Università degli Studi di Cagliari

Facoltà di Ingegneria

Dipartimento di Ingegneria Elettrica ed Elettronica

Ensemble of binary classifiers: combination techniques and design issues

Tesi di dottorato di

Roberto Tronci

Tutor: Prof. Giorgio Giacinto

Scuola di Dottorato in Ingegneria dell'Informazione

Dottorato di Ricerca in Ingegneria Elettronica e Informatica

XX ciclo

Settore Scientifico Disciplinare *ING-INF/05*

Alla mia famiglia

Contents

List of Figures

List of Tables

Preface

In this thesis the problem of the combination of binary classifiers ensamble is faced. For each pattern a binary classifier (or binary expert) assigns a similarity score, and according to a decision threshold a class is assigned to the pattern (i.e., if the score is higher than the threshold the pattern is assigned to the "positive" class, otherwise to the "negative" one). An example of this kind of classifier is an authentication biometric expert, where the expert must distinguish between the "genuine" users, and the "impostor" users. The combination of different experts is currently investigated by researchers to increase the reliability of the decision. Thus in this thesis the following two aspects are investigated: a score "selection" methodology, and diversity measures of ensemble effectiveness.

In particular, a theory on ideal score selection has been developed, and a number of selection techniques based on it have been deployed. Moreover some of them are based on the use of classifier as a selection support, thus different use of these classifier is analyzed.

The influence of the characteristics of the individual experts to the final performance of the combined experts have been investigated. To this end some measures based on the characteristics of the individual experts were developed to evaluate the ensemble effectiveness. The aim of these measures is to choose which of the individual experts from a bag of experts have to be used in the combination.

Finally the methodologies developed where extensively tested on biometric datasets.

Outlook of this thesis

This thesis is organised as follows.

In Chapter 1 an introduction on Pattern Recognition is outlined. Moreover the problem of binary classifiers (or experts) is faced, in particular the problem of the score based binary classifiers that is treated in this thesis. Finally an introduction on biometric authentication is given as it rappresents one of the most important research topics dealing with binary classifiers based on scores.

In Chapter 2 an ideal framework for score selection is given through the definition of an "ideal score selector". Moreover some properties of the "ideal score selector" are shown.

In Chapter 3 are shown the methods developed during this thesis to implement a "practice" score selection, and a possible use of a generic classifier to improve the score combination.

In Chapter 4 some performance measures and combination rules are described. In particular some measures to evaluate the ensemble effectiveness of a bag of experts are described. Moreover a description of some fixed score combination rules is given.

Finally Chapter 5 shows different experiments on the methodologies proposed on two biometric dataset. The conclusions are drawn in Chapter 6.

Chapter 1

Introduction

In this Chapter an introduction to the Pattern Recognition field is given. Afterwards the problem of binary experts is faced, in particular the case of expert that doesn't output a class label for a classified pattern. Finally a brief description of the biometric field is given as it is one of the most relevant field for binary classifiers (or binary experts).

1.1 Introduction to Pattern Recognition

Every human being is able to use his senses to recognize the world around him: for example every one of us is able to recognize a known person by its face or its voice. Pattern Recognition is the scientific discipline dealing with theories and methodologies for designing machines capable to automatically recognise "patterns" (i.e., objects) in noisy environments [1, 2]. Another definition is: Pattern Recognition studies "how machines can observe the environment, learn how to distinguish patterns of interest from their background, and make sound and reasonable decisions about the categories of the patterns" [3]. A pattern can be for example a fingerprint image, a human face, a voice signal, a text document, a fish type, a spam email etc. An example of recognising a pattern in noisy environments is finding the face of a known person in a picture. Some typical applications of Pattern Recognition are hand-written character recognition, remote-sensing image classification, people identification on the basis of biometric features such as fingerprints. At the end Pattern Recognition combines different disciplines (mathematics, statistic, physics, etc.) from the "engineer" point of view.

Statistical Pattern Recognition techniques have been the most studied and applied in practice [3]. The statistical Pattern Recognition studies and develops methodologies to create a classifier (or expert)[1] to recognize some patterns generalising the information retrieved from a set of example patterns. The learning approach can be distinguished into *supervised* and *unsupervised* learning[2]. The supervised approach learns the concepts from labeled examples: in practice for every example pattern its class label (or category) is known and the classifier is trained using these information[3]. Afterwards the classifier is used to label the unlabeled patterns, the class label is picked from those learned by the classifier. The unsupervised approach learns concepts from unlabeled data. In unsupervised classification new patterns are assigned to an unknown class, the unsupervised approach try to learn the classes by analyzing the feature of the example patterns and grouping them for similarity. In the following only the supervised approach will be taken into account.

Let be ω_k $(k = 1, \ldots, l)$ the possible classes, and C a classifier. The classifier C can be viewed as a function

$$C : \mathbb{R}^n \mapsto \Omega, \quad \Omega = \{\omega_1, \omega_2, .., \omega_l\}$$

Given a pattern x, the classifier C assigns it to a class $E(x) = \overline{\omega} \in \Omega$. Thus, given a set of labeled data $\mathcal{X} = \{x_i\}$ (with $i = 1, \ldots, m$), then the following

[1] The two terms are indistinctly used in the Pattern Recognition field, usually the term "classifier" is used if it associates a class to a pattern, while the term "expert" is generally used in a more generic context

[2] In literature exists also a *semi-supervised* that its a mixture of the two previous learning approach

[3] Usually the set of example patterns is named *training set*

set is available

$$\mathrm{D} = \{(x_1, L(x_1)), (x_2, L(x_2)), \ldots, (x_m, L(x_m))\}$$

where $L : \mathbb{R}^n \mapsto \Omega$ is an unknown function that assigns a pattern x_i to its true belonging class ω_{x_i}. The aim of a supervised learning algorithms is to build a classifier C that can correctly classify new patterns. In practice C is trained over D to replicate L. After that C has been trained, it can be used to classify a generic new pattern $y \notin \mathrm{D}$. In doing so, the classifier C computes *supports* $\mu_k(y)$ that rappresent how much y belongs to the class ω_k according to C. Afterwards, a decision rule is applied on the supports $\mu_k(y)$, in order to assign a label to the pattern y. These supports can be simply a score which indicates a "similarity" or they can be an estimation of the posterior probability $P(\omega_k|y)$ that y belongs to the class ω_k. Given a set of unlabeled data $\mathcal{Y} = \{y_p\}$ $(p = 1, \ldots, t)$, C assigns to all the patterns $\mathcal{Y}$ a class $\overline{\omega}_k \in \Omega$. If to a pattern y_p the assigned class $\overline{\omega}_k$ is equal to the true belonging class ω_{y_p}, the pattern is correctly classified. Thus, the performance of an expert C is measured through the accuracy who measures the percentage of the patterns correctly classified over the total number of the patterns classified by C.

$$Accuracy = \frac{num_{cor}}{num_{cor} + num_{mis}}$$

where num_{cor} is the number of the patterns correctly classified, and num_{mis} is the number of the patterns misclassified.

In Pattern Recognition the combination of different classifiers had been vastly studied in the last years [4, 5, 6, 7, 8, 9, 10, 11, 12], the combination of different classifiers is named Multiple Classifier System (MCS). Approaches based on ensemble of classifiers are widely used in many applications as they avoid the choice of the "best" classifier, and typically provide better performance than those provided by individual classifiers [5]. Ensemble approaches also allow "combining" classifiers based on different input sources, so that complementary information can be exploited, and the resulting classifier is robust with respect to noise [5]. Two main kind of classifier combination exist: the "fusion" approach, and the "selection" approach. The "fusion" approach combines the classes $\overline{\omega}_k^j$ assigned to a pattern y_p from an ensemble of classifiers E_j into a "new" class $\overline{\omega}_k^{fus}$, this class can be different from the classes $\overline{\omega}_k^j$ [5]. The "selection" approach combines the classes $\overline{\omega}_k^j$ assigned to a pattern y_p from an ensemble of classifiers E_j selecting a class among those classes, $\overline{\omega}_k^{sel} \in \overline{\omega}_k^j$ [13, 14].

1.2 Binary experts

One interesting type of Pattern Recognition problem is when only two class are involved. They can be two real class, or a single class vs "the rest of the world"[4]. Examples of binary (two-class) classification problem are biometric authentication, spam filtering, medical test, intrusion detection etc.

In the case of binary classification problem the two class are usually denoted with the terms *positive* class **p**, and *negative* class **n**. Thus, given a binary experts[5] four possible outcomes can be obtained. If a *positive* pattern is classified with a *positive* class, then it is *true positive* (TP); otherwise if it is classified with a *negative* class, then it is *false negative* (FN). If a *negative* pattern is classified with a *negative* class, then it is *true negative* (TN); otherwise if it is classified with a *positive* class, then it is *false positive* (FP).

$$true\ positive\ rate = \frac{positives\ correctly\ classified}{total\ positivies}$$

$$false\ positive\ rate = \frac{negatives\ incorrectly\ classified}{total\ negatives}$$

[4] This problem is also known as One-class classification problem, for this problem different techniques exists from those used for binary classifier

[5] In this case the term "expert" is preferred to "classifier" as in this situation a class is not always directly assigned to the pattern by the expert

it is worth noting that

$$true\ positive\ rate\ +\ false\ negative\ rate\ =\ 1$$

$$true\ negative\ rate\ +\ false\ positive\ rate\ =\ 1$$

Depending from the disciplines object of the study, different terminology are used. In some Pattern Recognition application the terms "hit rate" (*true positive* rate), and "false alarm rate" (*false positive* rate) are used. In biometric authentication the terms "false non matching rate" (*false negative* rate), "false matching rate" (*false positive* rate), and "true matching rate" or "genuine matching rate" (*true positive* rate) are used, as they are obtained by comparing the unknown pattern y_p to a known pattern x_i to verify if their identities match (if so, the pattern correspond to a "genuine" user).

For the binary experts another way to evaluate and compare the performance is by means of the Receiver Operating Characteristic curve (ROC) and the Area Under the ROC Curve (AUC) [15]. The ROC curve is a plot of the *false positive* rate against the *true positive* rate. The study of the relation between ROC curves and experts, and the combination of experts to improve the ROC is also known as ROC analysis [16].

Also for the binary experts the MCS are used to improve performance. A binary expert, as a generic expert, for a given pattern y_p can output

1. a class (i.e., *positive* or *negative*)[6]

2. a posterior probability $P(\mathbf{p}|y_p)$, and $P(\mathbf{n}|y_p) = 1 - P(\mathbf{p}|y_p)$, then the class is chosen according to these posterior probabilities

3. a support or a similarity score (usually) to the positive class, then the class is chosen according to this score

In the following the third case is considered, as it is the main subject of study of this thesis.

[6] In this case the term of classifier is usually preferred to the term expert

Let $E = \{E_1 \dots E_j \dots E_n\}$ be a set of n experts and $\mathcal{X} = \{x_i\}$ $(i = 1, \dots, m)$ be the set of patterns. Let also $f_j(\cdot)$ be the function associated to expert E_j that produces a score s_{ij} for each pattern x_i, $s_{ij} = f_j(u_i)$. Let s_j be the set of all the score produced by an expert E_j for all the patterns in $\mathcal{X}$. Let be th a decision threshold. If the score s_{ij} is higher than th then the pattern x_i is assigned to the *positive* class, otherwise is assigned to the *negative* class. This threshold is usually tuned to fulfil the requirements of the problem. Generally only two errors are taken into account the *false positive* rate (FPR), and the *false negative* rate (FNR)

$$FPR_j(th) = \int_{th}^{\infty} p(s_j|\mathbf{n})\mathrm{d}s_j = P(s_j \geq th|\mathbf{n}) \tag{1.1}$$

$$FNR_j(th) = 1 - TPR_j(th) = \int_{-\infty}^{th} p(s_j|\mathbf{p})\mathrm{d}s_j = P(s_j < th|\mathbf{p}) \tag{1.2}$$

where the probabilities of belonging to the *positive* or *negative* classes are taken into account to compute these errors as the decision threshold th varies. In this case, the higher the value of s_{ij}, the more similarly x_i belongs to the *positive* class. These errors are computed taking into account the two distinct distribution of the probabilities for the *positive* and the *negative* classes, or the two distribution of the similarity scores for the *positive* and the *negative* classes if the probabilities are not available.

Thus the ROC curve is a plot of the FPR against the TPR $(1 - FNR)$ as the threshold varies for all its possible values. An example of the ROC curve, and the ideal ROC curve are plotted in Figure 1.1. The dotted line in Figure 1.1 rappresents a situation where the *true positive* rate is equal to the *false positive* rate, this means that the two distributions of the scores of the *positive* and *negative* patterns are completely overlapped. In this case it is equivalent to a random classifier. Thus the more closer to the ideal ROC, the better the ROC. While the more closer to the "random" ROC, the worse the ROC. If a ROC curve lies beyond the "random" ROC its performance are very low, but it can be said that it has useful information, but it is applying

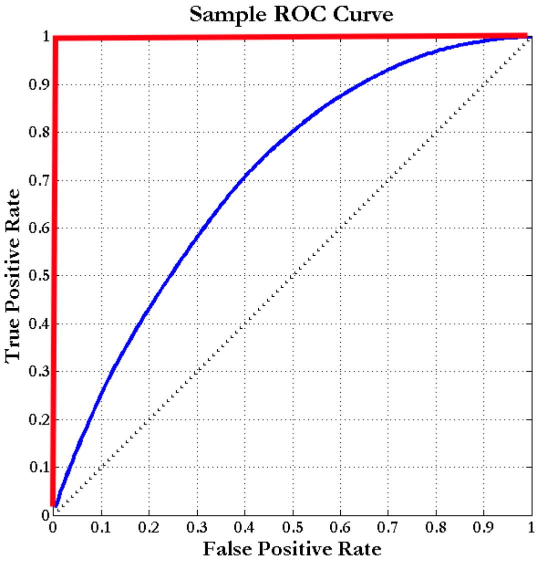


Figure 1.1: Examples of ROC curves: the ideal ROC curve (red), and a generic ROC curve (blue).

them incorrectly.

Another measure to compare two binary experts is the Area Under the ROC curve (AUC). This is a measure that summarizes the performance of the binary expert for all the values of the decision threshold.

$$AUC = \int (TPR(th)) \mathrm{d}FNR(th)$$

The area of the dotted ROC curve in Figure 1.1 is equal to 0.5, while the area of an ideal ROC curve is equal to 1. A typical ROC curve has an AUC between 0.5 and 1. Thus the higher the value of the AUC, the better the performance.

Another measure, typically used in the biometric field, is the *Equal Error Rate* (EER) that rappresent the point where the *false positive* rate and the *false negative* rate are equal. The ROC, the AUC, and the EER are summarized in Figure 1.2.

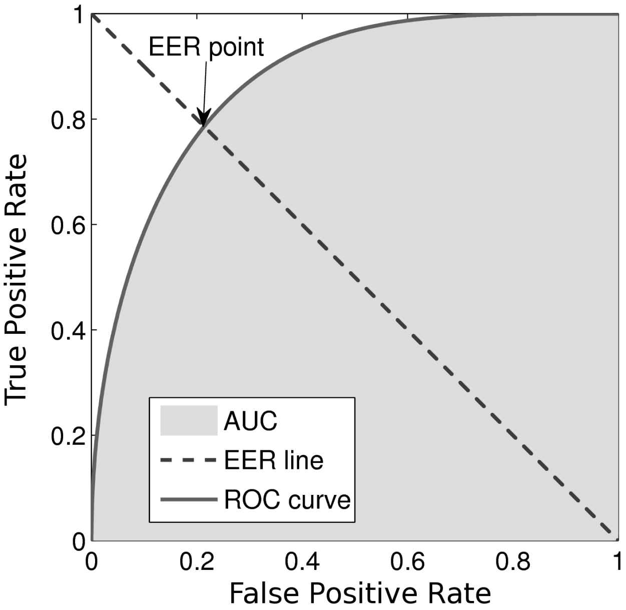


Figure 1.2: An example of a ROC curve, its AUC and its EER.

1.2.1 Area Under the ROC Curve

In ROC analysis the *Area Under the Curve* (AUC) is the most widely used measure for assessing the performance of a two-class system because it is a more discriminating measure than the accuracy [17]. The AUC can be computed by the numerical integration of the ROC curve, or by the Wilcoxon-Mann-Whitney (WMW) statistic [18]. In this thesis the WMW statistic is used to estimate the AUC as it is theoretically equivalent to the value computed by integrating the ROC curve, but in real cases (finite samples) it allows attaining a more reliable estimation of the AUC than that of the numerical integral of the ROC curve, as the integral computations depends on the numerical technique employed [19].

According to the WMW statistic, the AUC can be computed as follows. Let us divide into two sets all the scores $\{s_{ij}\}$ produced by an expert E_j for all the x_i patterns: $\{s^{pos}_{p,j}\}$, i.e. the set made up of the scores produced for the *positive* patterns, and $\{s^{neg}_{q,j}\}$, i.e. the set made up of the scores produced for the *negative* patterns.

$$AUC = \frac{\sum_{p=1}^{n_+} \sum_{q=1}^{n_-} I(s_{p,j}^{pos}, s_{q,j}^{neg})}{n_+ \cdot n_-}$$

where n_+ is the number of *positive* patterns and n_- is the number of *negative* patterns, and the function $I(s_{p,j}^{pos}, s_{q,j}^{neg})$ is[7]:

$$I(s_{p,j}^{pos}, s_{q,j}^{neg}) = \begin{cases} 1 & s_{p,j}^{pos} > s_{q,j}^{neg} \\ 0 & s_{p,j}^{pos} < s_{q,j}^{neg} \end{cases}$$

This formulation of the AUC can be also interpreted as follows: given two randomly chosen patterns, one from the set of *positive* patterns, and one from the set of *negative* patterns, the AUC is the probability $P(s_{p,j}^{pos} > s_{q,j}^{neg})$, i.e. the probability of correct pair-wise ranking [19].

1.3 The biometric problem

Nowadays the security problem is one of the hottest problems. One of the problems is to obtain a correct, and reliable verification of the identity of a person in today's networked society. However, "traditional" methods, like password, PIN (Personal Identification Number), ATM (Automatic Transaction Machine), are unreliable because a personal code (a sequence of letters or digits) can be stolen or duplicated, and used by other people for illegal aims. In this context, the biometric field is a very active research field. Its aim is to find reliable personal identification techniques based on human characteristics like face, fingerprint, retina, signature, iris, gait and so on. A biometric system assures a more reliable identification of a person, since fingerprint, face etc. are unique for each person and cannot be stolen or duplicated [20].

A biometric system can be built to face two different problems: the "authentication" (or verification) of a user, or the "recognition" (or identification) of a user. The authentication refers to the problem of confirming or

[7] for discrete values $I(s_{p,j}^{pos}, s_{q,j}^{neg}) = 0.5$ if $s_{p,j}^{pos} = s_{q,j}^{neg}$

denying a person's identity. Recognition refers to the problem of establishing a subject's identity: for example in forensic applications.

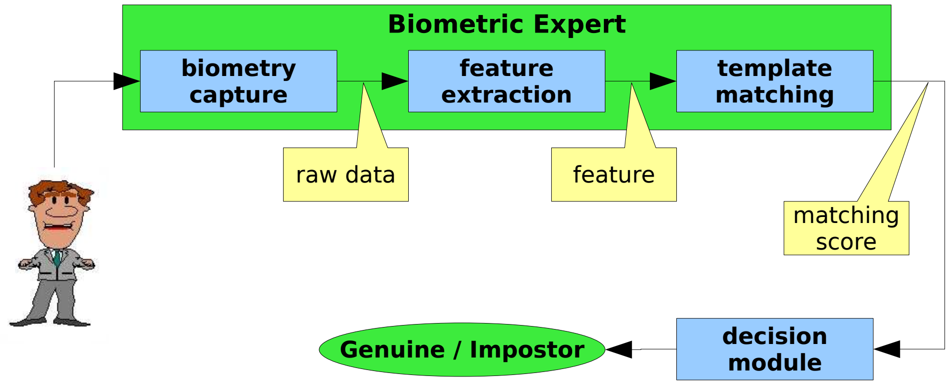


Figure 1.3: A schema of how an authentication biometric system works.

The functioning of a biometric expert for an "authentication" purpose is illustrated in Figure 1.3. At first a biometry is acquired by the biometric expert, and the raw data are extracted (e.g. the image of a fingerprint or a face). After the raw data acquired is enhanced through the use of different algorithms to improve the quality of the data and remove the noise. From this enhanced data the expert extracts the features (for example from a fingerprint image features regarding the minutiae can be extracted). When the features are available a matching algorithm is performed. This matching algorithm, with respect to the identity the user want to be authenticated, compares the features extracted to those stored as template. The output of a matcher is a score that indicates how the data acquired is similar to the template stored (i.e., a similarity score). The higher the value of a similarity score, the more similar are the data acquired to the template. Sometimes, instead of similarity score, distance scores are used, these scores indicates how much the data acquired is closer to the template. For a distance score the lower the value, the more similar are the data acquired to the template[8]. After that the biometric expert E_j outputs a score s_{ij} this is processed by a "decision module". In the decision module an acceptance threshold th is

[8] In the following only similarity scores are used.

stored: if $s_{ij} \geq th$ the user is accepted (i.e., assigned to the so-called *genuine* class), otherwise he is rejected (i.e., assigned to the so-called *impostor* class). At the score level, the performance of a biometric expert is evaluated in terms of the False Matching Rate (FMR, i.e., the percentage of impostors whose score is larger than the decision threshold) and the False Non-Matching Rate (FNMR, i.e., the percentage of genuines whose score is smaller than the decision threshold). Thus the Equations (1.1) and (1.2) can be rewritten as

$$FMR_j(th) = \int_{th}^{\infty} p(s_j|\textbf{impostor})\mathrm{d}s_j = P(s_j \geq th|\textbf{impostor}) \qquad (1.3)$$

$$FNMR_j(th) = \int_{-\infty}^{th} p(s_j|\textbf{genuine})\mathrm{d}s_j = P(s_j < th|\textbf{genuine}) \qquad (1.4)$$

then they can be reported to a Receiver Operating Characteristic (ROC) curve, where the value of *1 - FNMR* is plotted against the value of *FMR*. It is easy to see that the *genuine* class correspond to the *positive* class, and that the *impostor* class correspond to the *negative* class described for the binary experts in Section 1.2.

As it has been assessed above, in the Pattern Recognition field the combination of experts is widely used in many applications as it avoids the choice of the "best" expert, and typically provide better performance than those provided by individual experts [5]. The combination of experts also allows "fusing" experts based on different input sources, so that complementary information can be exploited, and the resulting combination is robust with respect to noise [5]. For the same reasons, in the biometric field there is an increasing interest in multi-biometrics, i.e., the combined use of different biometric traits and/or processing algorithms, as in many application the performance attained by individual sensors or processing algorithms does not provide the necessary reliability [21, 22, 23]. An example of all possible kinds of multi-biometric are illustrated in Figure 1.4.

