LEARNING CLASSIFICATION SYSTEMS MAXIMIZING THE AREA UNDER THE ROC CURVE

Claudio Marrocco

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
UNIVERSITÀ DEGLI STUDI DI CASSINO
VIA G. DI BIASIO, 43 I-03043 CASSINO (FR), ITALY
NOVEMBER 2006

UNIVERSITÀ DEGLI STUDI DI CASSINO

Date: **November 2006**

Author: **Claudio Marrocco**

Title: **Learning Classification Systems Maximizing the Area Under the ROC Curve**

Department: **Automazione, Elettromagnetismo, Ingegneria dell'Informazione e Matematica Industriale**

Degree: **Ph.D.**

Signature of Author

to my parents

Acknowledgements

This work would not have been possible without the support I received from many people. A big thank you to all who have helped me in some way or other to complete this thesis.

The most important person I would like to thank is my supervisor: Francesco Tortorella. In these three years he has guided me through this work. Having him as supervisor is a continuous source of motivation and gives you the feeling that you can do something useful. He is always ready to give you advice when you need, encouraging you to pursue your individual interest. I learnt a lot from our discussions on ROC curve and life. Thanks!

I would also like to thank prof. Bob Duin of the TU Delft. In Netherlands, I spent five beautiful months and his help was important to complete this thesis.

Then, my special thanks to all members of the LIT group, who ensured that it was always funny working there. Thanks for all the support during our "numerous" coffee breaks of everyday! They are friends more than colleagues.

Finally, my family, what would I have been without them? Thanks for everything.

Summary

This thesis is concerned with supervised classification problems in which the aim is to build a rule to assign objects to one of a finite set of classes. Systems able to perform these operations using a set of known examples are called classifiers. In particular, this work focuses on problems where we have to distinguish between two mutually exclusive classes. In this case, many distinct criteria for comparing performance of rules can be used. In this thesis an analysis of the Receiver Operating Characteristics (ROC) curve methodology in pattern recognition is performed and the use of the Area under the ROC curve (AUC) as performance measure for building dichotomizers and combination rules is proposed.

The thesis is organized as follows:

- *Chapter 1*: we introduce the framework of pattern recognition in which this work is placed. Starting from the basis of statistical pattern recognition we introduce the main problems of the topics of this thesis that are the two-class classification and in this context the combination of classifiers.

- *Chapter 2*: the ROC curve is introduced. We start by giving an overview of the performance measures depending on prior distributions and misclassification costs. Next, we present some topics of ROC analysis in pattern recognition and propose the AUC as a measure to evaluate the ranking of the classifier output on the two classes.

- *Chapter 2*: after an analysis of the linear discriminant functions in the light of the ROC analysis, we propose a non parametric classifier that performs a linear combination of features. In particular, a weight vector maximizing the ranking of the classifier through an iterative pairwise coupling of the features is evaluated. The proposed algorithm is tested on artificial and real data sets and compared with well known methods in literature.

- *Chapter 4*: we start with a brief review of the characteristics of classifier combination. Then, a technique for the optimal combination in terms of AUC between already trained dichotomizers is investigated. In particular, the dependence of the AUC on the weights of combination is analyzed and a method to find the optimal weight between two classifiers is proposed. Moreover, a greedy approach to extend the rule to several dichotomizers is presented and experiments on standard data sets are performed to confirm the effectiveness of the approach.

- *Chapter 5*: some conclusions and possible future works are presented.

- *Appendices*: three appendices are provided to render the work self-contained. Appendix A includes some notes on the data sets employed in the performed experiments. Appendix B is a review of the statistical tests used to assess a statistically significant difference in the performed while appendix C is a brief introduction to an algorithm of optimization used in the experiments.

Contents

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

List of Algorithms

Chapter 1

Introduction

The purpose of pattern recognition is to analyze a new object and assign it to one of a set of classes which are known beforehand. The complexity of this problem is hidden by the human capability in recognizing objects, understanding languages or reading characters. Systems able to perform these operations using a set of known examples are called classifiers. A common classification problem consists in distinguishing between two different mutually exclusive classes (two class or binary problem). In this thesis we will focus on the analysis of new techniques to better the performance of a classification system according to a particular performance measure that evaluates the probability of correct discrimination between the two classes.

In this first chapter we will give the framework in which this research is placed. Starting from the basis of statistical pattern recognition we will introduce the main problems of the topics of this work that are the binary classification and in this context the combination of classifiers. An outline of the thesis is presented in the last section of the chapter.

1.1 Learning from Data

Pattern recognition is a term used to cover the analysis of a problem through discrimination and classification imitating the ability of living organisms to learn. The goal of classification is to infer a rule which can assign the correct label and a function which outputs a class label for each input objects is called classifier. However, in many classification problems explicit rules do not exist but examples can be easily obtained. Therefore, in pattern recognition or machine learning we try to infer decision rules from a limited set of training examples. The examples are instances in some input space (pattern space) and the rules consist of general observations about the structure of the input space. We will use the

words object, pattern or sample to denote a Q-dimensional data vector $\mathbf{x} = (x_1, \ldots, x_Q)$ whose components $x_i \in \mathbb{R}$ are measurements of the features of an object. The features are the variables specified by the analyzer and thought to be important for classification (Webb, 2002). In our work we will assume that all the components in the vector are known and there are no missing values. In this way, each object can be represented as a point in a feature space X and the continuity assumption should hold: two objects that resemble each other in the real life should be near in the feature space (i.e. should belong to the same class).

In multiclass classification we assume that there exist C classes, say $\omega_1 \ldots \omega_C$ associated to each pattern $\mathbf{x}$. A general multiclass problem can always be decomposed in several binary classification problems (Fukunaga, 1990). Therefore, the two-class problem can be considered as the basic classification problem. In this case, we face with just two classes labelled as ω_1 and ω_2.

For the classification, a function $f(\mathbf{x})$ has to be evaluated from a training set. A training set can be defined as a set of objects (assumed to be independently distributed) where each one is associated with a label $\omega_i \in \{\omega_1, \omega_2\}$:

$$X^{\mathrm{tr}} = \{(\mathbf{x}_i, \omega_i) \, | i = 1 \ldots L\} \, . \tag{1.1}$$

The function f should be constructed such that for a given input vector $\mathbf{x}$ an estimate of the label is obtained:

$$\hat{\omega} = f(\mathbf{x}) \;\text{ with }\; f : \mathbb{R}^Q \rightarrow \{\omega_1, \omega_2\} \, . \tag{1.2}$$

Such a rule partitions the features space in two regions, one for the class ω_1 and the other for the class ω_2. Each region can be multiply connected, i.e. it may be made up of several disjoint regions. The boundaries between these regions are called decision boundaries (or decision surfaces). Generally, in the regions close to the boundaries there is the highest probability of misclassifications. An example of such a function is shown in fig. 1.1 where a linear function is assumed as classifier to separate the two classes ω_1 and ω_2 according to two measurements x_1 and x_2.

If we fix beforehand the decision function, in the training phase just some parameters for the function has to be determined. In every case, the most usual procedure to fix the correct set of parameters for the classifier is to minimize an error function on the training set. The classifier that exhibits the minimum error is the Bayes decision rule that assigns an object $\mathbf{x}$ to the class with the largest a posteriori probability $P(\omega_k|\mathbf{x})$ with $k = 1, 2$,

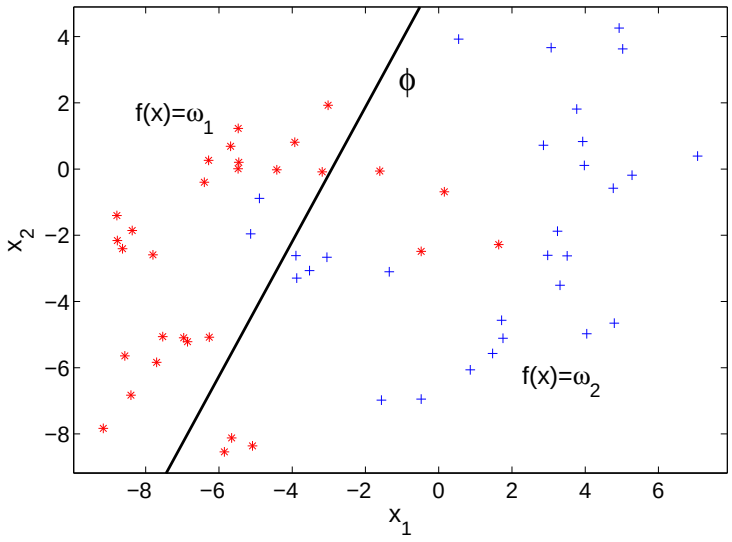


Figure 1.1: Plot of the training set available for a binary classification problem. A simple linear classifier is shown with its decision boundary. Some misclassifications are present due to the inability of the classifier to well separate the two classes.

i.e.:

$$f_{\text{Bayes}}(\mathbf{x}) = \begin{cases} \omega_1, & \text{if } P(\omega_1|\mathbf{x}) \geq P(\omega_2|\mathbf{x}), \\ \omega_2, & \text{if } P(\omega_1|\mathbf{x}) < P(\omega_2|\mathbf{x}). \end{cases} \tag{1.3}$$

The Bayes rule is the theoretical optimal rule, i.e. it has the best classification performance over all possible classifiers (Bishop, 1995). The problem is that this rule requires the real a posteriori probability of all the classes for all the samples, condition that is not verified in practice. However, it represents the best rule when there is a complete knowledge of the a posteriori probabilities and as a consequence it becomes useful in terms of comparison when artificial data distribution are used.

Indeed, since we have a finite number of examples, achieving optimal performance on the training set is not required. In general, the training set should be a characteristic set representative of the true operating conditions and moreover, the larger the sample size the better the characteristics of data can be determined. Therefore, the main goal of the classification is not to obtain optimal performance on the training data but it is the good classification of new and unseen objects. This property called generalization avoids to find an optimistic estimate of the performance of the classifier. As an example let us show in fig. 1.2 the same classes distribution of fig. 1.1 but now, instead of using a linear classifier, we choose a function that perfectly separates the two classes. In these conditions

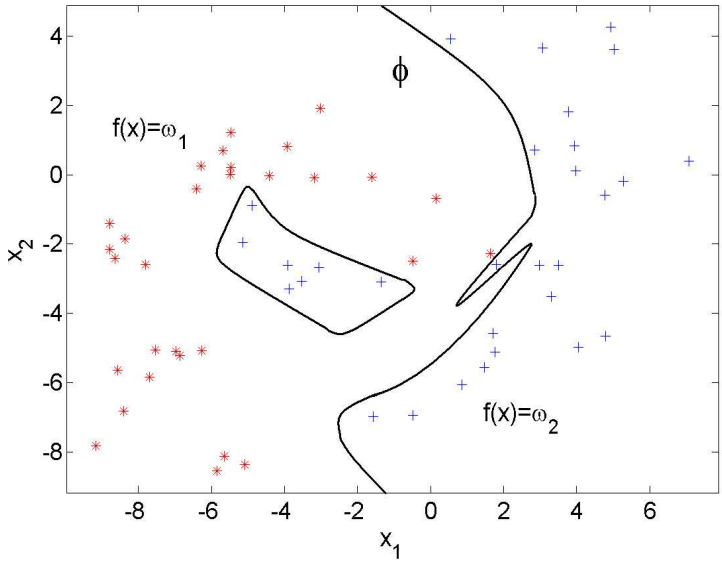


Figure 1.2: An example of overtrained classifier. This is able to have zero errors on the training set but it shows low generalization on the test set.

the function well describes the distribution of the training set but, in this way, it is not adapted to the real structure of data. This phenomenon is called overtraining or overfitting and it becomes more severe when a large number of features is used. Also the opposite problem, the underfitting, can occur when the function is not flexible enough to follow all characteristics in the data. Moreover, since the function $f(\mathbf{x})$ should be defined in the complete feature space, the volume that should be described increases exponentially in the number of features (the so called curse of dimensionality (Duda *et al.*, 2001)). Hence, if the classifier is too complex (i.e. there are too many free parameters) it may model the noise in the training set but if the classifier is not complex enough then it may fail to capture the real structure of the data (Webb, 2002).

In this situation although numerous classification functions are available, to find the real structure in the data is often very hard since it is difficult to find an optimal model that is able to handle all the details of the data. In fact, a classifier is optimal according to some quality measure but there are several ways of measuring classifier performance. In literature, the most employed measures are the error rate or the conditional risk that present several limitations since they depend on the priors of the classes. Therefore, a new topic in pattern recognition is to explore alternative criteria to build classifiers directly optimizing a desired measure. Moreover, it appears that the classification performance

can often be improved by extending one classifier to a combination of classifiers. The idea of combining classifiers is not a new one but it has received increasing attention in recent years. The use of different classifiers seems to be reasonable since using only an optimal classifier to solve a problem often disregards valuable information contained in other suboptimal classifiers that may be superior in some specific areas of the feature space.

In this work we focus on the Receiver Operating Characteristic (ROC) curve that is one of the most commonly used methods to summarize the discriminative ability of a classification system. In particular, in this thesis we will analyze methods to build a classification system trying to directly maximize the Area Under the ROC curve (AUC) that is a measure interpretable as the probability of correct discrimination between the two classes independently on priors and cost distributions of the classes. We focus on the search of both a classifier and a combination rule able to directly maximize the AUC. In the former case we propose a way to estimate the optimal weight of the linear combination of features so obtaining a ranker in the feature space able to maximize the AUC while in the latter case a similar method is proposed to combine two dichotomizers extending the method to several classifiers.

