certain application.

Analyzing the results obtained *after fusion*, one can see that the FAR is in our case always equal to zero. This could indicate that the generated hyper-planes are lying too close to the client prototypes.

6.2.6 Partial conclusions and future work

As shown above, this simple method performs well, only if the user-defined parameters α and Δ have been correctly set. But finding these *correct settings* is a very delicate problem, since it is application dependent and it requires enough representative training data to be available. One of the main reasons for this lack of robustness, is without any doubt the fact that the monotonicity property is used for throwing away valuable data points. This means that we make no use at all of the information which is available in the training data with respect to the probability densities of the different populations. Indeed, the determination of the optimal hyper-planes is completely based on the training data (empirical risk minimization), which can cause problems in the generalization phase. Taking this into account, a possibility to improve the results of this method could be to use Support Vector Machines (SVMs), which minimize the *structural risk* [168, 169].

Another possible improvement of this method could be to *learn* the user-defined parameters, for instance by means of a *genetic algorithm*. However, this has not been done, since the original purpose for developing this method was to design a very simple method.

Since the results of this very first fusion module are only relatively good when the user-defined parameters are correctly set, we decided to stop the development of this classifier, and to use for our further work methods which do take into account the available probability density information. This change of approach has thus led to the use of three classes of methods which are going to be presented hereafter: parametric, semi-parametric and non-parametric methods. The first class to be highlighted is the one of the parametric techniques.

6.3 A statistical framework for decision fusion

In this section, some elementary notions of *statistical* decision theory will be highlighted. More detailed information can be found in references such as [5, 6, 14, 15, 49, 56, 59, 61, 85, 86, 91, 94, 95, 104, 106, 107, 110, 118, 121, 128, 164, 167]. In the references cited above, the statistical decision theory has been explicitly or implicitly divided into two different subfields, which will be dealt with separately in what follows. The first subfield is the *Bayesian* decision theory and the second one is the *Neyman-Pearson* theory. The Bayesian decision theory itself can then again be subdivided into two different approaches, which will also be explained hereafter. The first Bayesian approach is the *minimization of the error probability* and the second one is the *minimization of the Bayes risk.*

Although in the next sections a complete overview of all subdivisions of the statistical decision theory will be given, in this work we only experimented the Bayesian strategy of minimizing the probability of error. We did neither experiment the more general strategy of minimizing the Bayes risk, nor the Neyman-Pearson approach. The reason of this choice is that we did not want to bias the results of this work by specifying different costs or constraints for the different class error probabilities.

6.3.1 Bayesian decision theory

In this section only a brief overview of the most important results of the Bayesian decision theory will be given in the specific case of a two-class problem [177]. These two classes are denoted by C_1 for *Clients* and C_2 for *Impostors*, or by $C_i,\ i = 1, 2$ if the expressions are valid for both classes.

Let X be a random observation coming from one of either classes. In the most general case X will be a multi-dimensional feature vector constructed by the concatenation of all feature vectors $\vec{M}_k$, given to all k experts $k = 1, \dots, n$. The decision problem is to classify correctly each observation in its respective class. Let $P(X, C_i)$ be the joint probability distribution of X and C_i. If we suppose that this joint probability distribution is known, then the connected marginal and conditional probabilities can be derived from it. To measure the performance of a classifier we define a *loss function* l_{ji}, which gives the cost of classifying a class i observation into a class j event. Using this general loss function and applying the definition of the *conditional loss* $R\{C_i|X\}$ for classifying observation X into a class i event found in [56] to our two-class problem, we obtain:

$$R\{C_i|X\} = \sum_{j=1}^{2} l_{ji}.P(C_j|X) \tag{6.1}$$

where $P(C_j|X)$ is the *a posteriori* probability of deciding C_j, given X. This conditional loss can then be used to define the *expected loss* L, also called the *Bayes risk* as follows:

$$L = \int R\{C(X)|X\}.p(X).dX \tag{6.2}$$

where $C(X)$ represents the decision of the classifier. This decision function $C(X)$ depends on the classifier design and it can be easily seen that the expected loss L will be minimized if the classifier is designed such that for each X we have the following:

$$C(X) = C_i \text{ such that } R\{C(X)|X\} = \min_i R\{C_i|X\}. \tag{6.3}$$

Minimizing the probability of error

In our application we have opted for the zero-one loss function defined by:

$$l_{ji} = \begin{cases} 0 & \text{if} \quad i = j \\ 1 & \text{if} \quad i \neq j \end{cases} \tag{6.4}$$

which assigns no loss to correct classification and a unit loss to any error, regardless of the class. With this type of loss function the expected loss L is equal to the *error probability of classification* and the conditional loss becomes:

$$R\{C_i|X\} = \sum_{i \neq j} P(C_j|X), \tag{6.5}$$

and since we obviously have that

$$\sum_i P(C_i|X) = 1 \tag{6.6}$$

we can rewrite the conditional loss as follows:

$$R\{C_i|X\} = 1 - P(C_i|X). \tag{6.7}$$

Under these assumptions, the optimal classifier, defined as the one that achieves minimum expected loss L, is the classifier that implements the following decision rule:

$$C(X) = C_i \text{ if } P(C_i|X) = \max_j P(C_j|X). \tag{6.8}$$

Stated otherwise, this means that the optimal classifier (which achieves the minimum *error probability of classification*) implements the *maximum a posteriori* (MAP) decision rule. The minimum error rate achieved by this optimal classifier is then called the *Bayes risk*. Using Bayes rule, the a posteriori probability $P(C_i|X)$ can be rewritten as:

$$P(C_i|X) = \frac{P(X|C_i).P(C_i)}{P(X)} \tag{6.9}$$

where $P(C_i)$ and $P(X)$ are the *a priori* probabilities of C_i and X respectively, and $P(X|C_i)$ is the conditional probability of X, given C_i. Since $P(X)$ does not depend on the class index, the MAP decision only depends on the numerator of the right-hand side of the previous equation.

$$\text{MAP} = \max_i P(C_i|X) \tag{6.10}$$

$$\text{MAP} = \max_i P(X|C_i).P(C_i) \tag{6.11}$$

By assuming that the a priori probabilities are equal for both classes (this is a strong assumption, which is going to be discussed in section 6.3.6), the MAP decision rule reduces to a maximum conditional probability (MCP) rule. $P(X|C_i)$ is often called the *likelihood* of X given C_i and a decision that maximizes $P(X|C_i)$ is hence also called a *maximum likelihood* (ML) decision.

$$\text{MCP} = \text{ML} = \max_i P(X|C_i) \tag{6.12}$$

It is interesting to see that implementing both these MAP and ML rules in our two-class application can be done by a so-called likelihood ratio test. Indeed, rewriting the MAP decision rule, we can easily obtain the following likelihood ratio decision rule:

$$l(X) \triangleq \frac{P(X|C_1)}{P(X|C_2)} \underset{impostor}{\overset{client}{\gtrless}} \frac{P(C_2)}{P(C_1)} \tag{6.13}$$

In the case of the ML decision rule where we suppose that the two classes have the same a priori probability, the likelihood ratio decision rule simplifies to:

$$l(X) = \underset{impostor}{\overset{client}{\gtrless}} 1 \tag{6.14}$$

From the above equations, it can be directly observed that the only thing that changes between the MAP and the ML decision rule, presented under the form of a likelihood ratio test, is the threshold.

Minimizing the Bayes risk

It can be shown that in the case a different loss-function l_{ji} is chosen (one which assigns for instance a different cost to either type of error FA and FR), one still obtains a likelihood ratio test in which the only thing that is changed is the threshold, this of course to be able to take into account the various costs. In [164], it is shown that the general MAP rule that *minimizes the Bayes risk, corresponding to the specified loss-function*, is given by the following likelihood ratio test:

$$l(X) = \underset{impostor}{\overset{client}{\gtrless}} \frac{P(C_2).(l_{12} - l_{22})}{P(C_1).(l_{21} - l_{11})} \tag{6.15}$$

where

$$\begin{aligned} l_{11} &= \text{cost of correctly accepting a client,} \\ l_{22} &= \text{cost of correctly rejecting an impostor,} \\ l_{12} &= \text{cost of wrongly accepting an impostor} = \text{FA,} \\ l_{21} &= \text{cost of wrongly rejecting a client} = \text{FR.} \end{aligned}$$

6.3.2 Neyman-Pearson theory

In many physical situations it is difficult to assign realistic costs l_{ji} or a priori probabilities. If this is the case, then the Bayesian decision theory can not be used as such. A simple procedure to bypass this difficulty is to work directly with the conditional probabilities and with the two error rates FAR and FRR. As explained in some detail previously, one can not minimize both error rates at the same time. An obvious criterion is then to constrain one of the error rates and to minimize the other one. This is

exactly what is done in the Neyman-Pearson theory. In what precedes it was shown that when one seeks to minimize the probability of error or the more general Bayes risk, one is led to a likelihood ratio test. Here it will be shown that applying the Neyman-Pearson theory also leads to a likelihood ratio test.
The Neyman-Pearson criterion fixes one of the class error probabilities, say FAR, to satisfy:

$$\mathrm{FAR} = \int_{C_1} P(X|C_2).dX = \alpha \tag{6.16}$$

where the integral is calculated over the area where the claim is accepted and α is some predetermined small number, and seeks to minimize the other class error probability:

$$\mathrm{FRR} = \int_{C_2} P(X|C_1).dX \tag{6.17}$$

where the integral is this time calculated over the area where the claim is rejected.
In order to minimize equation (6.17), subject to the constraint (6.16), the following quantity should be minimized:

$$\mathrm{FRR} + \lambda.(\mathrm{FAR} - \alpha) \tag{6.18}$$

where λ is a *Lagrange multiplier*. Doing this, it can be shown (see for instance [164]) that implementing the Neyman-Pearson criterion leads to the following decision rule:

$$l(X) = \underset{impostor}{\overset{client}{\gtrless}} \lambda \tag{6.19}$$

The only problem left now is to determine the threshold λ. To satisfy the constraint (6.16), we choose λ so that FAR $= \alpha$. If we now denote the density of the likelihood ratio $l(X)$ when the person under test is in reality an impostor as $P(l(X)|C_2)$, then we require the following:

$$\mathrm{FAR} = \int_{\lambda}^{\infty} P(l(X)|C_2).dl(X) = \alpha. \tag{6.20}$$

Solving equation (6.20) for λ, provides the threshold. Although equation (6.20) may sometimes be difficult to solve in applications, the philosophy of the Neyman-Pearson method is quite sound, since one often wishes to formulate the decision rule in terms of the desired class error probabilities.

6.3.3 Application of Bayesian decision theory to decision fusion

In a multi-expert decision fusion context, each expert k has access to a feature vector $\vec{M}_k$. Ideally, as explained in section 6.3.1, the decision should be based on $P(C_i|X)$ or, by expliciting X, on $P(C_i|\vec{M}_1, \vec{M}_2, \dots, \vec{M}_n)$, taking into account the loss function. However, this usually implies the direct use of the feature vectors $\vec{M}_k$, which might be undesirable or in some practical cases even impossible. The direct use of these feature vectors means that we completely deny the pertinence and usefulness of the available experts. Even if the theory obtained in section 6.3.1 states that the optimal classification should be based on $P(C_i|\vec{M}_1, \vec{M}_2, \dots, \vec{M}_n)$, this theory says nothing on how to obtain the best estimate of these probabilities.
A brute force approach, in which one tries to estimate directly the above probabilities using for instance Multi Layer Perceptrons (MLPs), might be appealing because it could lead to the optimum decision and it does not rely on any of the hypotheses that we will introduce in the next sections. However, if one would do that, this would only result in *estimates* of the real probabilities and these estimates could be so bad that they would be useless, because the corresponding MAP or ML decision would be far away from the desired optimum decision. Several reasons may hinder a good estimation of the underlying probabilities:

- First of all, large *multi-modal* databases are extremely rare and very expensive. It is more realistic to look for a large database for each *separate* modality.

- Furthermore, the best estimation of the desired probabilities may be obtained by making a good usage of the available databases and the a priori knowledge, and by limiting the parameter space by making appropriate assumptions. This a priori knowledge has to be introduced into the system by (human) experts. This might be easier at the level of each modality (mono-modal experts are needed) than at the fusion level. In most cases the introduction of this a priori knowledge has been done, either explicitly or implicitly, by the designer of the available mono-modal experts.

Good experts make an efficient use of the available databases and a priori knowledge. The different mono-modal experts could then be designed and tuned adapting their complexity (and thus their performance [57]) to the size of the respective mono-modal databases. In our opinion these kinds of experts should thus be used as they are and they should not be replaced by some kind of a pseudo-optimal probability density function estimator (such as an MLP) which denies their pertinence and expertise. Therefore our objective will be to make the best decision, based on the output (scalar) *scores* s_k, $k = 1, \ldots, n$ of the available experts, and not on their input *feature vectors*. Starting from this point, several approaches to estimate $P(C_i|s_1, s_2, \ldots, s_n)$ will be presented in the next sections.

6.3.4 The naive Bayes classifier

Formalization

Introducing the independence hypothesis transforms the general Bayesian approach presented above into the so-called *naive Bayes classifier* [119, 121]. This hypothesis is acceptable when looking at the results of the correlation analysis presented in section 3.4.3. If we suppose that the different experts are independent, then the scores of these experts are independent given either class. In our particular case, using the scores that are provided by the d experts $(s_1, s_2, \ldots, s_d)$, and denoting the two classes by C_1 and C_2 for clients and impostors respectively, this can be formalized by the following two specific hypotheses:

$$h1 : P(s_1, s_2, \ldots, s_d|C_1) = \prod_{k=1}^{d} P(s_k|C_1) \tag{6.21}$$

$$h2 : P(s_1, s_2, \ldots, s_d|C_2) = \prod_{k=1}^{d} P(s_k|C_2) \tag{6.22}$$

Under these two hypotheses, we show in appendix E that

$$P(C_1|s_1, s_{2,\ldots,} s_d) = \frac{1}{1 + \exp\left[-\left\{\left(\sum_{k=1}^{d} x_k\right) + x_0\right\}\right]} \tag{6.23}$$

where

$$x_k = ln\frac{P(s_k|C_1)}{P(s_k|C_2)} \tag{6.24}$$

$$x_0 = ln\frac{P(C_1)}{P(C_2)} \tag{6.25}$$

and s_k is the scalar score given by the $k-$th expert.
The interest of obtaining the mathematical expressions presented in equations (6.23), (6.24), and (6.25) is that in some specific cases they can be reduced to very simple expressions. This will be shown in section 6.3.5.

6.3.5 Applications of the naive Bayes classifier

Simple Gaussian distributions

In this section, we will particularize the general approach of the naive Bayes classifier under the hypothesis that all mono-variate conditional probabilities $P(s_j|C_i)$ are Gaussian. The only parameters to estimate are the mean and the variance of the mono-variate Gaussian distributions. Those parameters are estimated using the training data set. Note that no multi-modal database is needed to do this. Then the multi-variate conditional probability $P(s_1, s_2, \ldots, s_d|C_i)$ may be computed. Under the independence hypotheses we made in section 6.3.4 it is a product of Gaussian distributions, which is also a Gaussian with a diagonal covariance matrix. In the next section, we will show that if the Gaussian distributions have the same variance for the client and the impostor classes, then the a posteriori probability is a logistic function.
Figure 6.3 shows the modeled normal client and impostor distributions in our application, when using the vocal expert only. It can be seen that there exists an overlap between the two distributions, which will be responsible for the classification errors. This is again shown in Figure 6.4 for the profile and vocal experts. This Figure clearly illustrates that the fusion of (well-chosen) experts or modalities may significantly reduce the classification errors. The overlap indeed still exists, but it has become smaller.
If furthermore the a priori probabilities are equal, the ML decision rule from equation (6.12) may be used. The results obtained in our specific application using the three experts and the approach explained above, are shown in Table 6.5. Note that in our case, the equal prior hypothesis is questionable. Indeed, the a priori probabilities are (for the test set) equal to 1/37 and 36/37 for respectively clients and impostors. Using these numbers in the MAP decision rule from equation (6.11) it can be observed in Table 6.5 that changing the ML rule into a MAP rule leads to a decrease in the FAR and an increase in the FRR. This was expected. Indeed, since the a priori probability of an impostor is much higher than the one for a client, the major tendency will be to reject the claim (which is well adapted to impostors, hence the decrease in FAR), but which is not so good for clients

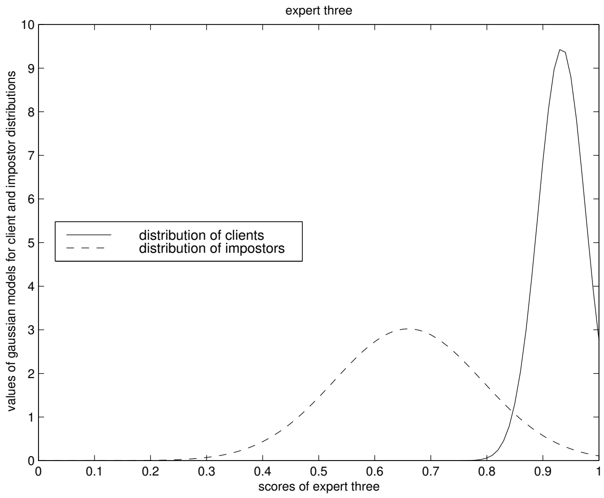

Figure 6.3: Typical overlap between modeled client and impostor distributions for the vocal expert.

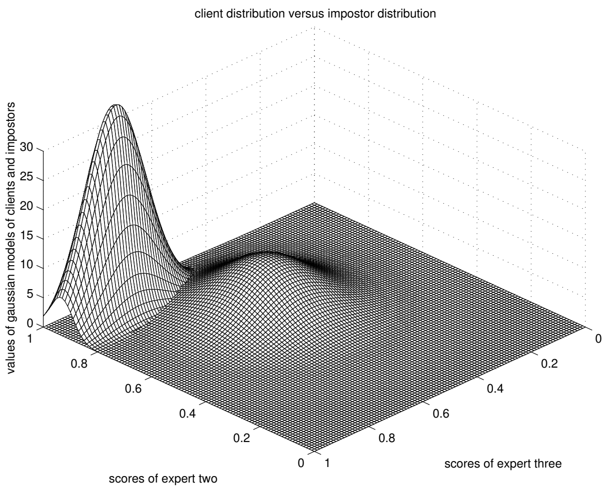

Figure 6.4: Overlap between modeled client and impostor distributions for the profile and the vocal experts.

(thus the increase in FRR). This is a global observation using Bayesian techniques, where the class with the smallest a priori probability tends to be underestimated. Since the only difference between the MAP and the ML decision rules is the decision threshold, these two decision rules implement the same discriminant or decision function, which is shown in the case of the profile and vocal experts in Figure 6.5.

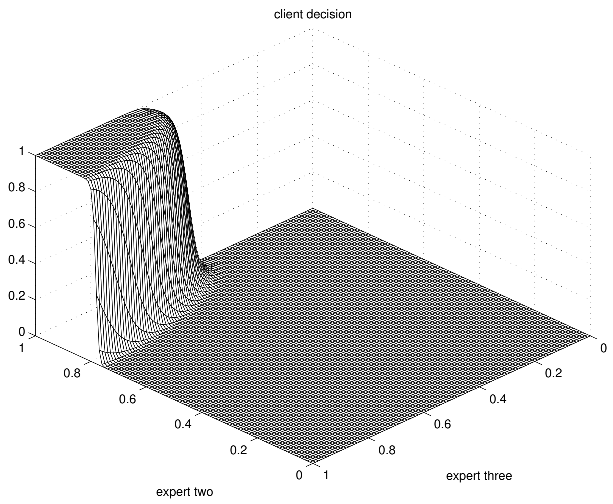

Figure 6.5: Shape of the MAP and ML decision function obtained using simple Gaussian distributions for the profile and the vocal experts.

It can be seen that although we had to make a number of hypotheses in order to reduce the computational complexity, the results given by the MAP and the ML rule supposing simple Gaussian distributions are relatively good.

The logistic regression model

Maintaining the independence hypothesis, another particularization of the general framework of section 6.3.4, can be obtained by assuming that for each expert ($k = 1, \dots, d$) the mono-modal conditional probability density functions for both classes are members of the exponential family with equal dispersion parameters. This assumption is formally expressed by the

Table 6.5: Verification results for a classifier using simple Gaussian distributions.

Rule	FRR (%) (37 tests)	FAR (%) (1332 tests)	TER (%) (1369 tests)
ML	2.7 [0.5,13.8]	0.7 [0.4,1.3]	0.7 [0.4,1.3]
MAP	5.4 [1.5,17.7]	0.0 [0.0,0.3]	0.1 [0.0,0.5]

following equations:

$$P(s_k|C_1) = f(s_k).\exp\left[(c_k.s_k + c_{k0})\right], \tag{6.26}$$

and

$$P(s_k|C_2) = f(s_k).\exp\left[(i_k.s_k + i_{k0})\right], \tag{6.27}$$

where $f(\cdot)$ is any monotonic function, and c_k, c_{k0}, i_k, i_{k0} are two sets of two class-dependent parameters for Clients and Impostors respectively. Using this, it is easy to see that equations (6.23), (6.24), and (6.25) of section 6.3.4 reduce to

$$P(C_1|s_1, s_2, \dots, s_d) = \frac{1}{1+\exp\left[-g(s)\right]} = \Pi(X) \tag{6.28}$$

where

$$g(s) = \beta_0 + \beta_1.s_1 + \dots + \beta_d.s_d, \tag{6.29}$$

$$\beta_0 = \sum_{k=1}^{d}(c_{k0} - i_{k0}) + ln\frac{P(C_1)}{P(C_2)}, \tag{6.30}$$

and

$$\beta_k = c_k - i_k \text{ for } k = 1, \dots, d. \tag{6.31}$$

Equation (6.28) is known as the logistic regression model or logistic distribution function and the linear expression in the scores of the experts presented in equation (6.29) is called the *logit* or *log-odds* transformation [83, 174]. This method performs a statistical analysis of the observed (training) data and the discrimination function it implements is the *logistic distribution function*, which is formalized hereafter:

$$E(Y|X) = \Pi(X) = \frac{\exp[g(s)]}{1 + \exp[g(s)]}$$

In this expression $E(Y|X)$ is the conditional probability for the (binary) output variable Y given the d-dimensional input vector X, with $g(s) = \beta_0 + \beta_1.s_1 + \ldots + \beta_d.s_d$ and $X = (s_1, s_2, \ldots, s_d)$. This equation gives as a result for the input pattern X, the probability $\Pi(X)$ of belonging to the class of clients ($Y = 1$) and, in an indirect manner, the probability $[1 - \Pi(X)]$ of belonging to the class of impostors ($Y = 0$). The typical non-linear shape of the discriminant function implemented by the logistic regression model is shown in Figure 6.6 for the specific case where there are only two experts.

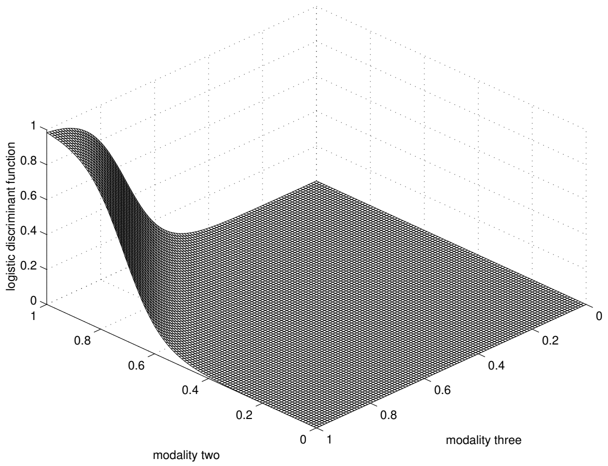

Figure 6.6: Shape of the discriminant function obtained using the logistic regression model for the profile and the vocal experts.

As it can be seen, the mathematical expressions of equations (6.23), (6.24), and (6.25) have been reduced to very simple expressions. Once the logistic regression parameters β_i have been obtained, $g(s)$ is calculated as a linear function of the scores $s_1, s_2, \ldots, s_d$ of the different experts. It can easily be shown that this linear function $g(s)$ is actually nothing else than the equation of the hyper-plane which acts as separation frontier in the d-dimensional space of the expert scores. To see this, we simply set the decision threshold to the MAP value of 0.5 (see further). In this case we obtain the following expressions for the separation frontier between the two classes:

$$\Pi(X) = \frac{\exp[g(s)]}{1+\exp[g(s)]} = 0.5, \tag{6.32}$$

which leads to

$$\frac{1}{1+\exp[-g(s)]} = 0.5, \tag{6.33}$$

which is equivalent to

$$\exp[-g(s)] = 1, \tag{6.34}$$

which leads to

$$g(s) = 0, \tag{6.35}$$

what we wanted to proof.
In [119], it is shown that the hyper-plane obtained by the logistic regression performs better over a large variety of databases than the one obtained when using the linear discriminant function. This is especially true in the case that the normality assumption (which has to be made when using linear discriminant analysis), is invalid [75]. And, as it has been shown in chapter 3, this is typically the case in our type of application.
To obtain the $(d+1)$ logistic regression parameters β_i with $i = 0, 1, \ldots, d$, the *maximal likelihood* principle is used, so as to maximize the probability of finding the observed training data. Since each β_i with $i \neq 0$ multiplies one of the d experts, and since *all* experts output scores in the same range *i.e.* between 0 and 1, the value of β_i is a measure of the importance of the i-th expert in the fusion process, under the hypothesis of equal dispersion parameters. A large β_i indicates an important expert, a small β_i indicates an expert that does not contribute very much. Table 6.6 shows the values for the parameters β_i obtained for the three experts in our application. These values have been calculated on the training data using the SPSS software package [162]. As can be seen, the vocal expert is the most important one for the logistic regression and the frontal expert is the least important one. This interesting property allows the designer to choose the most relevant experts very easily, without using Principal Component Analysis or other less convenient methods as suggested in [152].

Table 6.6: Parameter estimations for the logistic regression model.