Combination of multiple biometric systems can be performed at different representation levels, i.e, the sensor level, the feature level, the score (or

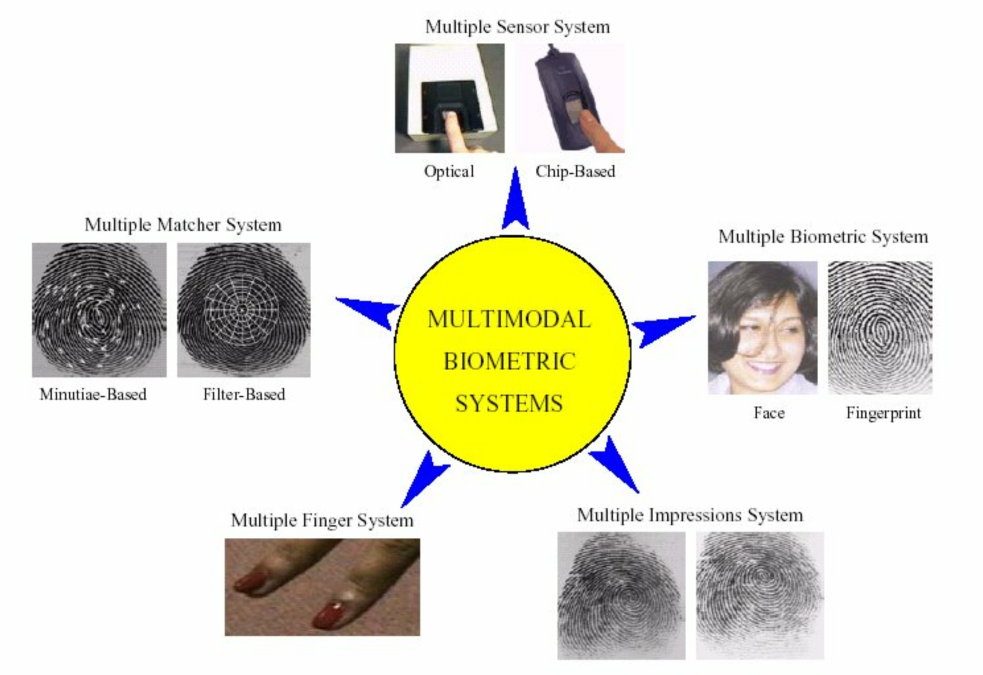


Figure 1.4: Different kind of multi-biometric systems.

rank) level, and the decision level. Sensor level combination is the combination of multiple raw data before they are subject to the feature extraction phase. This combination level can be used in those systems who capture multiple snapshot of the same biometric (e.g., multiple snapshot of the same fingerprint) and combine them into a new snapshot that is a combination of the others [24, 23]. Feature level combination combines different features sets extracted from multiple biometric sources. These features sets can be form the same feature extraction module, or they can be from different feature extraction modules, in this case same features can be incompatible [23]. Usually this level is low used because the biometric commercial systems don't provide access to the feature modules used in the their devices. Decision level combination is the simplest combination in the biometric problem as it relays only to the final class decision. Thus this combination level can be treated as generic expert combination that had been vastly treated by the Pattern Recognition literature [4, 5, 6, 7, 8, 9, 10, 11, 12].

The score level is the most used in multi-biometrics as it allows to combine completely different biometric traits in relatively easy way, and the score can be easily retrieved from a biometric system. A wide variety of score combination algorithms have been proposed in literature [23, 25, 26, 27, 22, 28, 29, 30, 31, 21]. The majority of them are based on the "fusion" combination strategy: i.e., for each user the scores from the individual biometric experts are combined through a fusion function into a "new" score, the aim of this function is to maximize the separation between the two distributions of *genuine* and *impostor* users. The main part of the research developed during this thesis was focused to the development of a "selection" combination strategy for scores [32, 25]. The aim of the score selection is to select for each user one score from the user's scores generated by the individual biometric experts. This selection strategy is described in Chapters 2, and 3.

Chapter 2

Ideal Score Selection

In the previous chapter the two class problem have been presented, and some performance measures used to evaluate the performance have been exposed. It was also pointed out that the combination of a bag of experts can improve the final performance. The aim of combining the scores is to produce "new" scores whose distributions for *positive* and *negative*[1] patterns exhibit a larger degree of separation than those produced by individual experts. Thus, if a "fusion" strategy is applied, its aim is to develop a fusion function that maximize the separation between the two distributions. In this way new scores are obtained, different from those of the individual experts. If a "selection" strategy is applied, the combined score have to be chosen among those of the individual experts [32, 25].

2.1 The Ideal Score Selector

Let $E = \{E_1, E_2, \dots E_j \dots E_N\}$ be a set of N experts and $X = \{x_i\}$ be the set of patterns, let also $f_j(\cdot)$ be the function associated to expert E_j that produces a score for each pattern x_i, $s_{ij} = f_j(x_i)$. Let s_j be the set of all the score produced by an expert E_j for all the patterns X. Let *th* be a decision

[1] For biometric issues the positive and negative class are named *genuine* and *impostor* respectively

threshold so that the patterns whose score is higher than *th* are assigned to the *positive* class, while the patterns whose score is lower than *th* are assigned to the *negative* class.

It is easy to see that, in the worst case, the largest degree of separation can be obtained by selecting for a *positive* pattern the highest of the scores assigned to that pattern by individual experts, and for a *negative* pattern the lowest of the scores assigned to that pattern by individual experts. In order to produce distribution of *positive* and *negative* scores that allows attaining lower errors than those of individual experts, the *ideal score selector* is defined as the ideal expert selector that selects the maximum score for the *positive* patterns, and the minimum score for the *negative* patterns [32, 25].

In this thesis similarity scores are used: i.e., a high value of score implies a high similarity of the pattern to the *positive* class. If distance scores are used what is described in the following can easily adapted inverting the selection of the scores (i.e., low values for *positive* patterns, and high values for *negative* patterns). It is worth noting that the same result in terms of degree of separation could be achieved not selecting the highest and the lowest score in some cases, but this is not generally true. In the following the ideal score selection algorithm based on the above goal of combination at the score level is described.

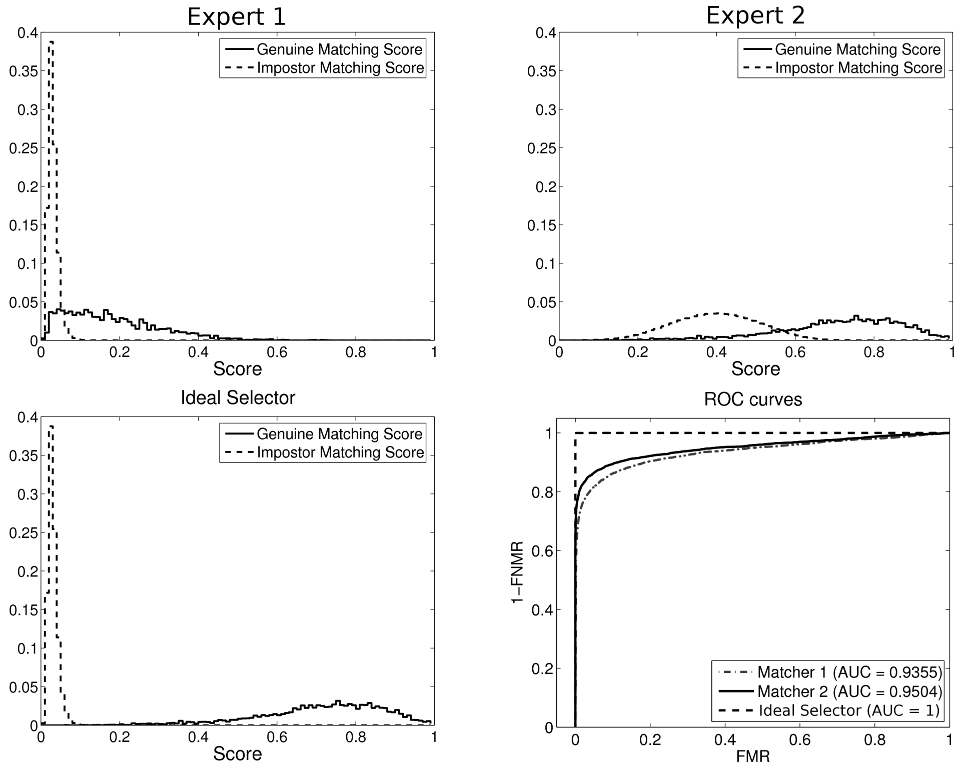
In other words, the output $s_{i,*}$ of the *ideal score selector* for the pattern x_i is computed as follows:

$$s_{i,*} = \begin{cases} \max\{s_{ij}\} & \text{if } x_i \text{ is a \textit{positive} pattern} \\ \min\{s_{ij}\} & \text{if } x_i \text{ is a \textit{negative} pattern} \end{cases} \tag{2.1}$$

An example of the result of this kind of selection is shown in figure 2.1.

Although the *ideal score selector* looks intuitively better than the individual experts, the proof is given in the following.

In the following the prove that the above defined *ideal score selector* allows attaining lower errors than those of the individual experts is given. In



Expert 1

Expert 2

Ideal Selector

ROC curves

Figure 2.1: An example of *ideal score selector* with two biometric experts from a real dataset.

particular, for any given value of FMR[2], the *ideal score selector* provides a value of FNMR lower than those provided by each individual expert.

By defining

$$s_{\max} = \max_j \left[s_j \right]$$

The following property holds for the distribution of the maximum of N random variables s_j [33]

$$P\left(s_{\max} \leq th\right) = P\left(s_1 \leq th; s_2 \leq th; \ldots\right) \leq P\left(s_j \leq th\right) \quad , \forall j$$

[2] In the following the terms FMR and FNMR are going to be used instead of FPR or FNR, as they are equivalent

By recalling the definition of FNMR

$$FNMR_j(th) = P(s_j \leq th | s_j \in \text{positive})$$

and the definition of the *ideal score selector* that always select the maximum score for *positive* patterns, the above property can be rewritten as follows for the *positive* patterns:

$$FNMR_*(th) \leq FNMR_j(th) \tag{2.2}$$

Analogously, let

$$s_{\min} = \min_j [s_j]$$

The following property holds for the distribution of the minimum of N random variables s_j [33]

$$P(s_{\min} > th) = P(s_1 > th; s_2 > th; \ldots) \leq P(s_j > th) \quad , \forall j$$

Thus, the above property can be rewritten as follows for the *negative* patterns:

$$FMR_*(th) \leq FMR_j(th) \tag{2.3}$$

It can be proved that for any value of FMR, the *ideal score selector* exhibit a value of FNMR lower than that provided by any individual expert. Given two threshold values, th' and th'', the following relationship holds

$$FMR_*(th') = FMR_j(th'')$$

then it is possible to state that

$$FNMR_*(th') \leq FNMR_j(th'') \qquad \forall j$$

From equations (1.3) and (2.3), it is easy to see that $th' \leq th''$, so that the proof can be subdivided into two cases:

1. $th' = th''$. This is the simplest case, from equation (2.2) the following relationship is obtained

$$FNMR_*(th') = FNMR_*(th'') \leq FNMR_j(th'')$$

2. $th' < th''$. Equation (1.4) implies that $FNMR_*(th') \leq FNMR_*(th'')$. By recalling equation (2.3), $FMR_*(th'') \leq FMR_j(th'')$ holds, consequently

$$FNMR_*(th') \leq FNMR_*(th'') \leq FNMR_j(th'')$$

Thus the proposed *ideal score selector* always perform better than any expert in the ensemble, and it provides a better ROC curve than the individual experts combined. By recalling that the Area Under the ROC Curve is computed as

$$AUC = \int (1 - FNMR(th))\mathrm{d}FMR(th)$$

it is possible to conclude that the AUC of the *ideal score selector* is always higher than that of any expert in the ensemble.

By using the above results, it is easy to see that the selection methodology described in equation (2.1) is the best score selection strategy. If with $\triangle$ another selection strategy is indicated, by using the above formulas it is easy to see that for a given threshold th the following relations holds:

$$FMR_*(th) \leq FMR_\triangle(th)$$

$$FNMR_*(th) \leq FNMR_\triangle(th)$$

and for a fixed value of FMR

$$FMR_*(th') = FMR_\triangle(th'')$$

we obtain that

$$FNMR_*(th') \leq FNMR_\triangle(th'')$$

2.2 Ideal Selector VS Linear Combination: combination of two experts

As it has been shown in Section 1.2.1 the Wilcoxon-Mann-Whitney statistic [18] can be used to compute the value of the AUC. Recalling, the AUC can be computed as follows:

$$AUC = \frac{\sum_{p=1}^{n_+} \sum_{q=1}^{n_-} I(s_{p,j}^{pos}, s_{q,j}^{neg})}{n_+ \cdot n_-} \tag{2.4}$$

This formulation of the AUC is also usefull to compare the AUC attained by the *ideal score selector* with the AUC attained by an optimal linear combiner. To this aim, let us consider two experts, E_1 and E_2, and all the possible pairs $\{\{s_{p,1}^{pos}, s_{q,1}^{neg}\}, \{s_{p,2}^{pos}, s_{q,2}^{neg}\}\}$ obtained from these experts. Let us divide these pairs into four subsets

$$S_{uv} = \left\{(p,q) | I(s_{p,1}^{pos}, s_{q,1}^{neg}) = u \quad and \quad I(s_{p,2}^{pos}, s_{q,2}^{neg}) = v\right\}$$

where $u, v \in \{0, 1\}$.

Thus, S_{11} is made up of all the pairs where $s_{p,1}^{pos} > s_{q,1}^{neg}$ and $s_{p,2}^{pos} > s_{q,2}^{neg}$, S_{00} is made up of all the pairs where $s_{p,1}^{pos} < s_{q,1}^{neg}$ and $s_{p,2}^{pos} < s_{q,2}^{neg}$, S_{10} is made up of all the pairs where $s_{p,1}^{pos} > s_{q,1}^{neg}$ and $s_{p,2}^{pos} < s_{q,2}^{neg}$, and S_{01} is made up of all the pairs where $s_{p,1}^{pos} < s_{q,1}^{neg}$ and $s_{p,2}^{pos} > s_{q,2}^{neg}$. These subdivision are summarized in Table 2.1.

Using the previous notation the AUC of the two experts, E_1 and E_2, can be written as follows:

$$AUC_1 = \frac{card(S_{11}) + card(S_{10})}{n_+ \cdot n_-} \ , \quad AUC_2 = \frac{card(S_{11}) + card(S_{01})}{n_+ \cdot n_-}$$

where $card(S_{uv})$ is the cardinality of the subset S_{uv}.

Subset	Pairs from Expert 1	Pairs from Expert 2
S_{11}	$s_{p,1}^{pos} > s_{q,1}^{neg}$	$s_{p,2}^{pos} > s_{q,2}^{neg}$
S_{00}	$s_{p,1}^{pos} < s_{q,1}^{neg}$	$s_{p,2}^{pos} < s_{q,2}^{neg}$
S_{10}	$s_{p,1}^{pos} > s_{q,1}^{neg}$	$s_{p,2}^{pos} < s_{q,2}^{neg}$
S_{01}	$s_{p,1}^{pos} < s_{q,1}^{neg}$	$s_{p,2}^{pos} > s_{q,2}^{neg}$

Table 2.1: Summary of the subdivision of the scores in the S_{uv} subsets.

Let us consider now this linear combination $f_{lc}(\cdot) = f_1(\cdot) + \alpha \cdot f_2(\cdot)$, where the fused output is computed as follows:

$$\xi_p = s_{p,1}^{pos} + \alpha \cdot s_{p,2}^{pos}$$
$$\eta_q = s_{q,1}^{neg} + \alpha \cdot s_{q,2}^{neg}$$

Subset	Relations	AUC contrib
S_{11}	$\xi_p > \eta_q \quad \forall\alpha$	$card(S_{11})$
S_{00}	$\xi_p < \eta_q \quad \forall\alpha$	0
S_{10}	$s_{p,1}^{pos} + \alpha \cdot s_{p,2}^{pos} > s_{q,1}^{neg} + \alpha \cdot s_{q,2}^{neg}$	α dependant max = $card(S_{10})$
S_{01}	$s_{p,1}^{pos} + \alpha \cdot s_{p,2}^{pos} > s_{q,1}^{neg} + \alpha \cdot s_{q,2}^{neg}$	α dependant max = $card(S_{01})$

Table 2.2: Summary of the contribution given from the S_{uv} subsets to the AUC using the *ideal linear combiner.*

The AUC attained by the *ideal linear combiner*, say AUC_{lc} can be computed by estimating the contribution of the pairs of outputs belonging to each of the four subsets $S_{uv}, u, v \in \{0, 1\}$ [34]. They are summarized in Table 2.2. Some cases are intuitively, some not. They are described in the follows. All the pairs belonging to S_{11} do not depend on the value of α, as $\xi_p > \eta_q$ is always verified, so that the contribution to the AUC_{lc} from the pairs belonging to S_{11} is equal to $card(S_{11})$. Similarly, it is easy to see that all the pairs belonging to S_{00} do not depend on the value of α, as $\xi_p < \eta_q$ is always verified, so that the pairs belonging S_{00} give a nil contribution to the AUC_{lc}. All the pairs belonging to S_{10} depend on α, and their contribution to the AUC_{lc} is equal to $card(S_{10})$ only if there is a value of α such that for all the pairs $s^{pos}_{p,1} + \alpha \cdot s^{pos}_{p,2} > s^{neg}_{q,1} + \alpha \cdot s^{neg}_{q,2}$. The same reasoning can be used to estimate the contribution to the AUC_{lc} of pairs in S_{01}. It is worth noting that the value of α such that the contributions of S_{10} and S_{01} are equal respectively to $card(S_{10})$ and $card(S_{01})$ may not exists. Summing up, the attainable value of AUC for the linear combination can be computed as follows:

$$AUC_{lc} \leq \frac{card(S_{11}) + card(S_{10}) + card(S_{01})}{n_+ \cdot n_-} \tag{2.5}$$

Let us now consider the *ideal score selector* defined according to equation (2.1), whose outputs are:

$$\begin{aligned} \varphi_p &= \max\left\{s^{pos}_{p,1}, s^{pos}_{p,2}\right\} \\ \psi_q &= \min\left\{s^{neg}_{q,1}, s^{neg}_{q,2}\right\} \end{aligned} \tag{2.6}$$

Subset	Relations	AUC contrib
S_{11}	$\varphi_p > \psi_q$	$card(S_{11})$
S_{00}	for some β cases: $s^{pos}_{p,1} < s^{neg}_{q,1} < s^{pos}_{p,2} < s^{neg}_{q,2}$ $s^{pos}_{p,2} < s^{neg}_{q,2} < s^{pos}_{p,1} < s^{neg}_{q,1}$	$card(\beta)$
S_{10}	$s^{neg}_{q,1} \geq s^{neg}_{q,2} \Rightarrow \max\left\{s^{pos}_{p,1}, s^{pos}_{p,2}\right\} > s^{neg}_{q,2}$ $s^{neg}_{q,1} < s^{neg}_{q,2} \Rightarrow \max\left\{s^{pos}_{p,1}, s^{pos}_{p,2}\right\} > s^{neg}_{q,1}$	$card(S_{10})$
S_{01}	$s^{neg}_{q,1} \leq s^{neg}_{q,2} \Rightarrow \max\left\{s^{pos}_{p,1}, s^{pos}_{p,2}\right\} > s^{neg}_{q,1}$ $s^{neg}_{q,1} > s^{neg}_{q,2} \Rightarrow \max\left\{s^{pos}_{p,1}, s^{pos}_{p,2}\right\} > s^{neg}_{q,2}$	$card(S_{01})$

Table 2.3: Summary of the contribution given from the S_{uv} subsets to the AUC using the *ideal score selector*.

In Table 2.3 the contributions given to the AUC by the *ideal score selection* , say AUC_{sel}. Some cases are intuitively, some not. They are described in the follows. It is easy to see that for all the pairs belonging to S_{11} the following relationship holds: $\varphi_p > \psi_q$. Thus the contribution to the AUC_{sel} of S_{11} is equal to $card(S_{11})$, as for the optimal linear combiner. By examining the pairs belonging to S_{00}, two cases have to be taken into account. One case is when φ_p and ψ_q come from the same expert. Thus it follows that

$\varphi_p < \psi_q$. The other case is when φ_p and ψ_q come from different experts. In this case, two subcases have to be considered. If $\varphi_p = s^{pos}_{p,1}$ and $\psi_q = s^{neg}_{q,2}$, then the following majority chain holds $s^{neg}_{q,1} > s^{pos}_{p,1} > s^{pos}_{p,2}$. In addition, if $s^{pos}_{p,1} > s^{neg}_{q,2}$ holds, then $\varphi_p > \psi_q$. Analogously if $\varphi_p = s^{pos}_{p,2}$ and $\psi_q = s^{neg}_{q,1}$, then the following majority chain holds $s^{neg}_{q,2} > s^{pos}_{p,2} > s^{pos}_{p,1}$. In addition, if $s^{pos}_{p,2} > s^{neg}_{q,1}$ holds, then $\varphi_p > \psi_q$. Let β be the ensemble of those pairs that verify the above relations. It is easy to see that the contribution to the AUC of the pairs belonging to β is equal to $card(\beta)$. For the subsets S_{10} and S_{01} using majority chains it is easy to see that the following relationship holds for every pair. $\varphi_p > \psi_q$. As a consequence the contribution of S_{10} and S_{01} to the AUC is always $card(S_{10}) + card(S_{01})$, while for the optimal linear combination this is an upper bound. Summing up, the AUC of the *ideal score selector* can be computed as follows:

$$AUC_{sel} = \frac{card(S_{11}) + card(S_{10}) + card(S_{01}) + card(\beta)}{n_+ \cdot n_-} \tag{2.7}$$

By comparing equations (2.5) and (2.7), it easy to see that

$$AUC_{sel} \geq AUC_{lc} \tag{2.8}$$

a comparison of the contributions given using the *ideal score selector*, and the *ideal linear combiner* are presented in Table 2.4.

Subset	Linear combination	Score selection
S_{11}	$card(S_{11})$	$card(S_{11})$
S_{00}	0	$card(\beta)$
S_{10}	depends on α, max=$card(S_{10})$	$card(S_{10})$
S_{01}	depends on α, max=$card(S_{01})$	$card(S_{01})$

Table 2.4: Contributions given to the AUC from the S_{uv} subsets when the ideal linear combination, and the ideal score selection are used.

2.3 Ideal Selector VS Linear Combination: combination of N experts

In the previous subsection the resulting AUC for ideal score selection and optimal linear combination when two experts are combined have been shown . In the following is shown what happens with this two ideal methods when more than two experts are used. Let us consider the case when three experts are available, (E_1, E_2, E_3). In this case the following subdivision into subsets is obtained:

$$S_{uvz} = \left\{(p,q) | I(s^{pos}_{p,1}, s^{neg}_{q,1}) = u \ , \quad I(s^{pos}_{p,2}, s^{neg}_{q,2}) = v \ , \quad I(s^{pos}_{p,3}, s^{neg}_{q,3}) = z\right\}$$

where $u, v, z \in \{0, 1\}$. Suppose now that the expert E_3 have been added to the pair (E_1, E_2). In this case is possible obtain the subsets S_{uvz} from S_{uv}, as it is shown by Table 2.5.