1.2 Outline of the thesis

In this chapter we introduced the basic problems of the pattern recognition and we gave some motivations for our work. In the next chapter the ROC methodology in pattern recognition is introduced and an analysis of the statistical properties of the AUC is presented. In the third chapter a non parametric classifier that performs a linear combination of features is presented and experiments to assess the reliability of the method are performed. Then, the fourth chapter focuses on the analysis of the combination rules and in particular, a linear combiner for the direct maximization of the AUC of a classification ensemble is proposed. Moreover, a new curve to evaluate the separability of the ranking of the classifier's output distributions is proposed and experimental results obtained for the comparison of our method with the literature are shown. Some conclusions and future developments are drawn in the last chapter. In the end, some appendices are reported including notes on the data sets and on the statistical tests employed in the performed experiments.

Chapter 2

The ROC Methodology in Pattern Recognition

Receiver Operating Characteristics (ROC) analysis is one of the most widely used methods to summarize the intrinsic properties of a classification system. ROC graphs are commonly used in medical decision making, and in recent years have been used increasingly in machine learning and pattern recognition research due to its capability to compare classifiers visualizing their performance. One of the most popular and convenient indices is the Area Under the ROC curve (AUC) that can be interpreted as the probability of correct discrimination between two different classes.

In this chapter we propose an introduction to the ROC curve and then to the AUC that is presented as a measure evaluating the ranking capability of the classifier output on the two classes. The goal is to give a view of these problems according to the research proposed in the next chapters.

2.1 Performance Measures

In this section we want to provide a general view of the several performance measures that have been proposed in literature to highlight the different facets explored using the AUC. To this aim let us consider a binary classification problem between two mutually exclusive classes (hereafter called *Positive* (P) and *Negative* (N)) class with priors π_P and π_N respectively. An evaluation of a trained model is based on the outcomes following the application on a test set. If we have a classifier f that provides for a given sample $\mathbf{x}$ an output $f(\mathbf{x}) \in \mathbb{R}$, without loss of generality we can say that $f(\mathbf{x})$ is a confidence degree that the sample belongs to one of the two classes, e.g. the class P. The sample should be

Table 2.1: The confusion matrix for a two class problem

		True Class	
		P	N
Predicted Class	P	True Positive	False Positive
	N	False Negative	True Negative

consequently assigned to the class N if $f(\mathbf{x}) \to -\infty$ and to the class P if $f(\mathbf{x}) \to +\infty$.

A threshold t is usually chosen, so as to attribute the sample $\mathbf{x}$ to the class N if $f(\mathbf{x}) \leq t$ and to the class P if $f(\mathbf{x}) > t$. For a given threshold value, two appropriate performance figures are given by the *True Positive Rate* ($TPR(t)$), i.e. the fraction of actually-positive cases correctly classified and by the *False Positive Rate* ($FPR(t)$), given by the fraction of actually-negative cases incorrectly classified as "positive". It is important to take into account both quantities for a particular choice of t since the consequences of false-negative and false-positive errors are often very different and hard to quantify.

Let us now suppose to know, besides the function $f(\mathbf{x})$, the class conditional probability densities on X, i.e. $p(\mathbf{x}|P)$ and $p(\mathbf{x}|N)$. In this case, we can also obtain the likelihoods of P and N with respect to f (i.e. the class conditional densities of the classifier score) $f_P(\tau) = p(f(\mathbf{x}) = \tau|\mathbf{x} \in P)$ and $f_N(\tau) = p(f(\mathbf{x}) = \tau|\mathbf{x} \in N)$. As a consequence, $TPR(t)$ and $FPR(t)$ are given by:

$$TPR(t) = \int_t^{+\infty} f_P(\tau)\, d\tau, \tag{2.1a}$$

$$FPR(t) = \int_t^{+\infty} f_N(\tau)\, d\tau. \tag{2.1b}$$

Taking into account samples with score less than the threshold it is also possible to define a *True Negative Rate* (TNR) and a *False Negative Rate* (FNR) as:

$$TNR(t) = \int_{-\infty}^{t} f_N(\tau)\, d\tau = 1 - FPR(t), \tag{2.2a}$$

$$FNR(t) = \int_{-\infty}^{t} f_P(\tau)\, d\tau = 1 - TPR(t). \tag{2.2b}$$

Although a confusion matrix as shown in table 2.1 can provide all of the information about the classifier quality, it is usual to extract measures from this matrix to illustrate specific aspects of the performance. As introduced in sec. 1.1, the most used measure is

the error rate Err defined as:

$$Err(t) = \pi_P FNR(t) + \pi_N FPR(t), \tag{2.3}$$

or its complementary measure, the accuracy Acc:

$$Acc(t) = \pi_P TNR(t) + \pi_N TPR(t) = 1 - Err(t). \tag{2.4}$$

The accuracy estimates the overall probability of correctly labelling a test sample, but combines the results for both classes in proportion to the priors (Swets, 1988).

In an imbalanced setting, where the prior probability of one class is significantly less than the others (we can say that the two classes are skewed defining as skew ϑ the ratio between π_P and π_N), accuracy is inadequate as a performance measure since it becomes biased towards one of the two classes (Provost & Fawcett, 1998), (Huang & Ling, 2005). In practice, when the skew increases accuracy loses the recognition capability towards the minority class. To some researchers, large class skews and large changes in class distributions may seem contrived and unrealistic (Fawcett, 2006). However, class skews of 10^1 and 10^2 are very common in real world domains, and skews up to 10^6 have been observed in some domains (Clearwater & Stern, 1991), (Fawcett & Provost, 1996), (Kubat *et al.*, 1998), (Saitta & Neri, 1998). Substantial changes in class distributions are not unrealistic either. For example, in medical decision making epidemics may cause the incidence of a disease to increase over time and in fraud detection, proportions of fraud varied significantly from month to month and place to place (Fawcett & Provost, 1997). In each of these examples the prevalence of a class may change drastically without altering the fundamental characteristic of the class.

In these situations, other performance measure that remains sensitive to the performance on each class can be proposed, an example is the precision Prec that expresses the fraction of the positives detected that are actually correct:

$$Prec(t) = \frac{TPR(t)}{TPR(t) + \vartheta FPR(t)}.$$

However, also this measure is related to a single decision threshold for a classification model.

A similar situation can be described for the misclassification costs related to each class.

In this case, a cost matrix can be defined as:

$$\mathbf{\Lambda} = \begin{pmatrix} \lambda_{PP} & \lambda_{NP} \\ \lambda_{PN} & \lambda_{NN} \end{pmatrix}, \tag{2.5}$$

where λ_{AB} is the cost of assigning a pattern to the class B when it actually belongs to the class A; the aim is to minimize the conditional risk (Risk) associated to the classification of a given sample $\mathbf{x}$ which in the two class case is defined as:

$$\begin{aligned} Risk(t) &= \pi_N \lambda_{NN} \int_{-\infty}^{t} f_N(\tau)\, d\tau + \pi_N \lambda_{NP} \int_{t}^{+\infty} f_N(\tau)\, d\tau \\ &+ \pi_P \lambda_{PN} \int_{-\infty}^{t} f_P(\tau)\, d\tau + \pi_P \lambda_{PP} \int_{t}^{+\infty} f_P(\tau)\, d\tau \\ &= \pi_N \lambda_{NN} TNR(t) + \pi_N \lambda_{NP} FPR(t) + \pi_P \lambda_{PN} FNR(t) \\ &+ \pi_P \lambda_{PP} TPR(t) = \pi_N (\lambda_{NP} - \lambda_{NN}) FPR(t) \\ &+ \pi_P (\lambda_{PP} - \lambda_{PN}) TPR(t) + \pi_P \lambda_{PN} + \pi_N \lambda_{NN}. \end{aligned} \tag{2.6}$$

If we assign the sample to one of the two classes, the Risk becomes:

$$Risk = \begin{cases} \frac{\lambda_{NN}\pi_N f_N(f(\mathbf{x})) + \lambda_{PN}\pi_P f_P(f(\mathbf{x}))}{\pi_N f_N(f(\mathbf{x})) + \pi_P f_P(f(\mathbf{x}))}, & \text{if } \mathbf{x} \text{ is assigned to } N, \\ \frac{\lambda_{NP}\pi_N f_N(f(\mathbf{x})) + \lambda_{PP}\pi_P f_P(f(\mathbf{x}))}{\pi_N f_N(f(\mathbf{x})) + \pi_P f_P(f(\mathbf{x}))}, & \text{if } \mathbf{x} \text{ is assigned to } P. \end{cases} \tag{2.7}$$

As a consequence, the risk is minimized when the sample $\mathbf{x}$ is assigned to the class P if:

$$l\left(f(\mathbf{x})\right) = \frac{f_P\left(f(\mathbf{x})\right)}{f_N\left(f(\mathbf{x})\right)} > \frac{(\lambda_{NP} - \lambda_{NN})\pi_N}{(\lambda_{PN} - \lambda_{PP})\pi_P}, \tag{2.8}$$

where $l(t)$ is the likelihood ratio:

$$l(t) = \frac{f_P(t)}{f_N(t)}. \tag{2.9}$$

In well defined environments, i.e. where class priors and misclassification costs are known, evaluation at a single operating point is appropriate. However, in imprecise environments or when comparing models operating at different points the ROC analysis becomes more appropriate (Provost & Fawcett, 2001).

2.2 The ROC Space

A ROC curve is a technique to visualize, organize and select classifiers based on their performance (Fawcett, 2006). This has been introduced in the signal theory during the second world war for the analysis of radar signals to depict the tradeoff between the rates of hit and false alarm of friendly and enemy airplanes (Egan, 1975), (Swets *et al.*, 2000). Then, the ROC analysis has been successfully extended to the visualization of the behavior of a diagnostic system (Metz, 1986), (Swets, 1988). One of the first paper in machine learning is Spackman (1989) that demonstrated the capability of ROC curves in comparing learning algorithms. Moreover, as introduced in the previous section, its capability to describe the classifier performance when dealing with skewed class distribution and unequal classification error costs makes it a very useful instrument in cost sensitive and unbalanced environments classification.

The ROC curve plots $TPR(t)$ vs. $FPR(t)$ by sweeping the threshold t into the whole range of f, thus providing a description of the performance of the dichotomizer at different operating points. As it is possible to note from equation (2.2), the pair $(FPR(t), TPR(t))$ is sufficient to completely characterize the performance of the classifier since the other indices are dependent on these. As an example, in fig. 2.1 the ROC space and the performance indices are shown for a given threshold value and gaussian confidence densities. If we have a perfect knowledge of the class conditional densities the ROC curve can be easily obtained estimating the likelihood ratio.

If we do not have knowledge of the class densities but we know the values of the classifier score, the value of the threshold t can be varied between $-\infty$ and $+\infty$ and the quantities in eq. (2.1) vary accordingly, thus defining the set of the operating points, given by the pairs $(FPR(t), TPR(t))$, achievable by the classifier. The two extreme points are reached when t tends to $-\infty$ or $+\infty$; in the former case, both $TPR(t)$ and $FPR(t)$ approach 1 since all the negative and positive samples are classified as belonging to the positive class while the contrary happens when $t \to +\infty$. In this way, we obtain an empirical estimator of the ROC curve by evaluating, for each possible value of t the empirical true and false positive rates as follows:

$$\widehat{TPR}(t) = \frac{1}{m_P} \sum_{i=1}^{m_P} S\left(f(\mathbf{p}_i) \geq t\right), \tag{2.10a}$$

$$\widehat{FPR}(t) = \frac{1}{m_N} \sum_{j=1}^{m_N} S\left(f(\mathbf{n}_j) \geq t\right), \tag{2.10b}$$

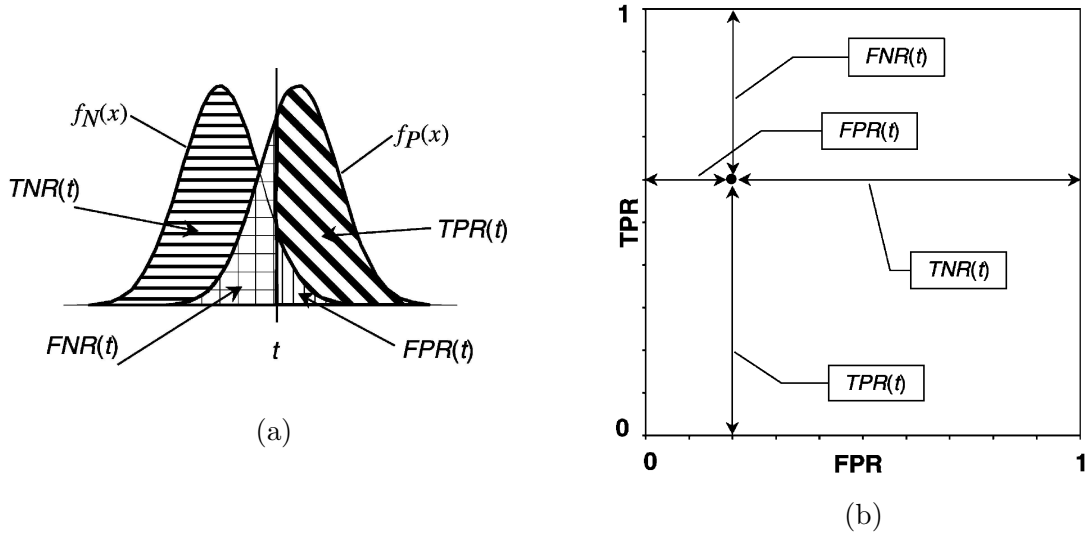


Figure 2.1: (a) The indices TPR, FPR, TNR, FNR evaluated on two gaussian shaped confidence densities for a given threshold value t. (b) The same quantities mapped on the (FPR, TPR) plane.