Expert	Parameter	Value
Constant	β_0	17.183
Profile	β_1	-0.4801
Frontal	β_2	-0.1865
Vocal	β_3	-0.6172

Note that the acceptable class of conditional probabilities, as defined by equations (6.26) and (6.27), is known as the exponential family with equal dispersion parameters for the two classes (clients and impostors) [91]. One particular case of this family is the well-known Gaussian distribution, which transforms equations (6.26) and (6.27) into:

$$P(s_k|C_1) = \frac{1}{\sqrt{2\pi}.\sigma_k}.\exp\left[-\frac{(s_k-\mu_k^c)^2}{2\sigma_k^2}\right] \tag{6.36}$$

and

$$P(s_k|C_2) = \frac{1}{\sqrt{2\pi}.\sigma_k}.\exp\left[-\frac{(s_k-\mu_k^i)^2}{2\sigma_k^2}\right], \tag{6.37}$$

where μ_k^c and μ_k^i represent the mean of respectively the class of clients and impostors and σ_k^2 is their *common* variance. In this particular case equations (6.30) and (6.31), give:

$$\beta_0 = \sum_{k=1}^{d} \frac{(\mu_k^i)^2-(\mu_k^c)^2}{2\sigma_k^2} + ln\frac{P(C_1)}{P(C_2)}, \tag{6.38}$$

and

$$\beta_k = \frac{\mu_k^c-\mu_k^i}{\sigma_k^2}, \tag{6.39}$$

where β_k is nothing else than the difference of the means of the distributions for the two classes for the k-th expert, divided by their common variance. The general observation that the value of β_k is a measure of the importance of the k-th expert in the fusion process, together with the expression of β_k in this particular case, is in accordance with our intuition which says that

an expert behaves better when the distributions of the two classes (clients and impostors) are more separated and when their variance is smaller.
It is also interesting to compare the approach using the strict Gaussian case of section 6.3.5 with this particular member of the exponential family with equal dispersion parameters. In the latter approach, the equality constraint on the dispersion parameter imposes the same variance for both classes, which is strictly spoken a restriction compared to the former approach, where the variances may differ. However, the latter approach also works when we assume that the class-conditional probability densities are members of the (same) exponential family distribution with equal dispersion parameters. Since this approach is invariant to a *family* of classification problems, it is more *robust* than the former approach, as we do not require here *one* particular distribution to be specified [91]. Another advantage of the approach based on the logistic regression model is that fewer parameters need to be estimated ($d+1$ instead of $4d$ (μ_k^i and σ_k^i for each class i and for each expert k) or $3d$ (if we suppose that the two classes have equal variances for each expert) for the approach in section 6.3.5).
Once the different parameters β_k have been estimated on the training data, an unknown test pattern is simply classified by calculating $\Pi(x)$. This result needs then to be compared with a threshold. In the case of the MAP decision rule (which minimizes the total number of errors) and in the simple case of our two-class problem, the optimal MAP threshold is 0.5. To show this, we consider the case in which we classify the unknown person as a client. According to the MAP decision rule, this will be the case if the a posteriori probability $P(C_1|X)$ is greater than the a posteriori probability $P(C_2|X)$ (6.11). Since $P(C_1|X) = \Pi(X)$ and $P(C_2|X) = 1 - \Pi(X)$, the MAP decision rule reduces to:

$$\Pi(X) > 1 - \Pi(X), \tag{6.40}$$

which leads to

$$\Pi(X) > 0.5 \tag{6.41}$$

what we wanted to proof.
If we use another optimization criterion, we obtain another decision threshold. Instead of the optimal threshold in the sense of Bayes, we can also choose other thresholds. One such a threshold is the EER threshold which is calculated on the training database, as explained in section 3.3. In our case, the value of this EER threshold is 0.012. The verification results for these two operating points are given in Table 6.7. It can be seen that the theoretical optimal threshold leads indeed to the smallest TER.

Table 6.7: Verification results for a classifier using a logistic regression model.

Threshold	FRR (%) (37 tests)	FAR (%) (1332 tests)	TER (%) (1369 tests)
0.500 (MAP)	2.7 [0.5,13.8]	0.0 [0.0,0.3]	0.1 [0.0,0.5]
0.012 (EER)	0.0 [0.0, 9.4]	1.1 [0.6,1.8]	1.1 [0.7,1.8]

6.3.6 The issue of the a priori probabilities

We have shown in section 6.3.1 that in the Bayesian framework, the optimal *Bayes* decision is a function of the a priori probabilities (note that this is not the case if one has chosen for a Neyman-Pearson approach). These probabilities may be fed in explicitly (e.g.: the MAP rule in the case of the simple Gaussian distributions) or may be learned (e.g.: the case of the logistic regression model) on a training database. In the latter case, it is the frequency on the learning database that is learned. However, ideally the frequency that the system will face during operational deployment should be used as the a priori probability.

Most often, the frequency on the training database is not representative of the frequency that the system will encounter during operational deployment (which is one of the problems encountered in the specific case when using MLPs). Indeed, this requirement is generally not taken into account when the database is built. As an example, the M2VTS contains 37 persons and all comparisons that are performed take into account a client frequency of 1/37. However, the frequency of clients that will present themselves to the system during deployment will be function of the application.

In some learning schemes (e.g. MLP, logistic regression), the database should be balanced to reach good results. Indeed, learning is often performed by means of a non linear optimization. With an unbalanced database, the poorly represented class has only a very week contribution to the error function leading to a near optimal solution for the trivial classifier that always assigns the pattern to the most represented class.

Since unfortunately, the deployment frequency is often difficult to estimate, this means that in many cases the training is done with a frequency that is not representative. This argument is often used to criticize the Bayesian approach and it has been presented as an advantage for other methods

which also try to minimize the number of classification errors and which do not need these a priori probabilities. However, in our opinion the optimal *Bayes* decision (typically the minimization of the number of errors) *does* depend on the a priori probability. This may be illustrated by the following example. If a man has to classify a person moving in the dark in his home using only visual information, it will probably be classified as his partner, although no detail is visible but the a priori probability for a person moving in the house is high for the partner. If the person was in reality a burglar, one could easily be mislead. On the other hand, that same man will have no difficulties using only visual information by daylight to classify a person moving in the house as a burglar, even if the a priori probability for the partner is higher. This hypothetical example shows that if the measurements are sufficiently discriminative, we do not (need to) rely on a priori probabilities. On the other hand, if the measurements are not discriminative enough, we do need the a priori probabilities.
This effect may be understood mathematically by analyzing equation (6.23). The discrimination power of the measures is represented by $\sum_k x_k$, whereas the a priori probabilities are represented by x_0. It is only in the case that the first term is significantly bigger than the second one that the a priori probabilities have no effect on the global sum.
The above reasoning shows that the a priori probabilities do indeed affect the optimum decision, especially when the discrimination power is low. Schemes that are argued to be preferable because they do not need these a priori probabilities as an input, try to solve this problem by for instance keeping ambiguities if discrimination power is low.
We believe that in practice a reasonable range for the a priori probabilities should be considered. The corresponding range of the posterior probabilities may then be computed and the effect on the optimal decision can then be analyzed.

6.3.7 Quadratic and Linear classifiers

Quadratic classifier

It has been shown in section 6.3 that various decision criteria led to a likelihood ratio test of the form

$$l(X) = \frac{P(X|C_1)}{P(X|C_2)} \underset{impostor}{\overset{client}{\gtrless}} \lambda \tag{6.42}$$

where the threshold λ is defined explicitly in terms of the a priori probabilities and the Bayes costs for criteria that minimize the probability of error or the Bayes risk (see section 6.3.1), and defined implicitly for the Neyman-Pearson criterion (see section 6.3.2).
Let us suppose now that the *multivariate class conditional probabilities* $P(X|C_i)$ are characterized by d-dimensional Gaussian densities with d being the number of experts and with respective means μ_i and covariance matrices Σ_i:

$$P(X|C_1) = \frac{1}{(2\pi)^{\frac{d}{2}}.\sqrt{|\Sigma_c|}}.\exp\left[-\frac{1}{2}(X-\mu_c)^T\Sigma_c^{-1}(X-\mu_c)\right] \tag{6.43}$$

and

$$P(X|C_2) = \frac{1}{(2\pi)^{\frac{d}{2}}.\sqrt{|\Sigma_i|}}.\exp\left[-\frac{1}{2}(X-\mu_i)^T\Sigma_i^{-1}(X-\mu_i)\right] \tag{6.44}$$

In this particular case, it is trivial to see that the likelihood ratio $l(X)$ from the decision rule in equation (6.42) becomes:

$$l(X) = \sqrt{\frac{|\Sigma_i|}{|\Sigma_c|}}.\exp\left[-\frac{1}{2}(X-\mu_c)^T\Sigma_c^{-1}(X-\mu_c)+\frac{1}{2}(X-\mu_i)^T\Sigma_i^{-1}(X-\mu_i)\right]. \tag{6.45}$$

If the quantity $h(X)$ is defined as:

$$h(X) = -2\ln l(X), \tag{6.46}$$

then the decision rule 6.42 can be expressed as:

$$h(X) = (X-\mu_c)^T\Sigma_c^{-1}(X-\mu_c) - (X-\mu_i)^T\Sigma_i^{-1}(X-\mu_i) + \ln\frac{|\Sigma_c|}{|\Sigma_i|} \underset{impostor}{\overset{client}{\lessgtr}} T \tag{6.47}$$

where

$$T = -2\ln\lambda. \tag{6.48}$$

The decision rule can then also be written as:

$$h(X) = X^TAX + b^TX + c \underset{impostor}{\overset{client}{\lessgtr}} T \tag{6.49}$$

where

$$
\begin{aligned}
A &= \Sigma_c^{-1} - \Sigma_i^{-1}, \\
b &= 2\left(\Sigma_i^{-1}\mu_i - \Sigma_c^{-1}\mu_c\right), \\
c &= \mu_c^T\Sigma_c^{-1}\mu_c - \mu_i^T\Sigma_i^{-1}\mu_i + \ln\frac{|\Sigma_c|}{|\Sigma_i|}.
\end{aligned}
$$

The decision rule obtained in this specific case is called a *Gaussian* or *quadratic* classifier. It can be shown easily (see for instance [164]) that the quadratic classifier computes the Mahalanobis distances to the mean of either class and compares the difference in those distances to a threshold. Another interesting property of this classifier is that the decision boundary that it defines is a general quadratic surface. The quadratic classifier can also be applied to scores that are not characterized by Gaussian density functions. But in this case, the probability of error (or the Bayes risk, or the Neyman-Pearson criterion, depending on which threshold λ has been chosen) is not necessarily minimized. Still one could then interpret the quadratic classifier as defining a decision boundary that is best matched to the second moment statistics of the scores.

It should be noted that this approach is different from the one that we have followed when applying the naive Bayes (NB) classifier to simple Gaussian distributions in section 6.3.5. Indeed, in the case of the naive Bayes classifier approach using simple Gaussian distributions, it was shown that the covariance matrices of the multivariate conditional probability densities were diagonal, due to the independence hypotheses. This means that in that particular case only $4d$ parameters have to be estimated (the mean μ and variance σ for each expert and for either class). In the case we are considering here, no independence hypotheses are made and $2[d + d(d+1)/2] = 2d[1 + (d+1)/2]$ parameters need to be estimated (the d-dimensional mean μ and the $d(d+1)/2$ elements of the *symmetric* covariance matrix Σ for either class). The difference between these two approaches increases rapidly with the number of experts d, as is shown in Table 6.8.

Linear classifier

In the case that the two covariance matrices are equal, that is $\Sigma_c = \Sigma_i = \Sigma$, the matrix A of equation (6.49) is identically zero and the decision rule

Table 6.8: Number of parameters to estimate as a function of the number of experts d.

d	NB using Gaussians	Quadratic classifier
1	4	4
2	8	10
3	12	18
4	16	28
5	20	40

becomes:

$$h(X) = b^T X + c \underset{impostor}{\overset{client}{\lessgtr}} T \tag{6.50}$$

where

$$\begin{aligned} b &= 2\Sigma^{-1}(\mu_i - \mu_c), \\ c &= \mu_c^T \Sigma^{-1} \mu_c - \mu_i^T \Sigma^{-1} \mu_i. \end{aligned}$$

The decision boundary thus reduces to a linear boundary (hyper-plane) and the classifier is correspondingly called a *linear* classifier. In the open literature, the term Linear Discriminant Analysis (LDA) is also often used [79, 118]. It is again important to note that this approach is different from the one that we have followed when applying the naive Bayes classifier to distributions coming from the exponential family with equal dispersion parameters in section 6.3.5. Indeed, in the case of the application of the naive Bayes (NB) classifier approach, the (diagonal) covariance matrices of the multivariate conditional probability densities had to be the same for either class. This means that in that approach only $3d$ parameters have to be estimated (the mean μ for each expert and for either class and the common variance σ for each expert), or even as few as $d+1$ if the approach of the logistic regression model is used. In the case we are considering here, no independence hypotheses are made and $2d + d(d+1)/2 = d[2 + (d+1)/2]$ parameters need to be estimated (the d-dimensional mean μ for either class and the $d(d+1)/2$ different elements of the common *symmetric* covariance matrix Σ). The difference between these two approaches still increases with the number of experts d, as is shown in Table 6.9.

Table 6.9: Number of parameters to estimate as a function of the number of experts d.

d	Logistic regression	NB using exponentials	linear classifier
1	2	3	3
2	3	6	7
3	4	9	12
4	5	12	18
5	6	15	25

The results obtained using quadratic and linear classifiers are presented in Table 6.10, where the results for the two applications of the Logistic regression are also repeated for comparison purposes.

Table 6.10: Performance of quadratic and linear classifiers versus the Logistic regression approach.

Fusion module	FRR (%) (37 tests)	FAR (%) (1332 tests)	TER (%) (1369 tests)
Quadratic classifier (EER)	2.7 [0.5,13.8]	10.1 [8.6,11.8]	9.9 [[8.4,11.6]
NB using Gaussians (MAP)	5.4 [1.5,17.7]	0.0 [0.0,0.3]	0.1 [0.0,0.5]
NB using Gaussians (ML)	2.7 [0.5,13.8]	0.7 [0.4,1.3]	0.7 [0.4,1.3]
Linear classifier (EER)	2.7 [0.5,13.8]	16.9 [15.0,19.0]	16.5 [14.6,18.6]
Logistic regression (MAP)	2.7 [0.5,13.8]	0.0 [0.0,0.3]	0.1 [0.0,0.5]
Logistic regression (EER)	0.0 [0.0, 9.4]	1.1 [0.6,1.8]	1.1 [0.7,1.8]

The differences in performance observed through the results in Table 6.10 between the quadratic and the linear classifier on the one hand and their respective naive Bayes counterpart on the other hand can be explained by several reasons:

1. In the quadratic and linear classifiers, no independence assumptions are made. This results in an important increase in the number of parameters to be estimated as compared to the naive Bayes classifier, where the independence assumptions are made. Since there is no large training database available, the methods based on the naive Bayes

classifier that need only to estimate a small number of parameters do have better generalization performances.

2. The quadratic and linear classifiers have been determined taking into account all values, including the extreme ones. This leads to a bias in the positioning of the decision surfaces.

3. Another very important difference between the various classifiers is the value of the threshold. In the MAP and ML cases, the value of the threshold is defined a priori, whereas in the EER case, the threshold has been calculated on the training database to satisfy the EER criterion.

6.4 The Multi-Layer Perceptron (MLP)

6.4.1 A neural network representative

To show the possibilities of neural networks, we have been using the most typical representative of this family: the Multi-layer Perceptron (MLP). Under certain conditions, a statistical justification can be found for this method [23]. A MLP is a neural classifier that separates the training data of the two classes by implementing a separation surface which can have any arbitrary "flexible" shape [23]. The "flexibility" of the separating surface is determined by the complexity of the architecture. We used a classical MLP architecture with 3 neurons on the input layer (three scores coming from three experts), 5 neurons on the hidden layer and one neuron (two classes) on the output layer, sigmoidal activation functions for all neurons and the Backpropagation training algorithm. Using sigmoidal activation functions, the value of the output neuron lies in the interval [0,1], and the optimal decision threshold is fixed at 0.5. Figure 6.7 shows the MLP architecture we have used in our application.
We also tried to use the same MLP but with two output nodes (one for each class), but this architecture does not give such good results as the one using only a single output node and a decision threshold on 0.5. There are two fundamental reasons that can explain this difference between performance [23]. The first is that in the MLP with two output nodes there are more weights that need to be estimated than in the MLP with only one output node, and since we only have a small database, this leads to a less optimal estimation. The second reason is that in the case of the MLP with two output nodes, there is a complementarity constraint which links the

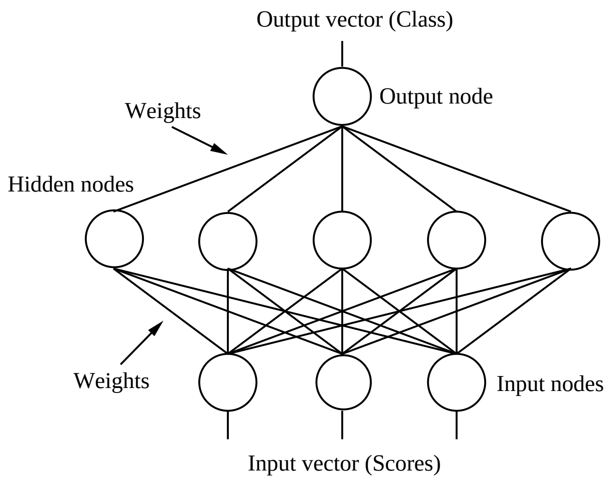

Figure 6.7: Example of the Multi-Layer Perceptron architecture we have used.

two outputs. This means that this constraint should be taken into account in the training algorithm. But in the standard software implementations of MLPs (such as the one available in Matlab's Neural Network Toolbox we used) this is usually not done, which again leads to sub-optimal performances of the MLP using two outputs as compared to the MLP with only one output.

6.4.2 Results

Table 6.11 shows the results obtained by using the method presented above.

Table 6.11: Performance of mono-modal multi-method fusion modules.

Fusion module	FRR (%) (37 tests)	FAR (%) (1332 tests)	TER (%) (1369 tests)
Multi-layer Perceptron	0.0 [0.0, 9.4]	0.4 [0.2,0.9]	0.4 [0.2,0.9]

6.4.3 Mixture of Experts

Another possible use of MLPs is in the framework of the so-called *Mixture of Experts* [87]. This method has not been tested on the M2VTS database, but we did investigate it for combining the outputs of segmental vocal experts [171]. This method is therefore not included in this chapter, but it is explained in section 9.4.4.

6.5 Comments

In theory [169], the normal usage for parametric statistical inference is when the investigator knows the problem to be analyzed rather well. He knows the physical laws that generate the stochastic properties of the data and the functions to be found up to a finite number of parameters. Estimating these parameters using the data is considered to be the essence of statistical inference. In practice however these parametric methods also have to be computationally feasible. Computational problems might occur in high-dimensional cases, due to the so-called "curse of dimensionality". We are not confronted with this problem, since we did opt for a decision fusion approach, which reduces the dimensionality of the problem. Furthermore, in our application we do not know the problem that well, so we had to limit ourselves to assume some of the most simple and most commonly used statistical distributions for estimating the underlying probability distributions. These favorite distributions are typically members of the exponential family.

When analyzing the results obtained by the parametric techniques we have experimented more closely, it is interesting to see that the best TER results are obtained by the naive Bayes classifier using the logistic regression model. This model assumes that the underlying conditional probability distributions are members of the exponential family (which is a very loose constraint), but with the same dispersion parameters for both classes (which is a very stringent constraint and we have indeed shown that in our application the different populations do not have the same dispersion parameters). On the other hand we also have tested the same naive Bayes classifier, but assuming this time that the underlying conditional probability distributions are Gaussians (and no other member of the exponential family), this time allowing for different dispersion parameters (i.e. variances) for the different populations. We also know that these assumptions are not satisfied, since the normality hypothesis is violated. The results obtained by this Gaussian naive Bayes classifier are not as good as those obtained by the naive Bayes

classifier using the logistic regression model. This suggests that at least in our application deviations from the "equality of dispersion parameters" assumption in the naive Bayes classifier using a logistic regression model are not as critical as deviations from the "Gaussian assumption" in the Gaussian naive Bayes classifier approach.
In any case, taking into account that for each method that we have experimented the assumptions we have made are not fulfilled, the results obtained by these parametric techniques in this application are surprisingly good. This is probably due to the fact that there is not a lot of data available, and therefore it is not possible to estimate a large number of parameters. This could explain the relative success of the methods requiring the estimation of only a small number of parameters, even if the assumptions regarding the underlying distributions are not completely fulfilled. This phenomenon can be seen as an application of Ljung's "pragmatic principle" observation that, in practice, the role of (model) identification is more often that of finding an *approximate* description, catching *some relevant* features, than that of determining the true, exact dynamics [108].

Chapter 7

Non-parametric methods

7.1 Introduction

The purpose of this chapter is to present a few non-parametric techniques, not to give an exhaustive overview of all existing methods. This chapter starts by presenting a very simple and popular family of non-parametric techniques. These *voting* techniques are sometimes referred to as k-out-of-n voting techniques, where k relates to the number of experts that have to decide that the person under test is a client, before the global voting classifier accepts the person under test as a client. After the voting methods, another simple but very popular technique, the k Nearest Neighbor (k-NN) technique, is presented with a number of variants. These variants include a distance weighted and a vector quantized version of the classical k-NN rule. This chapter ends by presenting the category of decision trees, by means of an implementation of the C4.5 algorithm, which is probably the most popular method in its kind.

7.2 Voting techniques

The first category of non-parametric or empirical techniques that we have studied is the class of the k-out-of-d *voting* techniques [46]. The global decision rule (obtained by fusing the *hard* decisions made by the d experts) implemented by this class is very simple. It says to accept the identity claim of the person under test if at least k-out-of-d experts decide that the person under test is indeed a client. These methods are really very simple, since they are only based on the actual hard *decisions*. They do not take into account the soft decisions or scores and they do not analyze the earlier behavior of the expert (by calculating for instance some statistical moments). For some values of k, particular decision fusion schemes are obtained:

1. $k = 1$. This is the so-called *OR* rule. The identity claim is accepted if at least one of the d experts decides that the person under test is a client. Intuitively this strategy leads to a very indulgent fusion scheme, which means that the acceptance will be fairly easy. This is good news for the clients since it means that the FRR will tend to be small, but bad news with respect to the protection against potential impostors since the FAR will tend to be higher.

2. $k = d$. This is the so-called *AND* rule. The identity claim is accepted only if all the d experts decide that the person under test is a client. Intuitively this strategy leads to a very severe fusion scheme, which means that the acceptance will be rather difficult. This is bad news for the clients since it means that the FRR will tend to increase, but good news with respect to the protection against potential impostors since the FAR will tend to be smaller.

3. $k = (d + 1)/2$ This is the so-called *MAJORITY* rule. It is obviously a compromise between the two previous rules.

Table 7.1 shows the results obtained by using the methods presented above using our three experts. The threshold settings for the different experts have been calculated for each individual expert on the training dataset using the EER criterion. There has been no attempt to optimize the threshold settings of the different experts in order to obtain the best performances using a specific voting scheme. This kind of optimization study has been undertaken in an exhaustive way in [135], but the generalization results on the test set were very disappointing.

Table 7.1: Performance of mono-modal multi-method fusion modules.

Voting scheme	FRR (%) (37 tests)	FAR (%) (1332 tests)	TER (%) (1369 tests)
OR	0.0 [0.0, 9.4]	7.4 [6.1,8.9]	7.2 [5.9,8.7]
AND	8.1 [2.8,21.3]	2.0 [1.4,2.9]	2.2 [1.6,3.1]
MAJORITY	0.0 [0.0, 9.4]	3.2 [2.4,4.3]	3.1 [2.3,4.2]

As has been announced, the OR rule leads to a small FRR, while the AND rule assures a small FAR. The compromise strategy of the MAJORITY rule offers in this particular case the best choice. When analyzing these results, one should bear in mind the extreme simplicity of these methods.

7.3 A classical k-NN classifier

The classical (also called *pooled*) k-NN classifier [56, 164] is a very simple classifier that needs no specific training phase. The only things needed are *reference* data points for both classes (clients, impostors). An unknown (test) data point y is then attributed the same class label as the label of the majority of its k nearest (reference) neighbors. To find these k nearest neighbors the Euclidean distance between the test point and all the reference points is calculated, the obtained distances are ranked in ascending order and the reference points corresponding to the k smallest Euclidean distances are taken. The exhaustive distance calculation step during the test phase leads rapidly to important computing times, which is the major drawback of this otherwise very simple algorithm. This computing time drawback can be solved for the testing phase by pre-calculating during the training phase a so-called *distance map*. This distance map is nothing else than a discrete version of the input space (all continuous expert score intervals are quantified) where with each point in this discrete space a class-label is attached using the chosen k-NN rule. An unlabeled test point can then labeled *immediately* by giving it the same label as that of the corresponding discrete point on the distance map. The statistical justification of this method can be found in [23]. A special case of the k-NN classifier can be obtained when $k = 1$. In this particular case, the classifier is called the Nearest Neighbor (NN) classifier.
Let k_1 and k_2 respectively be the number of client and impostor neighbors (such that k_1+k_2=k), the decision rule is then given by:

$$k_1 - k_2 \mathop{\gtrless}_{impostor}^{client} 0$$

Notice that there is no explicit notion of a continuous decision threshold for this type of classifier. The only threshold being k, but since this is a discrete number, it can not be used to trace a ROC. The results obtained with this fusion module are shown in Table 7.2 as a function of k.
Comparing these results with those obtained in Table 3.1 shows the benefits of combining several individual experts in a multi-modal system, even when

Table 7.2: Verification results for the k-NN classifier.

k	FRR (%) (37 tests)	FAR (%) (1332 tests)	TER (%) (1369 tests)
1	8.1 [2.8,21.3]	0.0 [0.0,0.3]	0.2 [0.1,0.6]
2	8.1 [2.8,21.3]	0.0 [0.0,0.3]	0.2 [0.1,0.6]
3	8.1 [2.8,21.3]	0.1 [0.0,0.5]	0.3 [0.1,0.8]
4	8.1 [2.8,21.3]	0.1 [0.0,0.5]	0.3 [0.1,0.8]

using such a simple fusion module as in this case. It can also be observed that in our typical application the number of neighbors k does not play an important role and the best results are obtained for $k = 1$, which is the NN-classifier. This can be explained by the fact that the different experts perform already well individually (i.e. the two classes are well separated: increasing the neighborhood then does indeed not improve performances). Another, more important, observation is that there is a great unbalance between the large FRR and the small FAR (a lot of clients are being rejected as impostors) [173]. A possible explanation for this undesired phenomenon is the fact that there are 1332 impostor but only 37 client reference points. This could indeed lead to the situation that for the "critical" *client* claims (i.e. the client test points which are lying close to the separation surface between the two classes), the impostor reference points in the chosen neighborhood outnumber the client reference points. If that would be the case, a possible solution to reduce the FRR could be to *weight* the importance of the different neighbors in the majority voting scheme.

7.4 A k-NN classifier using distance weighting

In a *distance weighted* k-NN classifier [164], the weighting of the k nearest neighbors is done as a function of their respective Euclidean distance to the test data point (the further a neighbor from the test data point, the smaller its weight). This could work if there is a relatively large "gap" between the two classes. In that case, a "critical" client access would have a small number of client reference points that are lying close and a large number of impostor reference points that are lying further away). The weighting function we have chosen here is represented in Figure 7.1. It is a sigmoidal function, which gives the relative weight of a neighbor as a function of its normalized distance (i.e. the Euclidean distance between the considered neighbor and the test point divided by the Euclidean distance between the $(k+1)$-th neighbor and the test point). Other weighting functions with the same general sigmoidal shape, but with a different position of the points of inflexion, have also been tested. Unfortunately the results obtained using this approach (whatever the chosen weighting function from the sigmoidal class) are exactly the same as the ones shown in Table 7.2. Since distance weighting does not seem to be able to counter the outnumbering problem, this means that there is not a big "gap" between the two classes in our application. Therefore we decided to use a more drastic approach by *reducing* the number of impostor reference points.