2 experts		**3 experts**
S_{11}	$\Longrightarrow$	S_{111} S_{110}
S_{00}	$\Longrightarrow$	S_{001} S_{000}
S_{10}	$\Longrightarrow$	S_{101} S_{100}
S_{01}	$\Longrightarrow$	S_{011} S_{010}

Table 2.5: How the S_{uv} subsets splits into S_{uvz} when a third z expert is added to the combination.

Now the linear combination becomes:

$$\xi_p = \alpha_1 \cdot s^{pos}_{p,1} + \alpha_2 \cdot s^{pos}_{p,2} + \alpha_3 \cdot s^{pos}_{p,3}$$
$$\eta_q = \alpha_1 \cdot s^{neg}_{q,1} + \alpha_2 \cdot s^{neg}_{q,2} + \alpha_3 \cdot s^{neg}_{q,3}$$

and the score selection changes into:

$$\varphi_p = \max\left\{s^{pos}_{p,1}, s^{pos}_{p,2}, s^{pos}_{p,3}\right\}$$
$$\psi_q = \min\left\{s^{neg}_{q,1}, s^{neg}_{q,2}, s^{neg}_{q,3}\right\}$$

Subset	**Linear combination**	**Score selection**
S_{111}	$card(S_{111})$	$card(S_{111})$
S_{110}	depends on $\{\alpha_1, \alpha_2, \alpha_3\}$, max=$card(S_{110})$	$card(S_{110})$
S_{001}	depends on $\{\alpha_1, \alpha_2, \alpha_3\}$, max=$card(S_{001})$	$card(S_{001})$
S_{000}	0	$card(\beta')$
S_{101}	depends on $\{\alpha_1, \alpha_2, \alpha_3\}$, max=$card(S_{101})$	$card(S_{101})$
S_{100}	depends on $\{\alpha_1, \alpha_2, \alpha_3\}$, max=$card(S_{100})$	$card(S_{100})$
S_{011}	depends on $\{\alpha_1, \alpha_2, \alpha_3\}$, max=$card(S_{011})$	$card(S_{011})$
S_{010}	depends on $\{\alpha_1, \alpha_2, \alpha_3\}$, max=$card(S_{010})$	$card(S_{010})$

Table 2.6: Contributions given to the AUC from the S_{uvz} subsets when the ideal linear combination, and the ideal score selection are used.

In Table 2.6 the contributions given to the AUC by the ideal linear combination are summarized, and the ideal score selection. Some cases are intuitively, some not. Let us describe them. For the ideal linear combination the contribution given to the AUC from the subset S_{101} (S_{011}) depends from $\{\alpha_1, \alpha_2, \alpha_3\}$ only if doesn't exists an $\{\alpha_1, \alpha_2\}$ that allow to fully recover the subset S_{10} (S_{01}) considering only the first two experts, in this case α_3 is meaningless for this combination. The other cases depends on $\{\alpha_1, \alpha_2, \alpha_3\}$, but S_{111} and S_{000}. When the *ideal score selector* is considered, it is easy to see what happens for the subsets that comes from S_{11}, S_{10}, and S_{01}. For the subset S_{00} and two experts, as already showed, some β cases could be present and they can be recovered. When S_{00} splits into S_{000} and S_{001}, also β splits: say $\beta_{(000)}$ those who go with S_{000}, and $\beta_{(001)}$ those who go with S_{001}. The subset S_{001} is fully recovered as $s^{pos}_{p,3} > s^{neg}_{q,3}$. While for S_{000} exist some β' cases so that $\max\left(s^{pos}_{p,1}, s^{pos}_{p,2}, s^{pos}_{p,3}\right) > \min\left(s^{neg}_{q,1}, s^{neg}_{q,2}, s^{neg}_{q,3}\right)$. It is worth noting that $\beta_{(000)} \subseteq \beta'$.

Thus, for three experts, the AUC obtained with ideal linear combination and ideal score selection in the following way can be written as follows:

$$AUC_{lc} \leq \frac{card(S_{111} \oplus S_{110} \oplus S_{001} \oplus S_{101} \oplus S_{100} \oplus S_{011} \oplus S_{010})}{n_{+} \cdot n_{-}} \tag{2.9}$$

$$AUC_{sel} = \frac{card(S_{111} \oplus S_{110} \oplus S_{001} \oplus \beta' \oplus S_{101} \oplus S_{100} \oplus S_{011} \oplus S_{010})}{n_{+} \cdot n_{-}} \tag{2.10}$$

$$= \frac{card(S_{11} \oplus S_{001} \oplus \beta' \oplus S_{10} \oplus S_{01})}{n_{+} \cdot n_{-}} \tag{2.11}$$

This reasoning can be iteratively repeated for k experts. It is easy to see that the following relations hold:

$$AUC_{lc}^{(2)} \lesseqgtr AUC_{lc}^{(3)} \lesseqgtr \ldots \lesseqgtr AUC_{lc}^{(k-1)} \lesseqgtr AUC_{lc}^{(k)}$$

$$AUC_{sel}^{(2)} \leq AUC_{sel}^{(3)} \leq \ldots \leq AUC_{sel}^{(k-1)} \leq AUC_{sel}^{(k)}$$

where $AUC_{*}^{(j)}$ is the AUC obtained combining j experts. Thus, when you combine more than two experts with the ideal score selector, what you can obtain is an increase in performance in terms of AUC. If you combine through the ideal linear combination you could obtain a worst AUC than using only two experts as there is an uncertainty due to the increase of the degree of freedom of the combination (i.e., the number of the α).

Chapter 3

Score Selection

In Chapter 2 has been defined the *ideal score selector*: an ideal methodology to combine the scores from different experts using a "selection" strategy. Thus the *ideal score selector* selects the maximum score value if the pattern belongs to the *positive* class, or selects the minimum score value if the pattern belongs to the *negative* class. The *ideal score selector* is based on the knowledge of the state of nature of the pattern (i.e., if the pattern belongs to the *positive* or to the *negative* class), but in practice the state of nature is unknown and it is the target of a classification task.

In this Chapter are shown the methods developed during this thesis to implement a "practice" score selection.

3.1 Error based estimation

In this section are explained two methods based on some measures whose aim is to estimate the error of wrongly assign the pattern to the *positive* or to the *negative* class. The methods described in the following are named *Dynamic Score Selection* (DSS) methods as they dynamically (i.e., for each pattern) select the scores. All these methods needs a training set to tune their parameters.

3.1.1 Minimum expected error

This methodology was presented in [32]. Given a pattern x_i, and a expert E_j for any value of $s_{ij} \in s_j$, the following relation holds

$$\int_0^{s_{ij}} p(s_j|\text{negative})\mathrm{d}s > \int_0^{s_{ij}} p(s_j|\text{positive})\mathrm{d}s \tag{3.1}$$

i.e., in the range $[0, s_{ij}]$ *negative* patterns outnumber *positive* patterns. This is usually true for a wide range of values of s_{ij}.

In order to compute the expected error in assigning a pattern to one of the classes *negative* or *positive* given the output score s_j of the $j-$ *th* expert, by setting the acceptance threshold to s_j and compute the difference

$$D_j = |FNMR(s_j) - FMR(s_j)| \tag{3.2}$$

By substituting Equations (1.3) and (1.4) in (3.2),

$$\left| \int_0^{s_{ij}} p(s_j|\text{positive})\mathrm{d}s - 1 + \int_0^{s_{ij}} p(s_j|\text{negative})\mathrm{d}s \right|$$

If $FNMR(s_j) > FMR(s_j)$, then $\int_0^{s_{ij}} p(s_j|\text{positive})\mathrm{d}s - 1 + \int_0^{s_{ij}} p(s_j|\text{negative})\mathrm{d}s = D_j > 0$. According to the assumption in Equation (3.1) the following relation is obtained

$$\int_0^{s_{ij}} p(s_j|\text{negative})\mathrm{d}s > \int_0^{s_{ij}} p(s_j|\text{positive})\mathrm{d}s = D_j + 1 - \int_0^{s_{ij}} p(s_j|\text{negative})\mathrm{d}s$$
$$\int_0^{s_{ij}} p(s_j|\text{negative})\mathrm{d}s > \tfrac{D_j+1}{2}$$
$$FMR(s_j) = 1 - \int_0^{s_{ij}} p(s_j|\text{negative})\mathrm{d}s < \tfrac{1-D_j}{2}$$

Thus it follows that by accepting the input pattern as a *positive* pattern an error smaller than $(1 - D_j)/2$ is expected. As a consequence the input pattern is likely to be a *positive* pattern.

Analogously, if $FMR(s_j) > FNMR(s_j)$, thus it follows that $FNMR(s_j) < (1 - D_j)/2$, and the input pattern is likely to be an*negative.* The highest the value of D_j computed according to Equation (3.2), the most likely the decision, as it leads to the minimum expected error. Summing up, the state of nature ω for pattern x can be estimated as follows:

1. for each expert E_j compute the value of D_j using the training set
2. let $k = \text{argmax}_j(D_j)$
3. then

$$\omega = \begin{cases} \text{positive} & \text{if} \quad FNMR(s_j) > FMR(s_j) \\ \text{negative} & \text{if} \quad FMR(s_j) > FNMR(s_j) \end{cases} \tag{3.3}$$

The difference in Equation (3.2) can be estimated by assuming Gaussian distributions for the score of *positive* patterns and *negative* patterns. Let μ_G (μ_I) and σ_G (σ_I) be the mean and the standard deviation of positive (negative) distribution estimated from the training set. By considering the first-order approximation of the integral used to compute the errors

$$P(s < s_{ij}) = \frac{1}{2}\left(1 + \frac{2}{\sqrt{\pi}}\frac{s_{ij} - \mu}{\sqrt{2}\sigma}\right) \tag{3.4}$$

Thus, substituting Equation (3.4) in Equation (3.2), the following relation is obtained:

$$D_j = |FNMR(s_j) - FMR(s_j)| =$$

$$= |P(s < s_j|\text{positive}) - P(s > s_j|\text{negative})| =$$

$$= |P(s < s_j|\text{positive}) - [1 - P(s < s_j|\text{negative})]| =$$

$$= \left|\frac{1}{2}\left(1 + \frac{2}{\sqrt{\pi}}\frac{s_j - \mu_G}{\sqrt{2}\sigma_G}\right) - 1 + \frac{1}{2}\left(1 + \frac{2}{\sqrt{\pi}}\frac{s_j - \mu_I}{\sqrt{2}\sigma_I}\right)\right| =$$

$$= \frac{1}{\sqrt{2\pi}}\left|\frac{s_j - \mu_G}{\sigma_G} + \frac{s_j - \mu_I}{\sigma_I}\right|$$

3.1.2 Relative minimum expected error

The previous method requires the type of distribution to estimate the necessary parameters for the score selection. The following method is an evolution of the previous one and uses an estimation measure different from the D_j measure described above. The estimation measure used is the *Relative Minimum Error* (RME) measure. The RME takes into account two terms: the error committed accepting a *negative* pattern, through the difference $FMR_j(-\infty) - FMR_j(s_{ij})$ (i.e., a measure of how likely x_i is a *positive* pattern), and the error committed when a *positive* pattern is rejected, through the difference $FNMR_j(\infty) - FNMR_j(s_{ij})$ (i.e., a measure of how likely x_i is a *negative* pattern). These quantities are estimated from a training set.

In detail, the *Relative Minimum Error* is computed as follows:

$$RME_{ij} = \frac{[FMR_j(-\infty) - FMR_j(s_{ij})] - [FNMR_j(\infty) - FNMR_j(s_{ij})]}{|[FMR_j(-\infty) - FMR_j(s_{ij})] + [FNMR_j(\infty) - FNMR_j(s_{ij})]|} =$$

$$= \frac{FNMR_j(s_{ij}) - FMR_j(s_{ij})}{FNMR_j(s_{ij}) + FMR_j(s_{ij})}$$

Summing up, the algorithm of *Dynamic Score Selection* is made up of the following steps:

1. Compute for each expert E_j the value of RME_{ij} for the pattern x_i
2. Estimate the most reliable state of nature for x_i by selecting the maximum value of $|RME_{ij}|$. Let $k = \text{argmax}_j(|RME_{ij}|)$
3. Select the score using RME_{ik} as

$$s_{sel} = \begin{cases} \max_j(s_{ij}) & \text{if} \quad RME_{ik} > 0 \\ \min_j(s_{ij}) & \text{if} \quad RME_{ik} < 0 \end{cases}$$

3.2 Using a classifier to estimate the state of nature

In the previous section two methods for the Dynamic Score Selection were presented. From these methods it is clear that the score selection faces the problem of the estimation of a pattern's state of nature. One way to overcome this problem could be the use of a classifier (i.e., k-Nearest Neighbour, Quadratic Bayes, Parzen Windows, etc). Through the use of a classifier it is possible to estimate both the state of nature of a pattern and the posterior probabilities of belonging to the *positive* or to the *negative* class.

So, for the estimation of the state of nature, a vector space could be built where, for each pattern, the vector components are the scores assigned to that pattern by an ensemble of experts. Thus, given an ensemble made up of N experts, for each pattern x_i, it is possible to construct the following feature vector $v_i = \{s_{i1}, \ldots s_{ij} \ldots s_{iN}\}$. Then, a classifier can be trained on this N-dimensional vector space, using a training set of scores related to *positive* and *negative* patterns. This classifier is thus used to estimate the state of nature of the patterns to be authenticated.

At this point,the outputs of a classifier can be used in different ways:

1. Using directly the state of nature of a pattern assigned by a classifier
2. Using the state of nature of a pattern assigned by a classifier to select the score through a Dynamic Score Selection
3. Using the posterior probabilities assigned to the pattern by a classifier

these methodologies are detailed in the following.

3.2.1 The direct use of a classifier's output

The direct use of the state of nature (i.e, the class) outputted by a classifier C for each pattern x_i, is likely if a threshold have been fixed, and singles fixed FMR and $FNMR$ are assigned to the combination of the bag of experts.

The advantage of this methodology is that a fixed classification in terms of *positive* and *negative* class is given, and no decision threshold have to be tuned. The disadvantage is that the use a decision threshold is usefull to fit the classification requirements in term of FMR and/or $FNMR$. Moreover if the classifier don't exhibit a classification accuracy of 100%, as usually appens in practice, the classification errors can't be "fixed" as it is possible tuning a decision threshold.

3.2.2 The use of a classifier for Dynamic Score Selection

This methodology uses the output of a classifier C to apply a Dynamic Score Selection [25]. Thus, the aim is to select one of the scores s_{ij} available for each pattern x_i using the state of nature of each pattern outputted by a classifier. After the computation of the state of nature, the pattern's score is selected according to Equation (2.1).

Summing up, the proposed algorithm for DSS is made up of the following steps:

1. Construct the N-dimensional vector space using the training scores dataset

2. Train a classifier C in the N-dimensional vector space

3. Classify the pattern to be authenticated with the classifier C

4. Select the score s_{sel} based on the output class of classifier C as follows

$$s_{sel} = \begin{cases} \max_j(s_{ij}) & \text{if} \quad class = positive \\ \min_j(s_{ij}) & \text{if} \quad class = negative \end{cases}$$

This methodology, as all the score combination algorithms, requires to tune a decision threshold.

3.2.3 The use of a classifier's posterior probabilities

This methodology can't be used with all kind of classifiers because not all classifiers computes the posterior probabilities for a pattern. In practice this methodology is less "general" than the previous two methodologies.

The posterior probabilities can be used in two ways:

- finding an optimal probability threshold to output a state of nature for the pattern, and after appling one of the two previous methodologies[1]
- using the posterior probabilities as a "fused" score

While the first point is a score selection methodology, the second is a score fusion methodology.

[1] This aspect is not treated in the experiments.

Chapter 4

Measures of performance, and Combination rules

In Chapter 1 the ROC curve have been introduced as a “graphic” performance measure, and the AUC, the FMR, and the FNMR as performance measures. The ROC and the AUC give a view of the global performance of an expert, infact they take into account all the possible values of the decision threshold. These measures give to the user an idea how good is globally the expert or the combination method used. Sometimes specific performance are required like a specific value of FMR or FNMR. In these cases it is usefull to use some measures that correspond to specific ROC points like: the Equal Error Rate (EER), the 1% FMR, the 1% FNMR, the 0% FMR, and the 0% FNMR.

The *Equal Error Rate* (EER) is the point of the ROC curve where the two errors, i.e. the FMR and the FNMR, are equal. The lower the value of EER, the better the performance of an expert. This performance measure is widely used in the biometric field to assess the performance of biometric systems [20], as it can be used to give a single performance measure. In certain biometric systems the EER is used to tune the threshold to be used in the system.

The 1% FMR is the value of the FNMR when the FMR is equal to 1%, respectively the 1% FNMR is the value of the FMR when the FNMR is equal to 1%. Same thing for the 0% FMR, and the 0% FNMR.

Another global performance measure is the d'. The d-prime (*d'*) is a measure of discrimination between the distributions of two signals, that has been proposed within the Signal Detection Theory [35] [2]. This measure has been also proposed in the biometric field to measure the separability of the distributions of genuine and impostor scores. Given the distributions of the scores produced respectively by *positive* and *negative* patterns, the *d'* is defined as

$$d' = \frac{|\mu_{pos} - \mu_{neg}|}{\sqrt{\frac{\sigma^2_{pos}}{2} + \frac{\sigma^2_{neg}}{2}}} \tag{4.1}$$

where μ_{pos} and μ_{neg} are the means of the two distributions, while σ_{pos} and σ_{neg} are the related standard deviations. It is easy to see that the larger the *d'*, the better the performance. In addition, if the scores are normally distributed, the d' can be directly related to the AUC and the ROC curve as it is shown in figure 4.

4.1 Measures of Ensemble Effectiveness

Besides the use of the previous measures as performance measures for a single expert or a combined system, they can be used also as measures of ensamble effectiveness. Thus, the combination of the performance measures of the single experts can be used to estimate the effectiveness of a bag of experts. One way is to use the mean value of the performance measure taken into account to estimate this effectiveness [36]. In the following for three of these measures (AUC, EER, and d') another way to measure the effectiveness is shown. It is worth noting that these way to measure can be easily adapted to every performance measure.

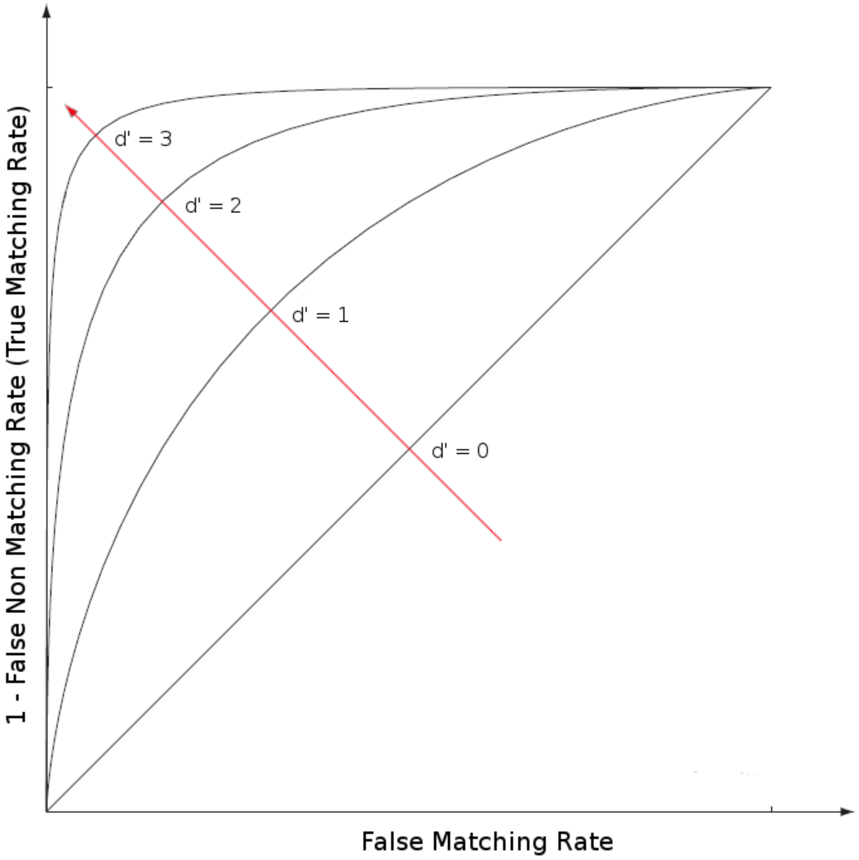


Figure 4.1: ROC curves with different d’ values derived from Gaussian distributions. The larger the d’, the better the ROC and consequently the AUC.

4.1.1 Ensemble Effectiveness based on AUC

The effectiveness of an ensemble of experts can be evaluated by the average AUC of the individual experts, the highest the average, the better the performances. However, the average value by itself does not take into account the difference in performance among the experts. Thus, we propose to weight the average AUC μ_{AUC} with the standard deviation of the AUC σ_{AUC}, according to the following definition:

$$AUC_{\delta} = \mu_{AUC} \times (1 - \tanh(\sigma_{AUC}))$$

where the hyperbolic tangent is used to "normalize" the value of σ_{AUC} between 0 and 1. This normalization is needed because the aim is to give a general formulation to this measure for every kind of performance measure (e.g., in some cases the standard deviation can be higher than 1):

$$pm_{\delta} = \mu_{pm} \times (1 - \tanh(\sigma_{pm}))$$

The choice of *tanh* it has been made after comparing it with other different normalization functions, and find it more suitable according to the extensive experiments made.

4.1.2 Ensemble Effectiveness based on EER

Another measure of the effectiveness of an ensemble is the average EER of the individual experts, the lowest the average, the better the performances. As the average value by itself does not take into account the difference in performance among the experts, we propose to weight the average EER μ_{EER} with the standard deviation of the EER σ_{EER}, according to the following definition:

$$EER_{\delta} = \mu_{EER} \times (1 - \tanh(\sigma_{EER}))$$

4.1.3 Ensemble Effectiveness based on d'

This measure is based on the average d' of the individual experts, the highest the average, the better the performances. To use an analogous formulation of those expressed above, instead of the use of the d' the following normalized value is used

$$D' = \log_b(1 + d')$$

the aim of this normalization is to take into account the nature of the d' that can assume values larger than 1, and so high variance can be present also if high d' values are taken into account. Thus by means of the normalization of the similar values of the d' this aspect can be taken into account (i.e., for high performance experts with high d' its standard deviation is less important than the same value obtained by low d' values). The value of the base b can be tuned, in the experiments a base equal to 10 is used according to the general values of d' obtained in the experiments.

In accordance to the previous reasoning, the proposed measure is based on the average d' $\mu_{D'}$ weighted by the standard deviation of the d' $\sigma_{D'}$, according to the following definition:

$$D'_{\delta} = \mu_{D'} \times (1 - \tanh(\sigma_{D'}))$$

4.2 Score Dissimilarity index

The *score dissimilarity* (SD) index [37] is an index based on the WMW formulation of the AUC, and is designed to measure the amount of improvement in AUC of the combination of an ensemble of experts with respect to the AUC of the individual experts. From this point of view, this index is a measure of the amount of AUC that can be "recovered" by exploiting the complementarity of the experts.