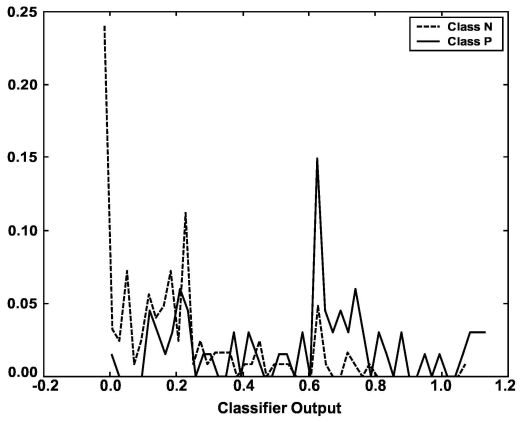
where $S(\cdot)$ is a predicate that is 1 when the argument is true and 0 otherwise, m_P and m_N are the number of samples in the positive and negative class respectively and $\mathbf{p}_i$ and $\mathbf{n}_j$ are the i-th and th j-th sample of the positive and negative class. Let us call the obtained curve *empirical ROC curve* in order to distinguish it from the *ideal ROC curve* that we obtain with a perfect knowledge of the class conditional densities. A perfectly discriminating classifier has an ROC curve that passes through the upper left corner (where $TPR = 1.0$ and $FPR = 0.0$), while a non discriminating classifier is represented by a diagonal line from the lower left to the upper right corner. Qualitatively, the closer the curve to the upper left corner, the better the classifier.

It is worth noting that once the two classes have been specified through their conditional densities, the ideal ROC is unique, while different classifiers trained on the same problem have different empirical ROCs. A typical empirical ROC curve is shown in fig. 2.2 together with the densities of the confidence degree for the two classes. It is worth noting that the threshold t varies between the minimum and the maximum value of the classifier output.

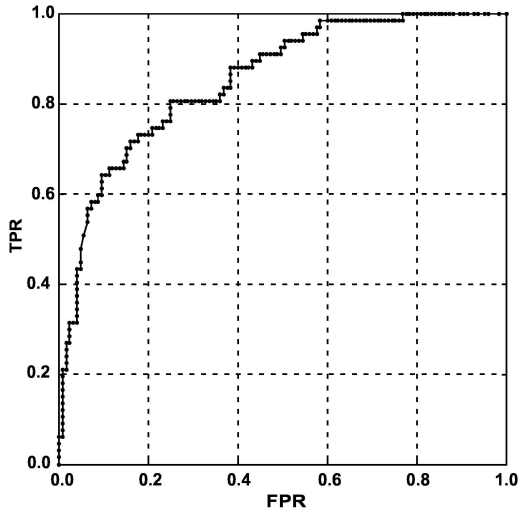
An important feature of the ideal ROC curve is stated in the following lemma:

Lemma 2.2.1. *The slope of the curve at any point $(FPR(t), TPR(t))$ is equal to the likelihood ratio:*

$$l(t) = \frac{f_P(t)}{f_N(t)}. \tag{2.11}$$



(a)



(b)

Figure 2.2: (a) The densities of the confidence degree obtained by the classifier output on real data and (b) the corresponding ROC curve.

Proof. The proof (Van Trees, 2001), (Green & Swets, 1966) is straightforward: consider that the slope in correspondence of the point $(FPR(t), TPR(t))$ is given by:

$$\frac{dTPR(\tau)}{dFPR(\tau)}\Big|_{\tau=t},$$

which, recalling equation (2.1), is equal to:

$$\frac{-f_P(t)}{-f_N(t)} = l(t).$$

□

This is a key result since two very popular decision criteria (risk minimization and Neyman-Pearson) are based on the likelihood ratio and thus the ROC curve can be profitably used to find the best operating point for both rules. Recalling eq. (2.8) the corresponding operating point on the ROC curve is that where the curve has gradient (Kanungo & Haralick, 1995):

$$\nabla_{ROC} = \frac{(\lambda_{NP} - \lambda_{NN})\pi_N}{(\lambda_{PN} - \lambda_{PP})\pi_P}.$$

Such point can be easily found moving down from above in the ROC plane a line with slope (see eq. (2.6)) $m = \frac{(\lambda_{NP}-\lambda_{NN})\pi_N}{(\lambda_{PN}-\lambda_{PP})\pi_P}$ and selecting the point in which the line touches

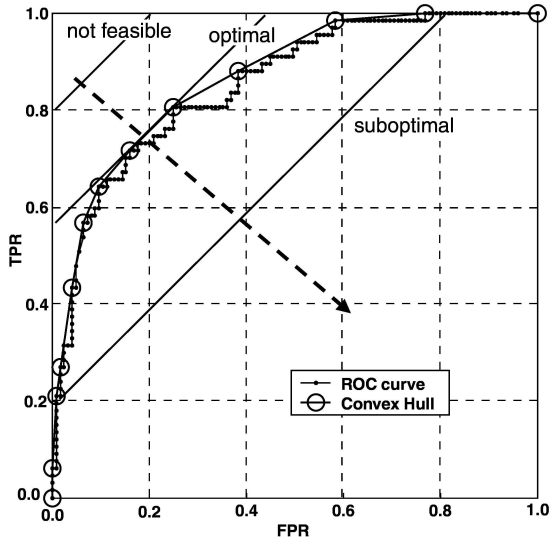


Figure 2.3: The ROC curve shown in fig. 2.2 and its convex hull. Three level lines with the same slope are also shown: the line touching the ROC convex hull determines the optimal operating point since it involves the minimum risk. The line above the optimal line does not determine any feasible point, while the line below identifies only suboptimal points.

the ROC curve (Webb, 2002). This is realizable for an ideal curve but if the ROC curve is defined by means of a finite number of experimental points connected with straight lines (such as the curve in fig. 2.2), the optimal operating point can be determined by the point where a line with slope m, moving down from above touches the ROC curve (Zweig & Campbell, 1993). Such point lies on the ROC Convex Hull, i.e. the smallest convex set containing the points of the ROC curve, as formally proved in Provost & Fawcett (2001).

This can be understood by looking at fig. 2.3, where the ROC curve of fig. 2.2 is shown together with its convex hull and some level lines (called isocost lines) with the same slope. The line touching the ROC curve determines the optimal point: in fact the line above does not determine any feasible operating point for the classifier, while the line below intersects the ROC curve in at least two points, but at the highest expected cost. Once the optimal operating point has been found, the optimal threshold t_{opt} is consequently determined by reading the value of t related to that point.

An alternative to the risk minimization is the Neyman-Pearson decision rule, where we assume to know the conditional densities, but not the costs and the prior probabilities. The goal here is to minimize the probability of error on one class (say P) subject to the constraint that the error probability on the other class is not larger than a given constant

ε. The optimal operating point on the ROC curve is the point with abscissa $FPR = \varepsilon$ corresponding to the decision rule $l\left(f(\mathbf{x})\right) > \tau$ where τ is the threshold generating the point. A major consequence of such property is that we can identify the optimal ROC curve, i.e. the curve which, for each $FPR \in [0, 1]$, has the highest TPR among all possible criteria based on $f(\mathbf{x})$. This is possible if we recall the Neyman-Pearson lemma (Neyman & Pearson, 1933) which in this case (the proof can be found in Mukhopadhyay (2000) and Garthwaite *et al.* (2002)) can be stated as:

Lemma 2.2.2. *Consider the decision rule $l\left(f(\mathbf{x})\right) > \tau$ with τ chosen to give $FPR = \varepsilon$. There is no other decision rule providing a TPR higher than $TPR(\tau)$ with a $FPR \leq \varepsilon$.*

Proof. Let $l_1\left(f(\mathbf{x})\right.$ (hereafter l_1) be a decision rule with:

$$FPR_{l_1} \leq \varepsilon,$$

The following identity holds for any decision rule l_1:

$$\int \left(l - l_1\right)\left(f_P(\mathbf{x}) - \tau f_N(\mathbf{x})\right)\, d\mathbf{x} \geq 0, \tag{2.12}$$

This can be seen by considering all possible values of $\mathbf{x}$: for those values for which $l \geq \tau$, $l - l_1 \geq 0$ and $f_P(\mathbf{x}) - \tau f_N(\mathbf{x}) \geq 0$; similarly, for those values for which $l \leq \tau$, $l - l_1 \leq 0$ and $f_P(\mathbf{x}) - \tau f_N(\mathbf{x}) < 0$. Multiplying out eq. (2.12) and writing the results in terms of true positive and false positive rate, we get:

$$\begin{aligned}
&\int \left(l - l_1\right)\left(f_P(\mathbf{x}) - \tau f_N(\mathbf{x})\right)\, d\mathbf{x} = \int l f_P(\mathbf{x})\, d\mathbf{x} - \int l_1 f_P(\mathbf{x})\, d\mathbf{x} \\
&- \tau \int l f_N(\mathbf{x})\, d\mathbf{x} + \tau \int l_1 f_N(\mathbf{x})\, d\mathbf{x} = \left(TPR_l - TPR_{l_1}\right) \\
&- \tau\left(FPR_l - FPR_{l_1}\right) \geq TPR_l - TPR_{l_1} \geq 0,
\end{aligned} \tag{2.13}$$

where the first inequality follows from the assumption that $FPR_{l_1} \leq \varepsilon = FPR_l$, and the second inequality follows from eq. (2.12). Thus, $TPR_l \geq TPR_{l_1}$ and l is a decision rule that maximizes the true positive rate for a given false positive rate. □

2.3 Generating an ROC Curve

Up to now we spoke about the ROC curve built from the output of a classifier on a test set. Indeed to generate an ROC curve we can refer to two alternative approaches:

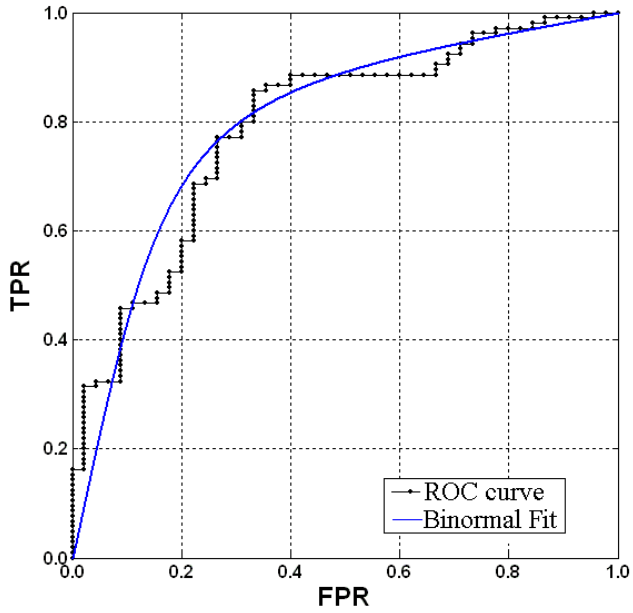


Figure 2.4: The ROC curve evaluated with the empirical and the parametric (binormal) approach.

parametric and non parametric. The latter, introduced in the previous section, is based on the exploitation of the monotonicity of thresholded classification, i.e. any instance that is classified positive with respect to a given threshold will be classified as positive for all lower thresholds as well (Fawcett, 2006). The parametric approach consist in techniques for fitting ROC curves to continuously-distributed data estimating the curve parameters on the basis of some assumptions concerning the form of the decision-variable distributions (Metz *et al.*, 1998). Both the methods present some disadvantages: the non parametric approach has high computational complexity and gives a non smooth curve while the parametric model can be wrong if the assumptions on the distributions do not correspond to the real data (Goddard & Hinberg, 1990). In this paragraph we report an efficient algorithm to generate the ROC curve in a non parametric way (that is used in the following of the work) and the most common parametric model, i.e. the binormal model. A parametric and non parametric ROC curve are compared for the same data in fig. 2.4.

2.3.1 The non parametric approach

A diffused way to obtain T points of the ROC curve of a classifier is to consider the outputs provided by the classifier on a labelled data set, compute T thresholds ranging

from the smallest to the largest values produced by the classifier and evaluate the resulting TPR and FPR for each of the T thresholds. Such kind of method is quite unsatisfactory because it is strictly dependent on the choice of T and when the discretization applied to the classifier output is too coarse (the T threshold values considered are few compared with the number of different values the classifier output assumes) the approximation can be poor and misrepresent the actual plot.

On the contrary, the ROC curves we use have been generated by employing all the scores returned by the classifier as possible decision thresholds, thus obtaining a faithful plot. To this aim we have used an efficient algorithm, described in Fawcett (2006) and reported in Algorithm 2.1, that simply sorts the test instances decreasing by f scores and move down the list, processing one instance at a time and updating true positives and false positives as it goes on. In this way an ROC curve can be created from a linear scan with complexity $O(nlogn)$ in the number of the data samples L.

The best ROC is generated when all the positive samples end up at the beginning of the sequence while the worst ROC is generated when the positive and negative samples are perfectly alternated in the sequence (starting with a positive sample). The principal problem of this algorithm is when for some samples we obtain the same score. In this case the algorithm averages the pessimistic and optimistic ROC not emitting any operating point until all instances of equal f have been processed.

Algorithm 2.1 Efficient method to generate an ROC curve

Input: Out: the set of the output of the classifier f on a data set; $m_P > 0$ and $m_N > 0$, the number of positive and negative examples.

Output: R, a list of ROC points increasing by FPR.