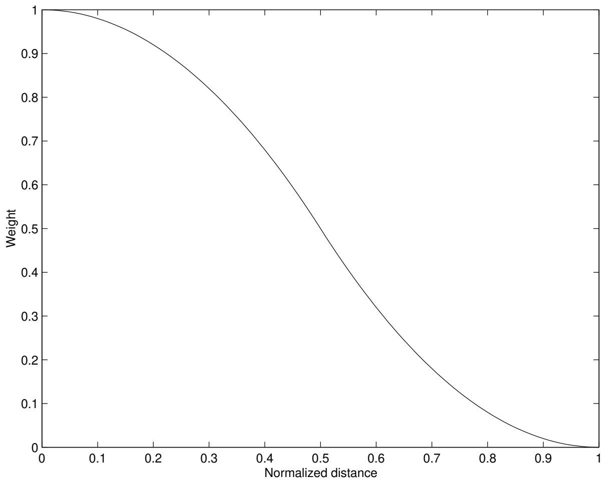

Figure 7.1: Typical distance weighting function.

7.5 A k-NN classifier using vector quantization

Obviously reducing the number of impostor data points is not without risk since this operation induces a loss of information. To try to minimize this loss, a clustering technique which replaces the actual impostor reference data points by a certain number of characteristic "codebook prototypes" has been chosen. The clustering is performed during an explicit training phase by the k-means algorithm [164], which has the advantage (compared to other classical clustering techniques) that it allows to fix *a priori* the number of prototypes P. This property allows to choose the ratio of the number of client and impostor prototypes. The k-means algorithm which has been implemented in this thesis, uses the Euclidean distance measure to divide the impostor reference data set into P clusters. Each cluster is then replaced by the *centroid* of its respective samples. The results obtained after this reduction operation are again the same for both previous discussed k-NN classifiers. They are shown in Table 7.3 for $k = 1$ (the value for which the best results have been obtained) as a function of P. The last line in this table (P=1332) corresponds to the classical k-NN classifier. This time, as opposed to the results obtained in the two previous sections, the FRR and the FAR are both very close to zero at the same time. It can also be observed that the FRR increases (and the FAR decreases) with P, as could be expected. It is interesting to observe the variation of the two errors

Table 7.3: Verification results as a function of P for a 1-NN classifier using vector quantization.

P	FRR (%) (37 tests)	FAR (%) (1332 tests)	TER (%) (1369 tests)
111	0.0 [0.0, 9.4]	0.5 [0.2,1.0]	0.5 [0.2,1.0]
333	2.7 [0.5,13.8]	0.4 [0.2,0.9]	0.5 [0.2,1.0]
666	2.7 [0.5,13.8]	0.4 [0.2,0.9]	0.5 [0.2,1.0]
1332	8.1 [2.8,21.3]	0.0 [0.0,0.3]	0.2 [0.1,0.6]

FRR and FAR as a function of P. At one extreme a FRR=0 is obtained for P=111, indicating that a small number of impostor reference points reduces the chances for a client to be rejected as an impostor. At the other extreme a FAR=0 is obtained for P=1332, indicating that using all available impostor reference points reduces the chances that an impostor is going to be accepted as a client. The optimal number of impostor prototypes P will depend on the cost-function as specified by the application. The main advantage for the vector quantization version, abstraction being made of the cost-function, is the fact that the number of distance calculations needed during the testing phase decreases rapidly with the reduction in the number of impostor reference points. Nevertheless the overall computing time still continues to be the major drawback for any kind of k-NN based classifier. Trying to reduce this amount of computing time has led us to use a decision tree based method. This is explained in the next section.

7.6 A decision tree based classifier

Decision tree learning is a supervised classification method in which the learned function (using the training data) is represented by a decision tree [121]. Learned trees can also be represented as sets of "if-then" rules to improve human readability. Decision trees classify unknown instances (*i.e.* test data) by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. Each node in the tree specifies a test of some *attribute* of the instance, and each branch descending from that node corresponds to one of the possible values for this attribute. In our specific case (use of a decision tree as a fusion module), these attributes are representing the scores of the instance obtained for each of the different experts. Figure 7.2 shows a typical binary decision tree, which is a specific kind of decision tree where each node has exactly two descending branches. This classifier is thus based on recursive parti-

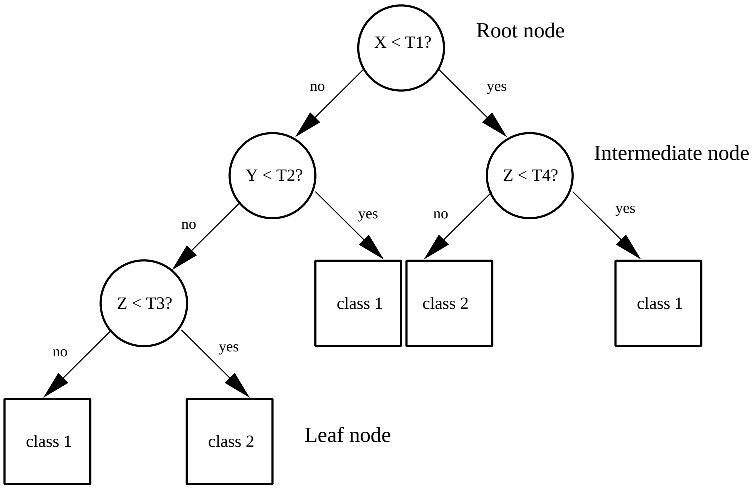

Figure 7.2: Typical binary decision tree.

tioning of the sample space. Space is divided into boxes, and at each stage in the procedure, each box is examined to see if it can be split into two boxes, the split being parallel to the coordinate axes. In our application, the C4.5 algorithm [143] has been chosen, which generates the decision tree

top-down, starting with the question "which attribute should be tested at the root of the tree?" To answer this question, each instance attribute is evaluated using a statistical test to determine how well it alone classifies the training examples. The best attribute is selected and used as the test at the root of the tree. A descendant of the root node is then created for each possible value of this attribute. In the C4.5 algorithm, the measure for determining the "best" attribute is the *information gain*. This measure is used to select the best attribute at each step in growing the tree and it is defined as the expected reduction in *entropy* caused by partitioning the examples according to this attribute.
Formally, the entropy $H(S)$ of a collection S containing both client and impostor examples is defined as:

$$H(S) = -p_c \log_2 p_c - p_i \log_2 p_i \tag{7.1}$$

where p_c is the proportion of client examples in S and p_i is the proportion of impostor examples in S.
The information gain $G(S, A)$ of an attribute A, relative to a collection of examples S, is then defined as:

$$G(S, A) = H(S) - H(S/A) \tag{7.2}$$

where the first term in the right hand side of the equation is the entropy of the original collection S and the second term is the expected value of the entropy after S is partitioned using attribute A. The expected entropy described by this second term is the sum of the entropies of each subset S_v, weighted by the fraction of examples that belong to S_v:

$$H(S/A) = \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H\left(S_V\right) \tag{7.3}$$

where $Values(A)$ is the set of all possible values for attribute A, and S_V is the subset of S for which attribute A has value V.
This implies that continuous-valued attributes need to be incorporated into the learning tree. This can be accomplished by dynamically defining new discrete-valued attributes that partition the continuous attribute value into a discrete set of intervals. In particular, for an attribute A that is continuous-valued, the algorithm can dynamically create a new boolean attribute A_c that is true if $A < c$ and false otherwise. The only question is how to select the best value for the threshold c. Clearly, one would like to pick a threshold c that produces the greatest information gain. By sorting

the examples according to the continuous attribute A, then identifying adjacent examples that differ in their classification, it is possible to generate a set of candidate thresholds midway between the corresponding values of A. It can be shown that the value of c that maximizes information gain must always lie at such a boundary [121]. These candidate thresholds can then be evaluated by computing the information gain associated with each.

In our specific application, the C4.5 algorithm generates a binary tree (which means that in each node a binary test is performed) using the training data. But before the thus generated tree for classifying unknown test points can actually be used, the tree needs to be pruned. This is an application of the well-known *Occam's razor* principle, which pleads for using the simplest hypothesis that fits the data to avoid *over-fitting* problems [23, 121]. Over-fitting reduces the generalization capability of a decision tree (it can be compared with *over-training* in the case of a neural network). After the generation of the decision tree using a *training* set, the pruning is implemented using a bottom-up strategy on a *validation* set. Starting at the leaf nodes, the tree is ascended and subtrees are removed in intermediate nodes according to a certain criterion. The intermediate node where the cut has been made becomes thus a leaf node. This new leaf node is assigned the most common classification of the *training* examples associated with it. The pruning criterion applied in the C4.5 algorithm is the *reduced-error pruning*, which stipulates that nodes are removed only if the resulting pruned tree performs no worse on the *validation* data than the original one.

Again there is no explicit presence of a decision threshold in this fusion module, which means that again only a single point of the ROC is found. The results obtained using the C4.5 algorithm are given in Table 7.4. As with the k-NN based classifier, the influence of reducing the number of impostor data points has been studied by using the k-means clustering algorithm.

The variation of FRR and FAR as a function of P shows very much the same behavior as in the case of the k-NN based classifier. The drawback of this decision tree based method is that for smaller values of P it is outperformed by the k-NN based classifier. The advantage of the decision tree with respect to the k-NN based classifier resides in the lower computing time.

Table 7.4: Verification results as a function of P for the C4.5 algorithm.

P	FRR (%) (37 tests)	FAR (%) (1332 tests)	TER (%) (1369 tests)
111	5.4 [1.5,17.7]	1.3 [0.8,2.1]	1.4 [0.9,2.2]
333	8.1 [2.8,21.3]	0.7 [0.4,1.3]	0.9 [0.5,1.6]
666	5.4 [1.5,17.7]	0.4 [0.2,0.9]	0.5 [0.2,1.0]
1332	8.1 [2.8,21.3]	0.3 [0.1,0.8]	0.5 [0.2,1.0]

7.7 Comments

In theory [169], the normal usage for non-parametric statistical inference is when one does not have reliable a priori information about the statistical law underlying the problem or about the function that one would like to approximate. In practice one might also opt for non-parametric methods if the parametric ones are not computationally feasible. The clear advantage of non-parametric methods is that one does not need to assume a certain distribution for the underlying probability distribution functions. This advantage is however at the same time their greatest drawback, since, to obtain similar performances, non-parametric methods tend to need more training data than parametric methods.

When analyzing the results obtained by the non-parametric techniques we have experimented more closely, it can be seen that the best TER results are obtained by the classical NN classifier and by the simple AND voting classifier. This is at least partially due to the typical application we are working with. Indeed, as already mentioned, these kind of applications always generate much more impostor than client examples. This means that methods which minimize FAR are always going to have a good TER (at least the way we have defined the TER). The AND method is typically such a technique, since it requires *all* experts to decide that the person under test is a client before the identity claim is accepted. The good TER results obtained in the case of the classical NN classifier are also easy to understand, since the number of impostor prototypes is so large that the probability of classifying an unknown and critical case as a client (which could lead to a FAR) is negligible. And, in the same context, this also explains why the TER results are getting worse when the number of impostor prototypes is reduced.

Chapter 8

Comparing the different methods

8.1 Introduction

This chapter starts with a general comparison between the different methods used. After that, the next section is devoted to the experimental comparison of classifiers, where we do not only present the results obtained during the test phase, but also during a subsequent validation phase. In the same section, we discuss the statistical significance of the differences observed between the presented methods. In the next section, we present a detailed comparison of the different methods. And finally, this section is followed by a visualization in a very simple bi-dimensional situation of the decision boundaries or the decision mechanisms implemented by most of the fusion modules.

8.2 Parametric versus non-parametric methods

As already mentioned in chapter 5, the optimal fusion modules are implemented by a Bayesian or a Neyman-Pearson test. These tests do however require the knowledge of the underlying probability distributions. Typically, parametric expressions for the probability distributions in a computationally convenient form are needed. Generally, parametric methods are preceded by a model verification step where the assumed distributions are tested, using for example, Kolmogorov-Smirnov goodness-of-fit type tests. If the verification tests are successful and if the parametric model is computationally convenient, then this parametric method should be used. But, even in the case where these verification tests are not successful, but where the deviations from the pre-supposed model are small, this method can be used to get at least a quick and approximated result. It is only in the case of computationally inconvenient forms or large deviations from a pre-supposed model, that parametric methods assuming specific distributions do not make sense. In that specific case, one can still try another type of parametric method, that does not pre-supposes any specific distribution: the Multi-Layer Perceptron. This kind of method can learn the underlying distributions, provided there are enough training data available. In our multi-modal context, this means that we would need huge multi-modal databases to be able to train correctly an MLP which learns the underlying multi-dimensional probability distributions. At this moment, such large multi-modal databases are not readily available. It is furthermore not always computationally attractive to try to train one very huge MLP. A possible way out of these very stringent requirements, is to assume that the different experts that are being used are class-conditionally independent. This assumption allows then to use one MLP per expert instead of one global MLP for all experts. Furthermore, this way the training of the different expert-based MLPs can be done using mono-modal databases, since this time the estimated probability distributions are mono-dimensional. In the case of computationally inconvenient forms or large deviations from a pre-supposed model, where either the independence hypothesis is not realistic and in which there are no large training databases available, or the global MLP is too complex, one can try to use so-called non-parametric methods.

8.3 Experimental comparison of classifiers

8.3.1 Test results

Table 8.1 gives a global view of the best verification results obtained on the test set with each classifier we have been using.
The first and most important observation we can make when looking at the results obtained by the fusion methods that we have experimented is that, in our application, fusion always improves the system performances beyond those of even the best single expert. The second observation is that these results seem to indicate that, generally speaking and again in our application, the class of the parametric methods does perform better than the class of non-parametric methods. Two indications that this statement is true in our case are that:

Table 8.1: Summary table of verification results on the test set of different fusion methods.

Method	FRR (%) (37 tests)	FAR (%) (1332 tests)	TER (%) (1369 tests)
ML	2.7 [0.5,13.8]	0.7 [0.4,1.3]	0.7 [0.4,1.3]
MAP	5.4 [1.5,17.7]	0.0 [0.0,0.3]	0.1 [0.0,0.5]
LR	2.7 [0.5,13.8]	0.0 [0.0,0.3]	0.1 [0.0,0.5]
QC	0.0 [0.0, 9.4]	2.4 [1.7,3.4]	2.3 [1.6,3.2]
LC	0.0 [0.0, 9.4]	3.1 [2.3,4.2]	3.0 [2.2,4.0]
MLP	0.0 [0.0, 9.4]	0.4 [0.2,0.9]	0.4 [0.2,0.9]
OR	0.0 [0.0, 9.4]	7.4 [6.1,8.9]	7.2 [5.9,8.7]
AND	8.1 [2.8,21.3]	0.0 [0.0,0.3]	0.2 [0.1,0.6]
MAJ	0.0 [0.0, 9.4]	3.2 [2.4,4.3]	3.1 [2.3,4.2]
k-NN	8.1 [2.8,21.3]	0.0 [0.0,0.3]	0.2 [0.1,0.6]
k-NN + VQ	0.0 [0.0, 9.4]	0.5 [0.2,1.0]	0.5 [0.2,1.0]
BDT	8.1 [2.8,21.3]	0.3 [0.1,0.8]	0.5 [0.2,1.0]

1. The mean TER calculated over the 6 parametric methods (1.10) is smaller than the mean TER calculated over the 6 non-parametric methods (1.95).

2. The mean rank ("1" being attributed to the "best" method (i.e. the method with the lowest TER), "2" to the "second best", and so on) calculated over the 6 parametric methods (5.83) is smaller than the mean rank calculated over the 6 non-parametric methods (7.17).

From a first, intuitive, analysis it would seem like we find here three groups of methods. A first group with TER values lying between 0.1 and 0.7, a second group with values between 2.3 and 3.1, and finally the "OR"-voting method with a TER result of 7.2. More specifically, the logistic regression method (a parametric method) gives the overall best TER results, and the "OR" voting scheme (a simple non-parametric method) gives the overall worst TER results. To verify the good results obtained with the logistic regression model, we did a *validation* test using this method.

8.3.2 Validation results

These validation results have been obtained on the same M2VTS database, but this time using a more sophisticated protocol: the so-called *leave-one-out* method [49]. In this case the M2VTS database has been split in two groups: group 1 consisting of 18 persons and group 2 containing 19 persons. These two groups have been used in turn respectively as training and testing data set for the fusion module, in such a way that if one group was used for training, the other one was used for testing. The purpose of this is split is to introduce a total separation between the training and the testing data sets. The fact that therefore not only the impostors, but also the clients are different in the training and the testing data sets, has as a direct consequence that the use of *individual* thresholds is not possible. For each group, client and impostor accesses are generated, rotating through the first four shots of the database. For group 1 this leads to 4×18=72 client and 4×18×17=1.224 impostor accesses. For group 2 the same method leads to 4×19=76 client and 4×19×18=1.368 impostor accesses. So this strategy produces in total 148 client and 2.592 impostor tests. This validation test protocol is visualized in Figure 8.1, and it is the same as the one described in [135].

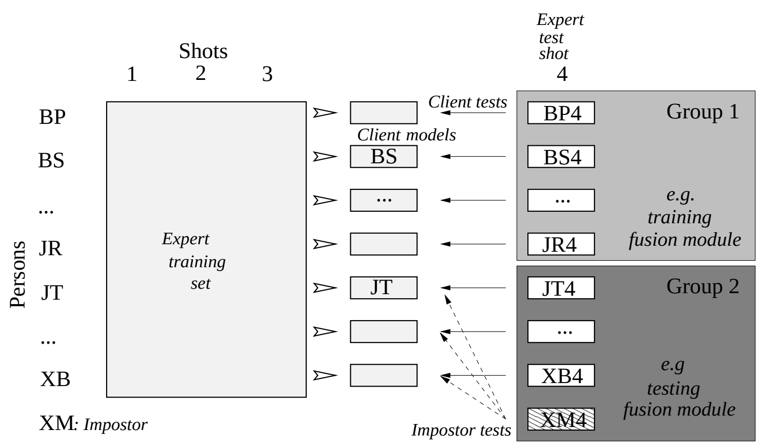

Figure 8.1: Visualization of the leave-one-out validation protocol.

The results obtained using this leave-one-out validation protocol are given in Table 8.2. This validation experiment shows indeed that the logistic regression does perform as well as predicted by the tests done using the original, more limited, test protocol. The verification performances obtained on the validation set extracted from this small database are extremely good, but when trying to generalize one should keep in mind the limitations of the described work.

Table 8.2: Validation of the logistic regression using a leave-one-out protocol on the M2VTS database.

Method	FRR (%) (148 tests)	FAR (%) (2.592 tests)	TER (%) (2.740 tests)
LR	0.0 [0.0,2.5]	0.0 [0.0,0.2]	0.0 [0.0,0.1]

8.3.3 Statistical significance

As can be observed in Table 8.1, the FAR, FRR and TER results from most of the methods used are lying very close to one another and they have confidence intervals which are overlapping each other in almost all cases. This means that, based on such FAR, FRR or TER results, it is not easy to decide without hesitation which method to use. That is why it would be very useful to have a systematic method which detects statistical significant differences between the FAR, FRR and/or TER results obtained by all the methods used.

In the case the scores obtained by the different fusion modules are independently drawn from normally distributed populations with the same variance, this problem can be solved by performing a basic analysis of variance (ANOVA), supplemented with so-called ad-hoc tests [123]. The ANOVA tells us *if* there are statistical significant differences between the different methods, and the ad-hoc tests (Least Significant difference (LSD), Duncan's Multiple Range Test and many others) tells us *where* the statistical significant differences exactly are. It is reminded here that the statistical comparison needs to be done between all the methods at the same time (as opposed to a series of pairs-wise comparisons), to avoid the in the statistical community well-known effect of dramatically increasing the type one error [114, 159].

Applying the ANOVA method in our case leads to a firm rejection of the hypothesis H_0, which means that there are significant differences between the presented fusion modules. To find out where these statistical significant differences are, we did use Duncan's Multiple Range Test. The result of this ad-hoc test with the highest power (*i.e.* the method with the lowest error of type II) is that three different groups of methods are significantly distinct. These groups are the following ones:

1. the first group (the one with the best performances) is formed by the following methods: LR, MAP, AND, k-NN, MLP, k-NN + VQ, BDT, and ML;
2. the second group consists of: QC, LC, and MAJ;
3. and finally the worst method in our case is the "OR"-vote.

However, since some of the fusion modules we use generate hard binary decisions as their output, we are, strictly speaking, not allowed to perform an ANOVA. Therefore we do have to use non-parametric methods. For the same reason as in the parametric case, it is here again absolutely necessary to compare all methods *at the same time.* In [51], five approximate statistical tests are presented for determining whether one learning algorithm out-performs another one on a particular learning task, but unfortunately these tests only allow a two-by-two comparison.

We believe that, depending on the discrete or continuous aspect of the output of the fusion module, two different non-parametric statistical tests can be applied to test the hypothesis H_0: all used fusion methods are of equal performance, against the alternative H_1: there are variations in performance. If one or more of the outputs of the fusion modules are *binary* or *hard* decisions, then *Cochran's Q test for binary responses* is suitable to solve this problem. If however the outputs of *all* fusion module are a continuous or *soft* decisions, then *Page's test for ordered alternatives* could be used. The latter test has the advantage that it has more *power* than the former [159, 161].

Since in this specific case there are several fusion modules with binary (in casu "0" for a reject and "1" for an acceptance) outputs, the only possibility is to use Cochran's Q test. In conventional terms of this test, the different fusion methods are called the *treatments* and the different access tests are called the *blocks.* If we have t treatments and b blocks with binary responses, the appropriate test statistic is:

$$Q = \frac{t\,(t-1)\sum_i T_i^2 - (t-1)\,N^2}{tN - \sum_j B_j^2}$$

where T_i is the total of 0's and 1's for treatment i, B_j is the total for block j and N is the grand total. The exact distribution of Q is difficult to obtain,

but for large samples Q has approximately a chi-squared distribution with $t-1$ degrees of freedom [161].
The application of Cochran's test for binary responses gives in our case a Q value which is much larger than the corresponding critical value of the corresponding chi-squared distribution, which leads us to reject the hypothesis H_0. This means that there are significant differences between the presented fusion methods, which is the same conclusion as the one obtained by performing the ANOVA. And although Cochran's test does not say where *exactly* these differences are, we did however establish one thing for sure: the best method (which in our case is the logistic regression) *is statistically significant better* than the worst method (which in our case is the "OR" voting scheme). This result is rather trivial, since for these two "extreme" methods the 95% confidence intervals do not overlap at all.
To conclude this section it can be seen that the results of the ANOVA and ad-hoc tests (although strictly speaking not allowed) do reinforce our first, intuitive approach. The allowed Cochran's Q test does confirm the results of the ANOVA, but unfortunately we did not find a non-parametric equivalent for the ad-hoc tests. Combining all this information leads us to the conclusion that there is no statistically justified evidence to prefer one specific method from the first (best performing) group above another one from the same group.

8.4 Visual interpretations

We have already pointed out that no single method is going to be optimal for all applications. A possible help for choosing the methods to investigate is the visual interpretation of the decision boundaries or mechanisms they implement. Indeed, if the different populations involved could be represented, then it would be possible to figure out which decision boundaries (and thus which methods) would fit best the application. This is obviously only possible in a simple two-dimensional scenario. In appendix F, some typical visual representations of decision boundaries and decision mechanisms of popular classifiers are presented.

8.5 Comments

The first and most important observation we can make when looking at the results obtained by the fusion methods that we have experimented is that, in our application, fusion always improves the system performances beyond

those of even the best single expert. The second observation is that these results seem to indicate that, generally speaking and again in our application, the class of the parametric methods does perform better than the class of non-parametric methods. More specifically, the logistic regression method (a parametric method) gives the overall best TER results, and the "OR" voting scheme (a simple non-parametric method) gives the overall worst TER results. Furthermore, the validation experiment which has been performed, has confirmed the good performances of the logistic regression model.
The strong preference we have developed in our case for the logistic regression is based on the following considerations:

1. logistic regression did obtain the least number of errors (one single error on 1369 access tests) on the M2VTS database;

2. logistic regression uses the *soft decision scores* of the different experts, which do contain more information than just the *binary hard decision*;

3. logistic regression is the parametric method that needs the smallest number of coefficients to be estimated. The fact that this is a good property can be justified by a combination of the "simplicity-favoring" idea of Occam's razor principle and the earlier mentioned "pragmatic principle" result obtained by Ljung.

In this chapter on comparing different fusion modules, we have added two-dimensional visualizations of the decision boundaries or decision mechanisms of most of the classifiers that we have discussed. This could be useful in order to determine which methods to select in a given application.

Chapter 9

Multi-level strategy

9.1 Introduction

In this chapter we present a multi-level decision fusion strategy that allows to gradually improve the performances of an automatic biometric identity verification system, while limiting the investments to the strictly necessary. This chapter is an improvement and an extension of what we have presented in [176, 178].

9.2 A multi-level decision fusion strategy

In the most general case, experts do have non-zero FA and/or FR errors. This is due to the fact that, generally speaking, the classes are not completely separated in the measurement space. Furthermore, both these errors are increased by the measurement noise. A solution to solve this problem, which is detailed in section 9.3, consists in combining the results obtained by one single expert over multiple instances obtained during multiple tests (temporal fusion). This way we can effectively reduce the variance of the results, which allows to improve the baseline system performances without having to modify the originally chosen approach. If these performances do not satisfy the end-user's needs, we can pass on to the next improvement. If we know the statistical distributions of the results obtained by each verification expert for either class (clients, impostors), then we can use the Bayesian decision rule which allows us to operate at the minimal error rate [56]. The classification error is going to be zero only in the case where the two distributions are not overlapping at all. We will only be able to reduce the two types of error rate (FAR, FRR) *at the same time* by

increasing the number of examples (training data), which also increases the *power* of the statistical test. In other words if we use more training data then we will be able to reduce the variances of the estimators of the real parameters of the distributions. Very often though we only have access to a limited set of training data, which generally means that using this approach, the possible improvements may be limited. Another way to reduce these two error rates is then to combine the results obtained by different experts using the same modality. This type of fusion allows to increase the system performance if the correlation of the errors that the different experts make is not equal to one. This improvement in performance will be bigger when the results (and the errors) obtained by the different experts are more de-correlated, since the information gain increases with the de-correlation. In section 9.4, we will present such a mono-modal multi-expert system based on the visual biometric modality. If system performance after both types of mono-modal fusion still does not meet the end-user's needs, it is possible to go on to the next step.
In a third step we can hope to improve the performances even more by finding characteristics which increase the separability of the hypotheses under test by increasing the dimensionality of the feature-space. To achieve this it is possible to use supplementary biometric modalities. The discrimination between the two distributions (clients, impostors) will be easier if the correlation between the different biometric modalities is small. We will study this case in section 9.5 for vocal and visual biometric modalities.