The SD index is computed according to the AUC formulation described in Equation (2.4). By considering two experts, E_1 and E_2, and all the possible pairs $\{\{s_{p,1}^{pos}, s_{q,1}^{neg}\}, \{s_{p,2}^{pos}, s_{q,2}^{neg}\}\}$ made up by the *positive* and the *negative* scores $\{s_{p,1}^{pos}, s_{q,1}^{neg}\}$ generated by the expert E_1, and the *positive* and the *negative* scores $\{s_{p,2}^{pos}, s_{q,2}^{neg}\}$ generated by the expert E_2, we can divide these pairs into four subsets S_{00}, S_{10}, S_{01}, and S_{11} as it have been shown in Section 2.1 and it is summarized in Table 2.1.

$$S_{uv} = \left\{(p,q)|I(s_{p,1}^{pos}, s_{q,1}^{neg}) = u \quad and \quad I(s_{p,2}^{pos}, s_{q,2}^{neg}) = v\right\} \quad u, v \in \{0,1\}$$

Remembering that using this notation, the AUC of the two individual experts can be expressed as:

$$AUC_1 = \frac{card(S_{11}) + card(S_{10})}{n_+ \cdot n_-}$$

$$AUC_2 = \frac{card(S_{11}) + card(S_{01})}{n_+ \cdot n_-}$$

where $card(S_{uv})$ is the cardinality of the subset S_{uv}.

The diversity of the two experts in terms of AUC is represented by the terms $card(S_{10})$ and $card(S_{10})$, as they are related to the pairs where the two experts disagree. Thus, SD index is defined as

$$SD = \frac{card(S_{10}) + card(S_{01})}{card(S_{11}) + card(S_{10}) + card(S_{01})} \tag{4.2}$$

this formulation differs from that proposed in [37]: i.e., the denominator have been added as a normalization factor to take into account also the performance of the experts in terms of AUC.

The higher the value of SD, the higher the maximum AUC that could be obtained by the combined scores with respect to the AUC of the individual experts. Otherwise, if the value of SD is low, the maximum AUC obtained by the combined scores is close to the AUC obtained by the individual experts. It is worth noting that actual increments of the value of the AUC depends on the combination method, and that very high values of SD are usually related to low performance experts.

In order to take into account the difference in AUC of the combined experts, a measure of ensemble effectiveness like those described previously can be obtained propose weighting the above SD value according to the standard deviation of the AUC values of the individual experts:

$$SD_\delta = SD \times (1 - \tanh(\sigma_{AUC}))$$

4.3 Combination Rules

In Chapter 1 it has pointed out that the combination of experts is used to improve the performance as they avoid the choice of the “best” expert, and typically provide better performance than those provided by individual experts [4, 5, 6, 7, 8, 9, 10, 11, 12, 23]. In the case of the experts that output a similarity score for a pattern the choice of the "best" expert is harder as the performance of the expert are not only related to the score, but also to the decision threshold needed for the application to be realized.

In the following some combination rules are described. These rules are rules usually used, especially in the biometric field, to compare the other methods with [23]. The following methods (i.e., Mean rule, Product rule, Max rule, and Min rule) where developed for a general combination of classifiers taking into account the posterior probabilities estimated for the patterns [4]. In the following they are described in the form that is generally used to combine scores.

4.3.1 Mean or Sum rule

The *Mean* (Sum) rule is applied directly to the scores produced by the set of N experts, and the resulting score is computed as follows:

$$s_{i,mean} = \frac{1}{N} \sum_{j=1}^{N} s_{ij}$$

or in the *Sum* form

$$s_{i,sum} = \sum_{j=1}^{N} s_{ij}$$

4.3.2 Product rule

Similarly to the *Mean* rule, this fusion rule is applied directly to the matching scores produced by the set of N experts:

$$s_{i,prod} = \frac{1}{N} \prod_{j=1}^{N} s_{ij}$$

4.3.3 Max rule

Similarly to the *Mean* rule, this fusion rule is applied directly to the matching scores produced by the set of N experts:

$$s_{i,max} = \max(s_{ij}) \qquad \forall j$$

4.3.4 Min rule

Similarly to the *Mean* rule, this fusion rule is applied directly to the matching scores produced by the set of N experts:

$$s_{i,min} = \min(s_{ij}) \qquad \forall j$$

4.3.5 Linear Combination by Linear Discriminant Analysis

The Linear Discriminant Analysis (LDA) can be used to compute the weights of a linear combination of the scores [2]. The goal of this fusion rule is to attain a fused score such that the within-class variations are minimised, and the between-class variations are maximised. The fused score is computed as follows:

$$s_{i,LDA} = W^t \cdot \mathbf{s}_i$$

where $\mathbf{s}_i$ is the vector of the scores assigned to the user u_i by all the experts E_j, and W^t is the transformation vector that takes into account the within and between class variations as

$$W = S_w^{-1}(\mu_{pos} - \mu_{neg})$$

where μ_{pos} is the mean of the genuine distribution, and μ_{neg} is the mean of the impostor distribution, and S_w is the within-class scatter matrix. The W transformation vector is computed using a training set.

Chapter 5

Experiments

For the experiments multimodal biometric datasets are used. The choice of this kind of dataset is driven by the fact that the biometrics are one of the most relevant application nowadays of binary classifiers. In the following the typical biometric terminology will be used, in particular, as explained in Section 1.3, the possible classes of an user (i.e., a pattern in the general case) are identified by the terms *genuine user* (i.e., the *positive* class), and *impostor user* (i.e., the *negative* class).

For the experiments the following combination methods are used[1]:

- the *ideal score selector* described in Chapter 2
- a *linear combiner* whose the "optimal" weight have been determined to maximize the AUC of the combined scores by performing an exhaustive search on the value of the combination weight α, using values between 0 and 100 with a step of 0.01.
- the Mean rule described in Chapter 4
- the Max rule described in Chapter 4
- the Min rule described in Chapter 4

[1] they are not all used in all the experiments

- the Product rule described in Chapter 4
- the linear combination based on the LDA described in Chapter 4
- the DSS based on RME described in Chapter 3
- the DSS based on classifiers described in Chapter 3
- using the posterior probabilities assigned by classifiers described in Chapter 3

As performance measures those described in Chapters 1 and 4 are used. In each experiment the measures used are specified.

5.1 Biometric datasets setup

In this section the datasets used in the experiments and some of the experiments modalities are described.

5.1.1 FVC2004 score dataset

This dataset is composed by a large number of experts (fingerprint matching algorithms) during the third Fingerprint Verification Competition[2] FVC2004 [38]. The competitors were divided into two categories *Open* and *Light*. The Open category is composed by 41 experts, while the Light category is composed by 26 experts with restricted computing and memory usage. Four databases of fingerprint images have been used in the competition: three of them were acquired with different sensors, while the fourth was created using a synthetic fingerprint generator. For each sensor and for each expert, a total of 7750 scores is available related to 2,800 authentication attempts of genuine users and to 4,950 authentication attempts of impostors. For the details on how the scores where obtained and normalised, the reader is referred to [38].

[2] Web site: *http://bias.csr.unibo.it/fvc2004/*

In order to create a training set for the LDA fusion rule, and the DSS algorithm, the set has been randomly divided into four subsets of the same size, each subset made up of 700 “genuines” and 1238 “impostors”. Each of the four subsets has been used for training, while the remaining three subsets have been used for testing. Using this partitioning of the dataset, an exhaustive multi-algorithmic combination experiment was performed: for each of the four partitioning, and for each sensor, all the possible pairs of experts are considered. Thus, for each partitioning 3,280 distinct pairs are obtained , so that the reported experiments are related to a total of 13,120 pairs of experts.

5.1.2 NIST BSSR1 score dataset

This score dataset is the Biometric Scores Set Release 1 (BSSR1) of the National Institute of Standards and Technology (NIST). This dataset is available from the NIST web site for free[3]. The dataset contains similarity scores from two face recognition systems and one fingerprint system on left and right index fingerprints. For aim of this thesis, the set containing the face and the fingerprint systems were used. In this multimodal dataset are present 517 subjects. For each subject, the set contains one score from the comparison of two right index fingerprints, one score from the comparison of two left index fingerprints, and two scores (from two separate matchers) from the comparison of two frontal faces. The non-matching scores from the full cross-comparison are also included in the dataset.

In the following the matchers are named as follows: FaceC, FaceG, FingerL, and FingerR, where C and G indicate two different face matchers, and L and R stand for the left and right fingerprint. In the experiments the scores were normalized using the *Min-Max* rule

$$s_{i,j}^{norm} = \frac{s_{i,j} - \min(s_j)}{\max(s_j) - \min(s_j)} \quad , \quad s_{i,j} \in s_j$$

[3] Web site: *http://www.itl.nist.gov/iad/894.03/biometricscores/*

As for the FVC2004 dataset, the BSSR1 dataset were subdivided in 4 parts, and one partition at a time was used as training set and the other three as test set. The performance of these individual matchers in the four test partitioning are exposed in Table 5.1.

	FaceC	**FaceG**	**FingerL**	**FingerR**
AUC	0.9891(±0.0025)	0.9828(±0.0023)	0.9629(±0.0062)	0.9823(±0.0035)
EER	0.0439(±0.0031)	0.0580(±0.0043)	0.0821(±0.0090)	0.0482(±0.0055)
d'	2.2301(±0.7404)	3.3264(±0.0688)	1.9027(±0.0588)	2.2127(±0.0383)
FMR 1%	0.0838(±0.0105)	0.1083(±0.0075)	0.1218(±0.0130)	0.0696(±0.0105)
FNMR 1%	0.2453(±0.0775)	0.4886(±0.0553)	0.7705(±0.0910)	0.6066(±0.1616)
FMR 0%	0.5519(±0.0525)	0.4030(±0.0049)	0.3469(±0.0428)	0.2037(±0.0257)
FNMR 0%	0.9086(±0.1829)	0.7745(±0.1292)	0.9362(±0.0528)	0.9985(±0.0030)

Table 5.1: Mean and standard deviation of different performance measures for the individual experts in NIST BSSR1 dataset.

5.2 Experimental results on the use of a classifier to estimate the state of nature

In Section 3.2 different methodologies on the use of a classifier to estimate the state of nature have been illustrated. This section presents a comparison between these methodologies to focus their properties.

5.2.1 Estimated state of nature vs Dynamic Score Selection

The first use of a classifier proposed in Section 3.2 is that of using the class assigned to an user x_i by a classifier C. Thus, the user is assigned to the *genuine* class or to the *impostor* class. In this case we can compute the accuracy of the classifier C, the "true positives" rate, and the "false positives" rate that are not threshold dependant. It is easy to see that if the classifier always exhibits an accuracy of 100%, than the direct use of the class assigned by a classifier is always better than the other methodologies described because they need always at least an extra step (e.g. using a decision threshold). In practise the accuracy of a classifier is lesser than 100%. In this case the use of the class assigned by classifier by itself is still the most convenient?

In Figures (5.1 - 5.2), the graphs on the experiments made using the FVC2004 dataset are shown. For these experiments four classifiers have been used: the k-Nearest Neighbour (kNN), the Linear Discriminant Classifier (LDC), the Quadratic Discriminant Classifier (QDC), and the Parzen windows. The comparison is made in the following way:

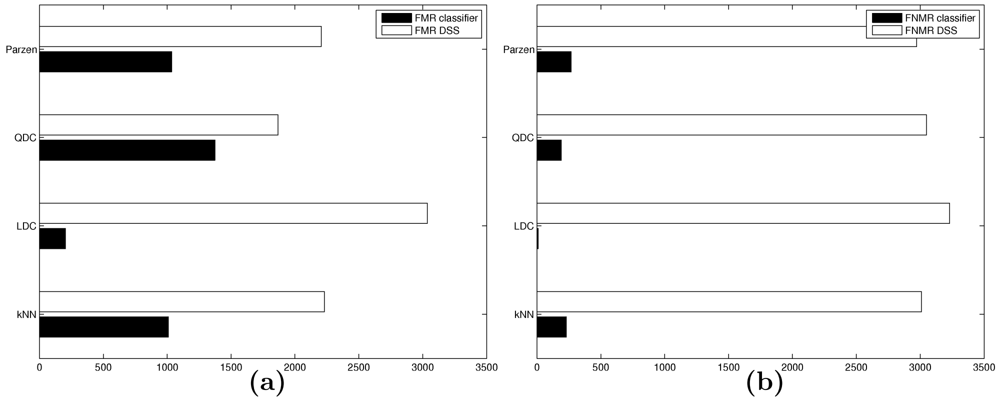
- each classifier is trained on the four subdivision of the dataset described above.
- for each classifier C the FMR and FNMR are computed as follows considering the estimated state of nature

$$FMR = \frac{impostors\ classified\ as\ genuines}{total\ impostors}$$

$$FNMR = \frac{genuines\ classified\ as\ impostors}{total\ genuines}$$

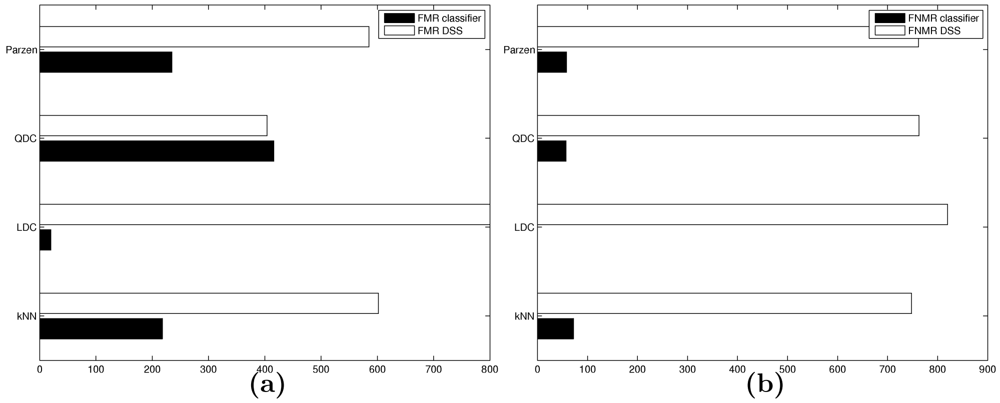
- for each classifier a DSS is built using the state of nature estimated
- for each classifier and subdivision, the value of FMR is computed for its DSS when the FNMR of the classifier and the DSS are equal. After this FMR is compared to those obtained by the classifier.
- for each classifier and subdivision, the value of $FNMR$ is computed for its DSS when the FMR of the classifier and the DSS are equal. After this $FNMR$ is compared to those obtained by the classifier.

In Figure (5.1) the results on the Open dataset are shown. While in Figure (5.2) the results on the Light dataset are reported. From these figures is clear that the use of a Dynamic Score Selector is generally preferable to the direct use of a classifier. Moreover from the figures is clear that this advantages are related to the classifier used. The more simple the classifier,



(a) (b)

Figure 5.1: Comparison between the estimated state of nature by a classifier and the Dynamic Score Selection using the Open FVC2004 dataset. In *white* are reported the occurrences when the DSS exhibits lower errors than the classifier. In *black* is reported the opposite situation. In (a) $FNMR_{class} = FNMR_{DSS}$. In (b) $FMR_{class} = FMR_{DSS}$.



(a) (b)

Figure 5.2: Comparison between the estimated state of nature by a classifier and the Dynamic Score Selection using the Light FVC2004 dataset. In *white* are reported the occurrences when the DSS exhibits lower errors than the classifier. In *black* is reported the opposite situation. In (a) $FNMR_{class} = FNMR_{DSS}$. In (b) $FMR_{class} = FMR_{DSS}$.

the better the performance of the DSS are with respect to the classifier: e.g. the performance of a LDC are very low respect its DSS, while for a QDC they are closer.

5.2.2 Dynamic Score Selection vs the use of the posterior probabilities as a "fused" score

If a classifier assigns to a pattern not only a class, but also estimates for that pattern a posterior probability of belonging to that class, these probabilities can be used as a "fused" score. These case differs from the previous because while in that case both the use are always possible, in this case are related to the potentiality of the classifier.

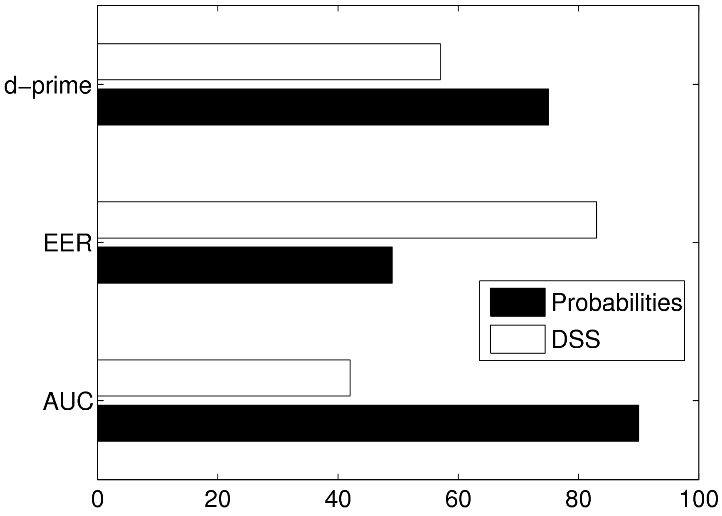


Figure 5.3: Comparison between the DSS and the use of the posterior probabilities.

In Figure (5.3), the graphs on the experiments made using the NIST Biometric Scores Set Release 1 dataset are shown. For these experiments three classifiers have been used: the k-Nearest Neighbour (kNN), the Linear Discriminant Classifier (LDC), and the Quadratic Discriminant Classifier (QDC). The comparison is made in the following way: in "white" are reported the number of times that the DSS exhibits better performance than the use of the posterior probabilities using the same classifier for all possible combinations, in "black" are reported the opposite situation. In this case it is clear that no one of the two methodologies is better than the other when taking into account the AUC and EER.

Finally it is important to remark that the DSS can be always used, also with a "crisp" classifier (i.e., a classifier that outputs only the class without other information), while the other methodology is necessary that the classifier is able to estimate the posterior probabilities.

5.3 Experimental results on measures of ensemble effectiveness

For these experiments the experts from the *Open* category of the FVC2004 dataset was used. The results on the Light category are not reported as the obtained results are equal to those obtained with the Open category.

For each pair of experts, the measures of effectiveness based on the values of the AUC, the EER, the d' and the SD index were computed, according to the formulation described in Section 4.1. Then, after combining the pairs of experts using four combination rules (i.e., the Mean rule, the Product Rule, the LDA, and a DSS based on a Quadratic Discriminant Classifier) the related values of AUC and EER was computed, as they better represent the performance.

The aim of the reported experiments is to investigate the correlation between the measures of the effectiveness of the ensemble, and the final performance achieved by the combined system. In order to evaluate this correlation a graphical representation of the results of the experiments was used. On the X axis is represented the performance of the pair of individual experts expressed in terms of AUC_δ, EER_δ, D'_δ, and SD_δ, while on the Y axis is reported the performance of their combination. The results are reported in Figures (5.4 - 5.11), where are shown the graphics of the four combination rules grouped by the ensemble effectiveness measures, and the performance measure of the combination rules. It is worth remarking that for the AUC_δ, and the D'_δ the higher the value the better the performance, while the reverse holds for the EER_δ. Moreover the higher the value of the

SD_δ index, the higher the maximum increment in terms AUC that can be achieved by the combination of the experts.

In Figures (5.4) and (5.5) the value of AUC_δ of any pair of experts is plotted against the AUC and the EER of all the considered combination methods, respectively. The inspection of Figure (5.4) allows to conclude that the value of AUC_δ is not an useful measure to select the pair of experts whose combination may provide performance improvements. In fact for all the combination rules , there is no clear relationship between the value of AUC_δ of the pair of experts and the AUC of their combination. It is easy to see that for very high values of AUC_δ the combination attains high values of AUC, but for other (lower) values of AUC_δ, the AUC of the combination is in a wide interval of values. This can be explained by the fact that high values of AUC_δ are related to pair of experts with high performance, and thus with similar behaviour. Finally, the graphics in Figure (5.4) show that AUC_δ is best correlated to the AUC of the combination when the Mean rule is employed. On the other hand, if the performance of combination in terms of the EER (Figure (5.5)) is evaluated, it is clear that the value of AUC_δ of the pair of experts is uncorrelated with the EER attained by the combination. In fact, for any value of AUC_δ, the EER of the combination spans over a wide range of values for all the combination techniques. Thus, is not possible to predict the performance of the combination in terms of EER by taking into account the value of AUC_δ.

In Figures (5.6) and (5.7) the value of EER_δ attained by each pair of experts is plotted against the AUC and the EER attained by the considered combination methods. The graphics in Figure (5.6) exhibit a better behaviour than those in Figure (5.4), but, as for AUC_δ, there is no clear relationship between the value of EER_δ and the AUC of their combination In this case too, the graphics show that EER_δ is best correlated to the AUC of the combination when the Mean Rule is employed. Finally, as far as the correlation of EER_δ with the EER of the combination is considered, the analysis of Figure (5.7) shows that there is no correlation between the two values. By comparing Figure (5.7) with Figure (5.5) it is easy to see that the corresponding graphics exhibit similar behaviour. Therefore, despite the fact that the AUC and the EER are widely used as performance measures to evaluate biometric systems, they are not suited as measures to select the experts to combine.

Figures (5.8) and (5.9) show the value of D'_δ of any pair of experts and the corresponding values of the AUC and the EER of their combinations, respectively. It is easy to see that for all combination methods, higher values of D'_δ guarantee smaller ranges of values of the performance of the combination. Thus, according to these graphics it can be concluded that the value of D'_δ is a good measure to evaluate the effectiveness of candidate ensembles of biometric experts. If we compare D'_δ with AUC_δ and EER_δ, it can be seen that D'_δ is more correlated to the performance of the combination than AUC_δ and EER_δ.