$Out_{\text{sort}} \leftarrow Out$ sorted by decreasing scores
$FP \leftarrow TP \leftarrow 0$
$R \leftarrow [\,]$
$f_{\text{prev}} \leftarrow -\infty$
$i \leftarrow 1$
while $i \leq \text{card}(Out_{\text{sort}})$ **do**
  if $f(i) \neq f_{\text{prev}}$ **then**
    put $\left(\frac{FP}{m_N}, \frac{TP}{m_P}\right)$ onto R
    $f_{\text{prev}} \leftarrow f(i)$
  end if
  if $Out_{\text{sort}}(i)$ is a positive example **then**
    $TP \leftarrow TP + 1$
  else /*i is a negative example*/
    $FP \leftarrow FP + 1$
  end if
  $i \leftarrow i + 1$
end while
put $\left(\frac{FP}{m_N}, \frac{TP}{m_P}\right)$ onto R

2.3.2 The Parametric Approach

To fit an ROC curve with a continuous data distribution different techniques have been proposed. An approach could be to estimate the mean and the variance of the data distribution and evaluating the parametric ROC on the basis of a chosen distribution function. However, one should not use this method unless the data are obtained from test distributions with known form, since the method is extremely sensitive to the validity of its distributional assumptions (Goddard & Hinberg, 1990).

Another possible approach less dependent on the choice of the distribution function is to assume that the data arises from a known distribution; the most common of this approaches is the *binormal model*. Such method has the advantage that the obtained fit is much less dependent upon the validity of assumptions concerning the form of the underlying distributions, due to the invariance of ROC curves under monotonic transfor-

mations of the decision-variable scale (Swets *et al.*, 1961), (Egan, 1975). The binormal model consists of choosing a normal distribution for both the classes and has been found empirically to provide satisfactory ROC fits to data generated in a very broad variety of situations (as shown in several papers (Hanley, 1988), (Swets, 1986), (Hajian-Tilaki *et al.*, 1996)). The approach consists of estimating the mean and the variance of the two normal distributions assumed for the two classes. Let us say (μ_P,σ_P) and (μ_N,σ_N) the parameters for the positive and negative class respectively, the two distribution functions are:

$$\begin{aligned} F_N(\mathbf{x}) &= \Phi\left(\frac{\mathbf{x}-\mu_N}{\sigma_N}\right), \\ F_P(\mathbf{x}) &= \Phi\left(\frac{\mathbf{x}-\mu_P}{\sigma_P}\right), \end{aligned} \tag{2.14}$$

where $\Phi(z) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{z} e^{-\zeta^2/2}\, d\zeta$ is the standard normal cumulative distribution function. If t is the threshold between positive and negative class we can evaluate the TPR and FPR as:

$$\begin{aligned} FPR(t) &= Prob(\mathbf{x} > t|\mathbf{x} \in N) = 1 - Prob(\mathbf{x} \leq t|\mathbf{x} \in N) \\ &= 1 - \Phi\left(\frac{t-\mu_N}{\sigma_N}\right) = \Phi\left(\frac{\mu_N - t}{\sigma_N}\right), \end{aligned} \tag{2.15a}$$

$$\begin{aligned} TPR(t) &= Prob(\mathbf{x} > t|\mathbf{x} \in P) = 1 - Prob(\mathbf{x} \leq t|\mathbf{x} \in N) \\ &= 1 - \Phi\left(\frac{t-\mu_P}{\sigma_P}\right) = \Phi\left(\frac{\mu_P - t}{\sigma_P}\right), \end{aligned} \tag{2.15b}$$

and we obtain:

$$\begin{aligned} -t = \sigma_N\Phi^{-1}(FPR) - \mu_N &= \sigma_P\Phi^{-1}(TPR) - \mu_P \\ &\Downarrow \\ \Phi^{-1}(TPR) &= \frac{\mu_P - \mu_N}{\sigma_P} + \frac{\sigma_N}{\sigma_P}\Phi^{-1}(FPR). \end{aligned} \tag{2.16}$$

If we define Metz *et al.* (1998):

$$a = \frac{|\mu_P - \mu_N|}{\sigma_P}, \qquad b = \frac{\sigma_N}{\sigma_P}, \tag{2.17}$$

substituting in eq. (2.16) we obtain every point of the ROC curve as:

$$\begin{gathered}\Phi^{-1}(TPR) = a + b\Phi^{-1}(FPR) \\ \Downarrow \\ TPR = \Phi\left(a + b\Phi^{-1}(FPR)\right),\end{gathered} \tag{2.18}$$

and the two normal distribution according to eq. (2.14) are:

$$F_N(\mathbf{x}) = \Phi(\mathbf{x}), \qquad F_P(\mathbf{x}) = \Phi(b\mathbf{x} - a). \tag{2.19}$$

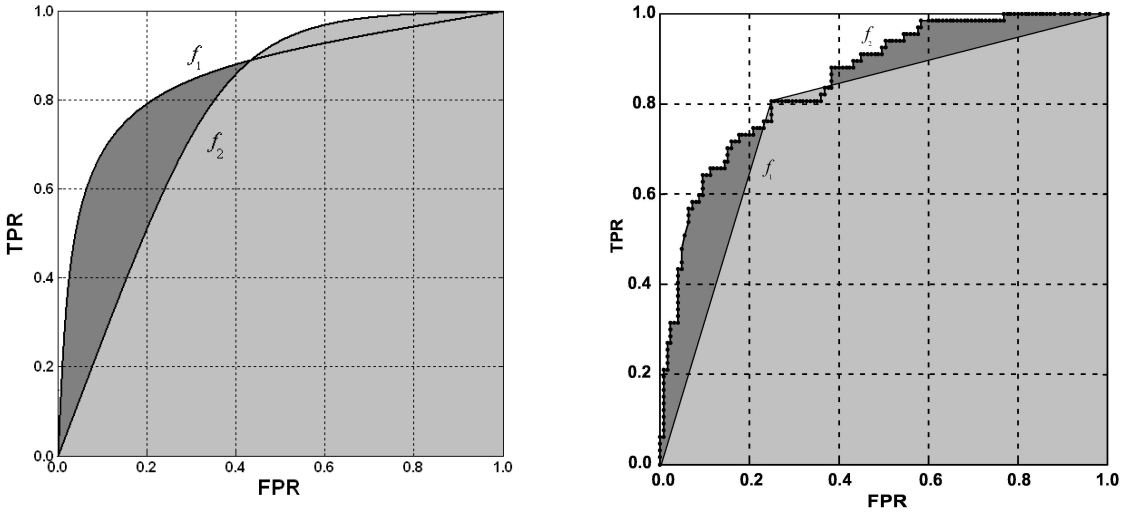
In practice, we have a parametrization that is still based on the binormal model but with only two parameters $\{a, b\}$ that better represent the data.

2.4 Comparing Classifiers: the AUC

In the previous section we have shown that the ROC curve is a two-dimensional depiction of classification performance. However, if we want to compare the average performance of different classifiers it can happen that ROC curves cross each other, and in general that one classifier can be superior for some values of the threshold and another superior for other values of the threshold. To evaluate the classifier performance independently of the threshold a single scalar value can be adopted: the Area Under the ROC curve (AUC) (Bamber, 1975). To put in evidence this behavior we report in fig. 2.5 the ROC curves of two classifiers (Fawcett, 2006), say f_1 and f_2. In the left graph we plot two classifiers with crossing ROC curves; in this case also if f_2 has greater AUC than f_1, it becomes worst for an $FPR > 0.6$. In the right graph another example is reported for a discrete classifier f_1 that reaches the performance of the probabilistic classifier f_2 just for one value of the threshold but it becomes inferior further from this point.

Also if AUC fails to take this into account, it is often used when a general measure of predictiveness is desired since it represents a good measure on how well the rule differentiates between the distributions of the two classes without being influenced by external factors. In practice it can be seen as an alternative parameter to the Bayes risk to evaluate the quality of the decision rule since it is not related to a particular prior probabilities and costs distributions.

Since the AUC is a portion of the area of the unit square, its value will always be between 0 and 1 (for perfectly discriminating classifiers). However, since random guessing produces the diagonal line between the points (0,0) and (1,1) (i.e. an AUC of 0.5), no



(a) (b)

Figure 2.5: Comparison of the ROC curves and the AUCs for two classifiers f_1 and f_2. The graph (b) shows the AUC for a discrete (f_1) and a probabilistic (f_2) classifier.

realistic classifiers should have an AUC less than 0.5. If this happens it means that our classifier is able to discriminate between the two classes but it exchanges the labels. For example if the AUC for a classifier is equal to zero it means that all the positive samples are classified as negative and all the negative as positive, i.e. our classifier perfectly discriminates the two classes but the decision is always wrong. In this case if we exchange the label, the AUC will be equal to 1.

It is worth noting that the AUC is an overall measure of how accurately the classifier ranks negative from positive patterns. In the ideal case, the following lemma states:

Lemma 2.4.1. *The AUC provides the probability that $f(\mathbf{p}_i) > f(\mathbf{n}_j)$ where $\mathbf{p}_i$ and $\mathbf{n}_j$ are two samples randomly extracted from the positive and the negative class respectively.*

Proof. We know that:

$$AUC = \int_0^1 TPR dFPR \tag{2.20}$$

Therefore:

$$AUC = \int_0^1 Prob\left(f(\mathbf{p}) > t | \mathbf{p} \in P\right) dProb\left(f(\mathbf{n}) > t | \mathbf{n} \in N\right).$$

Recalling eq. (2.1b) we obtain that:

$$dProb\left(f(\mathbf{n}) > t | \mathbf{n} \in N\right) = -f_N(t)dt;$$

moreover, we can perform a change of variable in the integral so obtaining:

$$\begin{aligned} AUC &= -\int_{-\infty}^{+\infty} Prob\left(f(\mathbf{p}) > t | \mathbf{p} \in P\right) f_N(t) dt \\ &= -\int_{-\infty}^{+\infty} Prob\left(f(\mathbf{p}) > t | \mathbf{p} \in P\right) f_N(t) dt. \end{aligned}$$

the last integral is actually the $Prob\left(f(\mathbf{p}_i) > f(\mathbf{n}_j) | \mathbf{p}_i \in P, \mathbf{n}_j \in N\right)$ and the lemma is proved. □

Nevertheless, some relations can be drawn between the AUC and the figures to be optimized in the decision criteria described in sec. 2.1. For the risk minimization criterion, it is simple to verify that, AUC provides the probability that $l(f(\mathbf{p}_i)) > l(f(\mathbf{n}_j))$ and the average Bayes risk becomes:

$$\begin{aligned} \langle R \rangle =& \int_{-\infty}^{+\infty} Risk(t) \phi_l(t) dt = \int_{-\infty}^{+\infty} Risk(t) \left(\pi_P f_P(t) + \pi_N f_N(t)\right) dt \\ =& \int_{-\infty}^{+\infty} \big(\pi_N(\lambda_{NP} - \lambda_{NN}) FPR(t) + \pi_P(\lambda_{PP} - \lambda_{PN}) TPR(t) \\ & + \pi_P \lambda_{PN} + \pi_N \lambda_{NN}\big) \left(\pi_P f_P(t) + \pi_N f_N(t)\right) \, dt \\ =& \pi_P \pi_N (\lambda_{NP} - \lambda_{NN}) \int_{-\infty}^{+\infty} FPR(t) \, dTPR + \pi_N^2 (\lambda_{NP} - \lambda_{NN}) \\ & \cdot \int_{-\infty}^{+\infty} FPR \left(-dFPR\right) + \pi_P^2 (\lambda_{PP} - \lambda_{PN}) \int_{-\infty}^{+\infty} TPR \, dTPR \\ & + \pi_P \pi_N (\lambda_{PP} - \lambda_{PN}) \int_{-\infty}^{+\infty} TPR \left(-dFPR\right) + \pi_P^2 \lambda_{PN} + \pi_P \pi_N \lambda_{PN} \\ & + \pi_N^2 \lambda_{NN} + \pi_P \pi_N \lambda_{NN} \\ =& \pi_P \pi_N (\lambda_{NP} - \lambda_{NN})(1 - AUC) + \frac{1}{2} \pi_N^2 (\lambda_{NP} - \lambda_{NN}) + \frac{1}{2} \pi_P^2 (\lambda_{PP} - \lambda_{PN}) \\ & - \pi_P \pi_N (\lambda_{PP} - \lambda_{PN}) AUC + \pi_P^2 \lambda_{PN} + \pi_P \pi_N \lambda_{PN} + \pi_N^2 \lambda_{NN} + \pi_P \pi_N \lambda_{NN} \\ =& \left(\lambda_{NP} + \lambda_{PN}\right) \pi_P \pi_N + \frac{1}{2} \left(\left(\lambda_{NP} + \lambda_{NN}\right) \pi_N^2 + \left(\lambda_{PN} + \lambda_{PP}\right) \pi_P^2\right) \\ & - \left(\lambda_{NP} + \lambda_{PN} - \lambda_{NN} - \lambda_{PP}\right) \pi_P \pi_N AUC. \end{aligned} \tag{2.21}$$

where $\phi_l(t)$ is the unconditional density of the likelihood ratio. This represents a more

general result than the one presented in Hand & Till (2001) that does not take into account the cost distributions. In summary, there exists a linear relation between the AUC and the Bayes risk: the higher the AUC for the decision rule, the lower the average Bayes risk. As a consequence, a learning algorithm which maximizes the AUC produces a classification system with minimal average Bayes risk. However, it is worth noting that this does not imply that, for a given set of costs and a priori probabilities, a system with a high AUC provides necessarily a Bayes risk lower than a system with smaller AUC (see again fig. 2.5).