9.3 Mono-modal mono-expert fusion

9.3.1 Introduction

The first step in the proposed strategy is to use a *temporal integration* in order to improve the performances of an identity verification system by reducing the *variance on the measurements*. An application of this technique is presented in [100]. The authors suppose in this paper on the one hand that the decision to accept (ω_1) or to reject (ω_2) a person is based on the *a posteriori* probability of classes $P(\omega_j|x_i), j = 1, 2$ and that, on the other hand, they dispose of multiple instances x_i of biometric measures coming from the single modality. It is also supposed that the different measurements x_i have been acquired under the same circumstances and that they can be considered to be multiple measurements which are different only by their noise component. This way the *a posteriori* class conditional probabilities can also be considered as noisy estimates of the nominal value of this probability, *i.e.*:

$$P(\omega_j|x_i) = P(\omega_j|x) + \epsilon(\omega_j|x_i).$$

This way, better estimations $\hat{P}(\omega_j|x)$ of this *a posteriori* probability can be obtained by combining the noisy estimations either in a linear manner or by using rang order statistics. According to this study, the most interesting noise-reducing temporal fusion methods are the mean and the median rules, which are respectively given by the following two equations :

$$\hat{P}(\omega_j|x) = \frac{1}{R}\sum_{i=1}^{R} P(\omega_j|x_i).$$

$$\hat{P}(\omega_j|x) = \mathrm{med}_{i=1}^{R} P(\omega_j|x_i).$$

9.3.2 Results

We have applied the principles exposed in the referenced paper [100] to the profile expert [138]. For these experiments the expert uses only the image part of the multi-modal M2VTS database. As can be observed, Table 9.1 shows indeed a reduction in the verification error when the number of instances N increases. The performance gain is very spectacular at the beginning of a sequence but levels of after the integration of the first instances.

Table 9.1: Verification error as a function of the number of instances N.

Method	N	FRR (%) (37 tests)	FAR (%) (1332 tests)	TER (%) (1369 tests)
Mean	1	18.9 [9.5,34.2]	15.6 [13.8,17.7]	15.7 [13.9,17.7]
Mean	2	8.1 [2.8,21.3]	11.8 [10.2,13.6]	11.7 [10.1,13.5]
Mean	3	8.1 [2.8,21.3]	9.5 [8.1,11.2]	9.5 [8.0,11.2]
Median	1	18.9 [9.5,34.2]	15.6 [13.8,17.7]	15.7 [13.9,17.7]
Median	2	13.5 [5.9,28.0]	10.4 [8.9,12.2]	10.5 [9.0,12.3]
Median	3	8.1 [2.8,21.3]	11.9 [10.3,13.8]	11.8 [10.2,13.7]

9.4 Mono-modal multi-expert fusion

9.4.1 Introduction

The combination of multiple experts using the same (biometric) modality is a widely used concept. We can cite for instance [8, 29, 58, 67, 90, 139, 160, 187].

The two experts we have used here are the previous *profile* image expert [138] which we have combined with the *frontal* image expert [116]. Although these two experts examine the same generic biometric modality (visual appearance), one can intuitively feel that they will probably be at least *partially* de-correlated. In fact an indication of this de-correlation can be given by the mean correlation coefficient between the scores of the profile and the frontal image experts for a same person (note that these are *class-conditional* scores). This mean value in our application is typically less than 25 %, which underpins our intuitive feeling. A graphical representation can be found in Figures 9.1 and 9.2, where a typical example of the class-conditional scores is plotted for both experts and for both classes.

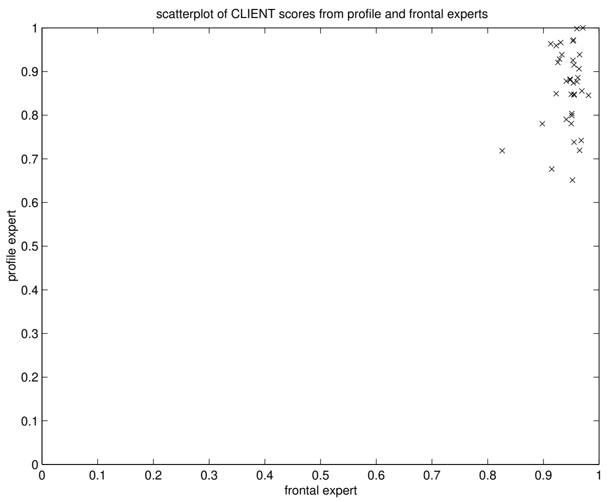

Figure 9.1: Typical scatter-plot of the *client* scores of the profile expert as a function of those of the frontal expert.

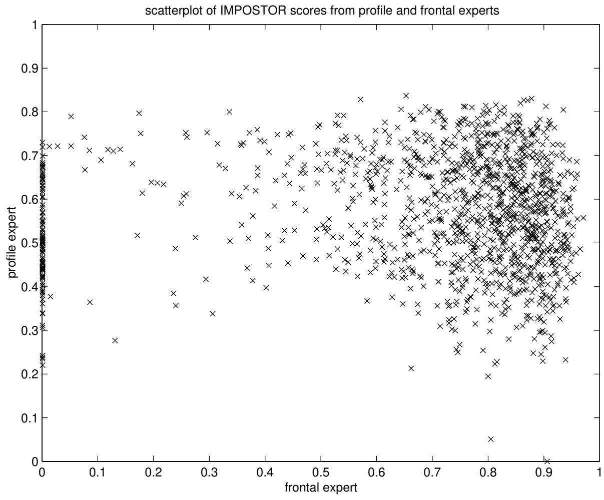

Figure 9.2: Typical scatter-plot of the *impostor* scores of the profile expert as a function of those of the frontal expert.

9.4.2 Methods

To show the possibilities of this mono-modal multi-expert fusion, we have been using five typical simple decision fusion modules such as: voting methods (AND, OR), linear (LC) and quadratic (QC) classifiers and a Multilayer Perceptron (MLP). Under certain conditions, a statistical justification can be found for all these methods [23, 46, 56, 61, 164].

QC This classifier separates the training data of the two classes by implementing a quadratic separation surface (see 6.3.7).

LC This classifier separates the training data of the two classes by implementing a linear separation surface (see 6.3.7).

MLP This classifier separates the training data of the two classes by implementing a separation surface which can have any arbitrary "flexible" shape (see 6.4).

OR To accept a person under test as a client, it is sufficient that *one* of the experts accepts that person as a client (see 7.2).

AND To accept a person under test as a client, it is necessary that *all* of the experts accept that person as a client (see 7.2).

9.4.3 Results

Table 9.2 shows the results obtained by using typical simple decision fusion modules such as: voting methods (AND, OR), linear (LC) and quadratic (QC) classifiers and a Multi-layer Perceptron (MLP).

Table 9.2: Performance of mono-modal multi-expert fusion modules.

Fusion module	FRR (%) (37 tests)	FAR (%) (1332 tests)	TER (%) (1369 tests)
QC	2.7 [0.5,13.8]	10.1 [8.6,11.8]	9.9 [[8.4,11.6]
LC	2.7 [0.5,13.8]	16.9 [15.0,19.0]	16.5 [14.6,18.6]
MLP	5.4 [1.5,17.7]	3.2 [2.4, 4.3]	3.3 [2.5, 4.4]
OR	0.0 [0.0, 9.4]	40.4 [37.8,43.1]	39.3 [36.7,41.9]
AND	8.1 [2.8,21.3]	2.0 [1.4, 2.9]	2.2 [1.6, 3.1]

By comparing the results obtained on the M2VTS database and presented in Table 9.2 with those of Table 9.1, it can be seen that this example of a mono-modal multi-expert data fusion effectively allows to improve the system performances beyond those of the previous lower level (mono-modal mono-expert) approach.

9.4.4 Combining the outputs of segmental vocal experts

Another example of mono-modal multi-expert data fusion is presented in [171]. In this case, *segmental* approaches to text-independent speaker verification are tested on a subset of the NIST-NSA'98 speaker evaluation database. Unlike the schemes based on Large Vocabulary Continuous Speech Recognition (LVCSR) with previously trained phone models, the systems used in this approach are based on units derived in an unsupervised manner using the ALISP (Automatic Language Independent Processing) tools [42, 134]. The speech segmentation is achieved using a *temporal decomposition* (TD) [4, 17] followed by *unsupervised clustering.* Among several available algorithms for performing this clustering (Ergodic HMM, self-organizing map, etc.), *Vector Quantization* (VQ) was chosen for its simplicity. The VQ codebook is trained by a k-means algorithm with binary splitting [68]. TD and VQ provide a symbolic transcription of the data in an unsupervised way. Each vector of the acoustic sequence is declared as a member of one class (which is a *hard* decision), determined

through the segmentation and the labeling. The number of classes is fixed by the number of centroids in the VQ codebook. In our experimental work, 8 classes have been used. Speaker modeling is then done independently for each class of speech sounds using the mixture of Gaussian distributions (GMM) [150, 151]. The next step to be performed is the fusion of the scores generated by these different models.
Among the techniques to merge the scores of the 8 class-dependent scores, logistic regression and a method based on the Mixture of Experts technique [87], were investigated.
The results obtained by the logistic regression are presented in Figure 9.3 for the training data and in Figure 9.4 for the test data.

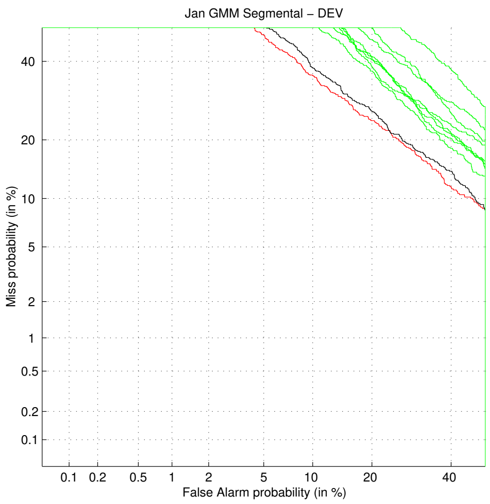

Figure 9.3: Comparison of the results of the 8 different classes (in green), the mean of these classes (in black) and the logistic regression (in red), obtained on the training data.

The underlying idea for the approach based on the paradigm of the mixture of experts is that we are not sure about the quality of the original classification of the different segments. Indeed, a given segment has been attributed the label of the nearest class prototype by a *hard* decision, but we did not use the information hidden in the distances towards the 7 other class prototypes. This leads us to think that it might be interesting to feed *all* segments to *all* segmental experts in parallel (which still have been trained on their respective labeled segments as before), whatever the label

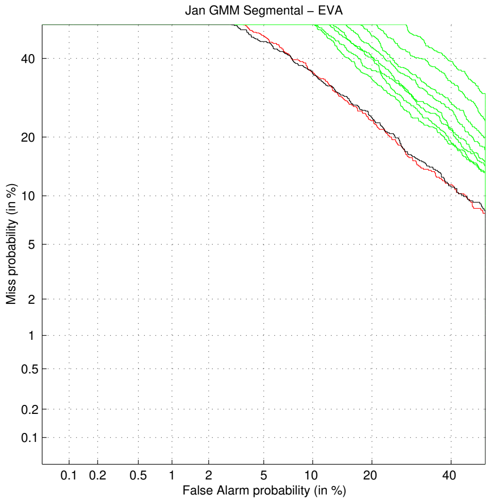

Figure 9.4: Comparison of the results of the 8 different classes (in green), the mean of these classes (in black) and the logistic regression (in red), obtained on the test data.

of that particular segment might be. To weight the likelihood ratio outputs LLR_i of each of the segmental experts, we add a MLP, which will serve as *gating network*. This gating network receives the same acoustic vectors as input as the segmental experts, has 20 hidden neurons with *tanh* activation functions, and has eight output neurons with *softmax* [23] activation functions. This softmax function assures that the outputs W_i of the gating network sum to unity and are non-negative, thus implementing the (soft) competition between the different segmental experts [122].
Formally, the i-th output W_i of the gating network is calculated as follows:

$$W_i = \frac{\exp(z_i)}{\sum_{j=1}^{8} \exp(z_j)}, \tag{9.1}$$

where the z_j are the gating network outputs before thresholding.
These 8 different output values W_i are then used to weight the 8 outputs LLR_i of the 8 segmental experts in the following manner:

$$\text{Total LLR} = \sum_{i=1}^{8} W_i * LLR_i$$

The gating network is trained using speech segments from the claimed speaker. For these speech segments, the target vector is 1 for the output neuron corresponding with the largest LLR_i, and 0 for the 7 other outputs. During the test phase, the 8 output neurons of the gating network are going to vary with the presented input segment. This means that if an input segment is lying close to k class segmentation prototypes, this will be translated by the fact that k different output neurons will tend to have significant outputs. In this manner, k segmental experts will significantly and proportionally contribute to the total LLR.
The structure of this data fusion module is represented in Figure 9.5.

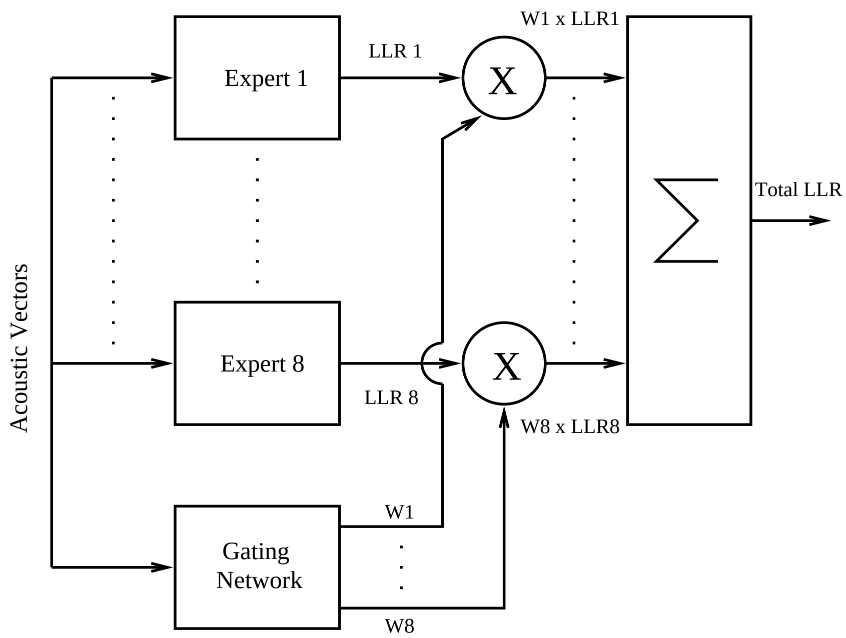

Figure 9.5: Combining the outputs of the different segmental experts.

The testing of this mixture of experts method is actually in progress.

9.4.5 Combining the outputs of global vocal experts

In the same context of mono-modal multi-expert fusion, the logistic regression model has been experimented on the data of the NIST-NSA'99 speaker verification evaluation campaign. This data set contains 1.311 male client accesses, 14.617 male impostor accesses, 1.846 female client accesses, and 19.846 female impostor accesses. In this context we have fused in total 12 vocal experts, 6 from the ELISA consortium (ENST, IDIAP, IRISA, LIA, RIMO, VERE) and 6 from other participants (A2RT, Dragon, Ensigma, IITM, MIT, OGI). The training of the logistic regression model has been done using the male data (15.928 accesses) and the testing has been performed on the female data (21.692 accesses). The EER results before and after fusion are shown in Table 9.3.

Table 9.3: Comparison of the EER performances of the best single expert versus those of a fusion module based on the logistic regression model (LR).

Experts	Best single expert EER (%) (21.692 tests)	LR Fusion EER (%) (21.692 tests)
ELISA (6)	12.0 [11.6,12.4]	12.0 [11.6,12.4]
Others (6)	10.0 [9.6,10.4]	8.0 [7.6, 8.4]
All (12)	10.0 [9.6,10.4]	7.0 [6.7, 7.3]

When looking at the results of the 6 submissions coming from the ELISA consortium, it can be seen that the EER for the best single expert is the same as the EER obtained after fusing the 6 experts.
This phenomenon can be explained by the following observations:

1. The 6 vocal experts coming from the ELISA consortium are very correlated, since most of them are directly derived from the same baseline GMM-based method. This implies that the de-correlation of the errors of these 6 experts will be rather small, so the potential gain of the fusion process (i.e. the distance between the DET-curves before and after fusion) is also likely to be small.

2. The coefficients of the logistic regression are optimized for the operating point specified by the minimization of the cost function C_{DET}, as defined by NIST[1]. This operating point is not the EER, but a point which is lying far more to the left, because of the extreme unbalance between the two weighting factors. This means that the DET-curve after fusion will be optimized in the neighborhood of the NIST operating point and not around the EER.

[1] The NIST cost function C_{DET} is a weighted sum of the FAR and FRR error rates: $C_{DET} = 0.1 \times \text{FRR} + 0.99 \times \text{FAR}$ [142].

The observation that the DET-curves representing the different systems used in the ELISA consortium are lying close together, combined with the observation that the coefficients of the logistic regression are optimized around the NIST operating point can indeed account for the fact that the EER after fusion is not better than the EER of the best single expert. This phenomenon can be observed in Figure 9.6, where the green curves represent the results of the 6 ELISA experts and the red curve shows the results obtained by the logistic regression fusion module. The NIST operating point for each DET-curve is shown by the respective black plus-sign.

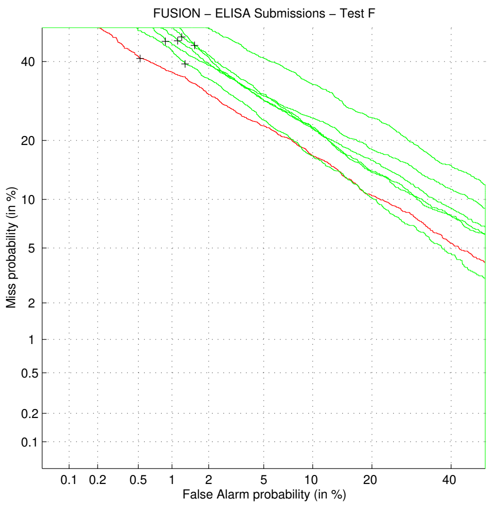

Figure 9.6: DET-curves presenting the results of the 6 ELISA experts (in green), and of the logistic regression fusion module (in red).

The DET-curves obtained in the testing phase using all 12 vocal experts are presented in Figure 9.7. The green curves represent the results of the 6 ELISA experts, the blue curves represent the results of the 6 other experts and the red curve shows the results obtained by the logistic regression fusion module. Observing these results it can be seen that fusing the results of 12 vocal experts using a logistic regression model improves the global system performances beyond those of the best single expert.

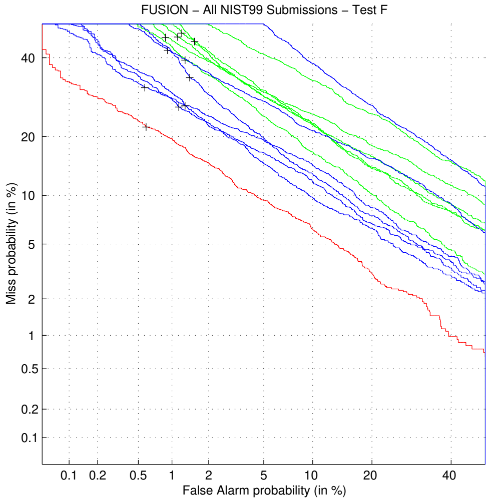

Figure 9.7: DET-curves presenting the results of the 6 ELISA experts (in green), of the 6 other experts (in blue), and of the logistic regression fusion module (in red).

9.5 Multi-modal multi-expert fusion

The third step in the proposed development strategy is to fuse multiple experts which are based on multiple (biometric) modalities. This approach has, for instance, been used in the following references [33, 40, 50, 54, 92, 100].

This step is in fact nothing else than what has been presented in this work in the first eight chapters. For the ease of the reader, Table 9.4 repeats the best verification results obtained on the test set with each classifier we have been using in this work.

By comparing the results obtained on the M2VTS database and presented in Table 9.4 with those of Table 9.2, it can be seen again that this example of a multi-modal multi-expert data fusion effectively allows to improve the system performances beyond those of the previous lower level (mono-modal multi-expert) approach.

Table 9.4: Summary table of verification results on the test set of different fusion methods.

Method	FRR (%) (37 tests)	FAR (%) (1332 tests)	TER (%) (1369 tests)
ML	2.7 [0.5,13.8]	0.7 [0.4,1.3]	0.7 [0.4,1.3]
MAP	5.4 [1.5,17.7]	0.0 [0.0,0.3]	0.1 [0.0,0.5]
LR	2.7 [0.5,13.8]	0.0 [0.0,0.3]	0.1 [0.0,0.5]
QC	0.0 [0.0, 9.4]	2.4 [1.7,3.4]	2.3 [1.6,3.2]
LC	0.0 [0.0, 9.4]	3.1 [2.3,4.2]	3.0 [2.2,4.0]
MLP	0.0 [0.0, 9.4]	0.4 [0.2,0.9]	0.4 [0.2,0.9]
OR	0.0 [0.0, 9.4]	7.4 [6.1,8.9]	7.2 [5.9,8.7]
AND	8.1 [2.8,21.3]	0.0 [0.0,0.3]	0.2 [0.1,0.6]
MAJ	0.0 [0.0, 9.4]	3.2 [2.4,4.3]	3.1 [2.3,4.2]
k-NN	8.1 [2.8,21.3]	0.0 [0.0,0.3]	0.2 [0.1,0.6]
k-NN + VQ	0.0 [0.0, 9.4]	0.5 [0.2,1.0]	0.5 [0.2,1.0]
BDT	8.1 [2.8,21.3]	0.3 [0.1,0.8]	0.5 [0.2,1.0]

9.6 Comments

From the verification results in Table 9.4, it can be seen that this example of multi-modal multi-expert fusion effectively allows to improve the system performances beyond those of the two previously presented lower level (mono-modal mono-expert and mono-modal multi-expert) approaches. This shows that on the M2VTS database and using our three experts, it is possible to gradually improve performances of biometric identity verification systems, by increasing the number of experts and, eventually, the number of modalities. Data fusion can thus be used in an identity verification system at different levels.

The first level is the temporal fusion of results obtained by a single expert (using only one biometric modality) to reduce the measurement variance.

A second level can be reached when combining the results obtained by different experts, working on the same biometric modality, to minimize the classification errors by relying on the de-correlation of the errors made by the different experts. In this context we did not only combine successfully the (highly de-correlated) profile and frontal face expert using the M2VTS database. Indeed, we also tested logistic regression on the combination of highly correlated segmental as well as global vocal experts using the NIST-NSA databases. And, although performance improvements are obviously not as spectacular as in the de-correlated case, we also did obtain significant performance improvements if we compare with to those of the best single expert.

Finally, the third level of application is to reduce the classification errors even more by trying to increase the separation between the distributions of the two populations by increasing the dimension of the space using different biometric modalities, which should be as de-correlated as possible.

This global strategy has the great advantage to *gradually* improve the performances of an identity verification system.

Chapter 10

Conclusions and future work

10.1 Conclusions

The objective of this thesis is to contribute to the multi-modal identity verification by the use of data fusion techniques. Three different experts derived from two biometric modalities (vocal and visual information) were used in this study. This choice was mainly dictated by the complementarity (behavioral versus physiological) of these two modalities and by the fact that these three experts are very easy to integrate in a low-cost platform such as a PC equipped with a microphone and a CCD camera. The outputs of the three experts, which we have called scores, are then presented in parallel to a fusion module. This fusion module implements either a parametric or a non-parametric classification method, and takes the final decision with respect to the acceptance or the rejection of the identity claim of the person under test.

After an introduction (chapter 1) stating the goal and the structure of this thesis, the remainder of this work has been divided in two parts.

In the first part general issues related to automatic multi-modal identity verification systems were presented in three chapters. In chapter 2 we have seen that each biometric technology has its strengths and limitations, and no single biometric is expected to effectively meet the needs of all applications. We have also seen that voice is one of the most popular behavioral biometrics, thanks to its high acceptability and its user-friendliness. And since in a multi-modal approach it is wise to complement a behavioral modality with a physiological one, we have chosen to add the visual modality. In chapter 3 we have analyzed the performances of three biometric experts (frontal, profile and voice) using the proposed protocol on the

M2VTS multi-modal database. Furthermore, the behavior of these experts has been statistically analyzed. This analysis did lead to the following observations: the normality hypotheses for underlying probability distributions for the different populations involved, are not satisfied, the three experts do show good discriminatory power, the variances of the scores of the different populations are not the same, the three experts are complimentary, and there is evidence that combining the three experts improves the performances onto a level that is better than those of the best expert. In chapter 4, we have presented different aspects of data fusion techniques. For all these aspects, we made motivated choices to come to the data fusion solution which suits best our application. The result of these choices was that we decided to implement a parallel decision fusion strategy as a particular classification problem, which gives us the enormous advantage of being to reuse directly the methods of the Pattern Recognition field.

In the second part of this thesis, the attention has been shifted towards the combination of the different experts. The first chapter in this part (chapter 5), justifies the analysis of both parametric and non-parametric methods as possible classification methods. In chapter 6, we present the parametric approach. In theory, the normal usage for parametric statistical inference is when the investigator knows the problem to be analyzed rather well. He knows the physical laws that generate the stochastic properties of the data and the functions to be found up to a finite number of parameters. Estimating these parameters using the data is considered to be the essence of statistical inference. In our application we do not know the problem that well, so we had to limit ourselves to assume some of the most simple and most commonly used statistical distributions for estimating the underlying probability distributions. These favorite distributions are typically members of the exponential family. The experiments presented in this chapter, show that in our application the best TER results are obtained by the logistic regression model. This model assumes that the underlying conditional probability distributions are members of the exponential family (which is a very loose constraint), but with the same dispersion parameters for both classes (which is a very stringent constraint and we have indeed shown that in our application the different populations do not have the same dispersion parameters). On the other hand we also have tested the naive Bayes classifier, assuming that the underlying conditional probability distributions are Gaussians, this time allowing for different dispersion parameters for the different populations. We also know that these assumptions are not satisfied, since the normality hypothesis is violated. The results obtained by this naive Bayes classifier are not as good as those obtained by the logistic regression model. This suggests that at least in our application deviations from the "equality of dispersion parameters" assumption in the logistic regression model are not as critical as deviations from the "Gaussian assumption" in the naive Bayes classifier approach. In any case, taking into account that for each method that we have experimented the assumptions we have made are not fulfilled, the results obtained by these parametric techniques in this application are surprisingly good. This is probably due to the fact that there is not a lot of data available, and therefore it is not possible to estimate a large number of parameters. This could explain the relative success of the methods requiring the estimation of only a small number of parameters, even if the assumptions regarding the underlying distributions are not completely fulfilled. This phenomenon can be seen as an application of Ljung's observation that, in practice, the role of (model) identification is more often that of finding an *approximate* description, catching *some relevant* features, than that of determining the true, exact dynamics [108]. In chapter 7 we have presented the non-parametric approach. In theory, the normal usage for non-parametric statistical inference is when one does not have reliable a priori information about the statistical law underlying the problem or about the function that one would like to approximate. The clear advantage of non-parametric methods is that one does not need to assume a certain distribution for the underlying probability distribution functions. This advantage is however at the same time their greatest drawback, since non-parametric methods tend to need more training (learning) data than parametric methods. When analyzing the results obtained by the non-parametric techniques we have experimented more closely, it can be seen that the best TER results are obtained by the classical 1-NN classifier and by the simple "N-out-of-N" voting classifier (AND). This is at least partially due to the typical application we are working with, since they always generate much more impostor than client examples. This means that methods which minimize FAR are always going to have a good TER (at least the way we have defined the TER). This also explains why the TER results are getting worse when the number of impostor prototypes is reduced. In chapter 8 we present a comparison between the results obtained by the parametric and the non-parametric methods.The first and most important observation we can make when looking at the results obtained by the fusion methods that we have experimented is that, in our application, fusion always improves the system performances beyond those of even the best single expert. The second observation is that these results seem to in-

dicate that, generally speaking and again in our application, the class of the parametric methods does perform better than the class of non-parametric methods. More specifically, the logistic regression method (a parametric method) gives the best overall best TER results, and the "OR" voting scheme (a simple non-parametric method) gives the overall worst TER results. Cochran's non-parametric Q test for binary responses has then been used to show that this result is statistically significant. Furthermore, the validation experiment which has been performed, has confirmed the good performances of the logistic regression model.
The strong preference we have developed in our case for the logistic regression is based on the following considerations:

1. logistic regression did obtain the least number of errors (one single error on 1369 access tests) on the M2VTS database;

2. logistic regression uses the *soft decision scores* of the different experts, which do contain more information than just the *binary hard decision*;

3. logistic regression is the parametric method that needs the smallest number of coefficients to be estimated. The fact that this is a good property can be justified by a combination of the "simplicity-favoring" idea of Occam's razor principle and the earlier mentioned "pragmatic principle" result obtained by Ljung.