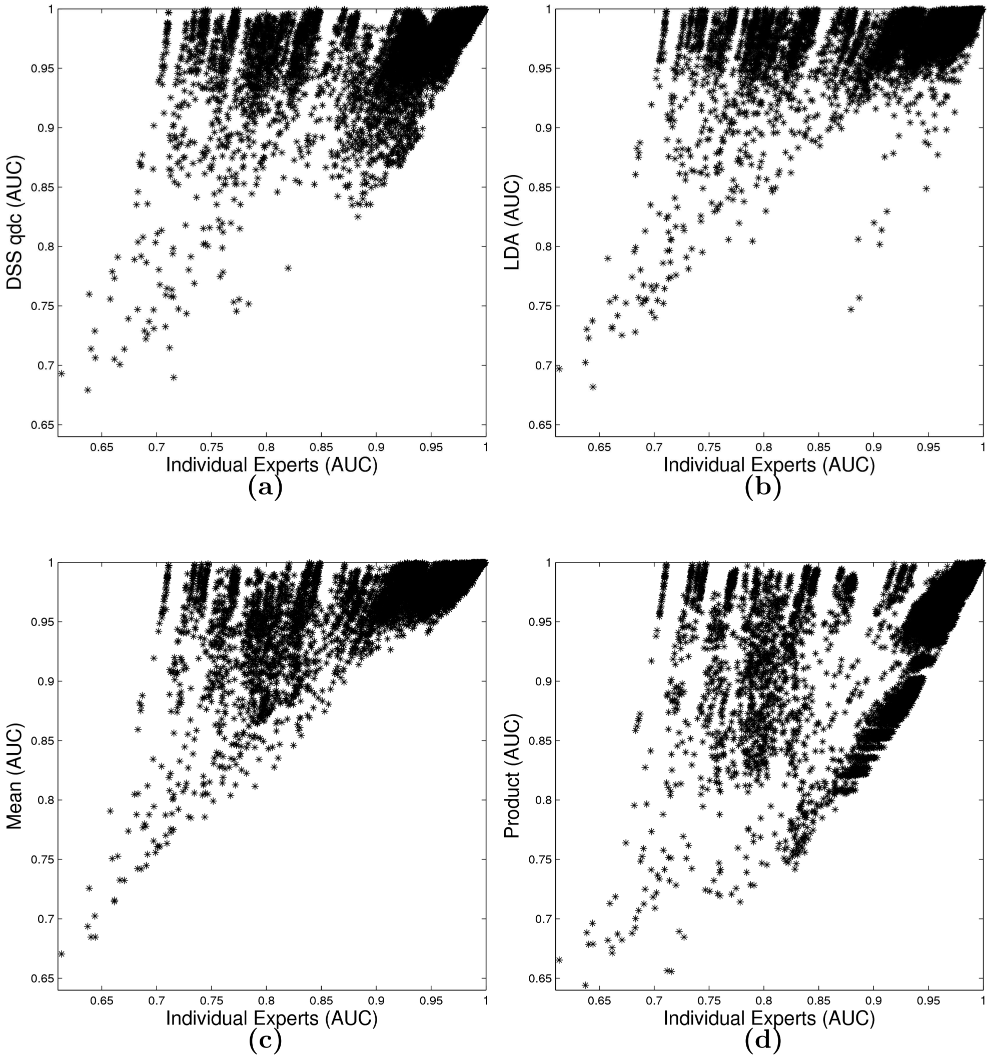
Figures (5.10) and (5.11) show the value of SD_δ attained by any pair of experts and the corresponding values of the AUC and the EER of their combinations, respectively. It seems that this measure exhibits a better correlation with the AUC rather than with the EER. This behaviour can be explained by the fact that the SD is a measure designed to predict the maximum improvement in AUC that could be attained by the combination of experts. By comparing these results with those obtained using AUC_δ as a measure of the effectiveness of the ensemble, it can be seen that they exhibit a similar behaviour. In particular, small values of SD_δ guarantee large values of performance of the ensemble, because they are related to pairs of individual experts that can "recovery" a small amount of AUC, and generally they are experts that have large values of AUC. As SD_δ measures the degree of complementarity of the experts, it is easy to see that the higher the AUC of the individual experts, the smaller the complementarity.

As far as the evaluation of the considered combination methods is concerned, Figures (5.4 - 5.11) allows to conclude that any combination technique allows attaining high performance regardless the performance of the individual experts. On the other hand, if the interest is in predicting the performance improvement, then it can be easily seen that the Product rule exhibits the worst performance, as in general the performance of the combination are not clearly correlated to the performance of the individual experts. On the other hand the Mean rule is the best combination method, while the LDA and the DSS[4] exhibit a similar behaviour not far from the performance achieved by the Mean rule.

In conclusion the D'_δ, among the measures analyzed, is the best measure to estimate the ensamble effectiveness. This results are similar to those proposed in [36] where the d' was the best measure. The results of [36] are not proposed in the thesis as the results previously described are more recent and complete[5].

[4] Remember that in this experiments the DSS is based on the state of nature estimated using a Quadratic Bayes classifier

[5] This new results are going to be published in a journal version of the paper [36] during the 2008



(a) (b) (c) (d)

Figure 5.4: In these figures the AUC_δ attained by each pair of experts is plotted against the AUC computed with the (a) DSS, (b) the LDA, (c) the Mean rule, and (d) the Product rule.

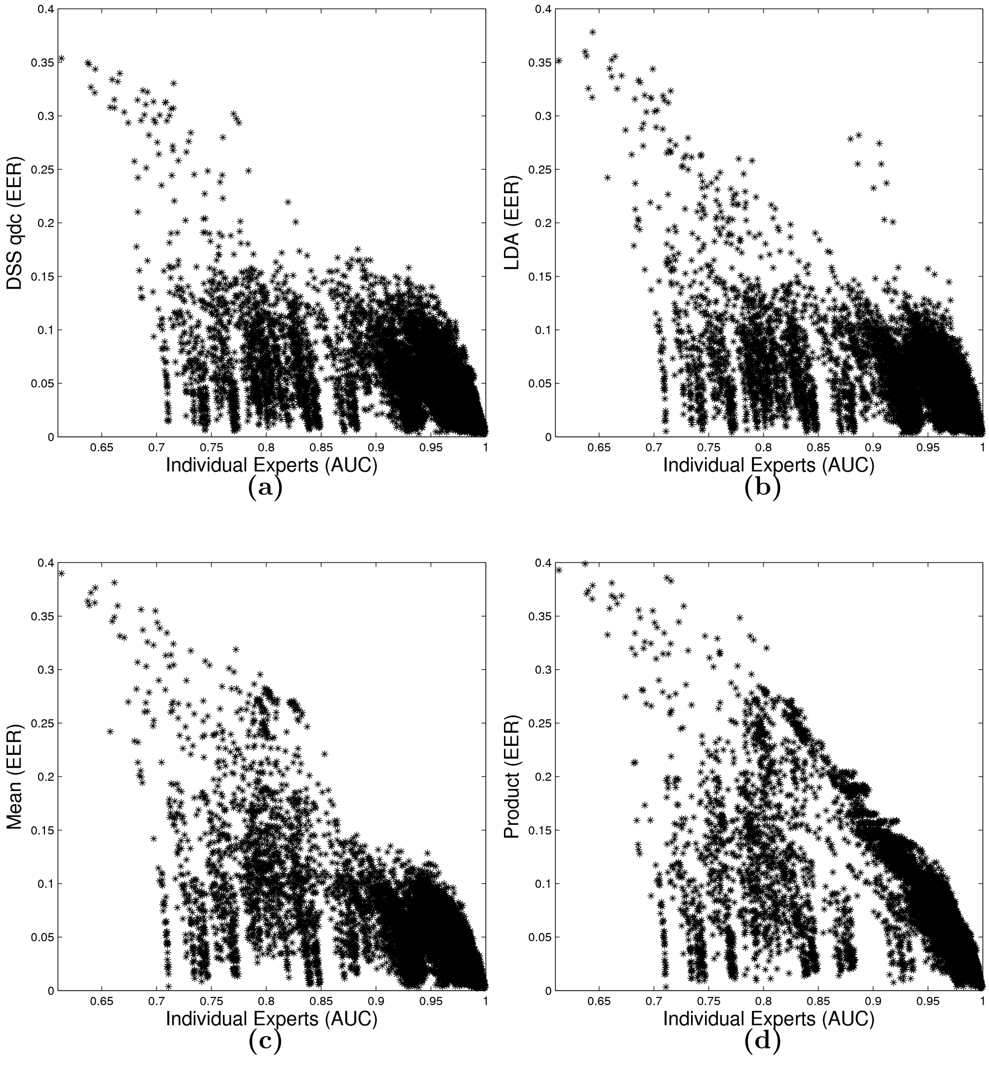
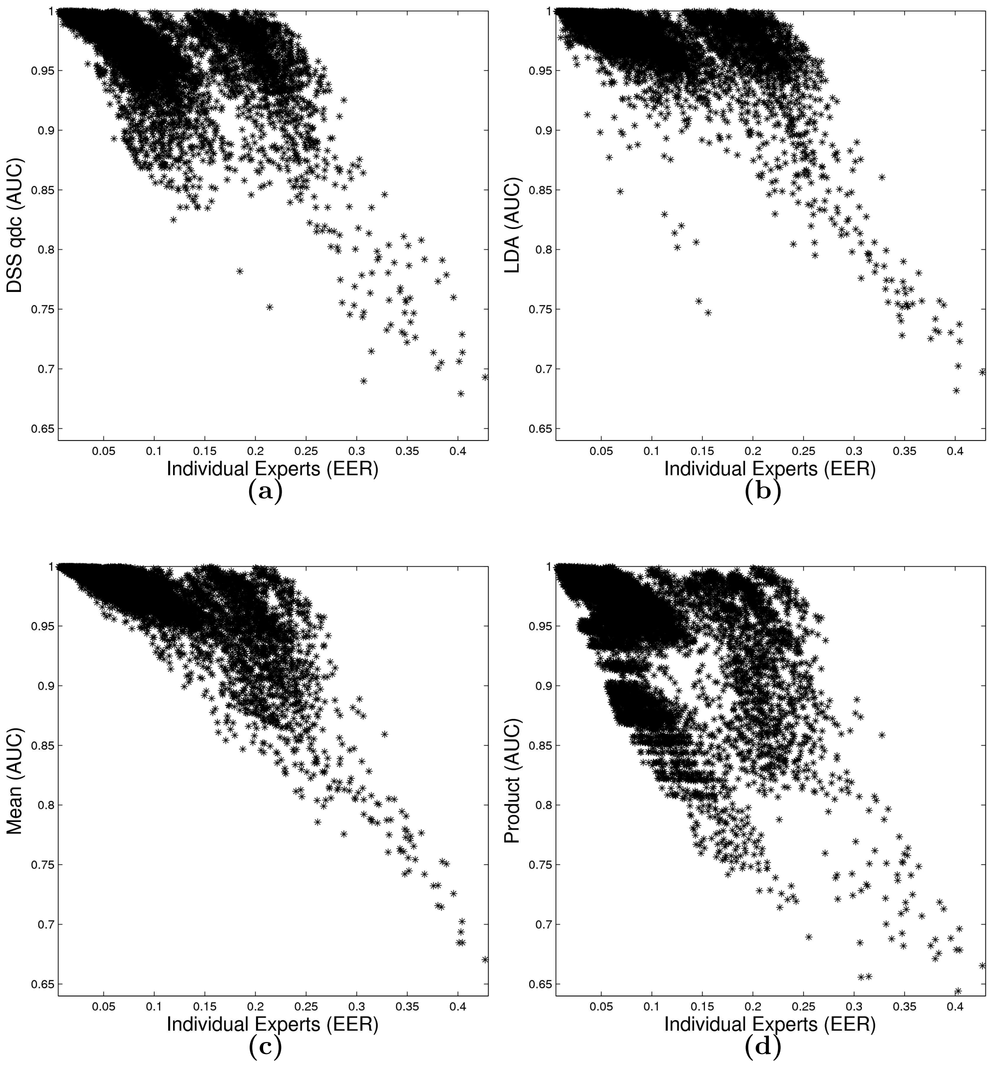


Figure 5.5: In these figures the AUC_{δ} attained by each pair of experts is plotted against the EER computed with the (a) DSS, (b) the LDA, (c) the Mean rule, and (d) the Product rule.



(a) (b) (c) (d)

Figure 5.6: In these figures the EER_δ attained by each pair of experts is plotted against the AUC computed with the (a) DSS, (b) the LDA, (c) the Mean rule, and (d) the Product rule.

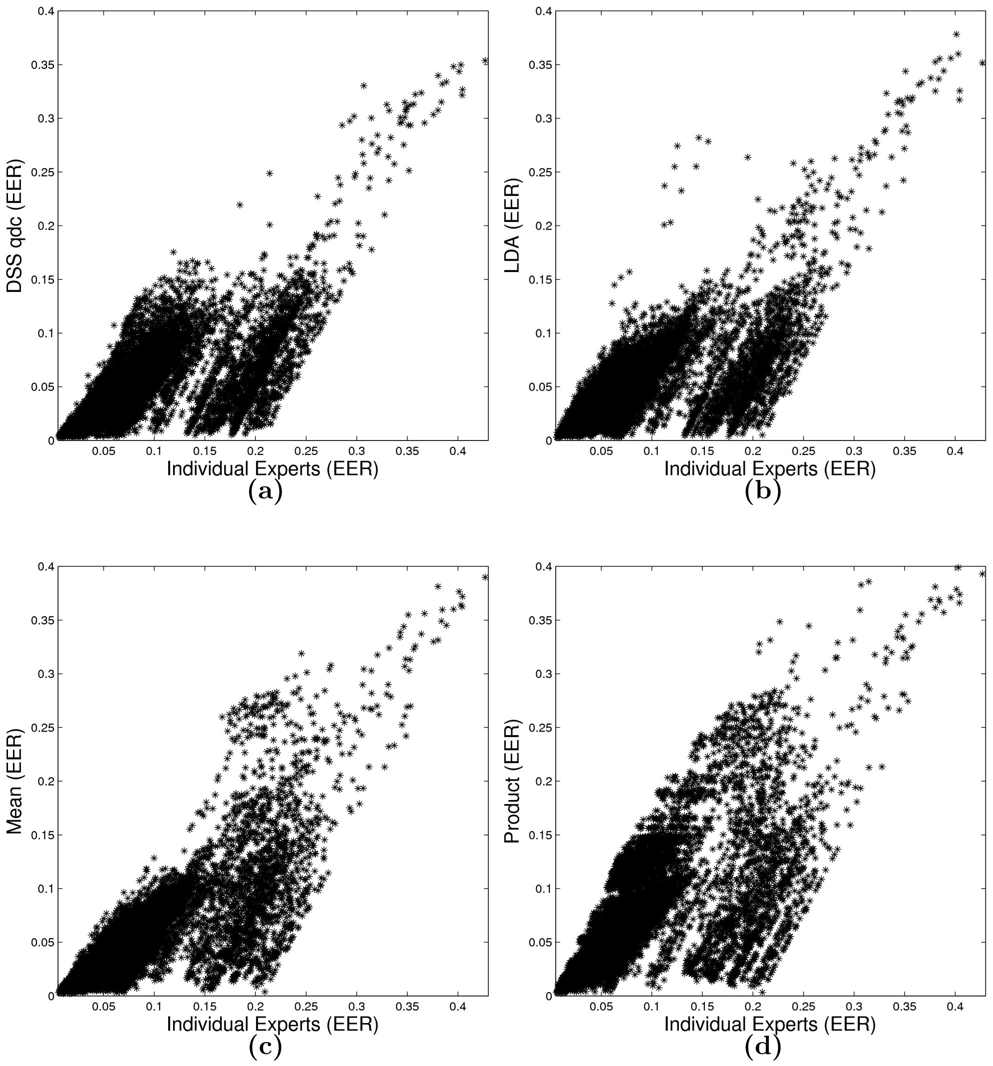
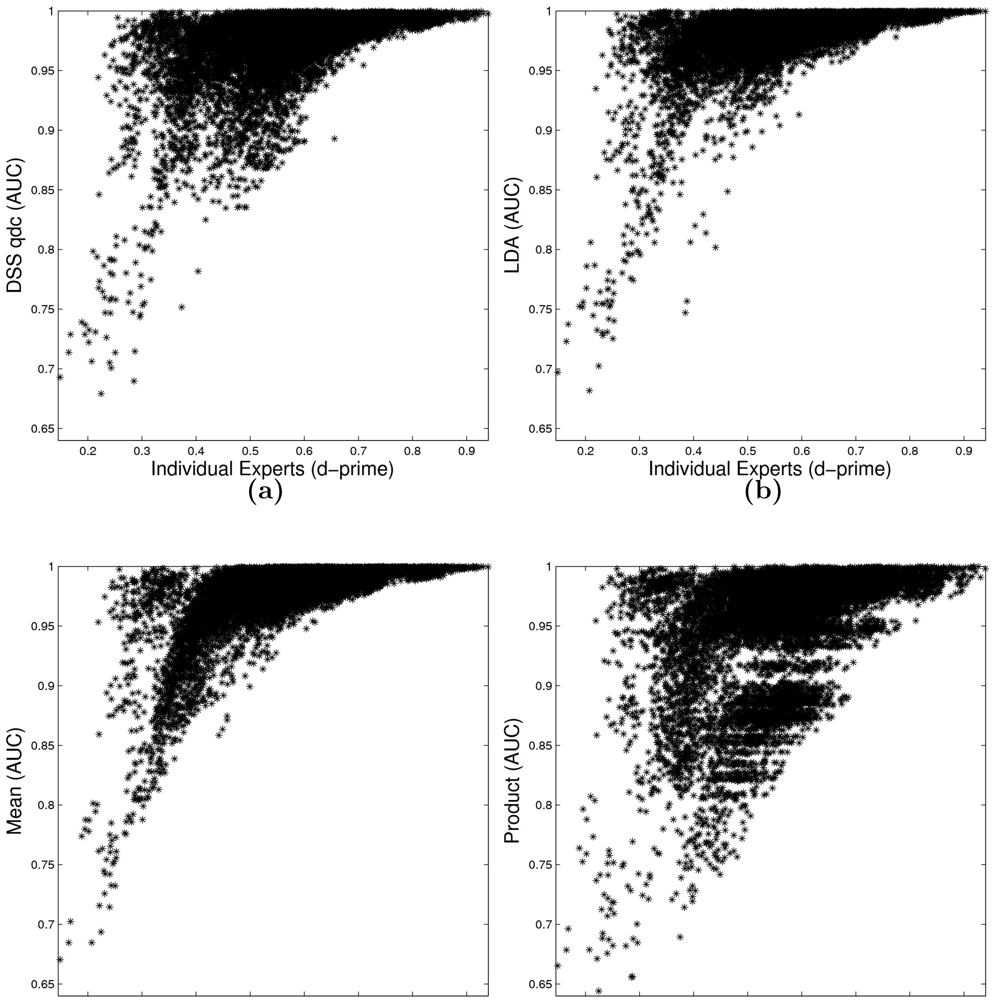


Figure 5.7: In these figures the EER_δ attained by each pair of experts is plotted against the EER computed with the (a) DSS, (b) the LDA, (c) the Mean rule, and (d) the Product rule.



(a) (b) (c) (d)

Figure 5.8: In these figures the D'_δ attained by each pair of experts is plotted against the AUC computed with the (a) DSS, (b) the LDA, (c) the Mean rule, and (d) the Product rule.

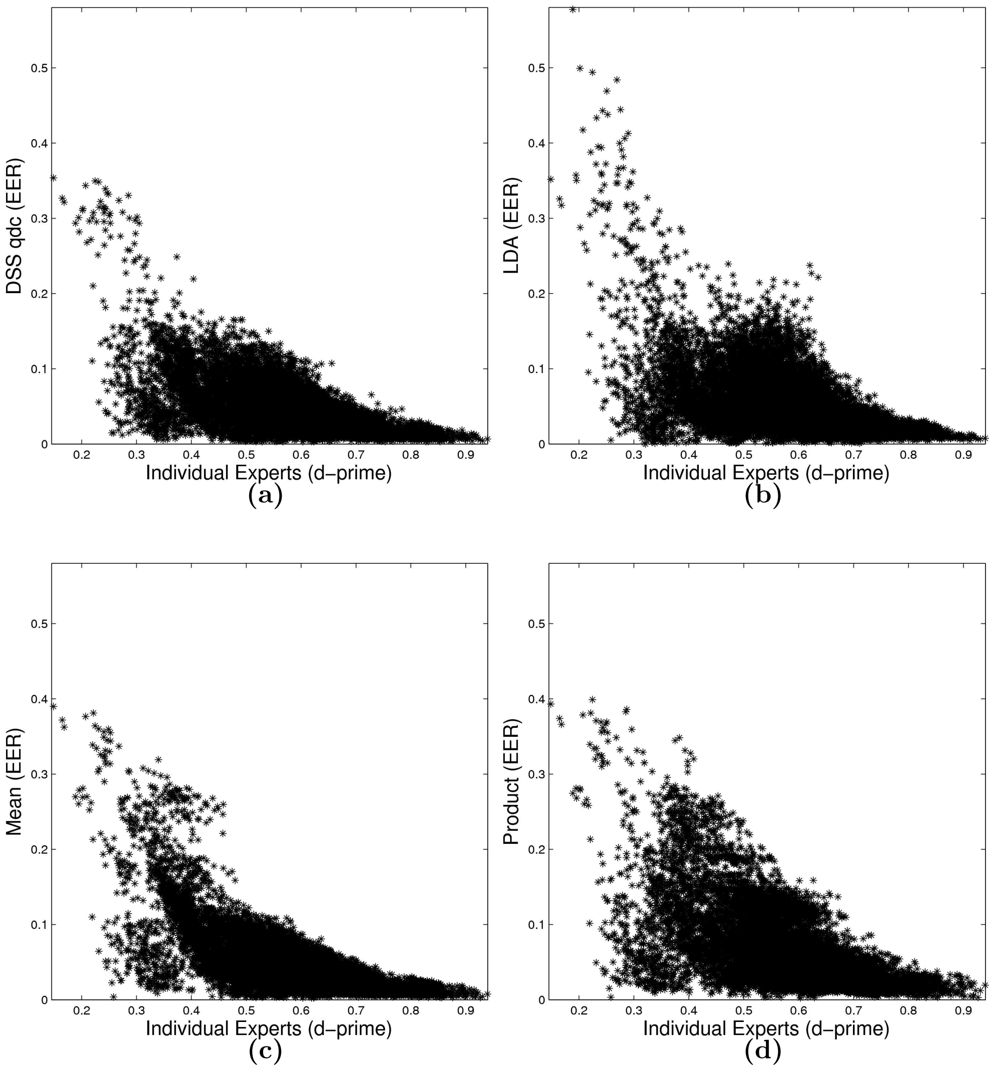
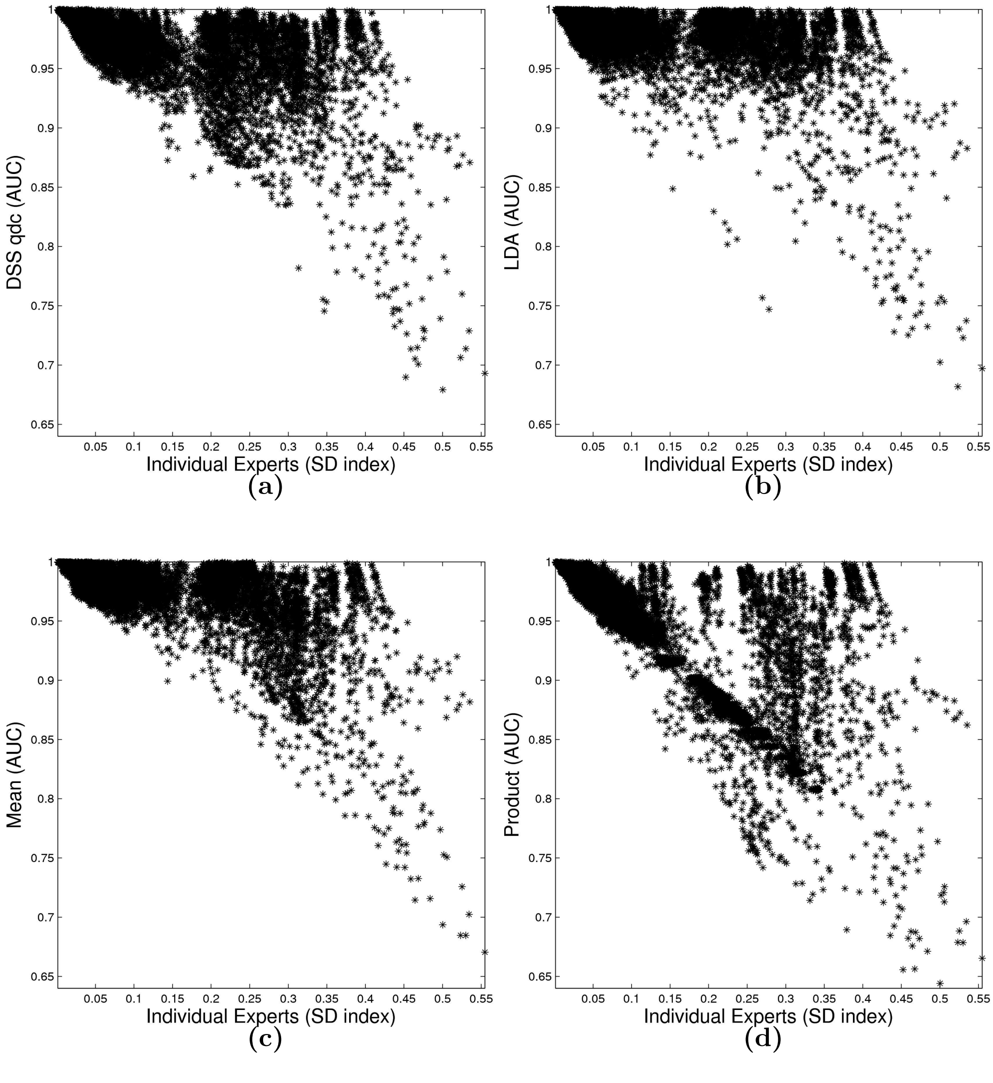


Figure 5.9: In these figures the D'_δ attained by each pair of experts is plotted against the EER computed with the (a) DSS, (b) the LDA, (c) the Mean rule, and (d) the Product rule.