As regards the Neyman-Pearson criterion, it is simple to verify that the AUC is related with the average error made on class P. In fact we have:

$$AUC = \int_0^1 TPR dFPR = \int_0^1 (1 - \varepsilon_P) d\varepsilon_N, \tag{2.22}$$

from which:

$$\langle \varepsilon_P \rangle = 1 - AUC,$$

where ε_N and ε_P are the error probabilities on class P and N respectively.

Due to its characteristics, the AUC has been recently proposed as an alternative single number measure for evaluating the predictive ability of learning algorithms. Huang & Ling (2005) have been shown theoretically and empirically that AUC is a better measure than accuracy and should replace it in comparing learning algorithms. Therefore, we can conclude that in every case, a classifier that maximizes the AUC also maximizes the average quality measure of the decision criterion.

2.5 How to Evaluate the AUC

Since the AUC is the area under a curve it can be numerically estimated by integrating the corresponding ROC, i.e. using eq. (2.22). Bradley (1997) proposed to construct an estimate of the ROC curve directly for specific classifiers by varying a threshold and then to use an integration rule (for example, the trapezium rule) to obtain an estimate of the area beneath the curve. Moreover, Provost & Fawcett (2001) elaborated an algorithm to evaluate the AUC with the trapezoid rule with a low computational complexity.

The estimation can also be performed using the binormal model presented in sec. 2.3.2. In this case the AUC can be evaluated as:

$$\widehat{AUC} = \Phi\left(\frac{a}{\sqrt{1+b^2}}\right),$$

where a and b have been defined in eq. (2.17).

The AUC, despite being defined as a geometric quantity, has two important statistical interpretations. First, the empirical AUC is equal to the *Wilcoxon-Mann-Whitney (WMW)* statistic (Mann & Whitney, 1947): recalling that $f(\mathbf{p_i})$ and $f(\mathbf{n_j})$ are the output of a classifier on the i-th positive sample $\mathbf{p}_i$ and on the j-th negative sample $\mathbf{n}_j$, we have:

$$R = \frac{\sum_{i=1}^{m_P} \sum_{j=1}^{m_N} I(f(\mathbf{p}_i), f(\mathbf{n}_j))}{m_P m_N}, \tag{2.23}$$

where $I(a, b)$ is an indicator function defined as

$$I(a,b) = \begin{cases} 1, & \text{if } a > b, \\ 0.5, & \text{if } a = b, \\ 0, & \text{if } a < b. \end{cases}$$

In this way, it is possible to evaluate the AUC of f directly through (2.23) without explicitly plotting the ROC curve and estimating the area with a numerical integration. Several papers try to maximize the AUC suggesting an approximation approach to the WMW statistic. For example in Yan *et al.* (2003) and Herschtal & Raskutti (2004) a continuous function is used so as it is possible to use the gradient methods to solve the optimization problem. The proof of this equivalence can be found in Pepe (2003) and Sing (2004).

Second, the AUC represents also the probability that a randomly chosen negative example will have a smaller estimated probability of belonging to the positive class than a randomly chosen positive example, i.e. the probability of correct pairwise ranking (Hanley & McNeil, 1982).

We can state it in the following proposition:

Proposition 2.5.1. *1. The area under an empirical ROC curve is equivalent to the Wilcoxon-Mann-Whitney statistic R in eq. (2.23).*

2. The area under an empirical ROC curve is equal to the probability that the learner will assign a higher score to a randomly drawn positive sample than to a randomly drawn negative sample: $AUC = \text{Prob}\,(f(\mathbf{p}_i) > f(\mathbf{n}_j) | \mathbf{p} \in P \text{ and } \mathbf{n} \in N)$.

Proof. Without loss of generality we assume that the values that the classifier f assigns to the samples are pairwise different.

1. If we refer to the algorithm 2.1 to build the ROC curve, we can see that the AUC is the sum of the area of the vertical columns under the curve evidenced in fig. 2.6.

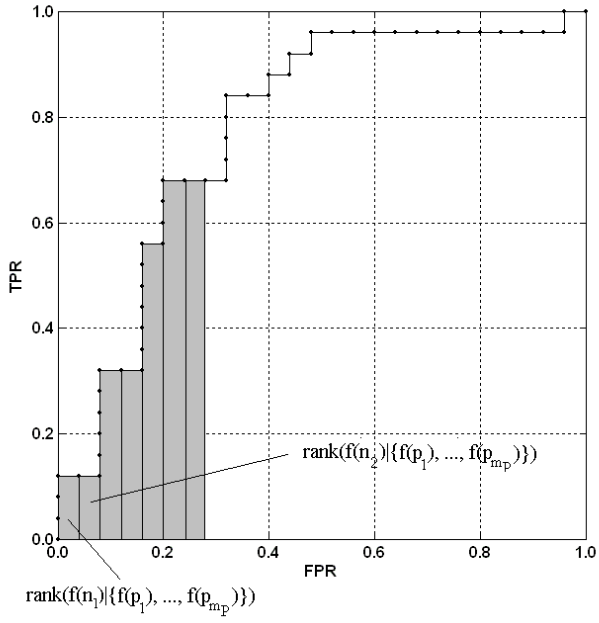


Figure 2.6: The ROC plot used in the proof of proposition 2.5.1.

The area for the column for the sample $f(\mathbf{n}_j)$ is equal to the number of positive samples that are scored higher, which is the relative rank of the negative sample with respect to the positive samples. For positive and negative samples labelled in descending order (i.e. $f(\mathbf{p}_1) > \ldots > f(\mathbf{p}_{m_P})$ and $f(\mathbf{n}_1) > \ldots > f(\mathbf{n}_{m_N})$), we have that the area under a column is the rank of a negative sample with respect to all the positive samples, i.e.:

$$A_{col} = \sum_{j=1}^{m_N} \text{rank}\left(f(\mathbf{n}_j)|\{f(\mathbf{p}_1), \ldots, f(\mathbf{p}_{m_P})\}\right).$$

Since the AUC is the sum on all positive samples scaled by $m_P \cdot m_N$, we can write:

$$\begin{aligned} AUC &= \frac{1}{m_P m_N} A_{col} = \frac{1}{m_P m_N} \sum_{j=1}^{m_N} \text{rank}\left(f(\mathbf{n}_j)|\{f(\mathbf{p}_1), \ldots, f(\mathbf{p}_{m_P})\}\right) \\ &= \frac{1}{m_P m_N} I(f(\mathbf{p}_i), f(\mathbf{n}_j)) = R. \end{aligned}$$

and the equivalence is proved.

2. From the proof of the point 1 we can see that the area of a column for the sample

n_j can also be evaluated as the Prob $(f(\mathbf{n}_j) < f(\mathbf{p})|\mathbf{p} \in P)$. Therefore, we have:

$$AUC = \sum_{j=1}^{m_N} \text{Prob}\left(f(\mathbf{n}_j) < f(\mathbf{p})|\mathbf{p} \in P\right)$$

$$= \text{Prob}\left(f(\mathbf{p}_i) > f(\mathbf{n}_j)|\mathbf{p} \in P \text{ and } \mathbf{n} \in N\right).$$

□

Moreover, since it can be demonstrated that the WMW statistic provides an unbiased estimate of the probability $Prob\left(f(\mathbf{p}_i) > f(\mathbf{n}_j)\right)\ \forall i = 1 \ldots m_P, j = 1 \ldots m_N$ (Lehmann & D'Abrera, 1975), the empirical AUC has a propriety similar to the ideal AUC shown in lemma 2.4.1 and it represents a measure of the quality of the ranking of the classifier: when AUC=1, the classifier correctly ranks all the pairs $(\mathbf{p}, \mathbf{n})$ while if AUC=0.5 the classifier is ineffective, i.e. its performance is equal to a random ranker.

2.6 AUC in Ranking Problems

In the previous sections we reported an analysis of the ROC curve in the context of pattern recognition. In particular, the AUC have been introduced as measure independent on priors and misclassification costs and its statistical properties in association to the ranking between positive and negative samples of the classifier's output has been shown.

Therefore, AUC can be helpful in many real cases where imbalanced environments are present or when the ranking is more useful than the categorization of patterns into classes. The former case is a fundamental aspect of medical detection problems or other screening applications where imbalanced class priors or misclassification costs are often present (Bradley, 1997).

The second aspect becomes important in many data mining applications where accuracy is not enough. As an example, consider a document retrieval application (Cortes & Mohri, 2003) where a search engine selects a prefixed number of documents from a huge database on the basis of some search criteria and prompts them to the user according to an estimated order of relevance. In this case, the actually significant outcome is the ranking of the documents rather than their categorization.

Another example is in direct marketing (Huang & Ling, 2005) where we need to promote the top percent customers during gradual roll-out, or we often deploy different promotion strategies to customers with different likelihood of purchasing. To accomplish these tasks a ranking of customers in terms of their likelihoods of buying is needed more than

a classification of buyers and non buyers. Thus, a ranking is much more desirable than just a classification (Ling & Li, 1998) and it can be easily obtained since most classifiers produce probability estimations that can be used for ranking examples.

It is worth noting that the difference between classifier and ranker is the threshold. If no threshold is fixed, the classifier can be considered as a ranker, since it orders the patterns in such a way that $f(\mathbf{x}_1) > f(\mathbf{x}_2)$ means that pattern $\mathbf{x}_1$ is more likely belonging to class P than pattern $\mathbf{x}_2$.

Cortes & Mohri (2003) have proved that algorithms designed to minimize the error rate may not lead to the best possible AUC thus motivating the use of algorithms and combiners directly optimizing the AUC. In literature, this problem has been recently analyzed and new algorithms focusing on the AUC have been proposed. In the next chapters we follow this approach and propose new combination rules and classifiers directly built to maximize the AUC.

Chapter 3

A Linear Classifier Maximizing the AUC

One of the most useful way to represent pattern classifiers is in terms of a set of discriminant functions. To fix such a function is equivalent to specify a decision rule without any assumptions on the class conditional densities. Discriminant functions that are linear combination of the features have a variety of pleasant analytical properties and result in linear decision boundaries. Different optimization schemes can be used to correctly estimate the parameters of this function according to some adopted performance measure. In literature the majority of the built classification systems try to minimize the error rate and only recently the AUC has become of interest in the project of a classifier.

In this chapter, after a brief review of the state of the art and an analysis of the linear discriminant functions with respect to the ROC curve, we propose a nonparametric classifier that performs a linear combination of features choosing weights suitable for the maximization of the AUC. The approach is based on the study of the WMW statistic of each single feature and on an iterative pairwise linear coupling of the features used to optimize the ranking of the combination.

3.1 Discriminant Functions and Ranking

In sec. 1.1 we introduced the search of a decision rule and a decision boundary to well separate two or more classes making assumptions on the class conditional probability densities. Another approach consists in making assumptions about the form of the discriminant functions.

The choice of a discriminant function may depend on prior knowledge about the pat-

terns to be classified or may be a particular functional form whose parameters are adjusted by a training procedure (Webb, 2002). In every case, a classifier can be viewed as a machine that computes a set of discriminant functions and assign the label to the corresponding largest discriminant. The choice of a discriminant function is not unique. We can always apply to the discriminant function a monotonically increasing function without influencing the resulting classification. Therefore, even though the discriminant can be written in different ways the decision rules and the partition of the feature space in decision regions are equivalent.

The problem of finding a discriminant function can be formulated as a problem of minimizing a criterion function, i.e. it depends on the performance measure we want to maximize. In the previous chapter, we introduced the AUC as a suitable measure to evaluate the ability of a classifier to rank instances in binary classification problems. Therefore, in our analysis we focus on learning algorithms that take into account this performance measure.

3.1.1 Learning Algorithms based on Ranking

Ranking is a popular topic in the machine learning field and on these bases several learning algorithms have been proposed in the recent literature.

A first approach can be based on the direct maximization of the WMW statistic. In particular, a method based on logistic regression is proposed in Herschtal & Raskutti (2004) where a continuous function is used to approximate the WMW statistic and the descent gradient method is applied to solve the optimization problem. However, such an approximation in the case of rank optimization has to be carefully handled since in this process information related to ranking may be easily lost.

A well performing learning algorithm is proposed in Freund *et al.* (2003) where a method to combine rankings based on the boosting approach has been introduced. Boosting is a method to produce highly accurate prediction rules by combining many weak rules which may be only moderately accurate (Freund & Schapire, 1997). Like all boosting algorithms, RankBoost operates in rounds assuming access to a separate procedure (i.e. the "weak learner") that, on each round, is called to produce a weak ranking. The algorithm maintains a distribution D_τ over $X \times X$ that is passed on round τ to the weak learner. Intuitively, RankBoost chooses D_τ to emphasize different parts of the training data. A high weight assigned to a pair of instances indicates a great importance that the weak learner orders that pair correctly. Weak rankings have the form $h_\tau : X \to \mathbb{R}$ and they are based on the given ranking features. In particular, Freund *et al.* (2003) derives a

weak ranking from the ranking of the feature x_i by comparing the score of x_i on a given instance to a threshold θ and assigning a default score to instances not correctly ranked by x_i. Then, the weak rankings are used in the boosting algorithm to update the distribution D_τ according to:

$$D_{\tau+1}(p_i, n_j) = \frac{D_\tau(p_i, n_j)\exp(\alpha_\tau(h(p_i) - h(n_j)))}{Z_\tau},$$

where Z_τ is a normalization factor given by:

$$Z_\tau = \sum_{p_i, n_j} D_\tau(p_i, n_j)\exp(\alpha_\tau(h(p_i) - h(n_j))).$$

In practice, supposing that for (p_i, n_j) we want p_i to be ranked higher than n_j (in all other cases D_τ will be zero) and assuming $\alpha_\tau > 0$, this rule decreases the weight $D_\tau(p_i, n_j)$ if h_τ gives a correct ranking (i.e if $h_\tau(p_i) > h_\tau(n_j)$) and increases the weight otherwise. Thus, D_τ will tend to concentrate on the pairs whose relative ranking is hardest to determine. The weight α_τ is chosen to be equal to:

$$\alpha = \frac{1}{2}\ln\left(\frac{1+r}{1-r}\right),$$

with

$$r = \sum_{p_i, n_j} D_\tau(p_i, n_j)(h(p_i) - h(n_j)),$$

and the final ranking H is a weighted sum of the weak rankings:

$$H(\mathbf{x}) = \sum_{\tau=1}^{T} \alpha_\tau h_\tau(\mathbf{x}).$$

A variation of RankBoost is proposed in Rudin *et al.* (2005). In particular, since in the boosting approach the margin is an important indicator of the classifier's generalization ability, they provide a general margin-based bound for ranking and derive an algorithm able to create large margins. In fact, RankBoost is not directly built to maximize the ranking margin and thus, it may not increase the margin at every iteration. Therefore, they introduced the Smooth Margin Ranking algorithm that is based on a different estimation of the weights α_τ at each iteration and is able to make progress in increasing the ranking margin.