In this chapter on comparing different fusion modules, we have added two-dimensional visualizations of the decision boundaries or decision mechanisms of most of the classifiers that we have discussed. This could be useful in order to determine which methods to select in a given application.
In chapter 9 we have presented a multi-level decision fusion strategy that allows to gradually improve the performances of an automatic biometric identity verification system, while limiting the investments to the strictly necessary. It has indeed been shown that on the M2VTS database and using our three experts, multi-modal data fusion effectively allows to improve the system performances beyond those of the mono-modal mono-method and mono-modal multi-method approaches. This strategy can then in theory be continued until the system performances meet the user requirements.
In the same chapter and using the highly correlated data coming from the experts which participated in the NIST-NSA'99 speaker verification evaluation campaign, it has been shown that, in the context of mono-modal multi-method fusion, the logistic regression fusion module outperforms even the best single expert.

With this thesis it has been shown that the problem of automatically verifying the identity of a person can be solved for a given application. To do so, one must use one or more biometric modalities, use expert knowledge to develop verification algorithms that use features derived from these biometrics as input and give scores as outputs, and, most importantly, one must combine the outputs of these different experts using a robust fusion module. In our application, using the M2VTS database and the three experts presented in this work, the best performing fusion module was based on the logistic regression model. The verification performances obtained on the validation set extracted from this small database were extremely good, but when trying to generalize one should keep in mind the limitations introduced in this work.

10.2 Future work

A first interesting research topic that was started but needs to be continued, is the test involving the gating network in the framework of the mono-modal multi-method fusion.

Another point which would be interesting to explore is the development of a non-parametric version of the so-called ad-hoc tests.

Furthermore, it would be interesting to compare the results of the multi-linear classifier (which is in fact an example of a classifier minimizing the *empirical risk*), with those of a classifier based on Support Vector Machines (SVMs). These SVMs have been introduced by Vapnik to improve the generalization capabilities, by minimizing the so-called *structural risk* [168, 169].

Instead of the binary {accept, reject} decision, it is possible to use a scheme in which a third decision, which could be called *undecided* could be added. This third decision could be used to define a set of margins around the decision threshold. In a Bayesian approach, where the goal is to minimize the risk, the straightforward way of implementing this {undecided} option, is to define a cost related to the fact that no decision is taken and to minimize the total cost in the same way as the one which is explained in section 6.3.1.

Obviously if the {undecided} option is used, something has to be done in the event of the outcome of a non-decision. Several approaches could be used, all linked with a particular cost which could be taken into account in the Bayes risk as mentioned above. A first possibility is to call a human operator to solve the {accept,reject} problem. This is clearly a possibility, but since we try to realize an automated biometric identity verification system, it is perhaps not the most desirable one. Another possibility is to implement the identity verification in a sequential way. Indeed, in the case of a non-decision pronounced by the first stage system that we have described until now, a second stage system could be used. This second stage system, that would only act in the case of a non-decision, could be based on the same biometric modalities as the first one (with respect to the vocal expert, the automated system could for instance ask the person under test to pronounce more sentences). It could also be based on other, slower or less popular but more performing biometric modalities, such as fingerprint analysis.

A possible improvement with respect to the system presented in this work, is the use of *personal characteristics*. This idea is based on the fact that the human recognition capabilities can also be guided by particular features that are very typical for one specific person (the extremely big eyes, the very pronounced chin, the oversized nose or the large ears of a certain person). This is an interesting observation, especially when seen in the light of the actual efforts to come to *robust* methods, in which extreme values, such as the ones we have been considering here, are very likely to be excluded! Using this kind of person-specific a priori knowledge will probably allow an increase in automated biometric identity verification systems. The only drawback of this idea is that enough training data has to be available *for each person*.

Another interesting future research topic is the integration of multi-modal biometrics on a so-called *smart-card* [148].

Last but not least, the association of a confidence measure with each score a certain expert is giving, could lead to an improvement in performances by using methods such as fuzzy logic and Dempster-Shafer evidence theory [26, 113, 156]. These forms of *uncertainty management* could also be used to improve the proposed multi-level strategy by formally integrating uncertainty in the transition strategy between the different hierarchical levels.

Bibliography

[1] ACTS. "M2VTS: Multi-modal verification for tele-services and security applications ". http://www.uk.infowin.org/ACTS/RUS/PROJECTS/ac102.htm, 1995.

[2] R. T. Antony. *Principles of Data Fusion Automation*. Artech House Publishing, 1995.

[3] B. S. Atal. "Automatic recognition of speakers from their voice". *Proceedings of the IEEE*, Vol. 64, no. 4, pp. 460–475, April 1976.

[4] B. S. Atal. "Efficient coding of LPC parameters by temporal decomposition". In *Proc. IEEE ICASSP 83*, pages 81–84, 1983.

[5] V. Barnett. *Comparative statistical inference*. John Wiley & Sons, 1973.

[6] M. D. Bedworth. "Less Certain, More Infallible". In *Proceedings of the International Conference on Multisource-Multisensor Information Fusion*, volume 2, pages 572–579, Las Vegas, USA, July 1998.

[7] S. Ben-Yacoub. "Multi-Modal Data Fusion for Person Authentication using SVM". IDIAP-RR 7, IDIAP, 1998.

[8] Y. Bennani. "Text-independent talker identification system combining connectionist and conventional models". In S. Y. Kung et al., editor, *Neural Networks for Signal Processing, Vol.2*. IEEE Service Center Press, 1992.

[9] Y. Bennani. "A modular and hybrid connectionist system for speaker identification". *Neural Computation*, Vol. 7, pp. 791–798, 1995.

[10] L. Besacier. *Un Modèle Parallèle pour la Reconnaissance Automatique du Locuteur*. PhD thesis, Université d'Avignon et des Pays de Vaucluse, April 1998.

[11] C. Beumier and M. Acheroy. "Automatic face identification". Technical report, Royal Military Academy, Department of Electrical Engineering, July 1995.

[12] C. Beumier and M. Acheroy. "Automatic profile identification". In *Proceedings of the first international conference on Audio- and Video-based Biometric Person Authentication*, Crans-Montana, Switzerland, March 1997.

[13] E. Bigün, J. Bigün, B. Duc, and S. Fisher. "Expert conciliation for multi modal person authentication systems by bayesian statistics". In *Proceedings of the first international conference on Audio- and Video-based Biometric Person Authentication*, pages 327–334, Crans-Montana, Switzerland, March 1997.

[14] E. S. Bigün. "Risk analysis of catastrophes using experts' judgments: An empirical study on risk analysis of major civil aircraft accidents in Europe". *European J. Operational research*, Vol. 87, pp. 599–612, 1995.

[15] E. S. Bigün. *Bayesian risk analysis of rare events, such as catastrophes, by means of expert assessments.* PhD thesis, Stockholm University, 1997.

[16] J. Bigün, G. Chollet, and G. Borgefors, editors. *First International Conference on Audio- and video-based biometric person authentication*, Crans-Montana, Switzerland, March 1997. Springer.

[17] F. Bimbot. "An evaluation of temporal decomposition". Technical report, Acoustic research departement AT&T Bell Labs, 1990.

[18] F. Bimbot and G. Chollet. "Assessment of speaker verification systems". In *Handbook of Standards and Resources for Spoken Language Systems.* Mouton de Gruyter, 1997.

[19] F. Bimbot, G. Chollet, and A. Paoloni. "Assessment methodology for speaker identification and verification systems". In *ESCA Workshop on automatic speaker recognition, identification and verification*, pages 75–82, Martigny, Switzerland, April 1994.

[20] F. Bimbot, H.-P. Hutter, C. Jaboulet, J. Koolwaaij, J. Lindberg, and J.-B. Pierrot. "An overview of the cave project research activities in speaker verification". In *RLA2C*, pages 215–220, Avignon, Paris, 1998.

[21] F. Bimbot, I. Magrin-Chagnolleau, and L. Mathan. "Second0order statistical measures for text-independent speaker identification". *Speech Communication*, Vol. 17, no. 1-2, pp. 177–192, 1995.

[22] F. Bimbot and L. Mathan. "Second-order statistical measures for text-independent speaker identification". In *ESCA workshop on automatic speaker recognition, identification and verification*, Martigny, Switzerland, April 1994.

[23] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford UK, 1995.

[24] S. S. Blackman. "Theoretical Approaches to Data Association and Fusion". In *SPIE Sensor Fusion*, volume 931, pages 50–55, 1988.

[25] I. Bloch. "Information combination operators for data fusion: a comparative review with classification". *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 26, no. 1, pp. 52–67, January 1996.

[26] I. Bloch. "Some aspects of Dempster-Shafer evidence theory for classification of multi-modality medical images taking partial volume effect into account". *Pattern recognition Letters*, Vol. 17, pp. 905–919, 1996.

[27] R. S. Blum, S. A. Kassam, and V. Poor. "Distributed detection with multiple sensors: Part ii - advanced topics". *Proceedings of the IEEE*, Vol. 85, no. 1, pp. 64–79, January 1997.

[28] G. Borgefors. "Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, no. 6, pp. 849–865, November 1988.

[29] D. Borghys, P. Verlinde, C. Perneel, and M. Acheroy. "Multi-level data fusion for the detection of targets using multi-spectral image sequences". *Optical Engineering*, Vol. 37, no. 2, 1998.

[30] E. Boros, P. L. Hammer, T. Ibaraki, A. Kogan, E. Mayoraz, and I. Muchnik. "An implementation of logical analysis of data". IDIAP-RR 5, Institut Dalle Molle d'Intelligence Artificielle Perceptive, Martigny, Switzerland, 1996.

[31] H. Bourlard and N. Morgan. "Speaker Verification A Quick Overview". Technical Report RR 98-12, IDIAP, Martigny, Switzerland, August 1998.

[32] A. P. Bradley. "ROC curves and the chi-square test". *Pattern recognition Letters*, Vol. 17, pp. 287–294, 1996.

[33] R. Brunelli and D. Falavigna. "Person identification using multiple cues". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, no. 10, pp. 955–966, October 1995.

[34] R. Brunelli and T. Poggio. "Face recognition: Features versus templates". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, no. 10, pp. 1042–1043, October 1993.

[35] J. P. Campbell. "Speaker Recognition: A Tutorial". *Proceedings of the IEEE*, Vol. 85, no. 9, pp. 1437–1462, September 1997.

[36] J. P. Campbell. *BIOMETRICS: Personal Identification in Networked Society*, chapter Speaker Recognition, pages 165–189. Kluwer Academic Publishers, 1999.

[37] Z. Chair and P. Varshney. "Optimal data fusion in multiple sensor detection systems". *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 22, no. 1, pp. 98–101, 1986.

[38] R. Chellapa, L. Davis, and P. Phillips, editors. *Proceedings of the Second International Conference on Audio- and Video-based Biometric Person Authentication*, Washington D.C., USA, March 1999.

[39] R. Chellappa, C.L. Wilson, and S. Sirohey. "Human and machine recognition of faces: A survey". *Proceedings of the IEEE*, Vol. 83, no. 5, pp. 705–740, May 1995.

[40] C.C. Chibelushi, J.S. Mason, and F. Deravi. "Integration of acoustic and visual speech for speaker recognition". In *EUROSPEECH'93*, pages 157–160, 1993.

[41] G. Chollet and C. Montacie. "Evaluating speech recognizers and data bases". In H. Niemann, M. Lang, and G. Sagerer, editors, *Recent advances in speech understanding and dialog systems*, volume 46 of *NATO ASI F: Computer and Systems Sciences*, pages 345–348. Springer-Verlag, 1988.

[42] G. Chollet, J. Černocký, A. Constantinescu, S. Deligne, and F. Bimbot. *"Towards ALISP: a proposal for Automatic Language Independent Speech Processing"*. "NATO ASI: Computational models of speech pattern processing". Springer Verlag, 1998.

[43] T. Choudhury, B. Clarkson, T. Jebara, and A. Pentland. "Multimodal Person Recognition using Unconstrained Audio and Video". In *Second International Conference on Audio- and Video-based Biometric Person Authentication*, pages 176–181, Washington D. C., USA, March 1999.

[44] K. Choukri. *Quelques approches pour l'adaptation aux locuteurs en reconnaissance automatique de la parole.* PhD thesis, ENST, Paris, November 1988.

[45] B. V. Dasarathy. "Decision fusion strategies in multisensor environments". *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 21, no. 5, pp. 1140–1154, September/October 1991.

[46] B. V. Dasarathy. *Decision Fusion.* IEEE Computer Society Press, 1994.

[47] B. V. Dasarathy. "Fusion strategies for enhancing decision reliability in multisensor environments". *Optical Engineering*, Vol. 35, no. 3, pp. 603–616, March 1996.

[48] B. V. Dasarathy. "Sensor fusion potential exploitation - innovative architectures and illustrative applications". *Proceedings of the IEEE*, Vol. 85, no. 1, pp. 24–38, January 1997.

[49] P. A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach.* Prentice-Hall Inc., Englewood Cliffs NJ, 1982.

[50] U. Dieckmann, P. Plankensteiner, and T. Wagner. "Sesam: A biometric person identification system using sensor fusion". *Pattern recognition letters*, Vol. 18, no. 9, pp. 827–833, September 1997.

[51] T. G. Dietterich. "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms". *Neural Computation*, Vol. 10, no. 7, 1998.

[52] G. R. Doddington. "Speaker recognition-identifying people from their voices". *Proceedings of the IEEE*, Vol. 73, no. 11, pp. 1651–1664, 1985.

[53] E. Drakopoulos and C. Lee. "Optimal Multisensor Fusion of Correlated Local Decisions". *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 27, no. 4, pp. 593–605, 1991.

[54] B. Duc, E. Bigün, J. Bigün, G. Maître, and S. Fischer. "Fusion of audio and video information for multi modal person authentication". *Pattern Recognition Letters*, Vol. 18, no. 9, pp. 835–843, September 1997.

[55] B. Duc, G. Maître, S. Fischer, and J. Bigün. "Person authentication by fusing face and speech information". In *Proceedings of the First International Conference on Audio- and Video-based Biometric Person Authentication*, Lecture Notes in Computer Science. Springer Verlag, 1997.

[56] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.

[57] R. P. W. Duin. "A note on comparing classifiers". *Pattern recognition Letters*, Vol. 17, pp. 529–536, 1996.

[58] K. R. Farrell. "Text-dependent speaker verification using data fusion". In *Proceedings ICASSP '95*, pages 349–352, Detroit, MI, May 1995.

[59] K. R. Farrell, R. J. Mammone, and K. T. Assaleh. "Speaker recognition using neural networks and conventional classifiers". *IEEE Transactions on Speech and Audio Processing*, Vol. 2, no. 1, Part II, pp. 194–205, January 1994.

[60] S. E. Fredrickson and L. Tarassenko. "Radial basis functions for speaker identification". In *ESCA Workshop on automatic speaker recognition, identification and verification*, pages 107–110, Martigny, Switzerland, April 1994.

[61] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, second edition, 1990.

[62] S. Furui. "Cepstral analysis technique for automatic speaker verification". *IEEE Transactions on ASSP*, Vol. 29, no. 2, pp. 254–272, 1981.

[63] S. Furui. "Comparison of speaker recognition methods using statistical features and dynamic features". *IEEE Transactions on ASSP*, Vol. 29, no. 3, pp. 342–350, 1981.

[64] S. Furui. "An overview of speaker recognition technology". In *ESCA Workshop on automatic speaker recognition, identification and verification*, pages 1–9, Martigny, Switzerland, April 1994.

[65] S. Furui. *Automatic speech and speaker recognition: advanced topics*, chapter An overview of speaker recognition technology, pages 31–56. Kluwer Academic publishers, 1996.

[66] S. Furui. "Recent advances in speaker recognition". *Pattern recognition letters*, Vol. 18, no. 9, pp. 859–872, September 1997.

[67] D. Genoud, F. Bimbot, G. Gravier, and G. Chollet. "Combining methods to improve speaker verification decision". In ICSLP, editor, *Proceedings of The Fourth International Conference on Spoken Language Processing*, Philadelphia, October 3-6 1996. ICSLP, ICSLP.

[68] Allen Gersho and Robert Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

[69] Y. Grenier. *Identification du locuteur et adaptation au locuteur d'un système de reconnaissance phonémique*. PhD thesis, Ecole Nationale Supérieure de Télécommunications, Paris, France, 1977.

[70] D. L. Hall. *Mathematical techniques in Multisensor Data Fusion*. Artech House, 1992.

[71] D. L. Hall and J. Llinas. "An introduction to multisensor data fusion". *Proceedings of the IEEE*, Vol. 85, no. 1, pp. 6–23, January 1997.

[72] J. Hennebert and D. Petrovska-Delacrétaz. "Phoneme based text-prompted speaker verification with multi-layer perceptrons". In *RLA2C*, pages 55–58, Avignon, France, 1998.

[73] A. Higgins, L. Bahler, and J. Porter. *Automatic speech and speaker recognition: advanced topics*, chapter Voice identification using non-parametric density matching, pages 211–232. Kluwer Academic publishers, 1996.

[74] R. Hill. *BIOMETRICS: Personal Identification in Networked Society*, chapter Retina Identification, pages 123–141. Kluwer Academic Publishers, 1999.

[75] T. K. Ho, J. J. Hull, and S. N. Srihari. "Decision combination in multiple classifier systems". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, no. 1, pp. 66–75, January 1994.

[76] H. Hollien and M. Jiang. "The challenge of effective speaker identification". In *RLA2C*, pages 2–9, Avignon, France, 1998.

[77] J. P. Holmes, R. L. Maxwell, and L. J. Wright. "A performance evaluation of biometric identification devices". Technical report, Sandia National Laboratories, Albuquerque NM 87185, July 1990.

[78] J. P. Holmes, R. L. Maxwell, and L. J. Wright. "A performance evaluation of biometric identification devices". Technical Report SAND91-0276, Sandia National Laboratories, Albuquerque NM 87185, June 1991.

[79] L. Holmstrom, P. Koistinen, J. Laaksonen, and E. Oja. "Neural and Statistical Classifiers-Taxonomy and Two Case Studies". *IEEE Transactions on Neural Networks*, Vol. 8, no. 1, pp. 5–17, January 1997.

[80] M. M. Homayounpour. *Vérification vocale d'identité: dépendante et indépendante du texte*. PhD thesis, Université de Paris XI Orsay, Paris, France, 1995.

[81] M. M. Homayounpour and G. Chollet. "A comparison of some relevant parametric representations for speaker verification". In *ESCA Workshop on automatic speaker recognition, identification and verification*, pages 185–188, Martigny, Switzerland, April 1994.

[82] L. Hong and A. Jain. "Integrating Faces and Fingerprints for Personal Identification". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, no. 12, pp. 1295–1307, December 1998.

[83] D. W. Hosner and S. Lemeshow. *Applied Logistic Regression*. John Wiley & Sons, 1989.

[84] Y. S. Huang and C. Y. Suen. "A method of combining multiple classifiers". In *Proceedings of the 12th international conference of pattern recognition*, volume 2, pages 473–475, Jerusalem, Israel, October 1994.

[85] G. R. Iversen. *Bayesian statistical inference*. SAGE publications, 1984.

[86] T. S. Jaakkola and M. I. Jordan. "A variational approach to Bayesian logistic regression models and their extensions". Technical report, Department of Brain and Cognitive Sciences, MIT, Cambridge, MA, 1996.

[87] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. "Adaptive mixtures of local experts". *Neural Computation*, Vol. 3, no. 1, pp. 79–87, 1991.

[88] A. Jain, R. Bolle, and S. Pankanti. *BIOMETRICS: Personal Identification in Networked Society*, chapter Introduction to Biometrics, pages 1–41. Kluwer Academic Publishing, 1999.

[89] A. K. Jain, L. Hong, S. Pankanti, and R. Bolle. "An Identity-Authentication System Using Fingerprints". *Proceedings of the IEEE*, Vol. 85, no. 9, pp. 1365–1388, September 1997.

[90] F. Jauquet. *Intégration des méthodes de vérification du locuteur dans une liaison par vocodeur*. PhD thesis, Faculté Polytechhnique de Mons, July 1998.

[91] M. I. Jordan. "Why the logistic function? A tutorial discussion on probabilities and neural networks". Computational Cognitive Science 9503, Massachusetts Institute of Technology, Cambridge MA, August 1995.

[92] P. Jourlin, J. Luettin, D. Genoud, and H. Wassner. "Acoustic-labial speaker verification". *Pattern Recognition Letters*, Vol. 18, no. 9, pp. 853–858, September 1997.

[93] P. Jourlin, J. Lüttin, D. Genoud, and H. Wassner. "Acoustic-labial speaker verification". In *Proceedings of the First International Conference on Audio- and Video-based Biometric Person Authentication*, Lecture Notes in Computer Science. Springer Verlag, 1997.

[94] B.-H. Juang, W. Chou, and C.-H. Lee. *Automatic speech and speaker recognition: advanced topics*, chapter Statistical and discriminative methods for speech recognition, pages 109–132. Kluwer Academic publishers, 1996.

[95] B.-H. Juang, W. Chou, and C.-H. Lee. "Minimum classification error rate methods for speech recognition". *IEEE Transactions on Speech and Audio Processing*, Vol. 5, no. 3, pp. 257–265, May 1997.

[96] G. K. Kanji. *100 Statistical tests*. SAGE Publications, 1993.

[97] J. Kittler, M. Hatef, and R. P. W. Duin. "Combining classifiers". In *Proceedings of 13th International Conference on Pattern Recognition*, pages 897–901, Vienna, Austria, 1996.

[98] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. "On combining classifiers". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, no. 3, pp. 226–239, March 1998.

[99] J. Kittler, Y. Li, J. Matas, and M. U. R. Sanchez. "Combining evidence in multimodal personal identity recognition systems". In *Proceedings of the first international conference on Audio- and Video-based Biometric Person Authentication*, Crans-Montana, Switzerland, March 1997.

[100] J. Kittler, G. Matas, K. Jonsson, and M. U. R. Sánchez. "Combining evidence in personal identity verification systems". *Pattern Recognition Letters*, Vol. 18, no. 9, pp. 845–852, September 1997.

[101] L. A. Klein. *Sensor and Data Fusion Concepts and Applications*, volume 14 of *Tutorial Texts Series*. SPIE Optical Engineering Press, Washington, 1993.

[102] C. Kotropoulos, I. Pitas, S. Fischer, B. Duc, and J. Bign. "Face authentication using morphological dynamic link architecture". In *Proceedings of the First International Conference on Audio- and Video-based Biometric Person Authentication*, Lecture Notes in Computer Science. Springer Verlag, 1997.

[103] L. Lam and C. Y. Suen. "Optimal combinations of pattern classifiers". *Pattern recognition Letters*, Vol. 16, pp. 945–954, 1995.

[104] C.-H. Lee and J.-L. Gauvain. *Automatic speech and speaker recognition: advanced topics*, chapter Bayesian adaptive learning and MAP estimation of HMM, pages 83–107. Kluwer Academic publishers, 1996.

[105] C.-H. Lee, F. K. Soong, and K. K. Paliwal. *Automatic speech and speaker recognition: advanced topics*. Kluwer Academic publishers, 1996.

[106] P. M. Lee. *Bayesian statistics: An introduction*. Arnold, second edition, 1997.

[107] D. V. Lindley. *Making Decisions*. Wiley, second edition, 1985.

[108] L. Ljung. "Convergence Analysis of Parametric Identification Methods". *IEEE Transactions on Automatic Control*, Vol. 23, no. 5, pp. 770–783, October 1978.

[109] R. C. Luo and M. G. Kay. "Multisensor integration and fusion in intelligent systems". *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 19, no. 5, September/October 1989.

[110] D. MacKay. "Bayesian interpolation". *Neural Computation*, Vol. 4, no. 3, pp. 415–447, 1992.

[111] I. Magrin-Chagnolleau. *Approches statistiques et filtrage vectoriel de trajectoires spectrales pour l'identification du locuteur indépendante du texte*. PhD thesis, Ecole Nationale Supérieure de Télécommunications, Paris, France, 1997.

[112] R. J. Mammone, X. Zhang, and R. P. Ramachandran. "Robust speaker recognition, a feature based approach". *IEEE Signal Processing Magazine*, Vol. 13, no. 5, pp. 58–71, September 1996.

[113] E. Mandler and J. Schürmann. "Combining the classification results of independent classifiers based on the dempster/shafer theory of evidence". In E. S. Gelsema and L. N. Kanal, editors, *Pattern Recognition and Artificial Intelligence*, pages 381–393. Elsevier Science Publishers, 1988.

[114] B. F. J. Manly. *Multivariate Statistical Methods*. Chapman & Hall, second edition, 1994.

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

[116] J. Matas, K. Jonsson, and J. Kittler. "Fast face localization and verification". In A. Clark, editor, *British Machine Vision Conference*, pages 152–161. BMVA Press, 1997.

[117] E. Mayoraz and F. Aviolat. "Constructive training methods for feed-forward neural networks with binary weights". *International Journal of Neural Systems*, Vol. 7, no. 2, pp. 149–166, May 1996.

[118] G. J. McLachlan. *Discriminant analysis and statistical pattern recognition*. John Wiley & Sons, 1992.

[119] D. Michie, D. J. Spiegelhalter, and C. C. Taylor. *Machine learning, neural and statistical classification*. Ellis Horwood, 1994.

[120] B. Miller. "Vital signs of identity". *IEEE Spectrum*, Vol. 31, no. 2, pp. 22–30, February 1994.

[121] T. M. Mitchell. *Machine learning*. Mc Graw-Hill, 1997.

[122] P. Moerland. "Mixtures of experts estimate a posteriori probabilities". In W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, editors, *Proceedings of the International Conference on Artificial Neural Networks (ICANN'97)*, number 1327 in Lecture Notes in Computer Science, pages 499–504, Berlin, 1997. Springer-Verlag. (IDIAP-RR 97-07).

[123] D. C. Montgomery. *Design and Analysis of Experiments.* John Wiley & Sons, fourth edition, 1997.

[124] M. Munich and P. Perona. "Visual-Based ID Verification by Signature Tracking". In *Proceedings of the Second International Conference on Audio- and Video-based Biometric Person Authentication*, pages 236–241, Washington D.C., USA, March 1999.