(a) **(b)** **(c)** **(d)**

Figure 5.10: In these figures the SD_δ attained by each pair of experts is plotted against the AUC computed with the (a) DSS, (b) the LDA, (c) the Mean rule, and (d) the Product rule.

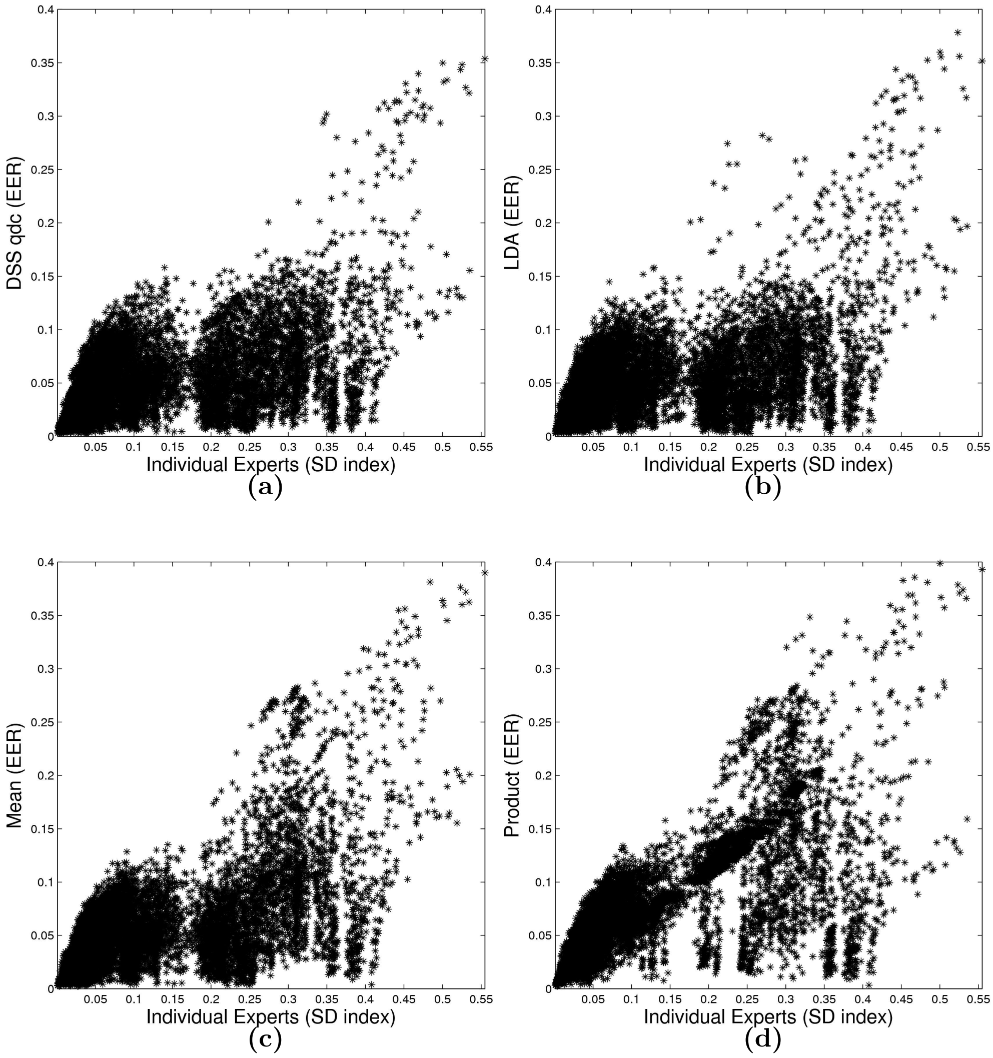


Figure 5.11: In these figures the SD_δ attained by each pair of experts is plotted against the EER computed with the (a) DSS, (b) the LDA, (c) the Mean rule, and (d) the Product rule.

5.4 Experimental results on FVC2004

In this section the methods and performance measures described in this thesis are applied to the FVC2004 score dataset. The setup of these experiments is the same of the experiments described in Section 5.3, but it is run both on the Open and Light category.

These experiments are made comparing the following methods: the average values computed on the two individual experts combined, the *ideal score selector*, the *optimal linear combiner*, the Mean rule,the Product rule, the LDA, the DSS based on the RME, the DSS based on four classifier (k-NN, Linear Discriminant, Quadratic Discriminant, and Parzen windows), and the use of posterior probabilities of the four classifier (k-NN, Linear Discriminant, Quadratic Discriminant, and Parzen windows). The performance are assessed using the following measures: the AUC, and the EER.

In these experiments the pairs of experts were sorted using the measures of effectiveness described in Section 4.1. For each measure the first ten pairs with the highest value of each measure are considered, and the last 10 pairs with the smallest value of each measure are taken into account.

Tables (5.2 - 5.5) show the results of the experiments run on the Open category. In Table (5.2) the pairs of experts are sorted according to their AUC_δ values. When the value of the AUC_δ is high the performance of all the combined methods (except the *ideal score selector* and the *optimal linear combiner*) is closer for both the performance measures taken into account. With respect of the average values computed from the individual experts an evident improvement is achieved when the EER is considered, while the improvements in terms of AUC (when present) are small. When the value of the AUC_δ is small the performance of all the combined methods present improvements with respect to the “average expert”. In this case all the DSS methods, and the combination methods based on the posterior probabilities (except those based on the LDC) have better performance than the other methods in terms of AUC and EER. In terms of AUC, the DSS based on k-NN and QDC exhibit closer performance to those of the *optimal linear*

large AUC_δ	**AUC**	**EER**
Average Expert	0.9938 (±0.0040)	0.0208 (±0.0094)
Ideal Selector	0.9996 (±0.0011)	0.0035 (±0.0041)
Optimal Linear Combiner	0.9981 (±0.0024)	0.0529 (±0.0707)
Mean	0.9977 (±0.0026)	0.0132 (±0.0085)
Product	0.9933 (±0.0057)	0.0169 (±0.0094)
LDA	0.9961 (±0.0056)	0.0164 (±0.0093)
DSS RME	0.9933 (±0.0051)	0.0165 (±0.0098)
DSS k-NN	0.9930 (±0.0055)	0.0168 (±0.0101)
DSS LDC	0.9931 (±0.0056)	0.0177 (±0.0102)
DSS QDC	0.9942 (±0.0051)	0.0156 (±0.0093)
DSS Parzen	0.9932 (±0.0054)	0.0164 (±0.0096)
k-NN Ppost	0.9933 (±0.0057)	0.0127 (±0.0069)
LDC Ppost	0.9929 (±0.0127)	0.0174 (±0.0115)
QDC Ppost	0.9960 (±0.0046)	0.0134 (±0.0072)
Parzen Ppost	0.9943 (±0.0072)	0.0150 (±0.0090)

small AUC_δ	**AUC**	**EER**
Average Expert	0.7108 (±0.0395)	0.3290 (±0.0483)
Ideal Selector	0.9823 (±0.0210)	0.0419 (±0.0483)
Optimal Linear Combiner	0.8173 (±0.0700)	0.3338 (±0.3253)
Mean	0.8084 (±0.0783)	0.2606 (±0.0847)
Product	0.7805 (±0.0862)	0.2597 (±0.1052)
LDA	0.8038 (±0.0696)	0.2583 (±0.1329)
DSS RME	0.7756 (±0.0842)	0.2515 (±0.0962)
DSS k-NN	0.8151 (±0.0759)	0.2391 (±0.0849)
DSS LDC	0.7794 (±0.0881)	0.2604 (±0.1055)
DSS QDC	0.8057 (±0.0720)	0.2371 (±0.0905)
DSS Parzen	0.8155 (±0.0776)	0.2310 (±0.0962)
k-NN Ppost	0.8548 (±0.0647)	0.2222 (±0.0660)
LDC Ppost	0.8040 (±0.0697)	0.2617 (±0.0759)
QDC Ppost	0.8487 (±0.0615)	0.2318 (±0.0851)
Parzen Ppost	0.8624 (±0.0661)	0.2103 (±0.0686)

Table 5.2: Open category, AUC_δ sorted. Mean and standard deviation among the experiments using for each experiment the 10 pairs of experts with the highest AUC_δ, and the 10 pairs of experts with the lowest AUC_δ.

large EER_δ	**AUC**	**EER**
Average Expert	0.6960 (±0.0360)	0.3502 (±0.0357)
Ideal Selector	0.9749 (±0.0217)	0.0605 (±0.0500)
Optimal Linear Combiner	0.7822 (±0.0444)	0.4415 (±0.3101)
Mean	0.7707 (±0.0490)	0.2996 (±0.0583)
Product	0.7367 (±0.0521)	0.3168 (±0.0529)
LDA	0.7709 (±0.0470)	0.3223 (±0.0971)
DSS RME	0.7311 (±0.0474)	0.3035 (±0.0457)
DSS k-NN	0.7791 (±0.0658)	0.2799 (±0.0642)
DSS LDC	0.7352 (±0.0579)	0.3156 (±0.0564)
DSS QDC	0.7674 (±0.0480)	0.2884 (±0.0475)
DSS Parzen	0.7786 (±0.0637)	0.2821 (±0.0631)
k-NN Ppost	0.8312 (±0.0662)	0.2420 (±0.0651)
LDC Ppost	0.7710 (±0.0469)	0.2934 (±0.0536)
QDC Ppost	0.8234 (±0.0578)	0.2531 (±0.0531)
Parzen Ppost	0.8364 (±0.0651)	0.2394 (±0.0631)

small EER_δ	**AUC**	**EER**
Average Expert	0.9914 (±0.0052)	0.0219 (±0.0111)
Ideal Selector	0.9991 (±0.0017)	0.0038 (±0.0049)
Optimal Linear Combiner	0.9968 (±0.0038)	0.0498 (±0.0453)
Mean	0.9966 (±0.0039)	0.0140 (±0.0094)
Product	0.9896 (±0.0068)	0.0202 (±0.0120)
LDA	0.9935 (±0.0074)	0.0178 (±0.0112)
DSS RME	0.9902 (±0.0060)	0.0187 (±0.0115)
DSS k-NN	0.9909 (±0.0061)	0.0175 (±0.0108)
DSS LDC	0.9895 (±0.0068)	0.0208 (±0.0129)
DSS QDC	0.9914 (±0.0062)	0.0176 (±0.0120)
DSS Parzen	0.9910 (±0.0060)	0.0175 (±0.0108)
k-NN Ppost	0.9913 (±0.0070)	0.0129 (±0.0072)
LDC Ppost	0.9880 (±0.0165)	0.0234 (±0.0151)
QDC Ppost	0.9940 (±0.0059)	0.0145 (±0.0090)
Parzen Ppost	0.9917 (±0.0090)	0.0157 (±0.0097)

Table 5.3: Open category, EER_δ sorted. Mean and standard deviation among the experiments using for each experiment the 10 pairs of experts with the highest EER_δ, and the 10 pairs of experts with the lowest EER_δ.

large SD_δ	**AUC**	**EER**
Average Expert	0.7156 (±0.0460)	0.3169 (±0.0581)
Ideal Selector	0.9899 (±0.0153)	0.0245 (±0.0387)
Optimal Linear Combiner	0.8401 (±0.0791)	0.2684 (±0.3224)
Mean	0.8312 (±0.0857)	0.2403 (±0.0877)
Product	0.8018 (±0.0904)	0.2283 (±0.1129)
LDA	0.8294 (±0.0810)	0.2369 (±0.1433)
DSS RME	0.7990 (±0.0912)	0.2227 (±0.1066)
DSS k-NN	0.8228 (±0.0764)	0.2245 (±0.0900)
DSS LDC	0.8051 (±0.0906)	0.2248 (±0.1128)
DSS QDC	0.8244 (±0.0777)	0.2076 (±0.1002)
DSS Parzen	0.8253 (±0.0800)	0.2066 (±0.1036)
k-NN Ppost	0.8609 (±0.0556)	0.2215 (±0.0571)
LDC Ppost	0.8298 (±0.0812)	0.2370 (±0.0871)
QDC Ppost	0.8539 (±0.0599)	0.2325 (±0.1021)
Parzen Ppost	0.8708 (±0.0608)	0.2031 (±0.0624)

small SD_δ	**AUC**	**EER**
Average Expert	0.9939 (±0.0040)	0.0207 (±0.0095)
Ideal Selector	0.9998 (±0.0005)	0.0027 (±0.0027)
Optimal Linear Combiner	0.9985 (±0.0016)	0.0752 (±0.1647)
Mean	0.9982 (±0.0019)	0.0121 (±0.0072)
Product	0.9943 (±0.0049)	0.0159 (±0.0081)
LDA	0.9965 (±0.0055)	0.0165 (±0.0091)
DSS RME	0.9940 (±0.0043)	0.0153 (±0.0079)
DSS k-NN	0.9942 (±0.0045)	0.0149 (±0.0079)
DSS LDC	0.9943 (±0.0047)	0.0156 (±0.0086)
DSS QDC	0.9953 (±0.0040)	0.0138 (±0.0071)
DSS Parzen	0.9943 (±0.0046)	0.0147 (±0.0078)
k-NN Ppost	0.9932 (±0.0059)	0.0121 (±0.0063)
LDC Ppost	0.9933 (±0.0127)	0.0171 (±0.0111)
QDC Ppost	0.9964 (±0.0044)	0.0129 (±0.0069)
Parzen Ppost	0.9946 (±0.0072)	0.0142 (±0.0085)

Table 5.4: Open category, SD_δ sorted. Mean and standard deviation among the experiments using for each experiment the 10 pairs of experts with the highest SD_δ, and the 10 pairs of experts with the lowest SD_δ.

large D'_δ	**AUC**	**EER**
Average Expert	0.9942 (±0.0033)	0.0171 (±0.0053)
Ideal Selector	1.0000 (±0.0000)	0.0026 (±0.0013)
Optimal Linear Combiner	0.9993 (±0.0005)	0.0912 (±0.1967)
Mean	0.9990 (±0.0006)	0.0098 (±0.0042)
Product	0.9918 (±0.0070)	0.0176 (±0.0089)
LDA	0.9992 (±0.0005)	0.0094 (±0.0020)
DSS RME	0.9948 (±0.0032)	0.0109 (±0.0042)
DSS k-NN	0.9962 (±0.0022)	0.0097 (±0.0029)
DSS LDC	0.9942 (±0.0035)	0.0143 (±0.0052)
DSS QDC	0.9964 (±0.0022)	0.0097 (±0.0032)
DSS Parzen	0.9962 (±0.0023)	0.0097 (±0.0031)
k-NN Ppost	0.9961 (±0.0017)	0.0092 (±0.0032)
LDC Ppost	0.9974 (±0.0017)	0.0096 (±0.0026)
QDC Ppost	0.9992 (±0.0005)	0.0082 (±0.0024)
Parzen Ppost	0.9988 (±0.0007)	0.0080 (±0.0025)

small D'_δ	**AUC**	**EER**
Average Expert	0.7049 (±0.0452)	0.3407 (±0.0471)
Ideal Selector	0.9820 (±0.0204)	0.0426 (±0.0506)
Optimal Linear Combiner	0.8062 (±0.0832)	0.3959 (±0.3387)
Mean	0.7944 (±0.0873)	0.2762 (±0.0945)
Product	0.7787 (±0.1010)	0.2775 (±0.1029)
LDA	0.7933 (±0.0824)	0.3068 (±0.1228)
DSS RME	0.7524 (±0.0900)	0.2775 (±0.0931)
DSS k-NN	0.7911 (±0.0919)	0.2634 (±0.0943)
DSS LDC	0.7690 (±0.1059)	0.2810 (±0.1045)
DSS QDC	0.7922 (±0.0856)	0.2605 (±0.0909)
DSS Parzen	0.7920 (±0.0915)	0.2646 (±0.0953)
k-NN Ppost	0.8314 (±0.0719)	0.2391 (±0.0785)
LDC Ppost	0.7935 (±0.0826)	0.2744 (±0.0850)
QDC Ppost	0.8276 (±0.0709)	0.2425 (±0.0788)
Parzen Ppost	0.8369 (±0.0722)	0.2337 (±0.0795)

Table 5.5: Open category, D'_δ sorted. Mean and standard deviation among the experiments using for each experiment the 10 pairs of experts with the highest D'_δ, and the 10 pairs of experts with the lowest D'_δ.

combiner, while the methods based on the posterior probabilities (i.e., k-NN, QDC, and Parzen) outperforms *optimal linear combiner*. By considering the two ideal methods the *ideal score selector* always outperforms the *optimal linear combiner*. Moreover, if the EER is considered, the *optimal linear combiner* is outperformed by all the methods. Remember that the *optimal linear combiner* is built to optimize the AUC, thus can happen, as in this case, that its performance in terms of EER can be lower than those achieved using other linear combination methods (e.g., Mean rule and LDA).

Table (5.3) shows the pairs of experts sorted using their EER_δ values. When the value of the EER_δ is high (low performance) the performance of all the combined methods present improvements with respect to the "average expert". In the combination methods based on the posterior probabilities (k-NN, QDC, and Parzen) have better performance than the other methods in terms of AUC and EER. While the DSS based on k-NN and Parzen exhibits better performance than the other rules except those cited above. When the value of the EER_δ is small (high performance) the performance of all the combined methods exhibits a similar behaviour when in table (5.2) AUC_δ is high. With respect of the average values computed from the individual experts a global improvement is achieved when the EER is considered, while the improvements in terms of AUC (when present) are small. Also in this case, by considering the two ideal methods the *ideal score selector* always outperforms the *optimal linear combiner*.

In Table (5.4) the pairs of experts sorted using their SD_δ values are reported. When the value of the SD_δ is high the performance of all the combined methods present improvements with respect to the "average expert". In the combination methods based on the posterior probabilities (based on k-NN, QDC, and Parzen) have better performance than the other methods in terms of AUC and also better than the *optimal linear combiner*. While in terms of EER both the methods based on the Parzen (DSS, and Ppost) and the DSS based on QDC exhibit the better performance with respect to the other combination rules. Moreover all the DSS methods have a better (lower) EER than the other methods used for comparison (i.e., the Mean rule, the Product rule, and the LDA) and than two methods based on the posterior probabilities (i.e., LDC, and QDC). When the value of the SD_δ is small the performance of all the combined methods exhibits closer performance in terms of AUC and EER. The vast majority of all the combined methods present improvements in terms of AUC with respect to the "average expert", while it is outperformed by all the combination methods in terms of EER. If these results are compared with those exposed in Table (5.2) for high values of AUC_δ it is clear that the *score dissimilarity* allows to achieve better results in terms of AUC than the use of the AUC itself. This result is due to the nature of the *score dissimilarity* index, that find the complementary of the experts in terms of AUC. In this case some of the DSS methods exhibits better performance than the correspondant methods with posterior probabilities (e.g., DSS LDC and Ppost LDC). Again, when considering the two ideal methods the *ideal score selector* always outperforms the *optimal linear combiner.*

Table (5.5) shows the pairs of experts sorted using their D'_δ values. When the value of the D'_δ is high the performance of all the combined methods, but the Product rule, present improvements in terms of AUC with respect to the "average expert". If the results obtained for an high value of the D'_δ are compred with the results exposed in Tables (5.2 - 5.4) for large AUC_δ, small EER_δ, and small SD_δ it is easy to see that the D'_δ allows to choose the best experts to be combined, and thus the combination methods achieve better performance (e.g., this is the only case than the *ideal score selector* achieves a perfect AUC as mean results among all the experiments run). Moreover the DSS methods, but that based on LDC, and the methods that use the posterior probabilities allow to obtain better results than the other methods in terms of EER. When the value of the D'_δ is small the performance of all the combined methods present improvements with respect to the "average expert". In this case the DSS methods (except that based on the LDC) have closer performance in terms of AUC to the Mean rule and the LDA, while in terms of EER the DSS are better than the Mean rule and the LDA. While the methods based on the posterior probabilities (i.e., k-NN, QDC, and Parzen) outperforms the other methods and the *optimal linear combiner* in terms of AUC. By considering the two ideal methods the *ideal score selector* always outperforms the *optimal linear combiner*. In terms of EER, the DSS methods and those that uses the posterior probabilities (except those based on the LDC) outperforms the other methods.

In Tables (5.6 - 5.9) the results of the experiments run on the Light category are presented. Table (5.6) shows the results on the pairs of experts sorted according to their AUC_δ values. When the value of the AUC_δ is high, the results exhibit a similar behaviour to those reported in Table (5.2) (obviously some value of the AUC and the EER are worse than those, because in the Light category the experts exhibits lower performance than in the Open category). In this case the posterior probabilities based on the QDC outperforms the other methods. All the combined methods, but the DSS based on LDC, exhibit an EER smaller than that of the "average expert". When the value of the AUC_δ is small the performance of all the combined methods present improvements with respect to the "average expert". The DSS based on the k-NN and Parzen exhibits better performance than the *optimal linear combiner*. Moreover all the posterior probabilities methods, but that based on LDC, outperform the *optimal linear combiner*. All the previous methods exhibit better performance in terms of AUC and EER than the other methods. In this case the best results in terms of AUC is reached by the posterior probabilities based on Parzen, while the best result in terms of EER are achieved by the *ideal score selector*.