Another approach for the maximization of AUC is applied to the well known decision

trees (Breiman *et al.*, 1984). In Ferri *et al.* (2002) a novel splitting criterion is proposed to choose the split that guarantees the highest local AUC. In particular, the AUC between two consecutive points of the ROC curve obtained sorting the leaves of each split by local positive accuracy is evaluated and the value maximizing this quantity is used to determine the best split in terms of AUC for the decision tree.

Recently, rank optimizing classifiers based on the well known Support Vector Machines (SVM) (Vapnik, 1998) have came into focus. In Rakotomamonjy (2004) rank optimizing kernels has been investigated leading to a formulation that produced comparatively inferior results to the regular SVM. Given a linear classifier $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$, it minimizes the l_2-norm of $\mathbf{w}$ with constraints on the ordering of the objects. The optimization problem is defined as follows:

$$\min \left\|\mathbf{w}^2\right\| + C\sum_{i=1}^{m_P}\sum_{j=1}^{m_N}\xi_{ij}$$

$$s.t. \forall i,j : f(\mathbf{p}_i) - f(\mathbf{n}_j) \geq 1 - \xi_{ij}, \xi_{ij} \geq 0,$$

where C is a trade off parameter between the two parts of the objective and the slack variables ξ_{ij} are used to approximate the indicator function in the WMW statistic. By introducing a kernel, thanks to the self duality of the l_2-norm, the above formulation remains valid for nonlinear SVM.

A similar kernel formulation which led to a better ranking performance compared to the previous work has been introduced in Brefeld & Scheffer (2005). In their algorithm they used regularized quadratic optimization to find a kernel structure that improves the ranking performance. Moreover, they provide a method to achieve a lower computational complexity of the algorithm reducing the number of constraints of the problem representing the $m_P m_N$ pairs in $m_P + m_N$ cluster centers.

A linear programming approach similar to l_1-norm SVM (Bennett & Mangasarian, 1992) has been developed in Ataman *et al.* (2006) while in Tax *et al.* (2006) a similar linear weighting of features (called AUC Linear Programming Classifier(AUC-LPC)) has been successfully applied to the interstitial lung disease. In this cases the optimization problem for a linear classifier can be written as:

$$\min \left\|\mathbf{w}_1\right\| + C\sum_{i=1}^{m_P}\sum_{j=1}^{m_N}\xi_{ij}$$

$$s.t. \forall i,j : \mathbf{w}^T(\mathbf{p}_i - \mathbf{n}_j) \geq 1 - \xi_{ij}, \xi_{ij} \geq 0.$$

This can be rewritten in a linear programming formulation as:

$$\min \sum_h (u_h + v_h) + C \sum_{i=1}^{m_P} \sum_{j=1}^{m_N} \xi_{ij}$$

$$s.t. \forall i, j : (\mathbf{u}^T - \mathbf{v}^T)(\mathbf{p}_i - \mathbf{n}_j) \geq 1 - \xi_{ij}, \xi_{ij} \geq 0, \forall h : u_h \geq 0, v_h \geq 0.$$

Since using the slack variables ξ_{ij} to approximate the indicator function in the WMW statistic has serious drawback (the number of constraints is quadratic in the number of the objects), different strategies to speed up the algorithm are also proposed: in the first paper the strategy consists of randomly sample the objects from both classes while the latter randomly subsample the constraints avoiding to focus on the local structure of the data.

3.2 Linear Discriminant Functions and ROC Curve

In this section we focus on the analysis of the linear classifiers and the relative decision boundary to put in evidence its relation with ROC curve and AUC. A linear discriminant function can be written as:

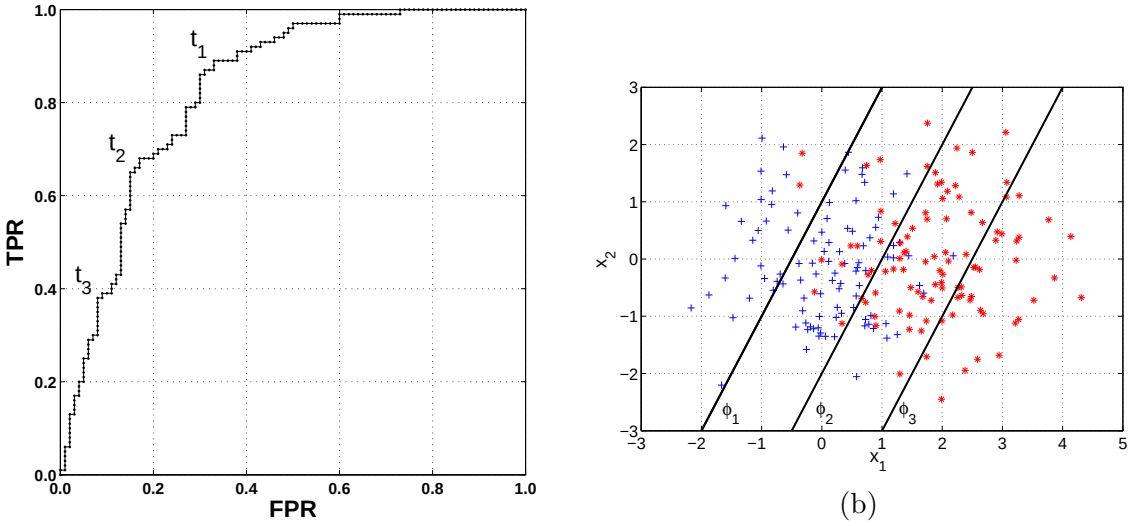
$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 = \sum_{i=1}^{L} w_i x_i + w_0, \tag{3.1}$$

where $\mathbf{w}$ is the weight vector and w_0 the threshold weight. We know that a sample $\mathbf{x}$ is assigned to the class P if $f(\mathbf{x}) > 0$ and to the class N if $f(\mathbf{x}) \leq 0$, i.e. $\mathbf{x}$ is assigned to P or to N if $\mathbf{w}^T \mathbf{x}$ exceeds or not the threshold $-w_0$. The equation $f(\mathbf{x}) = 0$ defines the decision boundary ϕ that separates the two decisions regions. In our case the decision boundary is a hyperplane. For a given threshold $w_0 = -t$ it is possible to define a *TPR* and a *FPR* as in eq. (2.10) and so it is possible to build an ROC curve and evaluate the corresponding AUC.

To put in evidence the relation between the discriminant linear function and the ROC curve let us focus on a two-dimensional problem. In this case the decision function simplifies in:

$$f(\mathbf{x}) = w_1 x_1 + w_2 x_2 + w_0, \tag{3.2}$$

and the sample $\mathbf{x}$ will be assigned to P if $w_1 x_1 + w_2 x_2 + w_0 > 0$ and to N otherwise. In this case the hyperplane collapses in a straight line with slope m equal to $-w_1/w_2$. Defining the two weights w_1 and w_2 is equivalent to fix the slope of the decision boundary



(a) (b)

Figure 3.1: Example of two-dimensional problem with two overlapping classes. Figure (a) shows the ROC curve for a linear classifier with three different operating points corresponding to the three straight lines shown in figure (b).

while varying the value $-w_0$ means that we are translating the boundary in the feature space. Once a particular value for the slope has been chosen, the distribution of the value w_0 produces a family of lines (decision boundaries) with the same slope. Each of them defines a particular classifier which produces a certain pair (TPR,FPR) corresponding to a particular point on the ROC curve. When the value of w_0 is varied (and thus the decision boundary is translated) the whole ROC curve is drawn. In summary, each value for m produces a particular ROC curve where each point is associated to one of the parallel lines with that slope in the feature space. In fig. 3.1 an example of a two-dimensional problem with two overlapping classes is shown. Fig. 3.1.a shows the ROC curve for a linear classifier with three different operating points corresponding to three different straight lines in the feature space represented in fig. 3.1.b. Each point (TPR,FPR) on the curve corresponds to a particular (operating point for the) classifier, i.e. to a particular value for the threshold $-w_0$ and, consequently, to one of the parallel lines defined by the chosen slope.

A same reasoning can be done for the AUC. In this case let us consider the decision function in eq. (3.2) and the decision boundary Φ. Given two points in the feature space $\mathbf{p} = (p_1, p_2)$ and $\mathbf{n} = (n_1, n_2)$ coming, respectively, from class P and N their signed

distance from the decision boundary are:

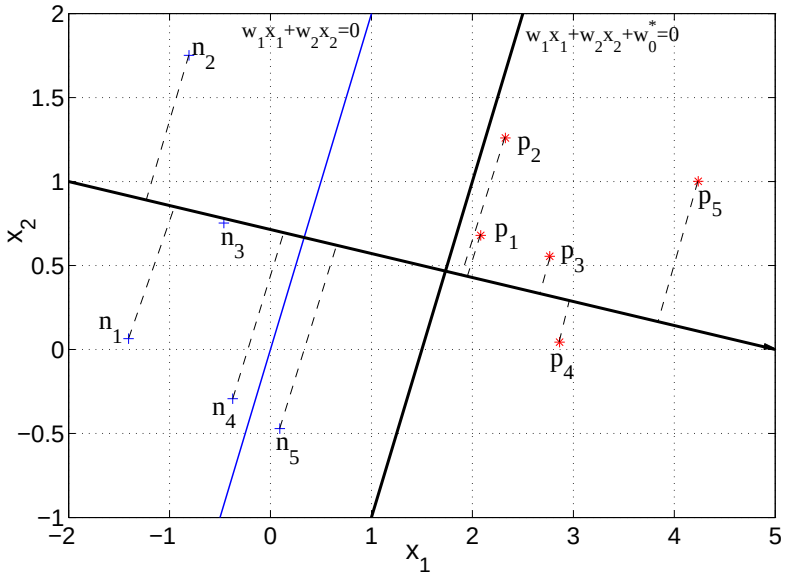
$$d(\mathbf{p}, \phi) = \frac{w_1 p_1 + w_2 p_2 + w_0}{\sqrt{w_1^2 + w_2^2}},$$
$$d(\mathbf{n}, \phi) = \frac{w_1 n_1 + w_2 n_2 + w_0}{\sqrt{w_1^2 + w_2^2}}.$$

Thus, for a correct ranking of the pair $\mathbf{p}$ and $\mathbf{n}$ we should have $d(\mathbf{p}, \phi) > d(\mathbf{n}, \phi)$; this means that the positive point follows the negative point on the line orthogonal to the decision boundary. As a consequence, the classification cannot be wrong for both samples: in the worst case, if the threshold is not adequately chosen, both the points lie on the same side of the decision boundary. However, a suitable shifting of the decision boundary allows the two points to be correctly classified. Hence, if $w_1 p_1 + w_2 p_2 > w_1 n_1 + w_2 n_2$, we can choose a threshold w_0^* such as:

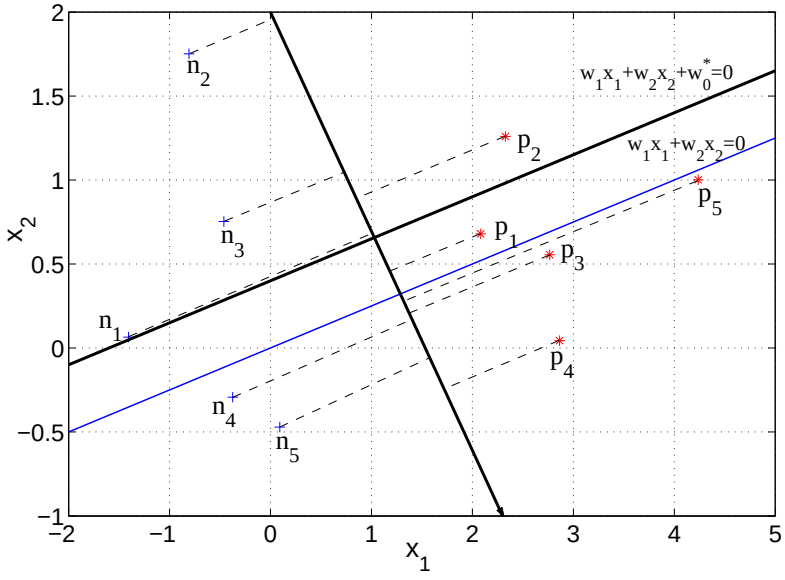
$$-(w_1 p_1 + w_2 p_2) \leq w_0^* \leq -(w_1 n_1 + w_2 n_2)$$
$$\Downarrow$$
$$w_1 p_1 + w_2 p_2 + w_0^* \geq 0 \text{ and } w_1 n_1 + w_2 n_2 + w_0^* \leq 0.$$

On the other side, if the pair $\mathbf{p}$ and $\mathbf{n}$ is not correctly ranked, there is no any value for the threshold w_0 which can correctly classify both the points. In summary, the slope maximizing the AUC is the slope for which there is the maximum number of pairs correctly ranked, i.e. of the pairs that could be correctly classified with a suitable choice of the threshold.