[125] V. S. Nalwa. *BIOMETRICS: Personal Identification in Networked Society*, chapter Automatic On-line Signature Verification, pages 143–163. Kluwer Academic Publishers, 1999.

[126] M. Nixon, J. Carter, D. Cunado, P. Huang, and S. Stevenage. *Biometrics: Personal Identification in a Networked Society*, chapter Automatic Gait Recognition, pages 231–250. Kluwer Academic Publishing, 1999.

[127] M. S. Obaidat and B. Sadoun. *BIOMETRICS: Personal Identification in Networked Society*, chapter Keystroke Dynamics Based Authentication, pages 213–229. Kluwer Academic Publishers, 1999.

[128] J. O'Brien. "An Algorithm for the Fusion of Correlated Probabilities". In *Proceedings of the International Conference on Multisource-Multisensor Information Fusion*, volume 2, pages 565–571, Las Vegas, USA, July 1998.

[129] J. Oglesby. "What's in a number?: Moving beyond the equal error rate". In *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pages 87–90, Martigny, April 1994.

[130] L. O'Gorman. *BIOMETRICS: Personal Identification in Networked Society*, chapter Fingerprint Verification, pages 43–64. Kluwer Academic Publishers, 1999.

[131] J. Olsen. *Phoneme based speaker recognition.* PhD thesis, Aalborg University, Aalborg, Denmark, 1997.

[132] J. Olsen. "A two-stage procedure for phone based speaker verification". *Pattern recognition letters*, Vol. 18, no. 9, pp. 889–897, September 1997.

[133] D. Petrovska-Delacrétaz and J. Hennebert. "Text-prompted speaker verification experiments with phoneme specific MLPs". In *ICASSP'98*, pages 777–780, Seattle, WA, 1998.

[134] D. Petrovska-Delacrétaz, J. Černocký, J. Hennebert, and G. Chollet. "Text-independent speaker verification using automatically labelled acoustic segments". In *International Conference on Spoken Language Processing (ICLSP)*, Sydney, Australia, December 1998.

[135] S. Pigeon. *Authentification multimodale d'identité.* PhD thesis, Université Catholique de Louvain, February 1999.

[136] S. Pigeon and L. Vandendorpe. "The M2VTS database (release 1.00)". http://www.tele.ucl.ac.be/M2VTS, 1996.

[137] S. Pigeon and L. Vandendorpe. "The m2vts multi-modal face database". In *Proceedings of the First International Conference on Audio- and Video-based Biometric Person Authentication*, Lecture Notes in Computer Science. Springer Verlag, 1997.

[138] S. Pigeon and L. Vandendorpe. "Profile authentication using a Chamfer matching algorithm". In *Proceedings of the First International Conference on Audio- and Video-based Biometric Person Authentication*, Lecture Notes in Computer Science, pages 185–192. Springer Verlag, 1997.

[139] S. Pigeon and L. Vandendorpe. "Multiple experts for robust face authentication". In SPIE, editor, *Optical security and counterfeit deterrence II*, volume 3314, pages 166–177, San Jose CA, January 1998.

[140] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C.* Cambridge University Press, second edition, 1992.

[141] F. J. Prokoski and R. B. Riedel. *BIOMETRICS: Personal Identification in Networked Society*, chapter Infrared Identification of Faces and Body Parts, pages 191–212. Kluwer Academic Publishers, 1999.

[142] M. A. Przybocki and A. F. Martin. "NIST speaker recognition evaluations". In *First international conference on language resources and evaluation*, volume I, pages 331–335, Granada, Spain, May 1998. ELRA.

[143] J. R. Quinlan. *C4.5: programs for machine learning.* Morgan Kaufmann, 1993.

[144] V. Radevski and Y. Bennani. "Combining structural and statistical features for handwritten digit recognition". In *International Joint Conference of Information Sciences*, pages 102–105, Research Triangle Park, USA, 1997.

[145] V. Radevski and Y. Bennani. "Committee neural classifiers for structural and statistical features combination". In *International Conference ANNIE'97*, Missouri, USA, 1997.

[146] N. Rao. "Distributed decision fusion using empirical estimation". *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 33, no. 4, pp. 1106–1114, 1997.

[147] N. S. V. Rao and S. S. Iyengar. "Distributed decision fusion under unknown distributions". *Optical Engineering*, Vol. 35, no. 3, pp. 617–624, March 1996.

[148] N. Ratha and R. Bolle. *BIOMETRICS: Personal Identification in Networked Society*, chapter Smartcard Based Authentication, pages 369–384. Kluwer Academic Publishers, 1999.

[149] A. Reibman and L. Nolte. "Design and Performance Comparison of Distributed Detection Networks". *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 23, no. 6, pp. 789–797, 1987.

[150] D. A. Reynolds. "Speaker identification and verification using gaussian mixture speaker models". In *ESCA Workshop on automatic speaker recognition, identification and verification*, pages 27–30, Martigny, Switzerland, April 1994.

[151] D. A. Reynolds and R. C. Rose. "Robust text-independent speaker identification using gaussian mixture speaker models". *IEEE Transactions on Speech and Audio Processing*, Vol. 3, no. 1, pp. 72–83, January 1995.

[152] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[153] A. R. Roddy and J. D. Stosz. "Fingerprint Features-Statistical Analysis and System Performance Estimates". *Proceedings of the IEEE*, Vol. 85, no. 9, pp. 1390–1421, September 1997.

[154] A. E. Rosenberg. "Automatic speaker verification: a review". *Proceedings of the IEEE*, Vol. 64, no. 4, pp. 475–487, April 1976.

[155] G. Saporta. *Probabilités, analyse des données et statistique*, volume I. Editions Technip, 1990.

[156] G. Shafer. *A mathematical theory of evidence*. Princeton University Press, 1976.

[157] S. Sharma, P. Vermeulen, and H. Hermansky. "Combining information from multiple classifiers for speaker verification". In *Proceedings of Speaker Recognition and its Commercial and Forensic Applications*, Avignon, France, April 1998.

[158] W. Shen, M. Surette, and R. Khanna. "Evaluation of Automated Biometrics-Based Identification and Verification Systems". *Proceedings of the IEEE*, Vol. 85, no. 9, pp. 1464–1478, September 1997.

[159] S. Siegel and N. J. Castellan. *Nonparametric statistics*. McGraw-Hill, 1988.

[160] F. K. Soong and A. E. Rosenberg. "On the use of instantaneous and transitional spectral information in speaker recognition". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 36, pp. 871–879, June 1988.

[161] P. Sprent. *Applied nonparametric statistical methods*. Chapman and Hall, 1989.

[162] SPSS. `http://www.spss.com`, 1998.

[163] R. Tenney and N. Sandell. "Detection With Distributed Sensors". *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 17, no. 4, pp. 98–101, 1981.

[164] C. W. Therrien. *Decision Estimation and Classification; An Introduction to Pattern Recognition and Related Topics*. Wiley, 1989.

[165] S. Thomopoulos, R. Viswanathan, and B. Bougoulias. "Optimal Distributed Decision Fusion". *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 25, no. 5, pp. 761–765, 1989.

[166] S. Thomopoulos, R. Viswanathan, and R. Tumuluri. "Optimal Serial Distributed Decision Fusion". *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 24, no. 4, pp. 366–376, 1988.

[167] H. L. Van Trees. *Detection, Estimation and Modulation Theory*, volume 1. John Wiley & Sons, New York, 1968.

[168] V. N. Vapnik. *The nature of statistical learning theory.* Springer-Verlag, New-York, USA, 1995.

[169] V. N. Vapnik. *Statistical learning theory.* John-Wiley & Sons, New-York, USA, 1998.

[170] P. K. Varshney. *Distributed detection and data fusion.* Springer, 1996.

[171] J. Černocký, D. Petrovska-Delacrétaz, P. Verlinde, and G. Chollet. "A segmental approach to text-independent speaker verification". In *EUROSPEECH'99*, Budapest, Hungary, September 1999. ESCA.

[172] P. Verlinde, D. Borghys, C. Perneel, and M. Acheroy. "Data fusion for long range target acquisition". In *7th symposium on Multi-Sensor Systems, and Data Fusion for Telecommunications, Remote Sensing and Radar*, Lisbon, October 1997. NATO RTO.

[173] P. Verlinde and G. Chollet. "Combining vocal and visual cues in an identity verification system using k-nn based classifiers". In *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, Los Angeles, USA, December 1998.

[174] P. Verlinde and G. Chollet. "Comparing decision fusion paradigms using k-NN based classifiers, decision trees and logistic regression in a multi-modal identity verification application". In *Proceedings of the Second International Conference on Audio- and Video-based Biometric Person Authentication*, pages 188–193, Washington D. C., USA, March 1999.

[175] P. Verlinde, G. Chollet, and M. Acheroy. "About Multi-Modal Identity Verification in Interactive Dialogue Systems". In *Interactive Dialogue in Multi-Modal Systems*, Kloster Irsee, Germany, June 1999. ESCA.

[176] P. Verlinde, P. Druyts, G. Chollet, and M. Acheroy. "A multi-level data fusion approach for gradually upgrading the performances of identity verification systems". In *Sensor Fusion: Architectures, Algorithms, and Applications III*, volume 3719, Orlando, USA, April 1999. SPIE Press.

[177] P. Verlinde, P. Druyts, G. Chollet, and M. Acheroy. "Applying Bayes based classifiers for decision fusion in a multi-modal identity verification system". In *International Symposium on Pattern Recognition "In Memoriam Pierre Devijver"*, Brussels, Belgium, February 1999.

[178] P. Verlinde, D. Genoud, G. Gravier, and G. Chollet. "Proposition d'une stratégie de fusion de données à trois niveaux pour la vérification d'identité". In *XXIIièmes Journées d'Etude sur la Parole*, Martigny, Switzerland, June 1998.

[179] P. Verlinde, G. Maître, and E. Mayoraz. "Decision fusion in a multi-modal identity verification system using a multi-linear classifier". IDIAP-RR 6, IDIAP, September 1997.

[180] P. Verlinde, G. Maître, and E. Mayoraz. "Decision Fusion using a Multi-Linear Classifier". In *Proceedings of the International Conference on Multisource-Multisensor Information Fusion*, volume 1, pages 47–53, Las Vegas, USA, July 1998.

[181] E. L. Waltz and J. Llinas. *Multisensor Data Fusion*. Artech House Publishing, 1990.

[182] J. L. Wayman. *BIOMETRICS: Personal Identification in Networked Society*, chapter Technical Testing and Evaluation of Biometric Identification Devices, pages 345–368. Kluwer Academic Publishers, 1999.

[183] J. J. Weng and D. L. Swets. *BIOMETRICS: Personal Identification in Networked Society*, chapter Face Recognition, pages 65–86. Kluwer Academic Publishers, 1999.

[184] R. P. Wildes. "Iris Recognition: An Emerging Biometric Technology". *Proceedings of the IEEE*, Vol. 85, no. 9, pp. 1348–1363, September 1997.

[185] J. D. Woodward. "Biometrics: Privacy's Foe or Privacy's Friend". *Proceedings of the IEEE*, Vol. 85, no. 9, pp. 1480–1492, September 1997.

[186] J. D. Woodward. *BIOMETRICS: Personal Identification in Networked Society*, chapter BIOMETRICS: Identifying Law & Policy Concerns, pages 385–405. Kluwer Academic Publishers, 1999.

[187] L. Xu, A. Krzyzak, and C.Y. Suen. "Methods of combining multiple classifiers and their applications to handwriting recognition". *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 22, no. 3, pp. 418–435, May/June 1992.

[188] D. Zhang and W. Shu. "Two novel characteristics in palmprint verification: datum point invariance and line feature matching". *Pattern Recognition*, Vol. 32, no. 4, pp. 691–702, April 1999.

[189] J. Zhang, Y. Yan, and M. Lades. "Face Recognition: Eigenface, Elastic Matchimg and Neural Nets". *Proceedings of the IEEE*, Vol. 85, no. 9, pp. 1423–1435, September 1997.

[190] R. L. Zunkel. *BIOMETRICS: Personal Identification in Networked Society*, chapter Hand Geometry Based Verification, pages 87–101. Kluwer Academic Publishers, 1999.

Appendix A

A monotone multi-linear classifier

Principle

We have developed a classifier determining regions in the d-dimensional space corresponding to the two classes, based on a combination of hyper-planes. We call this classifier *multi-linear classifier* in reference to the use of several hyper-planes, each one building a linear classifier.
The classifier training consists of a supervised phase in which the different hyper-planes are determined by hyper-planes which optimally separate pairs of points of either class and where the regions generated by these hyper-planes are labeled with the class identifier (accept, reject).
At testing, each data point from the test set is simply given the class label of the region it is belonging to.

Training

Overview

Given examples of the two classes, the goal is to find hyper-planes separating optimally all pairs of points of either class and to label the generated regions with the corresponding class identifier. Let's describe the samples available for training by the two sets:

- the set of *positive points* (representing the client claims) $\{\boldsymbol{a}^k\}_{k \in K} \in \mathbb{R}^d, |K|$ being the total number of positive points;

- the set of *negative points* (representing the client claims) $\{\boldsymbol{a}^l\}_{l \in L} \in \mathbb{R}^d, |L|$ being the total number of negative points;

The training of the multi-linear classifier consists of:

First step Reduction of training samples;

Second step Determination of hyper-planes;

Third step Class attribution to intersections of hyper-planes.

Each of these steps is going to be detailed separately hereafter.

Reduction of training samples

In a first step the classifier reduces the number of data points in the two classes by using the monotonicity hypothesis. In this specific case, the constraint of monotonicity implies that a given linear separator (*i.e.* a hyper-plane):

- has a positive normal vector, which can be formally expressed as $w_i^s \geq 0, \forall i = 1, \dots, d$;
- is considered to separate a particular pair of points only if the positive point (client) is on the positive side of the separator, and the negative point (impostor) on the negative side.

This monotonicity hypothesis allows a preprocessing of the input of the problem as follows: if there exists two points $\boldsymbol{x}$ and $\boldsymbol{y}$ in the positive set (respectively negative set) such that $x_i \leq y_i, \forall i = 1, \dots, d$, the point y (respectively x) can be suppressed from the input set.
As a result of this data reduction, only data points situated along the separation surface of the two classes are maintained. This technique reduces thus also the number of couples that can be formed consisting of one point from each class. These couples are the ones used in the next step.

Determination of hyper-planes

Principle The hyper-planes are determined by **maximizing** a separability (discrimination) measure of point pairs.
The goal is thus to determine a set of S hyper-planes given by $(\boldsymbol{w}^s, w_0^s) \in \mathbb{R}^d \times \mathbb{R}, s \in S$, with the following property for the discrimination between two points. A given hyper-plane $(\boldsymbol{w}^s, w_0^s)$ discriminates between two points

$\boldsymbol{x}$,$\boldsymbol{y} \in \mathbb{R}^d$ if $(\boldsymbol{x}\boldsymbol{w}^s - w_0^s)$ and $(\boldsymbol{y}\boldsymbol{w}^s - w_0^s)$ are both non-zero and of opposite signs. Because of the monotonicity constraint, it will be considered that $(\boldsymbol{w}^s, w_0^s)$ discriminates between $\boldsymbol{x}$,$\boldsymbol{y}$ only if $\boldsymbol{y}\boldsymbol{w}^s - w_0^s < 0 < \boldsymbol{x}\boldsymbol{w}^s - w_0^s$, and in this case, the quality of this discrimination is given by the minimum of the module of these two values, *i.e.* by $\min\{\boldsymbol{x}\boldsymbol{w}^s - w_0^s,\ -\boldsymbol{y}\boldsymbol{w}^s + w_0^s\}$.
The total discrimination for the *whole* set of separators for each pair of points is simply the sum of the discrimination obtained for each separator. This total discrimination for a pair is defined as Δ. A reference value for Δ is given by Δ_0, defined as half of the minimal Euclidean distance between a pair of positive/negative points $(\boldsymbol{x}, \boldsymbol{y})$. This is the discrimination for $(\boldsymbol{x}, \boldsymbol{y})$ that would obtain a single hyper-plane cutting orthogonally and at the middle, the segment $[\boldsymbol{x}, \boldsymbol{y}]$. It is obvious that the total number S of hyper-planes thus obtained will (amongst else) strongly depend on this user-defined value of Δ. The greater this value of Δ is chosen to be, the greater the total number of hyper-planes in the set will be. This dependency of the number of separators in the set on the choice of Δ, can be observed in the example of section A.
As we already have announced, we wish to be capable to introduce a *bias* in the classifier. This can be achieved by weighting differently the separation towards positive and negative points. Therefore the previous discrimination measure will be replaced by $\min\{\alpha(\boldsymbol{x}\boldsymbol{w}^s - w_0^s),\ -\boldsymbol{y}\boldsymbol{w}^s + w_0^s\}$, where α is any non-zero constant. It is clear that the value of α determines the bias that show the hyper-planes with respect to a certain class. In section A this "attraction tendency" can be observed. The reference value for α is "1", which corresponds to no bias at all.
One can thus see that the number S of hyper-planes generated and the bias they show towards one of either classes, are governed by two user-defined parameters respectively called Δ and α.

Proposed hybrid approach To solve this formal problem, we propose to use two successive phases: an iterative one followed by a global one. The purpose of the iterative phase is to generate iteratively a set of S linear separators (coarse tuning). The subsequent global phase is then used to locally optimize this set of S hyper-planes (fine tuning).

The iterative phase In this phase the total separability Δ to be achieved is fixed (by the user) and using this value a first hyper-plane is calculated. Subsequently, hyper-planes are continued to be inserted iteratively, until the total discrimination Δ is reached *for each pair of points*. At each iteration u the following problem has to be solved: given the two sets of points $\{\boldsymbol{a}^k\}_{k\in K}$, $\{\boldsymbol{a}^l\}_{l\in L} \subset \mathbb{R}^d$ and the hyper-planes $(\boldsymbol{w}^s, w_0^s) \in \mathbb{R}^d \times \mathbb{R}$, $s = 1, \ldots, u-1$ already determined before the current iteration, find $(\boldsymbol{w}^u, w_0^u)$, maximizing the following *iterative goal function*:

$$\text{maximize} \sum_{k\in K, l\in L} \min\{\Delta \ , \sum_{s\in S} \Delta_{kls}\} \tag{A.1}$$

$$\text{where} \ \ \Delta_{kls} = \max\{0, \min\{\alpha(\boldsymbol{a}^k \boldsymbol{w}^s - w_0^s), -\boldsymbol{a}^l \boldsymbol{w}^s + w_0^s\}\} \tag{A.2}$$

$$\text{under the normalization conditions } -1 \leq w_i^s \leq 1, \forall s \in S, \forall i = 0, \ldots, d \tag{A.3}$$

$$\text{and with } S = \{1, \ldots, u\}.$$

The advantage of this method is that the number of hyper-planes need not to be fixed a priori. The disadvantage is that the different hyper-planes are added sequentially to the total set of separators and once they have been entered they are not altered (fine-tuned) any more by the subsequent iterations.

As can be seen in equation (A.1), the maximal quality of the discrimination for a certain pair is limited to Δ. This has explicitly been done to limit the influence of distant pairs (which are easy to separate) on the determination of the current hyper-plane.

The iterative phase has been implemented using a gradient descent method. The computation of the gradient of the iterative goal function is detailed in appendix B. The initial points for this method are obtained in a hybrid manner. Some of the initial points are, as is usually the case, chosen at random. However, a certain number of those initial points are found using a heuristic approach, *i.c.* by calculating a hyper-plane that separates the n worst discriminated pairs at a certain moment. These hyper-planes are calculated using one of two simple classical linear classifiers: either a Fisher or a nearest-mean linear classifier [164, 56], depending on the convergence of the Fisher classifier. The number of random initialization points and the number n of worst discriminated pairs at a certain moment can both be varied by the user, to allow for the generation of a bigger and/or different set of initialization hyper-planes.

The reason why we have chosen this hybrid form of initialization is to be able to cope with the following phenomenon. After only a few iterations, the iterative goal function in equation (A.1) rapidly degenerates in this sense that it doesn't stay a smooth surface, where one can easily use a gradient descent method starting from a randomly chosen initialization point. Instead of the smooth initial surface, there soon appear very scarce and local peaks in the goal function. This is due to the very brutal non-linear behavior of our goal function, showing indeed a succession of *max* and *min* operators, which introduces discontinuities of the first kind. So to have more chances to place the initialization points at least somewhere in the neighborhood of the slopes of (one of) these peaks, the aforementioned heuristic with respect to the separation of the n worst separated points is used. The appearance of these peaks can be clearly observed in the simple two dimensional example of section A. This simple heuristic approach guarantees by no means that the *global* optimum (maximum) of the iterative goal function for the current iteration is going to be reached at each iteration.

Comment w.r.t. a smoother version of the iterative goal function We did try to improve the degree of smoothness of our iterative goal function by replacing the max/min operators in equation (A.1)by a sigmoidal function such as the *atanh*, but this only improves the smoothness of the slopes of the peaks and it doesn't change at all the highly undesirable fact that this goal function rapidly shows very large plateaus where the gradient descent method has absolutely no chance of working. So this sigmoidal like function didn't improve the behavior of the iterative goal function drastically, but it did increase the computing time severely, so we decided to fall back to the original max/min type of goal function, adding the heuristic approach for finding useful initial points.

The global phase In this global phase, the number S of separators is fixed a priori and the purpose of this phase is then to globally maximize the discrimination over all pairs of points by locally acting on all S hyperplanes at the same time (fine-tuning). The following *global goal function* has to be optimized: find $(\boldsymbol{w}^u, w_0^u)$, which

$$\text{maximize} \min_{k \in K, l \in L} \sum_{s \in S} \Delta_{kls} \tag{A.4}$$

where $\Delta_{kls}$is defined as in (A.2), under the constraints described in (A.3).

To try to optimize the set of S hyper-planes that has been found during the iterative phase, the global phase uses a new goal function, as can be seen when comparing equations (A.4) and (A.1). The main difference is that in the global phase we are optimizing the global separation for all pairs, which for a single pair is not limited any longer to the value of Δ, as it was the case during the iterative phase.
The global phase has also been implemented using a gradient descent method. The computation of the gradient of the global goal function is detailed in appendix C. An initial pair of points is selected at random at each iteration; if the total separation of this pair is above the current minimum, nothing is changed, otherwise, the parameters w_i^s are modified in the direction of the gradient of the objective function given in (A.4). This gradient is calculated in the point where the goal function in equation (A.4) is minimal. If there are $|N|$ such points instead of one, then the gradient is calculated in each point. But in this global approach we can use only one general direction for optimizing all S hyper-planes at the same time. To be able to find this best direction, the following problem needs to be solved:

$$\text{Using all } |N| \text{ global gradient vectors: } \nabla^S_{glob_1}, \dots, \nabla^S_{glob_{|N|}} \in \mathbb{R}^{(d+1).S},$$

$$\text{find an } \boldsymbol{x}^S \in \mathbb{R}^{(d+1).S} \text{ such that}$$

$$\forall n \in N : \boldsymbol{w}^S + \eta.\boldsymbol{x}^S \text{ maximizes expression (A.4) for all minimal pairs.}$$

$$\text{Where } d \text{ is the number of modalities to be combined,}$$

$$andS \text{ is the number of hyper-planes in the set,}$$

$$\boldsymbol{w} \in \mathbb{R}^{(d+1).S} \text{ is the vector that contains all } S \text{ hyper-planes,}$$

$$|N| \text{ is the number of pairs with minimal separation and } \eta \text{ is any positive number.}$$

To be able to solve this problem in an easy way, we have transformed it into an alternative form. In appendix D it is shown that the preceding problem is equivalent with the following one:

$$\text{Find } \boldsymbol{x}^S \in \mathbb{R}^{(d+1).S}$$

$$\text{Such that } \forall n \in N,\ \boldsymbol{x}^S.\nabla^S_{glob_n} > 0 \text{ is maximal.}$$

This problem can now be solved easily using linear programming, since all constraints are purely linear [140].

Class attribution to intersections of hyper-planes

The resulting set of S hyper-planes after training induces a partition of the d dimensional space. Each region of this partition is then coded by a word of S bits, indicating its membership to each hyper-plane. A "1" means the considered region is lying on the positive side of the considered separator and a "0" means on the negative side. Afterwards the label of one of either classes is attributed to each region, using the Logical Analysis of Data (LAD) [30] method.
One possibility offered by the flexibility of LAD is to attribute a "?" to a certain bit instead of a "1" or a "0", to express a certain doubt with respect to the classification. In our case we have decided to do this for the regions lying very close to (*i.e.* in a small zone determined by Δ_0 along both sides of) a certain separator.

Coding of training samples In the binarization phase, a data point of the training set is characterized by a word of S bits, according to its membership of a certain region of the partition.

Labeling of the partition After the binarization phase, one of either classes needs to be attributed to each region of the partition and this is done using LAD.

Testing

During testing the membership of each data point w.r.t. the S hyper-planes is calculated and each data point receives simply the class label of the region of the hyper-space it is lying in.

Synthetic two-dimensional example

Representation of the two classes

Figure A.1 shows the two (synthetic) classes of positive and negative points that are used to explain the basic ideas and mechanisms explained so far.

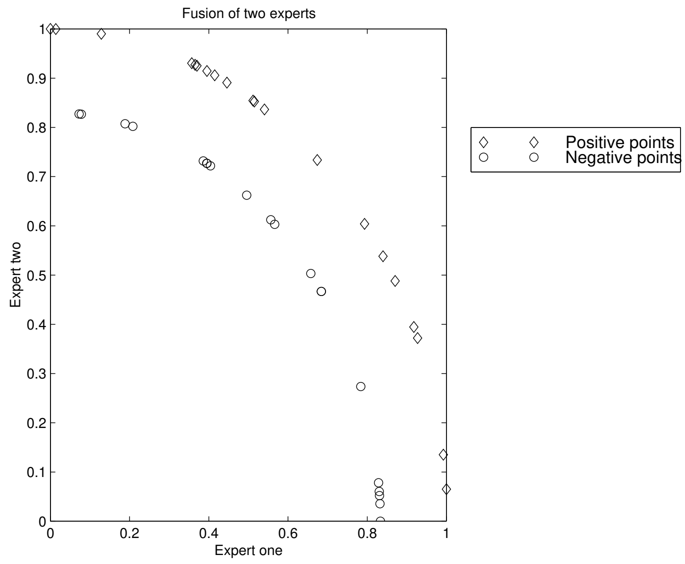

Figure A.1: Simple two dimensional two class problem

Appearance of the goal-function

Figures A.2 and A.3 show the appearance of the iterative goal function (A.1) in the case of this simple example, after respectively two and five iterations. To be able to represent the three components $w_0^s, w_1^s$and w_2^s of a hyper-plane, the two dimensional components $w_1^s$and w_2^s have been represented onto one axis by using the transformation "angle" $= \arccos(w_1^s) = \arcsin(w_2^s)$, the other axis being w_0^s. This transform has also the advantage that it satisfies automatically the normalization constraints (A.3).
It can be clearly seen by comparing Figures A.2 and A.3 that already after five iterations the goal function has enormous plateaus in which the classical gradient descent doesn't work.