In Table (5.7) the results on the pairs of experts sorted according to their EER_δ values are presented. When the value of the EER_δ is high the performance of all the combined methods present improvements with respect to the "average expert", but the Product rule and the DSS RME. Some of the DSS (based on k-NN, QDC, and Parzen) and posterior probabilities methods (based on k-NN, QDC, and Parzen) exhibits better performance

large AUC_δ	**AUC**	**EER**
Average Expert	0.9824 (±0.0048)	0.0449 (±0.0110)
Ideal Selector	0.9972 (±0.0022)	0.0139 (±0.0082)
Optimal Linear Combiner	0.9911 (±0.0049)	0.0722 (±0.0436)
Mean	0.9904 (±0.0048)	0.0331 (±0.0109)
Product	0.9821 (±0.0059)	0.0400 (±0.0112)
LDA	0.9883 (±0.0053)	0.0437 (±0.0140)
DSS RME	0.9831 (±0.0058)	0.0387 (±0.0121)
DSS k-NN	0.9840 (±0.0060)	0.0383 (±0.0127)
DSS LDC	0.9812 (±0.0059)	0.0450 (±0.0127)
DSS QDC	0.9865 (±0.0057)	0.0336 (±0.0127)
DSS Parzen	0.9844 (±0.0057)	0.0379 (±0.0125)
k-NN Ppost	0.9797 (±0.0075)	0.0313 (±0.0105)
LDC Ppost	0.9883 (±0.0053)	0.0370 (±0.0105)
QDC Ppost	0.9909 (±0.0049)	0.0299 (±0.0100)
Parzen Ppost	0.9883 (±0.0065)	0.0328 (±0.0104)

small AUC_δ	**AUC**	**EER**
Average Expert	0.7106 (±0.0715)	0.3307 (±0.0694)
Ideal Selector	0.9299 (±0.0904)	0.0765 (±0.0904)
Optimal Linear Combiner	0.8666 (±0.1456)	0.3437 (±0.2874)
Mean	0.8558 (±0.1415)	0.1945 (±0.1538)
Product	0.5752 (±0.0634)	0.4550 (±0.0710)
LDA	0.8642 (±0.1476)	0.1718 (±0.1495)
DSS RME	0.6489 (±0.0962)	0.3737 (±0.0910)
DSS k-NN	0.8680 (±0.1001)	0.1472 (±0.0970)
DSS LDC	0.8330 (±0.0874)	0.1830 (±0.0860)
DSS QDC	0.8504 (±0.0930)	0.1660 (±0.0978)
DSS Parzen	0.8683 (±0.0868)	0.1460 (±0.0842)
k-NN Ppost	0.9109 (±0.0721)	0.1335 (±0.0889)
LDC Ppost	0.8650 (±0.1458)	0.1817 (±0.1575)
QDC Ppost	0.9244 (±0.0664)	0.1385 (±0.1032)
Parzen Ppost	0.9359 (±0.0477)	0.1174 (±0.0654)

Table 5.6: Light category, AUC_δ sorted. Mean and standard deviation among the experiments using for each experiment the 10 pairs of experts with the highest AUC_δ, and the 10 pairs of experts with the lowest AUC_δ.

large EER_δ	**AUC**	**EER**
Average Expert	0.7031 (±0.0778)	0.3442 (±0.0699)
Ideal Selector	0.9088 (±0.0924)	0.1032 (±0.0941)
Optimal Linear Combiner	0.8195 (±0.1381)	0.3872 (±0.2281)
Mean	0.8113 (±0.1323)	0.2456 (±0.1351)
Product	0.6034 (±0.0622)	0.4405 (±0.0716)
LDA	0.8154 (±0.1424)	0.2383 (±0.1389)
DSS RME	0.6573 (±0.0723)	0.3798 (±0.0720)
DSS k-NN	0.8289 (±0.0912)	0.1953 (±0.0869)
DSS LDC	0.8059 (±0.0934)	0.2187 (±0.0972)
DSS QDC	0.8220 (±0.0823)	0.2033 (±0.0858)
DSS Parzen	0.8353 (±0.0818)	0.1885 (±0.0790)
k-NN Ppost	0.8951 (±0.0669)	0.1693 (±0.0815)
LDC Ppost	0.8162 (±0.1409)	0.2391 (±0.1453)
QDC Ppost	0.9042 (±0.0631)	0.1711 (±0.0963)
Parzen Ppost	0.9215 (±0.0431)	0.1459 (±0.0593)

small EER_δ	**AUC**	**EER**
Average Expert	0.9833 (±0.0063)	0.0409 (±0.0131)
Ideal Selector	0.9968 (±0.0025)	0.0144 (±0.0077)
Optimal Linear Combiner	0.9911 (±0.0052)	0.0702 (±0.0474)
Mean	0.9905 (±0.0052)	0.0307 (±0.0113)
Product	0.9818 (±0.0079)	0.0382 (±0.0146)
LDA	0.9877 (±0.0059)	0.0398 (±0.0135)
DSS RME	0.9831 (±0.0071)	0.0365 (±0.0139)
DSS k-NN	0.9841 (±0.0070)	0.0358 (±0.0134)
DSS LDC	0.9814 (±0.0076)	0.0418 (±0.0145)
DSS QDC	0.9867 (±0.0064)	0.0309 (±0.0129)
DSS Parzen	0.9845 (±0.0069)	0.0353 (±0.0136)
k-NN Ppost	0.9788 (±0.0098)	0.0288 (±0.0112)
LDC Ppost	0.9877 (±0.0059)	0.0355 (±0.0114)
QDC Ppost	0.9904 (±0.0061)	0.0288 (±0.0114)
Parzen Ppost	0.9882 (±0.0073)	0.0311 (±0.0120)

Table 5.7: Light category, EER_δ sorted. Mean and standard deviation among the experiments using for each experiment the 10 pairs of experts with the highest EER_δ, and the 10 pairs of experts with the lowest EER_δ.

large SD_δ	**AUC**	**EER**
Average Expert	0.7167 (±0.0628)	0.3049 (±0.0718)
Ideal Selector	0.9540 (±0.0815)	0.0467 (±0.0813)
Optimal Linear Combiner	0.9171 (±0.1252)	0.3106 (±0.3261)
Mean	0.9169 (±0.1252)	0.1253 (±0.1435)
Product	0.5173 (±0.0056)	0.4827 (±0.0056)
LDA	0.9135 (±0.1283)	0.1106 (±0.1304)
DSS RME	0.5718 (±0.0290)	0.4283 (±0.0290)
DSS k-NN	0.9059 (±0.0827)	0.1019 (±0.0809)
DSS LDC	0.8585 (±0.0746)	0.1445 (±0.0727)
DSS QDC	0.8942 (±0.0926)	0.1201 (±0.0963)
DSS Parzen	0.9018 (±0.0744)	0.1029 (±0.0732)
k-NN Ppost	0.9254 (±0.0573)	0.1074 (±0.0694)
LDC Ppost	0.9154 (±0.1261)	0.1273 (±0.1442)
QDC Ppost	0.9391 (±0.0580)	0.1084 (±0.0916)
Parzen Ppost	0.9428 (±0.0423)	0.0972 (±0.0527)

small SD_δ	**AUC**	**EER**
Average Expert	0.9828 (±0.0058)	0.0433 (±0.0128)
Ideal Selector	0.9968 (±0.0028)	0.0147 (±0.0096)
Optimal Linear Combiner	0.9907 (±0.0059)	0.0637 (±0.0371)
Mean	0.9900 (±0.0057)	0.0323 (±0.0121)
Product	0.9830 (±0.0063)	0.0377 (±0.0120)
LDA	0.9878 (±0.0059)	0.0406 (±0.0133)
DSS RME	0.9836 (±0.0066)	0.0367 (±0.0131)
DSS k-NN	0.9846 (±0.0063)	0.0365 (±0.0130)
DSS LDC	0.9820 (±0.0062)	0.0429 (±0.0134)
DSS QDC	0.9867 (±0.0063)	0.0321 (±0.0131)
DSS Parzen	0.9847 (±0.0063)	0.0363 (±0.0133)
k-NN Ppost	0.9789 (±0.0085)	0.0322 (±0.0139)
LDC Ppost	0.9878 (±0.0059)	0.0364 (±0.0121)
QDC Ppost	0.9907 (±0.0056)	0.0297 (±0.0117)
Parzen Ppost	0.9876 (±0.0068)	0.0320 (±0.0118)

Table 5.8: Light category, SD_δ sorted. Mean and standard deviation among the experiments using for each experiment the 10 pairs of experts with the highest SD_δ, and the 10 pairs of experts with the lowest SD_δ.

large D'_δ	**AUC**	**EER**
Average Expert	0.9906 (±0.0014)	0.0274 (±0.0044)
Ideal Selector	0.9985 (±0.0012)	0.0066 (±0.0025)
Optimal Linear Combiner	0.9962 (±0.0015)	0.0258 (±0.0132)
Mean	0.9960 (±0.0016)	0.0152 (±0.0039)
Product	0.9904 (±0.0022)	0.0205 (±0.0033)
LDA	0.9951 (±0.0024)	0.0215 (±0.0071)
DSS RME	0.9919 (±0.0021)	0.0173 (±0.0033)
DSS k-NN	0.9929 (±0.0020)	0.0169 (±0.0038)
DSS LDC	0.9908 (±0.0018)	0.0227 (±0.0041)
DSS QDC	0.9941 (±0.0020)	0.0147 (±0.0037)
DSS Parzen	0.9931 (±0.0020)	0.0165 (±0.0040)
k-NN Ppost	0.9912 (±0.0026)	0.0146 (±0.0067)
LDC Ppost	0.9951 (±0.0024)	0.0182 (±0.0065)
QDC Ppost	0.9958 (±0.0016)	0.0143 (±0.0035)
Parzen Ppost	0.9952 (±0.0022)	0.0151 (±0.0035)

small D'_δ	**AUC**	**EER**
Average Expert	0.6913 (±0.0664)	0.3304 (±0.0757)
Ideal Selector	0.9083 (±0.0912)	0.0967 (±0.0959)
Optimal Linear Combiner	0.8419 (±0.1525)	0.2798 (±0.1927)
Mean	0.8412 (±0.1527)	0.1968 (±0.1679)
Product	0.5436 (±0.0439)	0.4632 (±0.0392)
LDA	0.8352 (±0.1552)	0.1847 (±0.1701)
DSS RME	0.5995 (±0.0396)	0.4063 (±0.0329)
DSS k-NN	0.8539 (±0.1080)	0.1552 (±0.1116)
DSS LDC	0.7884 (±0.0814)	0.2188 (±0.0920)
DSS QDC	0.8591 (±0.1084)	0.1571 (±0.1148)
DSS Parzen	0.8529 (±0.0937)	0.1552 (±0.0991)
k-NN Ppost	0.9000 (±0.0691)	0.1380 (±0.0954)
LDC Ppost	0.8376 (±0.1547)	0.2016 (±0.1704)
QDC Ppost	0.9124 (±0.0700)	0.1441 (±0.1119)
Parzen Ppost	0.9240 (±0.0486)	0.1249 (±0.0719)

Table 5.9: Light category, D'_δ sorted. Mean and standard deviation among the experiments using for each experiment the 10 pairs of experts with the highest D'_δ, and the 10 pairs of experts with the lowest D'_δ.

than the *optimal linear combiner*, and consequently better than the other remaining methods. Also in this case the posterior probabilities method based on Parzen exhibits an AUC larger than the *ideal score selector*, but the *ideal score selector* is the best method if the EER is considered. If the small values of the EER_δ are taken into account the best combination methods in terms of AUC are the Mean rule and the posterior probabilities based on the QDC and the Mean rule. While in terms of EER all the combination rules outperform the "average expert", for this measure the best performance are achieved by the posterior probabilities based on the k-NN and the QDC.

In Table (5.8) the pairs of experts sorted using their SD_δ values are reported. When the value of the SD_δ is high the performance of all the combined methods present improvements with respect to the "average expert". In these results the best combination method is that one that uses the posterior probabilities based on the Parzen. Considering the remaining methods, the DSS based on the k-NN allows to obtain the highest performance in terms of EER. When the value of the SD_δ is small the vast majority of all the combined methods present improvements in terms of AUC with respect to the "average expert", while it is outperformed by all the combination methods in terms of EER. In this results the best method is the one that uses the posterior probabilities based on the QDC.

Table (5.9) shows the pairs of experts sorted using their D'_δ values. When the value of the D'_δ is high the performance of all the combined methods, but the Product rule, present improvements in terms of AUC with respect to the "average expert". If the results obtained for an high value of the D'_δ are compred with the results exposed in Tables (5.6 - 5.8) for large AUC_δ, small EER_δ, and small SD_δ it is easy to see that the D'_δ allows to choose the best experts to be combined, and thus the combination methods achieve better performance, as it happens when the Open category is taken into account. in these results the DSS methods, but that based on LDC, the methods that use the posterior probabilities, and the Mean rule have closer performance in terms of EER. When the value of the D'_δ is small the performance of all the combined methods present improvements with respect to the "average expert". In this case the DSS methods and the methods based on the posterior probabilities (except those based on the LDC) outperform the other combination rules and the *optimal linear combiner*.

Finally, reported results show that the *ideal score selector* always outperforms the *optimal linear combiner*, thus confirm that the selection strategy is an alternative to linear combination strategies. In some cases in the Light category, the methods based on the posterior probabilities outperform the *ideal score selector*, this fact can be explained as follows. The *ideal score selector* depends from the "quality" of the scores produced by the individual experts, and in the case of the Light category this "quality" is very low in some cases[6]. Thus, the influence of the score distribution is evident from Table (5.7) where very high values of EER implies that the two distributions are highly overlapped, and the *ideal score selector* works in a "one-dimensional" space (i.e., the score space). Instead the methods based on the posterior probabilities produce scores that can be different from those produced by the individual experts relying on a "N-dimensional" space (where N is the number of the experts combined). Thus, exploiting extra information in some cases the performance of this "fusion" methods can outperform the ideal "selection" methods that relies only on scores. It is also clear that this behaviour also depends by the sorting measure used, infact when the D'_δ is used the *ideal score selector* outperforms all the other methods as the "best" experts are chosen according to the results exposed in Section 5.3. Taking into account the DSS methods when the experts with high performance are combined (i.e., large AUC_δ, small EER_δ, small SD_δ, and large D'_δ) the best method is that based on the QDC. While in the case of low performance the best DSS methods are those based on the QDC and the Parzen windows. While when the methods that use the posterior probabilities are used the best methods are those based on the QDC and the Parzen windows. While these methods take advantages of the posterior probabilities obtained as "fused" score, the

[6] Cases of low "quality": high number of cases where the *negative* scores are higher than the *positive* scores, score distributions highly overlapped, etc.

DSS methods show their best performance especially when the low “quality” scores are used, and in some cases they also outperforms the *optimal linear combiner*. From these results it is also clear that their performance varies also according to the performance of the classifier used.

5.5 Experimental results on NIST BSSR1

In this section the methods and performance measures described in this thesis are applied to the NIST Biometric Scores Set Release 1.

These experiments are made comparing the following methods: the *ideal score selector*, the *linear combiner*, the Mean rule, the Max rule, the Min rule, the Product rule, the LDA, the DSS based on three classifier (k-NN, Linear Discriminant, Quadratic Discriminant), and the use of posterior probabilities of the three classifier (k-NN, Linear Discriminant, Quadratic Discriminant).

The performance are assessed using the following measures: the AUC, the EER, the d’, the FMR 1%, the FNMR 1%, the FMR 0%, and the FNMR 0%.

In these experiments the experts have not been sorted in a particular way because they are only four in number, so all the possible combinations have been tested. Thus, Tables (5.10) and (5.14) presents the results obtained combining two experts at a time. Tables (5.11) and (5.15) show the results when three experts at a time are combined. Tables (5.12) and (5.16) presents the results obtained combining all four the experts. Tables (5.13) and (5.17) contain the results obtained taking into account all the possible combinations (i.e., all the combination of two experts, all the combination of three experts, and all the combination of four) and give a summary snapshot of the global experiment. The experiments were run using the subdivisions described at the beginning of this chapter. The results are expressed by mean of the average value and the standard deviation among the experiments run.

Tables (5.10 - 5.12) show the results in terms of AUC, EER, and d’ when increasing number of experts is combined. The results exposed in the tables

	AUC	**EER**	**d'**
Ideal Selector	0.9996(±0.0010)	0.0026(±0.0061)	13.9975(±10.0984)
Optimal Linear Combiner	0.9968(±0.0035)	0.0162(±0.0121)	2.8193(±0.3729)
Mean	0.9951(±0.0043)	0.0217(±0.0151)	3.1184(±0.5626)
Max	0.9887(±0.0029)	0.0445(±0.0092)	2.9053(±0.5193)
Min	0.9732(±0.0098)	0.0655(±0.0159)	2.2459(±0.4712)
Product	0.9856(±0.0089)	0.0469(±0.0166)	2.1821(±0.6278)
LDA	0.9887(±0.0073)	0.0431(±0.0166)	2.2538(±0.3116)
DSS k-NN	0.9859(±0.0068)	0.0377(±0.0153)	4.6801(±1.4672)
DSS LDC	0.9743(±0.0103)	0.0634(±0.0162)	2.7359(±0.5278)
DSS QDC	0.9833(±0.0112)	0.0453(±0.0228)	5.3653(±2.9463)
k-NN Ppost	0.9699(±0.0226)	0.0340(±0.0238)	5.4099(±1.7641)
LDC Ppost	0.9891(±0.0076)	0.0429(±0.0164)	2.4899(±0.3568)
QDC Ppost	0.9953(±0.0047)	0.0214(±0.0161)	6.2254(±2.6977)

Table 5.10: Mean and standard deviation of AUC, EER and d' for the combination methods in NIST BSSR1 dataset using all possible score combinations using two experts at a time.

	AUC	**EER**	**d'**
Ideal Selector	1.0000(±0.0000)	0.0000(±0.0000)	25.4451(±8.7120)
Optimal Linear Combiner	0.9997(±0.0004)	0.0050(±0.0031)	3.1231(±0.2321)
Mean	0.9982(±0.0013)	0.0096(±0.0059)	3.6272(±0.4850)
Max	0.9892(±0.0022)	0.0450(±0.0048)	3.0608(±0.3803)
Min	0.9708(±0.0085)	0.0694(±0.0148)	2.0068(±0.1636)
Product	0.9919(±0.0036)	0.0309(±0.0102)	1.7205(±0.3705)
LDA	0.9945(±0.0040)	0.0296(±0.0123)	2.3802(±0.2036)
DSS k-NN	0.9917(±0.0050)	0.0236(±0.0072)	6.4094(±0.9884)
DSS LDC	0.9728(±0.0088)	0.0647(±0.0148)	2.7256(±0.2699)
DSS QDC	0.9919(±0.0086)	0.0243(±0.0182)	8.7278(±3.0557)
k-NN Ppost	0.9928(±0.0075)	0.0101(±0.0051)	7.4304(±1.0463)
LDC Ppost	0.9947(±0.0038)	0.0300(±0.0116)	2.5632(±0.2712)
QDC Ppost	0.9985(±0.0005)	0.0094(±0.0050)	9.3220(±2.6571)

Table 5.11: Mean and standard deviation of AUC, EER and d' for the combination methods in NIST BSSR1 dataset using all possible score combinations of three experts.

point out that when an increasing number of expert is combined generally there is an improvement of the performance if compared to the performance of the single experts exposed in Table (5.1). From the experiments it is also clear that the classifier based methodologies and the selection methodologies allows to increase the d' more than the other performance measures taken

	AUC	**EER**	**d'**
Ideal Selector	1.0000(±0.0000)	0.0000(±0.0000)	31.1087(±0.6621)
Optimal Linear Combiner	1.0000(±0.0000)	0.0017(±0.0010)	3.3325(±0.3414)
Mean	0.9996(±0.0003)	0.0045(±0.0013)	3.9444(±0.0599)
Max	0.9898(±0.0026)	0.0427(±0.0035)	3.0807(±0.1952)
Min	0.9700(±0.0054)	0.0742(±0.0080)	1.9082(±0.0421)
Product	0.9934(±0.0027)	0.0218(±0.0062)	1.3643(±0.0137)
LDA	0.9977(±0.0019)	0.0200(±0.0068)	2.5846(±0.0686)
DSS k-NN	0.9955(±0.0023)	0.0199(±0.0067)	7.4568(±1.1288)
DSS LDC	0.9744(±0.0055)	0.0632(±0.0098)	2.8457(±0.2316)
DSS QDC	0.9966(±0.0050)	0.0138(±0.0124)	11.1721(±3.1888)
k-NN Ppost	0.9973(±0.0051)	0.0058(±0.0033)	8.6911(±1.1626)
LDC Ppost	0.9973(±0.0017)	0.0206(±0.0069)	2.7273(±0.2211)
QDC Ppost	0.9986(±0.0001)	0.0070(±0.0013)	11.2798(±2.7403)

Table 5.12: Mean and standard deviation of AUC, EER and d' for the combination methods in NIST BSSR1 dataset using all possible score combinations from all four experts.

	AUC	**EER**	**d'**
Ideal Selector	0.9998(±0.0007)	0.0014(±0.0047)	19.7158(±11.1171)
Optimal Linear Combiner	0.9981(±0.0030)	0.0108(±0.0109)	2.9764(±0.3673)
Mean	0.9966(±0.0037)	0.0158(±0.0134)	3.3785(±0.5849)
Max	0.9890(±0.0026)	0.0445(±0.0074)	2.9778(±0.4515)
Min	0.9721(±0.0089)	0.0677(±0.0150)	2.1283(±0.3820)
Product	0.9886(±0.0076)	0.0388(±0.0165)	1.9399(±0.5833)
LDA	0.9916(±0.0068)	0.0361(±0.0165)	2.3298(±0.2774)
DSS k-NN	0.9889(±0.0068)	0.0310(±0.0143)	5.5614(±1.6175)
DSS LDC	0.9738(±0.0093)	0.0639(±0.0150)	2.7421(±0.4234)
DSS QDC	0.9876(±0.0110)	0.0348(±0.0234)	7.1159(±3.5801)
k-NN Ppost	0.9807(±0.0210)	0.0227(±0.0217)	6.4429(±1.8893)
LDC Ppost	0.9919(±0.0068)	0.0362(±0.0160)	2.5382(±0.3194)
QDC Ppost	0.9967(±0.0038)	0.0157(±0.0137)	7.8109(±3.2018)

Table 5.13: Mean and standard deviation of AUC, EER and d' for the combination methods in NIST BSSR1 dataset using all possible score combinations.

into account.