A two-dimensional example is shown in fig. 3.2. Five samples for the positive class and five for the negative class, i.e. twenty-five possible pairs, are plotted. In fig. 3.2.a the decision boundary ϕ is shown with a slope that maximizes the AUC. As an evidence, we consider the perpendicular to *phi* and project all the samples on that line; fixing a way to move along the line, it is possible to obtain a correct ranking among all the possible pairs ($AUC = 1$). If we choose a threshold on this line it is possible to build a linear classifier that is able to assign all the samples to the correct class. In fig. 3.2.b we have a similar situation but in this case the slope of the decision boundary does not lead to a perfect ranking ($AUC = 22/25$) and this reflects in errors in the class assignment.



(a)



(b)

Figure 3.2: Two-dimensional problem with five positive samples (asterisks) and five negative samples (plus signs). In (a) the decision boundary leading to a correct ranking, i.e. maximum AUC, is shown while in (b) no suitable threshold can be chosen to linearly separate the two classes.

3.3 AUC Maximization in the Two-Dimensional Feature Space

In the previous section we put in evidence the meaning of the AUC for a linear classifier. According to this, let us now focus on an approach to linearly combine features so as to maximize the AUC of the resulting linear classifier. To this aim, we firstly consider how to find an opportune slope for the linear combination of two features that maximizes the WMW statistic. Let X be the set of samples as defined in sec. 1.1 and let us consider two generic features x_h and x_k. Let us consider the values of the h-th and k-th features on the i-th positive sample p_i and the j-th negative sample n_j: p_i^h n_j^h p_i^k n_j^k and the relative ranking measures for the two features according to eq. (2.23) by:

$$R_h = \frac{\sum_{i=1}^{m_P}\sum_{j=1}^{m_N} I(p_i^h, n_j^h)}{m_P m_N} \qquad \text{and} \qquad R_k = \frac{\sum_{i=1}^{m_P}\sum_{j=1}^{m_N} I(p_i^k, n_j^k)}{m_P m_N}. \tag{3.3}$$

Since we want to maximize the AUC we are independent of the threshold. Hence, let us consider a linear combination of the two features:

$$x_{\text{lc}} = \alpha x_h + (1-\alpha) x_k, \tag{3.4}$$

where $\alpha/(1-\alpha)$ is the relative weight of the features x_k with respect to x_h. The value of x_{lc} on the positive and negative sample will be:

$$p_i^{\text{lc}} = \alpha p_i^h + (1-\alpha) p_i^k, \tag{3.5a}$$

$$n_j^{\text{lc}} = \alpha n_j^h + (1-\alpha) n_j^k. \tag{3.5b}$$

According to the WMW statistic the quality of the ranking of x_{lc} can be measured by

$$R_{\text{lc}} = \frac{\sum_{i=1}^{m_P}\sum_{j=1}^{m_N} I(p_i^{\text{lc}}, n_j^{\text{lc}})}{m_P m_N}, \tag{3.6}$$

and depends on the value of the weight α. We want to maximize the AUC relative to the pair of features; to this aim we have to analyze the term $I(p_i^{\text{lc}}, n_j^{\text{lc}})$ that depends on the value of $I(p_i^h, n_j^h)$ and $I(p_i^k, n_j^k)$. We can distinguish three different important cases[1]:

[1] It is worth noting that the cases relative to $I(a,b) = 1/2$ (that are only common in the nominal data) are treated as in the worst case, i.e. when the ranking between the features is wrong

- $I(p_i^h, n_j^h) = 1$ and $I(p_i^k, n_j^k) = 1$ means that according to both features the two samples are correctly ranked and $I(p_i^{\text{lc}}, n_j^{\text{lc}}) = 1$ independently of α.

- $I(p_i^h, n_j^h) = 0$ and $I(p_i^k, n_j^k) = 0$ means that neither feature correctly ranks the two samples and $I(p_i^{\text{lc}}, n_j^{\text{lc}}) = 0$ independently of α.

- $I(p_i^h, n_j^h) = 1$ and $I(p_i^k, n_j^k) = 0$ or $I(p_i^h, n_j^h) = 0$ and $I(p_i^k, n_j^k) = 1$ means that only one feature correctly ranks the two samples and the value of $I(p_i^{\text{lc}}, n_j^{\text{lc}})$ is dependent on α.

According to these cases we can subdivide the set of samples in four different subsets defined as:

$$X_{rs} = \left\{ (i,j) | I(p_i^h, n_j^h) = r \text{ and } I(p_i^k, n_j^k) = s \right\}. \tag{3.7}$$

As a consequence the expression of the ranking of the combined features will be:

$$\begin{aligned} R_{\text{lc}} &= \frac{1}{m_P m_N} \left[\sum_{(i,j)\in X_{00}} I(p_i^{\text{lc}}, n_j^{\text{lc}}) + \sum_{(i,j)\in X_{11}} I(p_i^{\text{lc}}, n_j^{\text{lc}}) + \sum_{(i,j)\in X_{10}\cup X_{01}} I(p_i^{\text{lc}}, n_j^{\text{lc}}) \right] \\ &= \frac{1}{m_P m_N} \left[0 + \text{card}(X_{11}) + \nu(\alpha) \right]. \end{aligned} \tag{3.8}$$

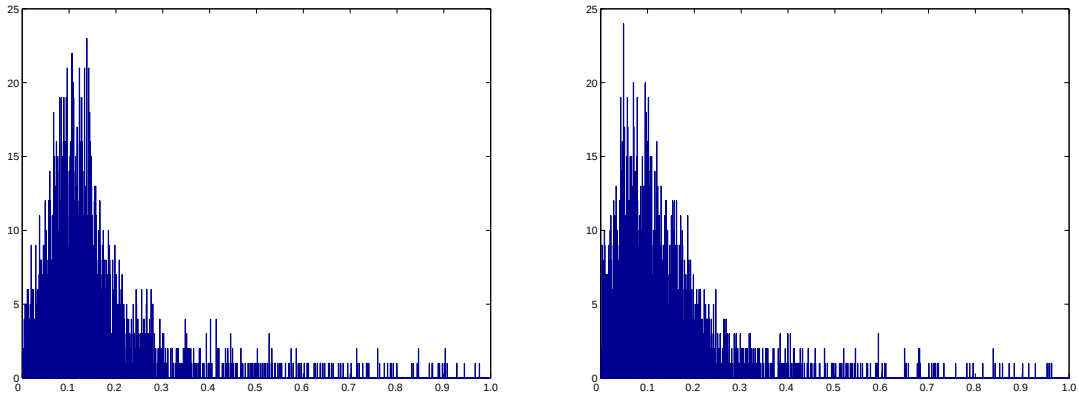
Hence, we have to focus on the pairs on which the features are differently ranked, i.e. on the sets X_{10} and X_{01}. In order to find the value of α that maximizes the ranking we have to study the term $\sum_{(i,j)\in X_{10}\cup X_{01}} I(\xi_i, \eta_j)$ looking at the weight for which $I(\xi_i, \eta_j) = 1 \Rightarrow \xi_i > \eta_j$, i.e. :

$$\begin{gathered} \alpha p_i^h + (1-\alpha) p_i^k > \alpha n_j^h + (1-\alpha) n_j^k \\ \Downarrow \\ \alpha \Delta_{ij}^h + (1-\alpha)\, \Delta_{ij}^k > 0, \end{gathered} \tag{3.9}$$

where $\Delta_{ij}^h = p_i^h - n_j^h$ and $\Delta_{ij}^k = p_i^k - n_j^k$. From eq. (3.9) we can obtain two different constraints on α according to which set we are considering; we have:

$$\alpha < \frac{\Delta_{ij}^k}{\Delta_{ij}^k - \Delta_{ij}^h} \quad \text{if } (i,j) \in X_{10}, \tag{3.10a}$$

$$\alpha > \frac{\Delta_{ij}^k}{\Delta_{ij}^k - \Delta_{ij}^h} \quad \text{if } (i,j) \in X_{01}. \tag{3.10b}$$



(a) (b)

Figure 3.3: Example of the distributions of the ratio $\frac{\Delta_{ij}^k}{\Delta_{ij}^k - \Delta_{ij}^h}$ evaluated on the sets X_{10} (a) and X_{01} (b)

If eqs. (3.10) is verified for each pair $(i,j) \in X_{10} \cup X_{01}$, it is possible to obtain the maximum value for the function $\nu(\alpha)$, i.e. the cardinality of $X_{10} \cup X_{01}$. In this case, we can find an optimum α:

$$\max_{(i,j)\in X_{01}} \frac{\Delta_{ij}^k}{\Delta_{ij}^k - \Delta_{ij}^h} < \alpha_{\text{opt}} < \min_{(i,j)\in X_{10}} \frac{\Delta_{ij}^k}{\Delta_{ij}^k - \Delta_{ij}^h}. \tag{3.11}$$

Anyway, this condition is verified only if the two sets are completely disjoint, i.e. if the two features are highly complementary in the ranking evaluation. When the two sets are not separated we have to evaluate the weight α using the cumulative distributions:

$$F_{10} = \text{card}\left((i,j) \in X_{10} \middle| \frac{\Delta_{ij}^k}{\Delta_{ij}^k - \Delta_{ij}^h} > \alpha \right), \tag{3.12a}$$

$$F_{01} = \text{card}\left((i,j) \in X_{01} \middle| \frac{\Delta_{ij}^k}{\Delta_{ij}^k - \Delta_{ij}^h} < \alpha \right). \tag{3.12b}$$

Hence, the function that has to be maximized is:

$$\nu(\alpha) = F_{10}(\alpha) + F_{10}(\alpha), \tag{3.13}$$

and the optimal value of α can be found by means of a linear search.

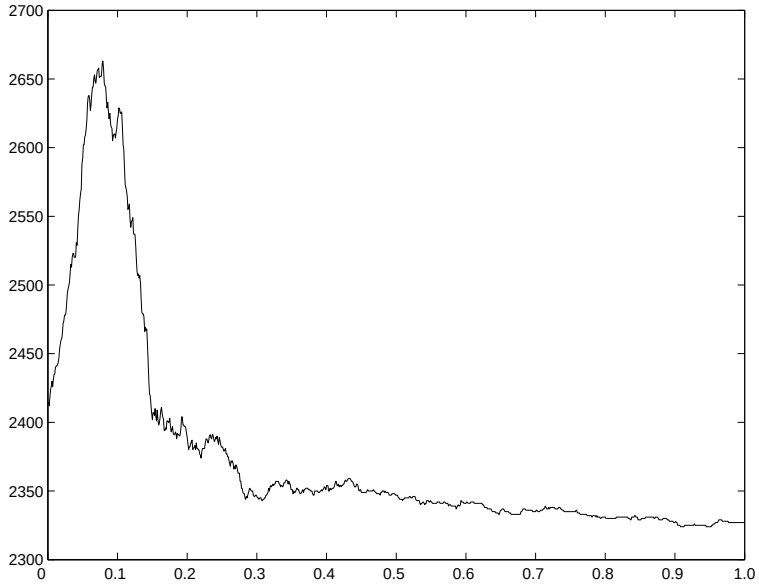


Figure 3.4: The trend of the function $\nu(\alpha)$ obtained by the two distributions shown in fig. 3.3

An example of the real distributions of the ratios $\Delta_{ij}^k \Big/ (\Delta_{ij}^k - \Delta_{ij}^h)$ in the two sets X_{10} and X_{01} is shown in fig. 3.3 while the function $\nu(\alpha)$ obtained by these two distributions is plotted in fig. 3.4.

3.4 AUC Maximization in the Multidimensional Feature Space

The next step of our method consists in extending the procedure described in the previous section to a higher number of features. To this aim, let us consider Q features $x_1 \dots x_Q$ and their linear combination:

$$x_{\text{lc}} = \alpha_1 x_1 + \cdots + \alpha_Q x_Q = \sum_{i=1}^{Q} \alpha_i x_i = \alpha^T \mathbf{x}. \tag{3.14}$$

The goal is to find the weight vector:

$$\alpha_{\text{opt}} = (\alpha_1 \dots \alpha_Q)\,, \tag{3.15}$$

maximizing the WMW statistic associated with the ranker described by x_{lc}. However, it is not possible to extend our approach to this case since the direct optimization of the

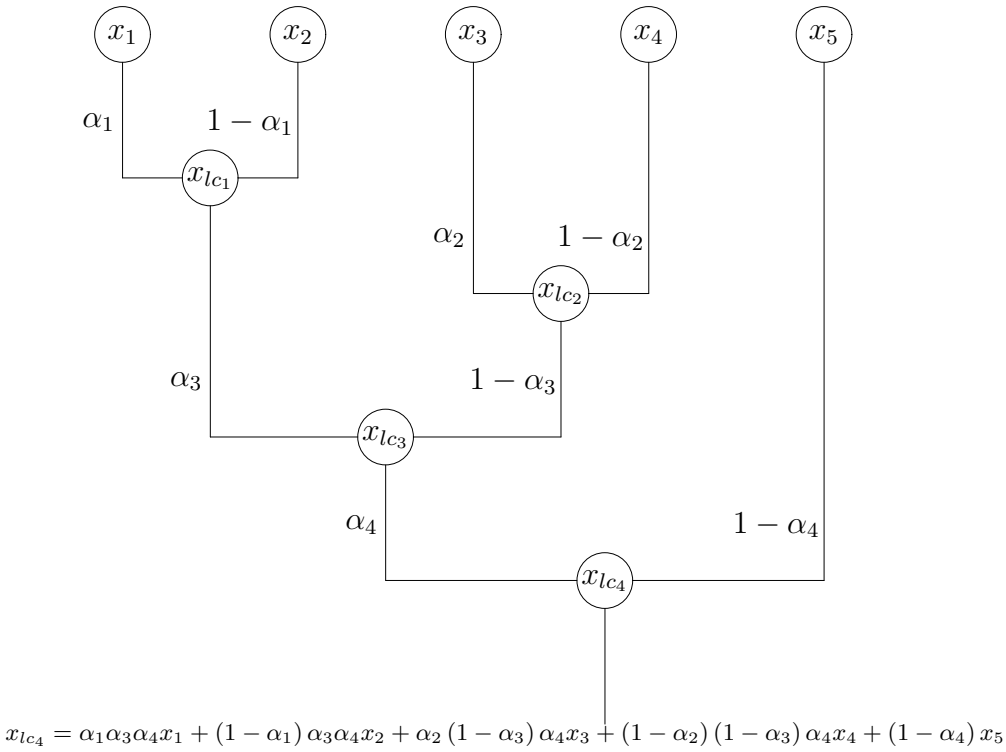


Figure 3.5: Example of the tree used to rebuild the weight vector. Moving from each leave to the root and multiplying the values on the edges we can recover the weight associated to each feature.

function in eq. (3.6) is intractable.