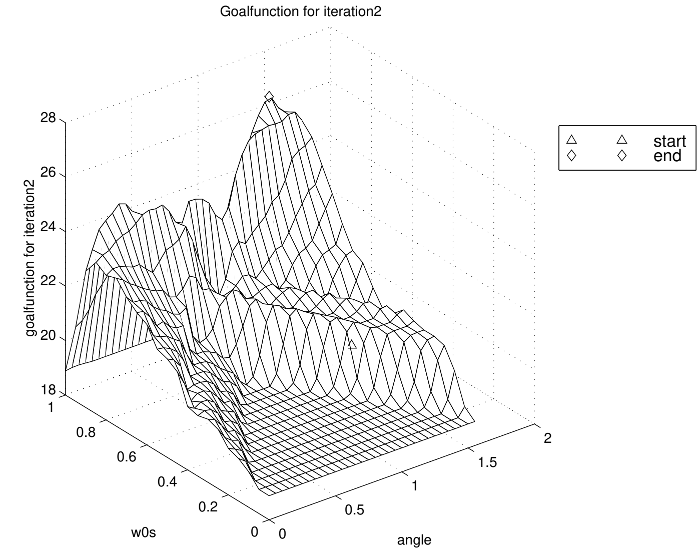

Figure A.2: Example of the iterative goal function after two iterations

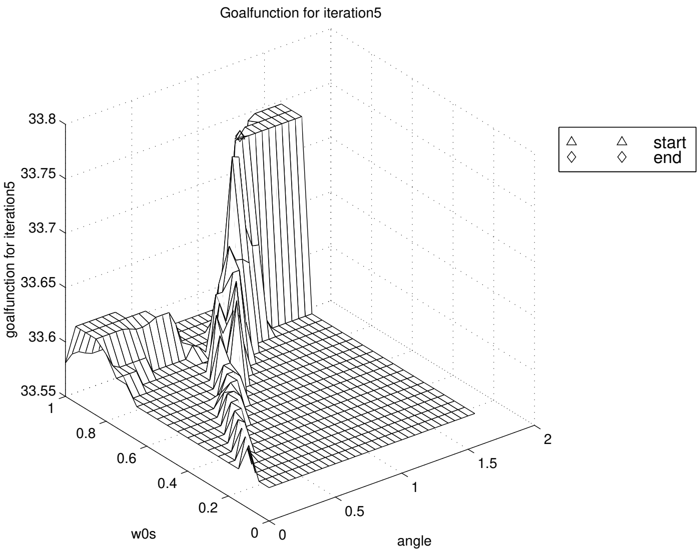

Figure A.3: Example of the iterative goal function after five iterations

Determination of hyper-planes

Figure A.4 shows the set of hyper-planes that the multi-linear classifier has found in the case of this example ($S = 5$). This set of hyper-planes has been found for the reference values for α and Δ and will be used as a reference case to be compared with the following Figures..

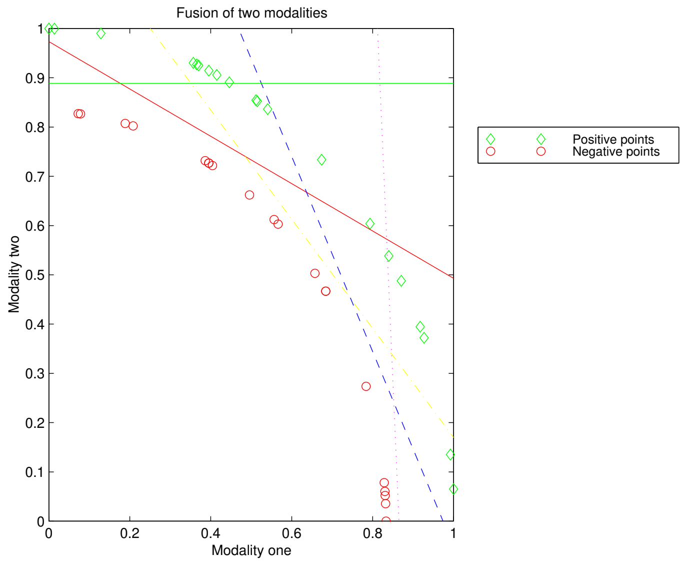

Figure A.4: Set of hyper-planes generated for $\alpha = 1$ and $\Delta = \Delta_0$

Influence of α

Figure A.5 and A.6 show the influence of α on the *attraction tendency* of the set of hyper-planes towards one of either classes. Figure A.5 has been obtained for $\alpha = 0.9$ and Figure A.6 for $\alpha = 1.1$. When α becomes smaller than the reference value, an attraction tendency towards the negative points can be observed and when α on the other hand becomes larger than the reference value, an attraction towards the positive points can be seen. Both these sets of hyper-planes have been found for the reference value for Δ. From the comparison of these two figures with our reference case, it can be seen that the value of α has also an influence on the *number* of hyper-planes that are generated. When α is chosen smaller than the reference value, the number of hyper-planes decreases w.r.t. our reference case ($S = 4 < 5$)

and when α becomes larger than the reference, the number of hyper-planes increases w.r.t. our reference case ($S = 6 > 5$).
This effect could have been expected since, when taking a closer look at equation (A.2), we see that α has a direct influence on the actually calculated discrimination. In the case the two classes are well separated, a value of α close to the reference value should generate the lowest number of hyper-planes. The more α differs from the reference value, the more the hyper-planes are approaching the points of one of either classes and the more hyper-planes will therefore be needed to "zig-zag" around these points. This is a drawback of this method, since ideally spoken the number of hyper-planes generated should only be influenced by Δ. The interdependence of α and Δ makes it more difficult to fine-tune the method for a specific application.

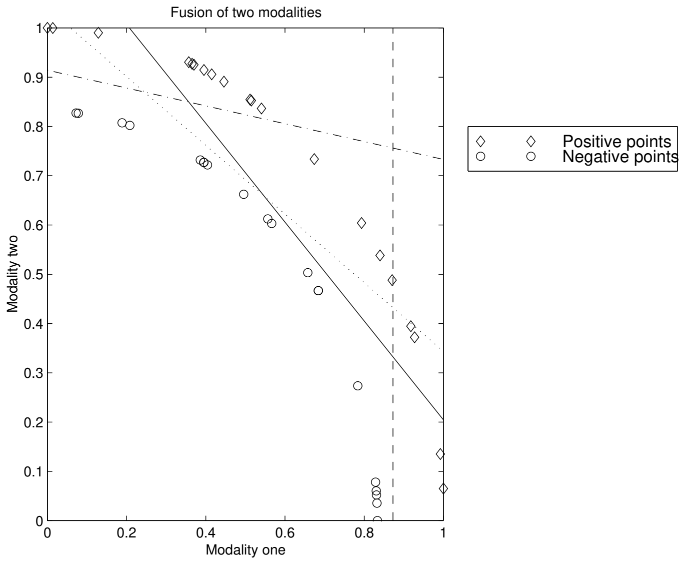

Figure A.5: Set of hyper-planes generated for $\alpha = 0.9$ and $\Delta = \Delta_0$

Influence of Δ

Figure A.7 and A.8 show the influence of Δ on the number of generated hyper-planes. Figure A.7 has been obtained for $\Delta = 0.25 * \Delta_0$ and Figure A.8 for $\Delta = 4 * \Delta_0$. These sets of hyper-planes have both been found

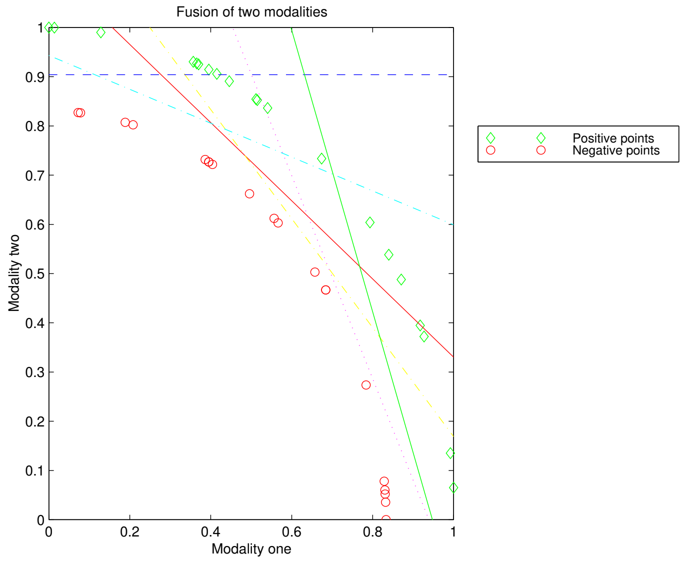

Figure A.6: Set of hyper-planes generated for $\alpha = 1.1$ and $\Delta = \Delta_0$

for the reference value for α. When Δ becomes smaller than the reference value, the number of hyper-planes generated decreases w.r.t. our reference case ($S = 2 < 5$). When on the other hand Δ becomes larger than the reference value, the number of hyper-planes generated increases w.r.t. our reference case ($S = 19 > 5$).

Discussion

It is important to realize that there shouldn't be too many hyper-planes. Indeed, the ideal number of separators results from the classical trade-off between the *robustness* and the *sensitivity* of a classifier. In the specific case of our multi-linear classifier this compromise can be explained as follows. The more hyper-planes there are, the larger the number of regions of the partition of the d dimensional space. This means that the number of training data points that are likely to fall in a single region becomes smaller. This means that the attribution of the class label to the different regions is going to be more and more influenced by isolated training data points. This makes the classifier on the one hand more sensitive, but on the other hand at the same time also less robust. This duality is explained

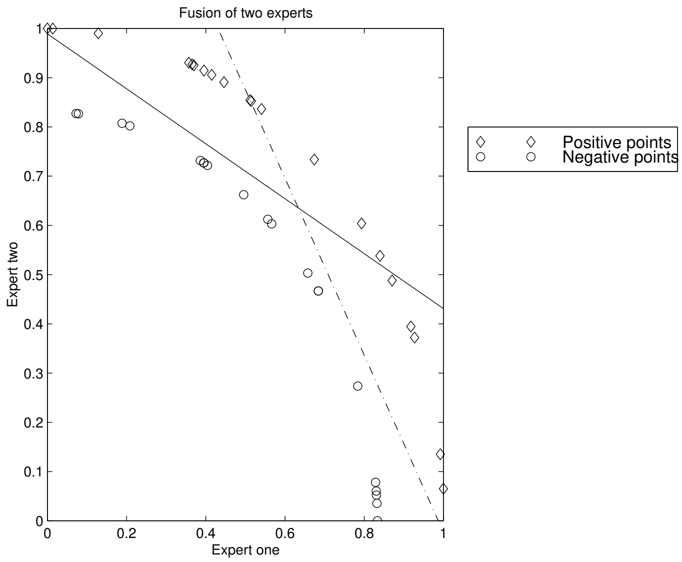

Figure A.7: Set of hyper-planes generated for $\alpha = 1$ and $\Delta = 0.25 * \Delta_0$

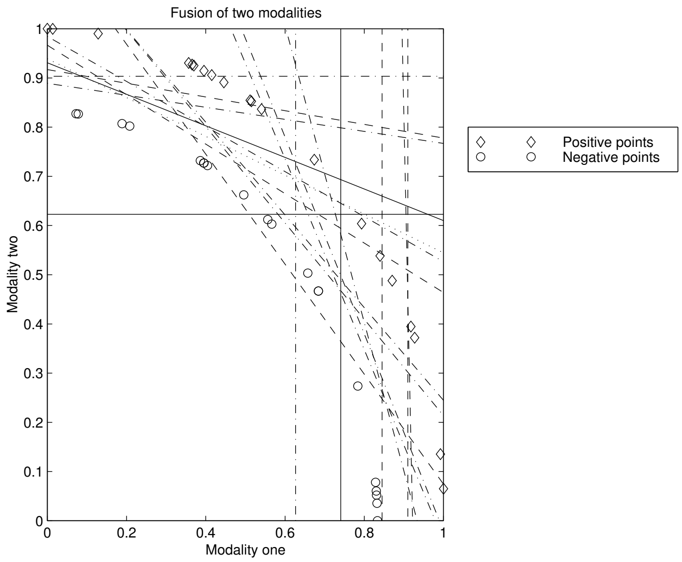

Figure A.8: Set of hyper-planes generated for $\alpha = 1$ and $\Delta = 4 * \Delta_0$

below:

Greater sensitivity If those isolated training data points are representative for the real (unknown) characteristics of the rest of the population then the classifier did a good job on capturing this level of detail.

Smaller robustness If those isolated training data points are outliers who's characteristics are only marginal related to those of the rest of the population, then the classifier is going to accumulate a lot of errors.

This duality can also be expressed in terms of "over-training" and "under-training" the classifier. Over-training the classifier means that we are in fact modeling the noise on the training data, which leads to a bad generalization capability. This happens when we generate a lot of hyper-planes but the isolated training points are in fact outliers or extreme values. Under-training the classifier means that we are not modeling enough significant variations in the training data, which means that we are generalizing too much. This happens when we generate only a small number of hyper-planes such that meaningful isolated training data points are grouped with the bulk of the training data.
This compromise can be fixed using a supplementary data set (which can be seen as a kind of validation set).

Appendix B

The iterative goal function

The iterative goal function has been defined in expression (A.1). To simplify this expression, we introduce the following notations:

$$\tilde{\boldsymbol{w}}^s = \begin{pmatrix} \boldsymbol{w}^s \\ w_0^s \end{pmatrix}, \ \tilde{\boldsymbol{a}}^k = \begin{pmatrix} \boldsymbol{a}^k \\ -1 \end{pmatrix}, \ \tilde{\boldsymbol{a}}^l = \begin{pmatrix} \boldsymbol{a}^l \\ -1 \end{pmatrix} \in \mathbb{R}^{d+1}$$

$$\text{with } \ \boldsymbol{w}^s = \begin{pmatrix} w_1^s \\ \vdots \\ w_d^s \end{pmatrix}, \ \boldsymbol{a}^k = \begin{pmatrix} a_1^k \\ \vdots \\ a_d^k \end{pmatrix}, \ \boldsymbol{a}^l = \begin{pmatrix} a_1^l \\ \vdots \\ a_d^l \end{pmatrix} \in \mathbb{R}^d, \ s \in \{1, \ldots, S\}$$

In these expressions d is the number of modalities, S is the number of hyper-planes in the set, k is the number of positive and l the number of negative points, $\tilde{\boldsymbol{w}}^s$ is the hyper-plane being added during the current iteration s.

Furthermore we split Δ_{kls} into two parts: a first part Δ_{kl} that represents the separation on each pair (k, l) obtained during the previous iterations $\{1, \ldots, s-1\}$ (note that this part is independent of the currently added hyper-plane $\tilde{\boldsymbol{w}}^s$), taking into account the "clipping" effect induced by the max operator in the expression (A.1) and a second part δ_{kls} that represents the dynamic portion of the separation, added during the current iteration s (this part depends of course of the current hyper-plane $\tilde{\boldsymbol{w}}^s$). This convention can be written as follows:

$$\Delta_{kls}(\tilde{\boldsymbol{w}}^s) = \Delta_{kl} + \delta_{kls}(\tilde{\boldsymbol{w}}^s)$$

Using these new notations, the iterative goal function for the iteration s becomes:

$$\text{goal}_{iter} = \sum_{k,l} \min\{\Delta, \Delta_{kl} + \delta_{kls}(\tilde{\boldsymbol{w}}^s)\},$$

$$\text{where} \quad \delta_{kls}(\tilde{\boldsymbol{w}}^s) = \max\{0, \min\{\alpha.\delta_k(\tilde{\boldsymbol{w}}^s), -\delta_l(\tilde{\boldsymbol{w}}^s)\}\},$$

$$\text{with} \quad \delta_x(\tilde{\boldsymbol{w}}^s) = \tilde{\boldsymbol{w}}^s.\tilde{\boldsymbol{a}}^x, x \in \{k, l\}.$$

Rewriting this goal function to eliminate the max and the min gives:

$$\text{goal}_{iter} = \left(\sum_{\substack{k,l \\ s.t.\ \Delta_{kls} \geq \Delta}} \Delta + \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta}} \Delta_{kl} + \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta}} \delta_{kls}(\tilde{\boldsymbol{w}}^s) \right)$$

Replacing the first two terms (which, as already has been stated, are independent of $\tilde{\boldsymbol{w}}^s$) respectively by A and B, the expression becomes:

$$\text{goal}_{iter} = \left(A + B + \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta}} \delta_{kls}(\tilde{\boldsymbol{w}}^s) \right)$$

Replacing $\delta_{kls}(\tilde{\boldsymbol{w}}^s)$ by its expression and eliminating the max operator leads to the following expression:

$$\text{goal}_{iter} = \left(A + B + \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta \\ \text{and } \min\{\alpha.\delta_k, -\delta_l\} > 0}} \min\{\alpha.\delta_k(\tilde{\boldsymbol{w}}^s), -\delta_l(\tilde{\boldsymbol{w}}^s)\} \right)$$

Eliminating the min operator gives us then:

$$\text{goal}_{iter} = \left(A + B + \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta \\ \text{and } 0 < \alpha.\delta_k \leq -\delta_l}} \alpha.\delta_k(\tilde{\boldsymbol{w}}^s) + \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta \\ \text{and } 0 < -\delta_l < \alpha.\delta_k}} -\delta_l(\tilde{\boldsymbol{w}}^s) \right)$$

Deriving with respect to $\tilde{\boldsymbol{w}}^s$ yields:

$$\nabla_{iter}^{s} = \left(0 + 0 + \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta \\ \text{and } 0 < \alpha.\delta_k \leq -\delta_l}} \alpha.\tilde{\boldsymbol{a}}^k + \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta \\ \text{and } 0 < -\delta_l < \alpha.\delta_k}} -\tilde{\boldsymbol{a}}^l \right)$$

This gradient can then be rewritten in its final form as:

$$\nabla_{iter}^{s} = \left(\sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta \\ \text{and } 0 < \alpha.\delta_k \leq -\delta_l}} \alpha.\tilde{\boldsymbol{a}}^k - \sum_{\substack{k,l \\ s.t.\ \Delta_{kls} < \Delta \\ \text{and } 0 < -\delta_l < \alpha.\delta_k}} \tilde{\boldsymbol{a}}^l \right)$$

Appendix C

The global goal function

The global goal function has been defined in expression (A.4). To simplify this expression, we introduce the following new notations:

$$\tilde{\boldsymbol{w}}^S = \begin{pmatrix} \tilde{\boldsymbol{w}}^1 \\ \vdots \\ \tilde{\boldsymbol{w}}^S \end{pmatrix} \in \mathbb{R}^{(d+1).S}$$

$$\text{where} \quad \tilde{\boldsymbol{w}}^s = \begin{pmatrix} \boldsymbol{w}^s \\ w_0^s \end{pmatrix}, \; \tilde{\boldsymbol{a}}^k = \begin{pmatrix} \boldsymbol{a}^k \\ -1 \end{pmatrix}, \; \tilde{\boldsymbol{a}}^l = \begin{pmatrix} \boldsymbol{a}^l \\ -1 \end{pmatrix} \in \mathbb{R}^{d+1}$$

$$\text{and} \quad \boldsymbol{w}^s = \begin{pmatrix} w_1^s \\ \vdots \\ w_d^s \end{pmatrix}, \; \boldsymbol{a}^k = \begin{pmatrix} a_1^k \\ \vdots \\ a_d^k \end{pmatrix}, \; \boldsymbol{a}^l = \begin{pmatrix} a_1^l \\ \vdots \\ a_d^l \end{pmatrix} \in \mathbb{R}^d, \; \text{with } s \in \{1, \ldots, S\}$$

In these expressions d is the number of modalities, S is the number of hyper-planes in the set, k is the number of positive and l the number of negative points, $\tilde{\boldsymbol{w}}^S$ is the vector containing all S hyper-planes.

What is particular for this global approach is that *all* hyper-planes are used at the same time (this was not the case for the iterative approach, where only the last added hyper-plane was used). With these new notations, the global goal function becomes:

$$\text{goal}_{glob} = \min_{k,l} \left(\sum_s \Delta_{kls}(\tilde{\boldsymbol{w}}^s) \right)$$

$$\text{where} \quad \Delta_{kls}(\tilde{\boldsymbol{w}}^s) = \max\{0, \min\{\alpha.\delta_k(\tilde{\boldsymbol{w}}^s), -\delta_l(\tilde{\boldsymbol{w}}^s)\}\},$$

$$\text{with} \quad \delta_x(\tilde{\boldsymbol{w}}^s) = \tilde{\boldsymbol{w}}^s.\tilde{\boldsymbol{a}}^x, x \in \{k, l\}.$$

This time however there is no need to split Δ_{kls} as we have done in the iterative approach, since in this global goal function, the separation for the pair (k, l) is calculated directly for all S hyper-planes of the set, without having to deal with the "clipping" effect of the iterative goal function. Taking this into account and rewriting the global goal function replacing Δ_{kls} by its value, gives:

$$\text{goal}_{glob} = \min_{k,l} \left(\sum_s \max\{0, \min\{\alpha.\delta_k(\tilde{\boldsymbol{w}}^s), -\delta_l(\tilde{\boldsymbol{w}}^s)\}\} \right)$$

Eliminating the max operator gives us then the following expression:

$$\text{goal}_{glob} = \min_{k,l} \left(\sum_{\substack{s \\ s.t.\ 0<\min\{\alpha.\delta_k, -\delta_l\}}} \min\{\alpha.\delta_k(\tilde{\boldsymbol{w}}^s), -\delta_l(\tilde{\boldsymbol{w}}^s)\} \right)$$

Eliminating the min operator gives:

$$\text{goal}_{glob} = \min_{k,l} \left(\sum_{\substack{s \\ s.t.\ 0<\alpha.\delta_k \leq -\delta_l}} \alpha.\delta_k(\tilde{\boldsymbol{w}}^s) + \sum_{\substack{s \\ s.t.\ 0<-\delta_l<\alpha.\delta_k}} -\delta_l(\tilde{\boldsymbol{w}}^s) \right)$$

Deriving with respect to $\tilde{\boldsymbol{w}}^s$ yields then the gradient for the pair (k, l):

$$\nabla^S_{glob} = \min_{k,l} \left(\sum_{\substack{s \\ s.t.\ 0<\alpha.\delta_k \leq -\delta_l}} \alpha.\tilde{\boldsymbol{a}}^k - \sum_{\substack{s \\ s.t.\ 0<-\delta_l<\alpha.\delta_k}} \tilde{\boldsymbol{a}}^l \right)$$

Appendix D

Proof of equivalence

To simplify the development of this proof of equivalence between two alternative problem formulations, we introduce again the following notations:

$$\tilde{\boldsymbol{x}}^S = \begin{pmatrix} \tilde{\boldsymbol{x}}^1 \\ \vdots \\ \tilde{\boldsymbol{x}}^S \end{pmatrix}, \ \tilde{\boldsymbol{w}}^S = \begin{pmatrix} \tilde{\boldsymbol{w}}^1 \\ \vdots \\ \tilde{\boldsymbol{w}}^S \end{pmatrix} \in \mathbb{R}^{(d+1).S}$$

$$\text{where} \ \ \tilde{\boldsymbol{x}}^s = \begin{pmatrix} \boldsymbol{x}^s \\ x_0^s \end{pmatrix}, \tilde{\boldsymbol{w}}^s = \begin{pmatrix} \boldsymbol{w}^s \\ w_0^s \end{pmatrix}, \ \tilde{\boldsymbol{a}}^k = \begin{pmatrix} \boldsymbol{a}^k \\ -1 \end{pmatrix}, \ \tilde{\boldsymbol{a}}^l = \begin{pmatrix} \boldsymbol{a}^l \\ -1 \end{pmatrix} \in \mathbb{R}^{d+1}$$

$$\text{and} \ \ \boldsymbol{w}^s = \begin{pmatrix} w_1^s \\ \vdots \\ w_d^s \end{pmatrix}, \ \boldsymbol{a}^k = \begin{pmatrix} a_1^k \\ \vdots \\ a_d^k \end{pmatrix}, \ \boldsymbol{a}^l = \begin{pmatrix} a_1^l \\ \vdots \\ a_d^l \end{pmatrix} \in \mathbb{R}^d, \ \text{with } s \in \{1, \ldots, S\}$$

In this expression d is the number of modalities, S is the number of hyper-planes in the set, k is the number of positive and l the number of negative points.

Using these conventions, the original problem can be stated as follows:

Using all $|N|$ global gradient vectors: $\nabla^S_{glob_1}, \ldots, \nabla^S_{glob_{|N|}} \in \mathbb{R}^{(d+1).S}$,

find an $\boldsymbol{x}^S \in \mathbb{R}^{(d+1).S}$ such that

$\forall n \in N : \boldsymbol{w}^S + \eta.\boldsymbol{x}^S$ maximizes expression (A.4) for all minimal pairs.

In this expression $|N|$ is the number of pairs with minimal separation and η is any positive number.

$\forall n \in N$, we can rewrite the global goal function as follows:

For k, l fixed, $k \in K, l \in L$:

$$\text{goal}_{glob}(\tilde{\boldsymbol{w}}^S) = \sum_{s=1}^{S} \Delta_{kls}$$

where Δ_{kls} is the same as in expression (A.2).