In Tables (5.10) and (5.11) the best combination method (excluding the *ideal score selector* and the *optimal linear combiner*) is the QDC using its posterior probabilities as combined score, followed by the Mean rule. While generally all the other methods exhibits closer performance in terms of AUC and EER. In Table (5.12) the best results in terms of AUC and EER are achieved by the Mean rule followed by the QDC using its posterior probabilities as “fused” score. Table (5.13) gives a global snapshot of all the experiments in terms of AUC, EER, and d’. Thus among the “fixed”(i.e., Mean rule, Max rule, Min rule, Product rule) combination rules and the LDA the best performance are achieved by the Mean rule followed by the LDA. Among the DSS methodologies the global better performance in terms of AUC and EER are obtained when a k-NN classifier is used (but the QDC based is closer in performance), while in terms of d’ better performance are achieved when a QDC classifier is used. Between the methods that use the estimated posterior probabilities the best performance are achieved by those estimated using a QDC classifier.

Tables (5.14 - 5.16) show the results in terms of FMR 1%, FNMR 1%, FMR 0%, and FNMR 0% when increasing number of experts is combined[7]. As for the results exposed in Tables (5.10 - 5.13), these results exposed in the tables point out that when an increasing number of expert is combined generally there is an improvement of the performance if compared to the performance of the single experts exposed in Table (5.1). Also in this case the best performance (except the *ideal score selector* and the *optimal linear combiner*) are achieved by the QDC using its posterior probabilities as combined score, and the Mean rule.

In Table (5.14) the best combination method (excluding the *ideal score selector* and the *optimal linear combiner*) in terms of FMR 1% is Mean rule, followed by the QDC using its posterior probabilities as combined score. Moreover the DSS based on k-NN and QDC, the k-NN with the posterior probabilities exhibits similar behaviour, they are followed in performance by the others. In the case of FMR 0% the DSS methods exhibits better performance than the methods based on the posterior probabilities. Tables (5.15 - 5.16) exhibit a similar behaviour than Table (5.14), in these tables better performance are achieved by using the posterior probabilities using the

[7] For these performance measures, as for the EER, the smaller the value, the better the performance

QDC in terms of FMR 1% and FNMR 0%, instead the Mean rule have better performance in term of FMR 0%. Table (5.17) gives a global snapshot of all the experiments in terms of FMR 1%, FNMR 1%, FMR 0%, and FNMR 0%. Thus among the “fixed”(i.e., Mean rule, Max rule, Min rule, Product rule) combination rules and the LDA the best performance are achieved by the Mean rule. Among the DSS methodologies the global better performance in terms of FMR 1%, FNMR 1% and FMR 0% are obtained when a k-NN classifier is used, while in terms of FNMR 0% better performance are achieved when a QDC classifier is used. Between the methods that use the estimated posterior probabilities the best performance are achieved by those estimated using a QDC classifier if the FMR 1%, FNMR 1% and FNMR 0% are considered, while the best performance in terms of FMR 0% are achieved when a LDC classifier is used. Moreover in Tables (5.16 - 5.17) in some cases the Mean rule exhibits better performance than the linear combiner, this is due to the fact that the linear combiner is built to optimize the AUC, as explained in the previous section.

From all the Tables (5.10 - 5.17) it is clear that the *ideal score selector* always outperforms the *optimal linear combiner*. For the other combination methods, globally it can be said that method based on the posterior probabilities using the QDC is the best combination method of those based on the use of classifiers, and is mostly comparable with the Mean rule that for this dataset is a good combination rule. Moreover generally the DSS based on k-NN and QDC exhibits better performance than the Max rule and the Min rule, this aspect is important because the ideal selection can be viewed as a function that “switches” between these two combination rules.

	FMR 1%	**FNMR 1%**	**FMR 0%**	**FNMR 0%**
Ideal Selector	0.0029(±0.0069)	0.0136(±0.0422)	0.0099(±0.0224)	0.0646(±0.1584)
Optimal Linear Combiner	0.0242(±0.0260)	0.0711(±0.1204)	0.1703(±0.1530)	0.4973(±0.4085)
Mean	0.0288(±0.0257)	0.1182(±0.1669)	0.1315(±0.1014)	0.6354(±0.3531)
Max	0.0822(±0.0218)	0.2957(±0.1278)	0.4282(±0.1537)	0.7749(±0.1998)
Min	0.1012(±0.0248)	0.6634(±0.1461)	0.2875(±0.0869)	0.9803(±0.0432)
Product	0.0675(±0.0259)	0.3660(±0.2217)	0.2172(±0.0794)	0.8618(±0.2251)
LDA	0.0722(±0.0279)	0.2581(±0.1798)	0.2593(±0.1327)	0.7711(±0.2433)
DSS k-NN	0.0548(±0.0305)	0.4437(±0.2507)	0.3610(±0.1874)	0.9017(±0.1443)
DSS LDC	0.0957(±0.0246)	0.6555(±0.1614)	0.2830(±0.1238)	0.9803(±0.0432)
DSS QDC	0.0607(±0.0346)	0.4782(±0.2634)	0.3884(±0.1552)	0.8038(±0.2693)
k-NN Ppost	0.0629(±0.0453)	0.8386(±0.3690)	0.5712(±0.4282)	0.9427(±0.1830)
LDC Ppost	0.0737(±0.0268)	0.2563(±0.1809)	0.2712(±0.1300)	0.5947(±0.2785)
QDC Ppost	0.0357(±0.0367)	0.0919(±0.1386)	0.7917(±0.3393)	0.3554(±0.2796)

Table 5.14: Mean and standard deviation of FMR 1%, FNMR 1%, FMR 0%, and FNMR 0% for the combination methods in NIST BSSR1 dataset using all possible score combinations of two experts.

	FMR 1%	**FNMR 1%**	**FMR 0%**	**FNMR 0%**
Ideal Selector	0.0000(±0.0000)	0.0000(±0.0000)	0.0000(±0.0000)	0.0000(±0.0000)
Optimal Linear Combiner	0.0027(±0.0047)	0.0025(±0.0037)	0.0741(±0.0328)	0.0920(±0.1232)
Mean	0.0090(±0.0075)	0.0175(±0.0245)	0.0530(±0.0210)	0.4899(±0.4020)
Max	0.0851(±0.0118)	0.2304(±0.0841)	0.5055(±0.0839)	0.8172(±0.2128)
Min	0.1044(±0.0229)	0.6994(±0.1220)	0.2471(±0.0643)	0.9964(±0.0093)
Product	0.0406(±0.0145)	0.1938(±0.1452)	0.1420(±0.0330)	0.8802(±0.2352)
LDA	0.0483(±0.0270)	0.1300(±0.1026)	0.1575(±0.0867)	0.5325(±0.3085)
DSS k-NN	0.0276(±0.0077)	0.3055(±0.2234)	0.3539(±0.1945)	0.7331(±0.2427)
DSS LDC	0.0980(±0.0227)	0.6853(±0.1237)	0.2081(±0.0577)	0.9964(±0.0093)
DSS QDC	0.0276(±0.0218)	0.2677(±0.2733)	0.4293(±0.1395)	0.6231(±0.3737)
k-NN Ppost	0.0158(±0.0137)	0.4389(±0.5110)	0.2504(±0.3734)	0.5657(±0.4619)
LDC Ppost	0.0498(±0.0266)	0.1294(±0.1024)	0.2031(±0.0893)	0.3798(±0.2264)
QDC Ppost	0.0087(±0.0101)	0.0103(±0.0176)	0.6856(±0.4212)	0.1493(±0.1880)

Table 5.15: Mean and standard deviation of FMR 1%, FNMR 1%, FMR 0%, and FNMR 0% for the combination methods in NIST BSSR1 dataset using all possible score combinations of three experts.

	FMR 1%	**FNMR 1%**	**FMR 0%**	**FNMR 0%**
Ideal Selector	0.0000(±0.0000)	0.0000(±0.0000)	0.0000(±0.0000)	0.0000(±0.0000)
Optimal Linear Combiner	0.0000(±0.0000)	0.0003(±0.0003)	0.0354(±0.0226)	0.0014(±0.0007)
Mean	0.0039(±0.0026)	0.0001(±0.0001)	0.0251(±0.0044)	0.1525(±0.0985)
Max	0.0800(±0.0122)	0.2096(±0.1180)	0.5442(±0.0509)	0.9060(±0.1821)
Min	0.1122(±0.0121)	0.6942(±0.0925)	0.2186(±0.0107)	0.9981(±0.0038)
Product	0.0309(±0.0076)	0.1225(±0.0587)	0.1102(±0.0060)	0.8510(±0.2980)
LDA	0.0251(±0.0108)	0.0743(±0.0578)	0.0825(±0.0075)	0.2342(±0.2240)
DSS k-NN	0.0226(±0.0080)	0.1394(±0.0680)	0.2996(±0.2202)	0.5669(±0.0032)
DSS LDC	0.0974(±0.0155)	0.6378(±0.0818)	0.1663(±0.0188)	0.9981(±0.0038)
DSS QDC	0.0142(±0.0131)	0.0903(±0.1739)	0.4669(±0.1577)	0.3998(±0.3950)
k-NN Ppost	0.0058(±0.0100)	0.2514(±0.4990)	0.0574(±0.0180)	0.2566(±0.4956)
LDC Ppost	0.0271(±0.0100)	0.0760(±0.0567)	0.1967(±0.0751)	0.2352(±0.2225)
QDC Ppost	0.0019(±0.0025)	0.0040(±0.0013)	0.3067(±0.4624)	0.1125(±0.2065)

Table 5.16: Mean and standard deviation of FMR 1%, FNMR 1%, FMR 0%, and FNMR 0% for the combination methods in NIST BSSR1 dataset using all possible score combinations of four experts.

	FMR 1%	**FNMR 1%**	**FMR 0%**	**FNMR 0%**
Ideal Selector	0.0016(±0.0053)	0.0074(±0.0316)	0.0054(±0.0171)	0.0353(±0.1203)
Optimal Linear Combiner	0.0142(±0.0222)	0.0398(±0.0947)	0.1231(±0.1256)	0.3048(±0.3750)
Mean	0.0193(±0.0220)	0.0709(±0.1337)	0.0933(±0.0866)	0.5386(±0.3791)
Max	0.0831(±0.0178)	0.2642(±0.1159)	0.4669(±0.1312)	0.8022(±0.2024)
Min	0.1034(±0.0231)	0.6793(±0.1323)	0.2666(±0.0780)	0.9877(±0.0331)
Product	0.0544(±0.0255)	0.2813(±0.2076)	0.1801(±0.0743)	0.8675(±0.2296)
LDA	0.0592(±0.0304)	0.1948(±0.1623)	0.2062(±0.1262)	0.6355(±0.3123)
DSS k-NN	0.0420(±0.0270)	0.3658(±0.2467)	0.3528(±0.1888)	0.8100(±0.2100)
DSS LDC	0.0967(±0.0229)	0.6648(±0.1415)	0.2452(±0.1062)	0.9877(±0.0331)
DSS QDC	0.0444(±0.0340)	0.3664(±0.2880)	0.4104(±0.1487)	0.7013(±0.3391)
k-NN Ppost	0.0406(±0.0423)	0.6399(±0.4821)	0.4078(±0.4269)	0.7433(±0.4067)
LDC Ppost	0.0608(±0.0297)	0.1938(±0.1625)	0.2397(±0.1159)	0.4839(±0.2819)
QDC Ppost	0.0228(±0.0310)	0.0543(±0.1101)	0.7090(±0.3968)	0.2584(±0.2623)

Table 5.17: Mean and standard deviation of FMR 1%, FNMR 1%, FMR 0%, and FNMR 0% for the combination methods in NIST BSSR1 dataset using all possible score combinations.

Chapter 6

Conclusions and future work

Binary classifiers (or binary experts) is one of the interesting problems of the Pattern Recognition field. In particular, this thesis focuses on the cases when a similarity score is assigned by binary experts to a pattern. Examples of this situation are biometric authentication, spam filtering, medical test etc. In the experimental phase, this thesis focus its attention on the biometric authentication problem.

Ensemble of binary experts are used to improve the performance of a system. Infact the combination of different experts is generally used to exploit different information provided by the individual experts. Two main combination approaches exist: "fusion" and "selection". Fusion approaches aim at producing a new output as a function of the outputs (crisp or continuous) of the experts. On the other hand, selection approaches aim at selecting, for each input pattern, the most suited expert for that pattern. Usually for ensemble of experts that output similarity scores, "fusion" approach is used.

This thesis introduce the selection approach for this kind of experts. In particular the problem of selecting scores is outlined. Thus, an ideal framework of the selection of scores is proposed by means of the *ideal score selector* (Chapter 2). This *ideal score selector* selects the maximum score for the *positive* patterns, and the minimum score for the *negative* patterns. In particular, the properties of this *ideal selector* are derived, showing that the *ideal score* *selector* always outperforms the *optimal linear combiner* developed to maximize the value of the Area Under the ROC Curve in a linear combination of scores. These properties are also evident in the experimental phase. This confirms that the selection strategy is an alternative to linear combination strategies.

In order to implement the score selection, different *Dynamic Score Selection* methods are proposed. These methods estimate the state of nature of a pattern, and select the scores accordingly. The state of nature is estimated in two ways: one estimating the error of making the wrong selection (e.g., selecting the highest score for a *negative* pattern), the other use a classifier trained on a feature space where the features of each pattern are the scores assigned to that pattern by the experts. Among the methods proposed, those that rely on the use of classifier are the best suited. The proposed Dynamic Score Selection prove its effectiveness especially in the case of low "quality" scores are used (e.g., high number of cases where the *negative* scores are higher than the *positive* scores, score distributions highly overlapped, etc.), and in some cases they also outperforms the *optimal linear combiner*. In the majority of cases the performance generally are close to those achieved by the other methods used for comparison. From the results it is also clear that their performance varies also according to the performance of the classifier used.

Moreover, the use of the posterior probabilities assigned by a classifier as a "fused" score is proposed and investigated. Generally these methods achieve better results than the other methods. In some experiments, these methods outperform the *ideal score selector*. This fact can be explained as follows. The *ideal score selector* depends from the "quality" of the scores produced by the individual experts, and in that cases the "quality" of the score is very low. This means that the two score distributions are highly overlapped. While the *ideal score selector* works in a "one-dimensional" space (i.e., the score space), the methods based on the posterior probabilities can exploit extra information because they rely on a "N-dimensional" space (where N is the number of the experts combined). This behaviour also depends by the potential of the classifier used, and by the effectiveness of the ensemble of experts to be combined.

In this thesis it is also investigated the problem of how to measure the effectiveness of the ensemble of experts to be combined. This aspect is highly relevant when, instead of all the experts in the ensemble, only a subset of the ensemble is used (e.g., in practice only a few number of experts is combined). Thus, different measures of ensemble effectiveness have been proposed (Section 4.1). The proposed measures are based on the Area Under the ROC Curve, the Equal Error Rate, the d', and the Score Dissimilarity index. These measures are tested in an extensive experimental test. This experimental test clearly indicates that the measure based on d' is a good measure to estimate the ensamble effectiveness, as the larger the measure, the better the performance of the methods that combine the ensemble of experts.

Moreover from the experiments can be observed that the "selection" and the "fusion" approach are two alternatives. No one on them is better than the other, as it is well known in the Pattern Recognition field. Infact both of them have pros and cons, and the use of one or the other approach it is mainly characterized by the faced problem.

6.1 Future work

Future work is needed along all the methods and the theories developed during this thesis. Although the *ideal score selector* described in Chapter 2 seems mature as an ideal "selection" methodology, other work is needed to improve the some parts of the theoretical aspects of the methods that approximate this methodology. Along with this research newer Dynamic Score Selection methodologies have to be researched.

As it has pointed out above, from the experiments is clear that neither the "selection" approach, neither the "fusion" approach is the best generally speaking. Thus, one direction for future research is to develop methods that combine these approaches. The starting point is to use some of the measures developed to measure of ensemble effectiveness to decided when “select” or when “fuse” the scores.

Moreover the study of the generic use of classifiers for the combination of scores need a more deep study, in particular a better way of exploiting the information of the posterior probabilities when they are present. The aim of this future study is to obtain improvements in the Dynamic Score Selection or, as already said, to develop “mixed” methodologies that use both the “selection” and the “fusion” approach.

Bibliography

[1] R.P.W. Duin, F. Roli, and D. de Ridder. A note on core research issues for statistical pattern recognition. *Pattern Recognition Letters*, 23(4):493–499, 2002.

[2] R.O. Duda, P.E. Hart, and D.G Stork. *Pattern Classification*. John Wiley & Sons, Inc., 2001.

[3] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[4] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[5] L.I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons, Inc., 2004.

[6] J. Kittler and F. Roli, editors. *Multiple Classifier Systems - First International Workshop (MCS2000)*, volume 1857 of *Lecture Notes in Computer Science*, Cagliari, Italy, 2000. Springer.

[7] J. Kittler and F. Roli, editors. *Multiple Classifier Systems - Second International Workshop (MCS2001)*, volume 2096 of *Lecture Notes in Computer Science*, Cambridge, UK, 2001. Springer.

[8] F. Roli and J Kittler, editors. *Multiple Classifier Systems - Third International Workshop (MCS2002)*, volume 2364 of *Lecture Notes in Computer Science*, Cagliari, Italy, 2002. Springer.

[9] T. Windeatt and F. Roli, editors. *Multiple Classifier Systems - 4th International Workshop (MCS2004)*, volume 2709 of *Lecture Notes in Computer Science*, Guilford, UK, 2003. Springer.

[10] F. Roli, J. Kittler, and T. Windeatt, editors. *Multiple Classifier Systems - 5th International Workshop (MCS2004)*, volume 3077 of *Lecture Notes in Computer Science*, Cagliari, Italy, 2004. Springer.

[11] N.C. Oza, R. Polikar, J. Kittler, and F. Roli, editors. *Multiple Classifier Systems - 6th International Workshop (MCS2005)*, volume 3541 of *Lecture Notes in Computer Science*, Seaside, CA, USA, 2005. Springer.

[12] M. Haindl, J. Kittler, and F. Roli, editors. *Multiple Classifier Systems - 7th International Workshop (MCS2007)*, volume 4472 of *Lecture Notes in Computer Science*, Prague, Czech Republic, 2007. Springer.

[13] G. Giacinto and F. Roli. Dynamic Classifier Selection. In *Proceedings of the First International Workshop on MCS*, volume LNCS 1857, pages 177–189. Springer-Verlang Berlin Heidelberg, 2000.

[14] K. Woods, W.P. Kegelmeyer, and K. Bowyer. Combination of Multiple Classifiers Using Local Accuracy Estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):405–410, 1997.

[15] A.P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[16] T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[17] J. Huang and C.X. Ling. Using AUC and Accuracy in Evaluating Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17:299–310, 2005.

[18] H.B. Mann and D.R. Whitney. On a test whether one or two random variable is stochastically larger than the other. *Annals of Mathematical Statistics*, 18:50 – 60, 1947.

[19] J.A. Hanley and B.J. McNeil. The meaning and the use of the area under a receiver operanting charateristic curve. *Radiology*, 143:29 – 36, 1982.

[20] A.K. Jain, R. Bolle, and S. Pankanti. *BIOMETRICS: Personal Identification in Networked society.* Kluwer Academic Publishers, 1999.

[21] L. Hong, A.K. Jain, and S. Pankanti. Can Multibiometrics Improve Performance? In *AutoID'99*, pages 59–64, 1999.

[22] A.K. Jain and A. Ross. Multibiometric Systems. *Communications of the ACM, Special Issue on Multimodal Interfaces*, 47(1):34–40, January 2004.

[23] A. Ross, K. Nandakumar, and A.K. Jain. *Handbook of Multibiometrics.* Springer-Verlang Berlin Heidelberg, 2006.

[24] A.K. Jain and A. Ross. Fingerprint mosaicking. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 4, pages 4064 – 4067, 2002.

[25] R. Tronci, G. Giacinto, and F. Roli. Dynamic Score Selection for Fusion of Multiple Biometric Matchers. In *ICIAP 2007: 14th International Conference on Image Analysis and Processing*, pages 15–20. IEEE Computer Society, 2007.

[26] J. Bigun, J. Fierrez-Aguilar, J. Ortega-Garcia, and J. Gonzales-Rodriguez. Multimodal Biometric Authentication using Quality Signals

in Mobile Communications. In *Proceedings of IAPR ICIAP*, pages 2–13, Mantova, Italy, 2003.

[27] J. Fierrez-Aguilar, L. Nanni, J. Ortega-Garcia, R. Cappelli, and D. Maltoni. Combining Multiple Matchers for Fingerprint Verification: A Case Study in FVC2004. In F. Roli and S Vitulano, editors, *Image Analysis and Processing - ICIAP 2005*, volume LNCS 3617, pages 1035 – 1042. Springer-Verlang Berlin Heidelberg, 2005.

[28] A.K. Jain, L. Hong, and Y. Kulkarni. A Multimodal Biometric System using Fingerprint, Face and Speech. In *Proceedings of Second International Conference on AVBPA*, pages 182–187, 1999.

[29] G. L. Marcialis and F. Roli. Fingerprint Verification by Fusion of Optical and Capacitive Sensors. *Pattern Recognition Letters*, 25(11):1315–1322, 2004.

[30] G.L. Marcialis and F. Roli. Fusion of multiple fingerprint matchers by single-layer perceptron with class-separation loss function. *Pattern Recognition Letters*, 26(12):1830–1839, 2005.

[31] S. Prabhakar and A. K. Jain. Decision-level Fusion in Biometric Verification. *Pattern Recognition*, 35(4):861–874, 2002.

[32] G. Giacinto, F. Roli, and R. Tronci. Score Selection Techniques for Fingerprint Multi-modal Biometric Authentication. In F. Roli and S. Vitulano, editors, *Image Analysis and Processing - ICIAP 2005*, pages 1018 – 1025. Springer-Verlang Berlin Heidelberg, 2005.

[33] A. Mood, F. Graybill, and D. Boes. *Introduction to the Theory of Statistics*. McGraw-Hill, 1974.

[34] C. Marrocco, M. Molinara, and F. Tortorella. Exploiting AUC for optimal linear combinations of dichotomizers. *Pattern Recognition Letters*, 27(8):900 – 907, 2006.

[35] G.O. Williams. The Use of d' As a "Decidability" Index. In *30th International Carnahan Conference on Security Technology*, pages 65–71, 1996.

[36] R. Tronci, G. Giacinto, and F. Roli. Selection of experts for the design of multiple biometric systems. In P. Perner, editor, *Machine Learning and Data Mining in Pattern Recognition*, volume LNAI 4571, pages 795–809. Springer-Verlang Berlin Heidelberg, 2007.

[37] R. Tronci, G. Giacinto, and F. Roli. Index Driven Combination of Multiple Biometric Experts for AUC Maximisation. In M. Haindl, J. Kittler, and F. Roli, editors, *Multiple Classifiers Systems*, volume LNCS 4472, pages 357–366. Springer-Verlang Berlin Heidelberg, 2007.

[38] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman, and A.K. Jain. FVC2004: Third Fingerprint Verification Competition. In *Proceedings ICBA*, pages 1 – 7, Hong Kong, 2004.