Rewriting this gives:

$$\text{goal}_{glob}(\tilde{\boldsymbol{w}}^S) = \sum_{s=1}^{S} \max\{0, \min\{\boldsymbol{w}^s.\boldsymbol{a}^k, -\boldsymbol{w}^s.\boldsymbol{a}^l\}\}$$

Returning to our problem, we can introduce $\boldsymbol{x}^S$ in the previous expression:

$$\text{goal}_{glob}(\tilde{\boldsymbol{w}}^S + \eta.\boldsymbol{x}^S) = \sum_{s=1}^{S} \max\{0, \min\{(\boldsymbol{w}^s + \eta.\boldsymbol{x}^s).\boldsymbol{a}^k, -(\boldsymbol{w}^s + \eta.\boldsymbol{x}^s).\boldsymbol{a}^l\}\}$$

We can eliminate the max by restricting the summation to only those hyperplanes that separate the considered minimal pair. This gives us the following:

$$\text{goal}_{glob}(\boldsymbol{w}^S + \eta.\boldsymbol{x}^S) = \sum_{s\ separates\ k,l} \min\{(\boldsymbol{w}^s + \eta.\boldsymbol{x}^s).\boldsymbol{a}^k, -(\boldsymbol{w}^s + \eta.\boldsymbol{x}^s).\boldsymbol{a}^l\}$$

Since we want to maximize the additional separation on the minimal pair introduced by going in the direction $\boldsymbol{x}^S$), maximizing the previous expression is equivalent with the following expression:

$$\text{goal}_{glob}(\boldsymbol{w}^S + \eta.\boldsymbol{x}^S)\text{ - goal}_{glob}(\boldsymbol{w}^S) =$$

$$\sum_{s\ separates\ k,l} \Big(\min\{(\boldsymbol{w}^s + \eta.\boldsymbol{x}^s).\boldsymbol{a}^k, -(\boldsymbol{w}^s + \eta.\boldsymbol{x}^s).\boldsymbol{a}^l\} - \min\{\boldsymbol{w}^s.\boldsymbol{a}^k, -\boldsymbol{w}^s.\boldsymbol{a}^l\}\Big)$$

This expression can be rewritten as:

$$\text{goal}_{glob}(\boldsymbol{w}^S + \eta.\boldsymbol{x}^S)\text{ - goal}_{glob}(\boldsymbol{w}^S) =$$

$$\sum_{s\, separates\, k,l} \Big(\min\{\boldsymbol{w}^s.\boldsymbol{a}^k + \eta.\boldsymbol{x}^s.\boldsymbol{a}^k, -\boldsymbol{w}^s.\boldsymbol{a}^l - \eta.\boldsymbol{x}^s.\boldsymbol{a}^l\} - \min\{\boldsymbol{w}^s.\boldsymbol{a}^k, -\boldsymbol{w}^s.\boldsymbol{a}^l\} \Big)$$

Since η is any positive number and since all other quantities involved are positive, the minimum of both expressions between brackets will not shift. This reduces the previous expression to:

$$\text{goal}_{glob}(\boldsymbol{w}^S + \eta.\boldsymbol{x}^S) \text{ - } \text{goal}_{glob}(\boldsymbol{w}^S) = \sum_{s\, separates\, k,l} \min\{\eta.\boldsymbol{x}^s.\boldsymbol{a}^k, -\eta.\boldsymbol{x}^s.\boldsymbol{a}^l\}$$

Since η doesn't depend on s, we can place η before the summation:

$$\text{goal}_{glob}(\boldsymbol{w}^S + \eta.\boldsymbol{x}^S) \text{ - } \text{goal}_{glob}(\boldsymbol{w}^S) = \eta. \sum_{s\, separates\, k,l} \min\{\boldsymbol{x}^s.\boldsymbol{a}^k, -\eta.\boldsymbol{x}^s.\boldsymbol{a}^l\}$$

As we want to maximize this expression and since η is positive, this is equivalent with the following:

$$\text{goal}_{glob}(\boldsymbol{w}^S + \eta.\boldsymbol{x}^S) \text{ - } \text{goal}_{glob}(\boldsymbol{w}^S) = \sum_{s\, separates\, k,l} \min\{\boldsymbol{x}^s.\boldsymbol{a}^k, -\eta.\boldsymbol{x}^s.\boldsymbol{a}^l\}$$

Posing $\boldsymbol{x}^s.\boldsymbol{a}^k = \delta_k$ and $-\boldsymbol{x}^s.\boldsymbol{a}^l = -\delta_l$, and knowing that both these quantities are strictly positive (because we only consider the hyper-planes that actually do separate k, l), we can rewrite the goal we are after as maximizing the RHS (Right Hand Side) of the previous expression. This gives us the following new formulation of the problem to be solved:

$$\text{Maximize} \sum_{\substack{s \\ s.t.\; 0<\delta_k, -\delta_l}} \min\{\boldsymbol{x}^s.\boldsymbol{a}^k, -\boldsymbol{x}^s.\boldsymbol{a}^l\}$$

Splitting the min function gives us the following:

$$\text{Maximize} \left(\sum_{\substack{s \\ s.t.\; 0<\delta_k \le -\delta_l}} \boldsymbol{x}^s.\boldsymbol{a}^k - \sum_{\substack{s \\ s.t.\; 0<-\delta_l<\delta_k}} \boldsymbol{x}^s.\boldsymbol{a}^l \right)$$

Rewriting this expression using the complete vector of hyper-planes $\boldsymbol{x}^S$, this expression becomes:

$$\text{Maximize } \boldsymbol{x}^S. \left(\sum_{\substack{s \\ s.t.\; 0<\delta_k \le -\delta_l}} \boldsymbol{a}^{k^s} - \sum_{\substack{s \\ s.t.\; 0<-\delta_l<\delta_k}} \boldsymbol{a}^{l^s} \right)$$

The expression between brackets is nothing else than the gradient $\nabla^S_{glob_n}$ of the global goal function for the considered minimal pair. This is shown in Appendix C. Using that knowledge, the previous line gives:

$$\text{Maximize} \;\; \boldsymbol{x}^S.\nabla^S_{glob_n}$$

So the original problem can be restated as follows:

$$\text{Find } \boldsymbol{x}^S \in \mathbb{R}^{(d+1).S}$$

$$\text{Such that } \forall n \in N, \;\; \boldsymbol{x}^S.\nabla^S_{glob_n} > 0 \text{ is maximal.}$$

Appendix E

Expression of the conditional probabilities

Theorem

In this appendix we show that under hypothesis h:

$$P(C_1|s_1, s_2, \dots, s_n) = \frac{1}{1 + \exp\left[-\left\{(\sum_{k=1}^{n} x_k) + x_0\right\}\right]} \tag{E.1}$$

where

$$x_k = ln\frac{P(s_k|C_1)}{P(s_k|C_2)} \tag{E.2}$$

$$x_0 = ln\frac{P(C_1)}{P(C_2)} \tag{E.3}$$

- C_1 and C_2 stand respectively for Client and Impostor
- s_k is the scalar score related to the $k-$th expert
- Hypothesis h is composed of two sub-hypotheses $h1$ and $h2$, which are defined as follows:

$$h1 = P(s_1, s_2, \dots, s_n|C_1) = \prod_{k}^{n} P(s_k|C_1) \tag{E.4}$$

$$h2 = P(s_1, s_2, \dots, s_n|C_2) = \prod_{k}^{n} P(s_k|C_2) \tag{E.5}$$

Proof

$$P_{C_1} = P(C_1|s_1, \dots, s_n) = \frac{P(s_1, \dots, s_n|C_1).P(C_1)}{P(s_1, \dots, s_n)} \tag{E.6}$$

$$P_{C_1} = \frac{P(s_1, \dots, s_n|C_1).P(C_1)}{P(s_1, \dots, s_n|C_1).P(C_1) + P(s_1, \dots, s_n|C_2).P(C_2)} \tag{E.7}$$

$$P_{C_1} = \frac{1}{1 + \frac{1}{D}} \tag{E.8}$$

where

$$D = \frac{P(s_1, \dots, s_n|C_1).P(C_1)}{P(s_1, \dots, s_n|C_2).P(C_2)} \tag{E.9}$$

$$D \stackrel{h}{=} \frac{P(s_1|C_1)}{P(s_1|C_2)}. \dots .\frac{P(s_n|C_1)}{P(s_n|C_2)}.\frac{P(C_1)}{P(C_2)} \tag{E.10}$$

and the announced result is obtained by substituting:

$$\frac{P(C_1)}{P(C_2)} = \exp\left[x_0\right] \tag{E.11}$$

$$\frac{P(s_k|C_1)}{P(s_k|C_2)} = \exp\left[x_k\right] \tag{E.12}$$

Corrolarium

It is very easy to extend the previous theorem, which was valid for the class of clients, to the class of impostors. This extension can be formalized, under the same hypothesis h and with the same conventions as in the case of the theorem, as follows:

$$P(C_2|s_1, s_2, \dots, s_n) = \frac{1}{1 + \exp\left[+ \{(\sum_{k=1}^{n} x_k) + x_0\}\right]} \tag{E.13}$$

The proof of this corrolarium is analogous to the proof of the theorem and very straightforward. To explain the change of sign in the exponential, it is sufficient to see that

$$ln\frac{P(s_k|C_2)}{P(s_k|C_1)} = -ln\frac{P(s_k|C_1)}{P(s_k|C_2)} \tag{E.14}$$

and

$$ln\frac{P(C_2)}{P(C_1)} = -ln\frac{P(C_1)}{P(C_2)} \tag{E.15}$$

And of course, it can be easily shown that

$$P(C_1|s_1,\ldots,s_n) + P(C_2|s_1,s_2,\ldots,s_n) = 1 \tag{E.16}$$

Appendix F

Visual interpretations

In this appendix, some typical visual representations of decision boundaries and decision mechanisms of popular classifiers are presented. They are based on a well-known bi-dimensional classification example, inspired by the important work performed in the STATLOG project and presented in [119].

Linear classifier

A typical example of the decision boundary that a linear classifier (see sections 6.3.5 and 6.3.7) induces, is given in Figure F.1.

Piece-wise linear classifier

A typical example of the decision boundary that a multi-linear classifier (see section 6.2) induces, is given in Figure F.2.

Quadratic classifier

A typical example of the decision boundary that a quadratic classifier (see section 6.3.7) induces, is given in Figure F.3.

MLP classifier

A typical example of the decision boundary that an MLP classifier (see section 6.4) induces, is given in Figure F.4.

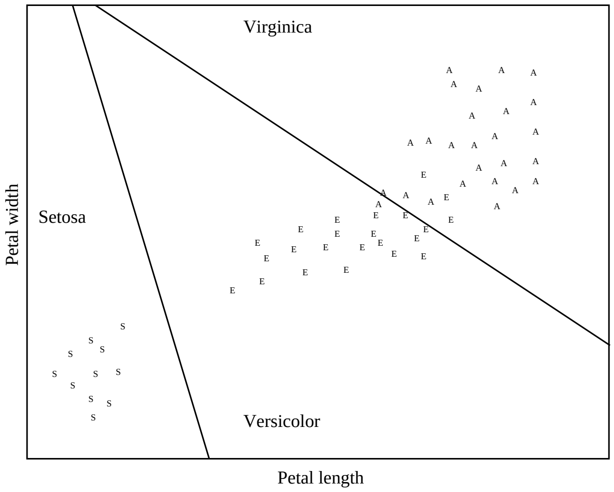

Figure F.1: Typical example of the decision boundary generated by a linear classifier.

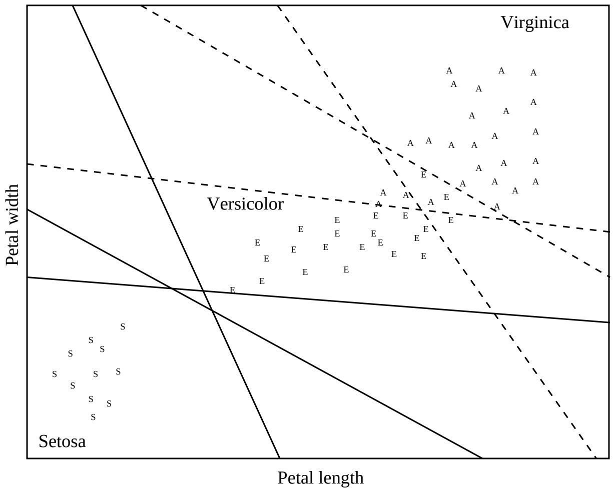

Figure F.2: Typical example of the decision boundary generated by a piecewise linear classifier.

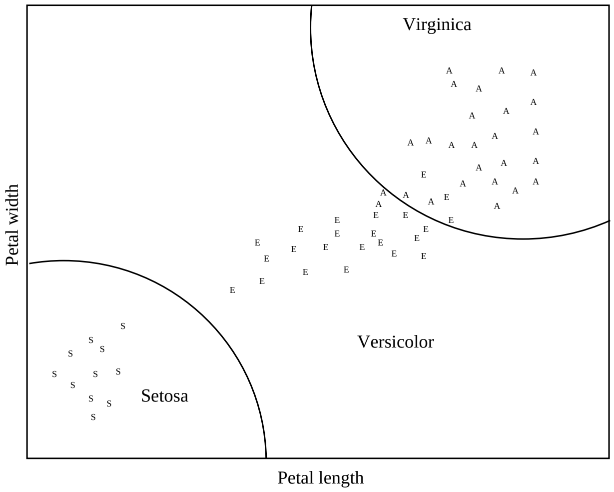

Figure F.3: Typical example of the decision boundary generated by a quadratic classifier.

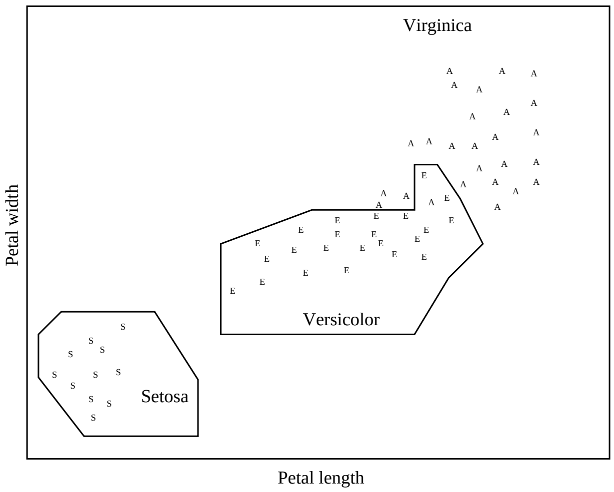

Figure F.4: Typical example of the decision boundary generated by a MLP.

k-NN classifier

A typical example of the classification mechanism that a k-NN classifier (see section 7.3), is given in Figure F.5.

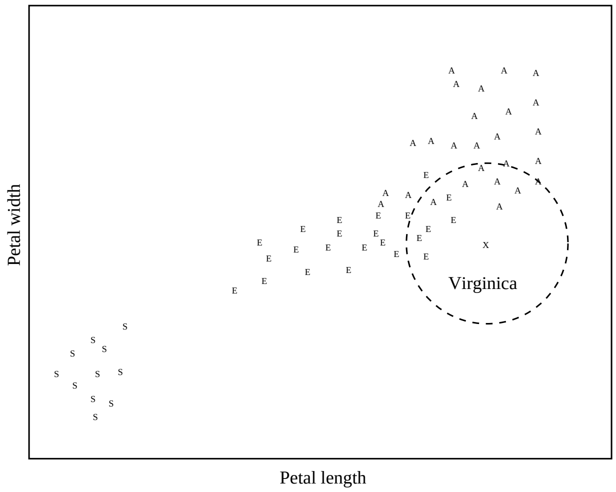

Figure F.5: Typical example of the decision mechanism used by a k-NN classifier using a Euclidean metric.

If the Euclidean metric is replaced by the Mahalanobis distance measure, then the decision mechanism changes in the most general case from searching the k nearest neighbors in a circle to looking for them in an ellipse. A typical example of the classification mechanism that this kind of Mahalanobis based classifier uses, is given in Figure F.6.

Binary tree classifier

A typical example of the decision boundary that a binary tree classifier induces (see section 7.6), is given in Figure F.7.

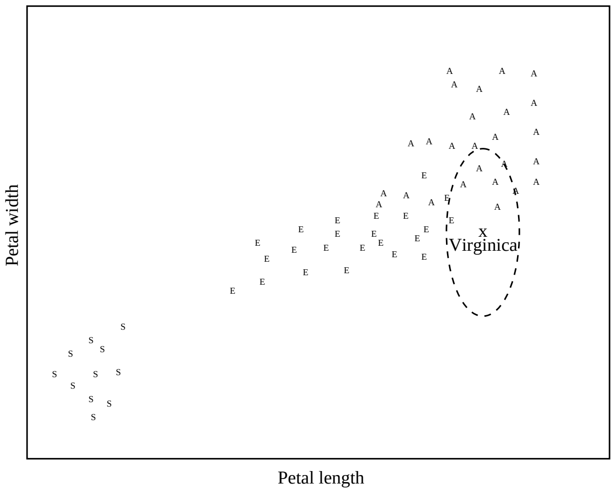

Figure F.6: Typical example of the decision boundary generated by a k-NN classifier using a Mahalanobis metric.

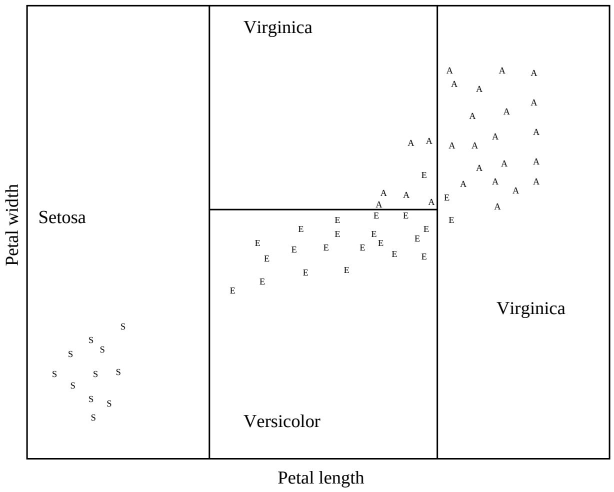

Figure F.7: Typical example of the decision boundary generated by a binary decision tree classifier.

Appendix G

Résumé

Introduction

Cette thèse traite de la vérification automatique de l'identité d'une personne coopérative, en combinant les résultats d'analyses d'images de face et de profil, et d'analyses vocales. Cette spécificité, qui a été utilisée tout au long de ce travail, a été définie dans le cadre du projet M2VTS (Multi-Modal Verification for Tele-services and Security applications) du programme ACTS de l'Union Européenne.
L'idée principale dans cette thèse est d'analyser les possibilités des techniques de fusion de données afin de combiner les résultats obtenus par les différents experts biométriques (face, profil, voix), chaque expert ayant pris une décision concernant l'identité proclamée. Dans ce travail on a volontairement omis de traiter des sujets délicats tels que l'éthique, la responsabilité ou la protection de la vie privée.
La vérification automatique de l'identité d'une personne devient de plus en plus un outil important dans plusieurs applications telles que l'accès contrôlé à des environnements (physiques et virtuels) restreints.
Un certain nombre de techniques matures, telles que des mots de passe, des cartes à bande magnétique et des codes d'identification personnels (PIN), sont déjà largement utilisées dans ce contexte, mais la seule chose réellement vérifiée est, dans le meilleur des cas, une combinaison d'une certaine *possession* (par exemple la possession de la bonne carte à bande magnétique) et d'une certaine *connaissance*, par le biais de la reproduction correcte du code alphabétique et/ou numérique. On sait que ces mécanismes très simples de contrôle (d'accès) peuvent facilement amener à des abus, induits par exemple par la perte ou le vol de la carte magnétique et du PIN cor-

respondant. De ce fait, un nouveau type de méthodes fait son apparition, basé sur ce que l'on appelle des caractéristiques ou mesures biométriques, telles que la voix, le visage, le profil, ou une autre information physiologique ou comportementale mesurable et (de préférence) propre à la personne à vérifier.

Les caractéristiques biométriques en général, et les mesures biométriques non-invasives et conviviales (voix, image) en particulier, sont très attrayantes car elles ont le grand avantage de ne pouvoir être perdues ou oubliées, et elles sont vraiment personnelles (on ne peut pas les transmettre à quelqu'un d'autre).

Lorsque l'on n'utilise qu'une seule mesure biométrique (conviviale), les résultats obtenus ne sont peut-être pas suffisamment satisfaisants. Ceci est du au fait que ces mesures biométriques conviviales ont tendance à varier avec le temps pour une même personne, et pour rendre cela encore plus problématique, l'importance même de cette variation varie d'une personne à l'autre. Ceci est particulièrement vrai pour la modalité vocale, qui montre une variabilité intra-locuteur importante. Une solution possible pour essayer de résoudre ce problème est de combiner ou fusionner les résultats de différentes modalités ou experts. Actuellement, il y a un intérêt international significatif pour ce thème de recherche. L'organisation, récemment, de deux conférences internationales sur ce sujet précis *(Audio- and Video-based Biometric Person Authentication: AVBPA)*, en est probablement la meilleure preuve.

Combiner les résultats de différents experts peut être fait en utilisant les techniques classiques de fusion de données, mais le désavantage majeur de la plupart de ces techniques est leur degré de complexité relativement élevé. Cette complexité s'exprime - entre autre - par le fait que ces méthodes ont tendance à incorporer un grand nombre de paramètres qui doivent être estimés. Si cette estimation n'est pas faite en utilisant suffisamment de données d'entraînement (i.e. si l'estimation n'est pas faite correctement), ceci place une contrainte sérieuse sur la capacité du système à généraliser correctement. Mais, actuellement, une des difficultés majeures de ce problème d'estimation est la pénurie de données d'entraînement multimodales. En effet, afin de garder un système de vérification automatique convivial, l'enregistrement d'un (nouveau) client ne doit pas prendre trop de temps, ce qui a pour conséquence directe que les données d'entraînement pour les clients sont plutôt limitées.

Afin de combler ce manque de données d'entraînement, une possibilité est de développer des méthodes de fusion (classificateurs) simples, c.a.d. des

classificateurs qui n'utilisent que très peu de paramètres.
Dans ce travail, on se limite à l'utilisation de techniques de fusion de décisions. On considèrera que tous les experts prennent leur décision locale en générant un score dans l'intervalle [0,1]. Ces scores sont une mesure de leur croyance respective de l'acceptabilité de la proclamation d'identité : plus haut est le score, plus grande est la croyance que la proclamation d'identité est authentique. Cette manière de travailler a le grand avantage de séparer la conception des experts spécialisés (ce qui est manifestement très dépendant de l'application) du problème de fusion. Ceci permet de développer des règles de fusion de décision très génériques, qui ne dépendent pas de l'application. Une autre raison de choisir la fusion de décisions à la place d'une fusion de caractéristiques est que ce choix décroît la dimensionalité du problème. Cette réduction en dimensionalité est très bénéfique, puisqu'elle va de pair avec une réduction du nombre de données d'entraînement nécessaire pour entraîner les différents modules de fusion.
Finalement, on a décidé d'implémenter une stratégie de fusion de décisions parallèle sous forme d'un problème de classification, ce qui a comme énorme avantage de pouvoir utiliser directement les méthodes du domaine de la Reconnaissance de Forme.

Méthodes utilisées

Dans ce travail, on a utilisé aussi bien des méthodes paramétriques que des méthodes non-paramétriques. Ceci peut se justifier, entre autre, par le fait que ces deux types de méthodes représentent en fait les deux approches possibles pour aborder l'inférence statistique:

1. l'inférence particulière (paramétrique), qui a comme but de créer des méthodes d'inférence statistique simples, qui peuvent être utilisées pour résoudre des problèmes de la vie réelle;

2. l'inférence générale (non-paramétrique), qui a comme but de trouver une seule méthode d'induction, pour tous les problèmes d'inférence statistique.

Méthodes paramétriques

En théorie, on utilise habituellement l'inférence statistique paramétrique lorsque l'on connaît relativement bien le problème à analyser. On connaît les lois physiques qui génèrent les propriétés stochastiques des données

et des fonctions à trouver, jusqu'à un certain nombre fini de paramètres. Estimer ces paramètres en utilisant les données est considéré comme l'essentiel de l'inférence statistique. Pourtant, en pratique, ces méthodes paramétriques doivent également être calculables. Des problèmes de calcul peuvent se présenter dans les cas de grande dimensionalité. On n'est pas confronté à ce genre de problèmes dans cette application, puisqu'on a opté pour une approche de fusion de décisions, qui réduit la dimensionalité du problème. De plus, le problème n'étant pas parfaitement défini, on a opté pour les distributions statistiques les plus simples et les plus utilisées pour estimer les distributions de probabilité sous-jacentes. Ces distributions favorites sont typiquement des membres de la famille exponentielle.
En analysant les résultats obtenus par les techniques paramétriques expérimentées d'un peu plus près, il est intéressant de voir que les meilleurs résultats sont obtenus par le classificateur Bayesien naïf, utilisant le modèle de la régression logistique. Ce modèle suppose que les distributions de probabilité sous-jacentes sont des membres de la famille exponentielle (ce qui est une contrainte très faible), mais avec les mêmes paramètres de dispersion pour les deux classes (ce qui est une contrainte très stricte, puisqu'on a démontré que, dans notre application, les différentes populations n'ont pas les mêmes paramètres de dispersion). D'un autre côté, on a également testé ce même classificateur Bayesien naïf, cette fois-ci en supposant que les distributions de probabilités sous-jacentes sont Gaussiennes (et pas un autre membre de la famille exponentielle), et en permettant aux différentes populations d'avoir des paramètres de dispersion (i.e. variances) différents. On sait également que ces suppositions ne sont pas valides, vu que l'hypothèse de normalité n'est pas satisfaite. Les résultats obtenus par ce classificateur Bayesien naïf ne sont pas aussi bons que ceux obtenus par le classificateur Bayesien naïf qui utilise le modèle de la régression logistique. Ceci suggère que, au moins dans notre application, l'hypothèse d'égalité des paramètres de dispersion pour la régression logistique, n'est pas aussi critique que celle de normalité dans le classificateur Bayesien naïf Gaussien.
De toute façon, compte tenu que pour toute méthode expérimentée les hypothèses sous-jacentes ne sont pas remplies, les résultats obtenus par les méthodes paramétriques dans cette application, sont très bons.

Méthodes non-paramétriques

En théorie, on utilise habituellement l'inférence statistique non-paramétrique quand on n'a pas d'information a priori concernant les lois statistiques

sous-jacentes au problème ou concernant la fonction qu'on aimerait approcher. En pratique, on pourrait également opter pour des méthodes non-paramétriques lorsque les méthodes paramétriques ne sont pas calculables. L'avantage majeur des méthodes non-paramétriques est qu'on ne doit pas présupposer une certaine distribution pour les distributions de probabilité sous-jacentes. Cet avantage est en même temps le plus gros inconvénient puisque, à performances égales, ces méthodes non-paramétriques ont tendance à avoir besoin de plus de données (d'entraînement) que des méthodes paramétriques. Ceci s'explique facilement car, dans le premier cas il faut estimer toute la distribution, tandis que dans le deuxième cas il suffit d'estimer quelques paramètres d'une distribution préchoisie.
En analysant de plus près les résultats obtenus par les techniques non-paramétriques expérimentées, on voit que les meilleurs résultats TER (Total Error Rate) sont obtenus par le classificateur 1-PPV classique (ou simplement PPV: Plus Proche Voisin) et par la simple méthode de vote unanime (ET). Ceci est au moins partiellement du à l'application typique avec laquelle on travaille. En effet, ce genre d'applications génère toujours plus de données imposteur que de données client. Ceci veut dire que les méthodes qui minimisent le FAR (False Acceptance Rate) auront toujours un bon TER (au moins avec notre définition du TER, qui n'est rien d'autre que la somme des FAR et FRR (False Rejection Rate), pondérées par le nombre d'exemples respectifs qui a servi à calculer ces taux d'erreurs). La méthode "ET" est typiquement une méthode de ce type, car elle exige que tous les experts décident que la personne testée soit un client, avant d'accepter la proclamation d'identité. Les bons résultats en TER du classificateur PPV classique sont également facile à comprendre, vu que le nombre d'exemples imposteur est si grand que la probabilité de classer comme client une observation inconnue, qui tomberait près de la frontière entre les deux populations (ce qui pourrait amener a une augmentation du FAR), est négligeable. Et, dans le même contexte, ceci explique également pourquoi les résultats TER deviennent moins bons lorsque le nombre d'exemples imposteur diminue.

Conclusions

Dans cette thèse, on a montré que le problème de la vérification automatique d'identité d'une personne peut se résoudre pour une application donnée. Pour le faire, il faut utiliser une ou plusieurs modalités biométriques, mettre en oeuvre l'expertise disponible afin de développer des algorithmes de vérification basés sur les caractéristiques dérivées de ces modalités biométriques, et combiner les sorties de ces différents experts en utilisant un module de fusion robuste. Dans notre application, en utilisant la base de données M2VTS et les trois experts présentés dans ce travail, le module de fusion donnant les meilleures performances est basé sur l'utilisation d'un modèle de régression logistique. Les performances obtenues sur la base de validation, extrait de cette petite base de données, sont extrêmement bons, mais avant de généraliser il faut bien se rendre compte des limitations du travail décrit.