Therefore, a suboptimal algorithm that approximates the solution using a greedy approach has been adopted. Greedy methods build solutions piece by piece. Each step increases the size of the partial solution and is based on local optimization: the selected choice is the one that produces the largest immediate gain, i.e. the best ranking, maintaining the feasibility of the problem. In our case, instead of finding a weight vector in one step we iteratively find the optimal weight of the linear combination of two features (as described in the previous section) so as to evaluate all the combination weights in at most $Q-1$ steps.

In this context an important role is played by the order of combination, i.e. which pair of features should be combined at first, to avoid considering every possible combination. From eq. (3.11) we know that the more separated are the two distributions relative to the sets X_{10} and X_{01} the greater is the improvement to the ranking of the combined features. Therefore, it is possible to combine the features choosing the pair that exhibits

the maximum diversity in the ranking, i.e. the minimum rank correlation coefficient between features (Kuncheva *et al.*, 2000). To this aim, we choose the Spearman's rank correlation coefficient, a nonparametric measure of correlation that assesses how well an arbitrary monotonic function could describe the relationship between two features without making any assumptions about the frequency distribution (Lehmann & D'Abrera, 1975). It is:

$$\rho_{hk} = 1 - 6\frac{\sum_{i=1}^{L}\left(r_i^h - r_i^k\right)^2}{L\left(L^2-1\right)}, \tag{3.16}$$

where r_i^h and r_i^k are the rankings on the two considered features h and k.

Once the procedure has been repeated until a single feature is obtained (i.e. $Q-1$ times), it is necessary to recover the weight for each of the features to be combined. To this aim, a combination tree is built during the evaluation of the weight vector. The original features are the leaves of the tree and a parent node is added when a pair of features is combined. The edges are labelled with the weights assigned to each feature in each step. Multiplying the values found on the edges by running through the tree (from the leaves down to the root) it is possible to recover the weight for each single feature. Fig. 3.5 shows an example of such a tree for a classification problems with five features. In the first step the pair (x_1, x_2) is combined obtaining a new feature $x_{lc_1} = \alpha_1 x_1 + (1-\alpha_1)x_2$; in the second step x_3 and x_4 are combined and so on until we obtain a single feature x_{lc_4}. To recover the weights associated to $x_1 \dots x_5$, we multiply the weights on the edges that we encounter on the path from the single features to x_{lc_4}. As an example, moving from the feature x_3 towards x_{lc_4} we encounter the weights $\alpha_3, 1-\alpha_2$ and α_4 and so the weight relative to x_3 is $\alpha_3(1-\alpha_2)\alpha_4$.

A pseudo code of the whole algorithm is reported in 3.1.

3.5 Experiments

In this section some experiments are reported to assess the reliability of the proposed method. To evaluate our approach a comparison with other classifiers that work on ranking has been conducted. In particular, three classifiers has been used: SVM (Vapnik, 1998), (Cristianini & Shawe-Taylor, 2000), RankBoost (Freund *et al.*, 2003) and AUC-LPC (Tax *et al.*, 2006).

SVM is a well known classifier not directly built to maximize the ranking performance but to minimize the error rate; nevertheless, in literature it is considered to be a good ranker among classifiers methods. In our experiments a linear kernel has been considered

Algorithm 3.1 The Maximum AUC Linear Classifier (MALC)

Input: A L-dimensional matrix representing the training set with Q features $\mathbf{x}_1, \ldots, \mathbf{x}_Q$; the number of classifiers to be combined; $m_P > 0$ and $m_N > 0$, the number of positive and negative samples with $L = m_P + m_N$.

Output: α, the weight vector of the linear combination of features.

for $h = 1$ to Q **do**
  $r_h^{(0)} \leftarrow x_h^{(0)}$ sorted by decreasing values and ranked
end for
for $h = 1$ to $Q - 1$ **do**
  for $k = h + 1$ to Q **do**
    $\rho_{h,k}^{(0)} = 1 - \frac{6}{L(L^2-1)} \sum_{i=1}^{L} \left(r_i^h - r_i^k\right)^2$ /*evaluate the rank coefficient matrix at step 0*/
  end for
end for
for $m = 1$ to $Q - 1$ **do**
  $(\sigma, \tau) \leftarrow \arg\min_{h,k} \rho^{(m-1)}$ /*find the pair of classifiers with the minimum rank coefficient*/
  for $i = 1$ to m_P **do**
    for $j = 1$ to m_N **do**
      $X_{rs} \leftarrow \left\{(i,j) \middle| I\left(p_i^\sigma, n_j^\sigma\right) = r \text{ and } I\left(p_i^\tau, n_j^\tau\right) = s\right\}$ with $r, s = 0, 1$
      $\Delta_{i,j}^\sigma = p_i^\sigma - n_j^\sigma$
      $\Delta_{i,j}^\tau = p_i^\tau - n_j^\tau$
    end for
  end for
  evaluate $F_{10}^{\sigma\tau}$ and $F_{01}^{\sigma\tau}$
  $\nu(\alpha) \leftarrow F_{10}^{\sigma\tau} + F_{01}^{\sigma\tau}$
  $\alpha_{opt} \leftarrow \max_\alpha \nu(\alpha)$
  update the combination tree
  $r_{\sigma+\tau}^{(m)} \leftarrow \alpha x_\sigma + (1 - \alpha) x_\tau$ sorted by decreasing values and ranked
  $n \leftarrow 2$
  while ($m < Q - 1$ and $n < Q - m$) **do**
    $\rho_{\sigma+\tau,n}^{(m)} = 1 - \frac{6}{L(L^2-1)} \sum_{i=1}^{L} \left(r_i^{\sigma+\tau} - r_i^n\right)^2$
    $n \leftarrow n + 1$
  end while
end for
$\rho^{(m)} \leftarrow \rho^{(m-1)}$ eliminating $\rho_\sigma^{(m-1)}$ and $\rho_\tau^{(m-1)}$ and adding $\rho_{\sigma+\tau}^{(m)}$ /*update the rank correlation matrix*/
evaluate α by multiplying the values on the edges of the tree

due to the linear behavior of the proposed approach. A linear classifier similar to SVM is the AUC-LPC that is specifically setup to maximize ranking performance. As we introduced in subsec. 3.1.1 a serious drawback of this method is the number of constraints (quadratic in the number of the considered objects) that leads to a non optimal solution. Another technique well known in literature is the RankBoost that utilizes a multistage approach to combine preferences according to the boosting method (Freund & Schapire, 1997). Even if RankBoost is a multistage non-linear approach the comparison with this method is performed since it is considered to be the state of the art algorithm.

In order to evaluate the performance of the proposed method (hereafter called Maximum AUC Linear Classifier (MALC)), experiments on both artificial and real data sets has been performed. The former approach has been used to put in evidence the behavior of the MALC on known data distributions while experiments on real data have been performed to verify the utility of our method even when dealing with real problems. In particular, the admissibility of MALC on some data sets has been proved in a statistical way with respect to the employed methods.

To avoid any bias in the comparison a 10-fold cross validation procedure (Duda *et al.*, 2001) has been performed on all data sets. In each run 9 folds have been used as training set to train the classifiers and the remaining fold as test set to evaluate the classifiers performance.

All the classifiers have been implemented by means of PRTools (Duin, 2000) (van der Heijden *et al.*, 2004) toolbox. Since SVM and AUC-LPC are parametric classifiers, different architectures of these algorithms have been employed. In particular, we have varied the C parameter (see subsec. 3.1.1) for both the SVM and the AUC-LPC between 0.1 and 1000. For the sake of readability we report in the following tables only the best results obtained for these classifiers.

It is worth repeating that the comparison has been performed in terms of AUC since we are aiming at the maximization of the ranking quality of the classifier and not at the evaluation of the error rate (or other measures depending on a threshold value). Hence, in our experiments only the value of the AUC has been evaluated using the WMW statistic according to eq. (2.23).

3.5.1 The Artificial Data

The first part of our experiments focuses on synthetic data: in particular, some data sets have been built to analyze the characteristics of the classifier. To this aim, a Gaussian distribution for both classes has been used to generate 500 samples varying three param-

Table 3.1: Results on the test set for a Gaussian data set with uncorrelated class distributions, Δ_μ equal to 0.3 and variable number of features.

Q \ Classifiers	MALC	SVM	AUC-LPC	RankBoost
5	0.616 (0.073)	0.615 (0.074)	0.615 (0.073)	0.594 (0.105)
10	0.575 (0.054)	0.552 (0.045)	0.562 (0.060)	0.573 (0.063)
30	0.584 (0.095)	0.559 (0.082)	0.570 (0.110)	0.558 (0.075)
50	0.555 (0.067)	0.550 (0.072)	0.548 (0.048)	0.596 (0.061)
75	0.540 (0.042)	0.538 (0.068)	0.522 (0.071)	0.586 (0.070)
100	0.533 (0.044)	0.528 (0.068)	0.512 (0.047)	0.551 (0.062)

Table 3.2: Results on the test set for a Gaussian data set with uncorrelated class distributions, Δ_μ equal to 0.5 and variable number of features.

Q \ Classifiers	MALC	SVM	AUC-LPC	RankBoost
5	0.685 (0.073)	0.674 (0.065)	0.681 (0.068)	0.654 (0.084)
10	0.600 (0.060)	0.593 (0.059)	0.582 (0.062)	0.570 (0.094)
30	0.611 (0.094)	0.605 (0.091)	0.604 (0.099)	0.582 (0.075)
50	0.610 (0.057)	0.611 (0.075)	0.598 (0.052)	0.619 (0.072)
75	0.575 (0.065)	0.572 (0.066)	0.567 (0.054)	0.577 (0.083)
100	0.561 (0.043)	0.560 (0.049)	0.552 (0.047)	0.583 (0.075)

eters: the number of features Q to put in evidence the behavior of the greedy approach for high dimensionality problems, the difference Δ_μ between the means of the two classes distributions to evaluate the performance for different overlapping of data and the covariance matrix of the classes distributions to consider both correlated and uncorrelated data. Q has been varied between 5 and 100 while Δ_μ has been varied between 0.3 to 1 for uncorrelated data and between 1 and 3 for correlated data. In our case, greater values of this parameter have no significance since the two classes become easily separable. More characteristic of the employed data sets are reported in appendix A.

The results of the comparison of the four employed classifiers are presented in tables 3.1-3.3 for uncorrelated data and in tables 3.4-3.6 for correlated class distributions. Each cell of the tables contains a value corresponding to the mean (and the standard deviation in parentheses) of the AUC relative to the performance of each classifier on each data set for the relative number of features.

Firstly, let us analyze the results obtained on uncorrelated data. In this case, it is possible to highlight the good behavior of MALC when the dimensionality of the feature space is not high. In fact, MALC gives the highest mean value among the four classifiers

Table 3.3: Results on the test set for a Gaussian data set with uncorrelated class distributions, Δ_μ equal to 1 and variable number of features.

Q \ Classifiers	MALC	SVM	AUC-LPC	RankBoost
5	0.939 (0.019)	0.937 (0.023)	0.936 (0.022)	0.931 (0.022)
10	0.920 (0.047)	0.914 (0.047)	0.910 (0.049)	0.913 (0.047)
30	0.929 (0.032)	0.921 (0.038)	0.918 (0.037)	0.931 (0.025)
50	0.915 (0.028)	0.880 (0.038)	0.888 (0.024)	0.919 (0.027)
75	0.741 (0.059)	0.726 (0.079)	0.701 (0.074)	0.800 (0.048)
100	0.886 (0.037)	0.843 (0.058)	0.841 (0.051)	0.908 (0.022)

Table 3.4: Results on the test set for a Gaussian data set with correlated class distributions, Δ_μ equal to 1 and variable number of features.

Q \ Classifiers	MALC	SVM	AUC-LPC	RankBoost
5	0.809 (0.055)	0.771 (0.052)	0.767 (0.053)	0.711 (0.070)
10	0.796 (0.015)	0.773 (0.077)	0.771 (0.069)	0.662 (0.026)
30	0.756 (0.041)	0.740 (0.052)	0.733 (0.054)	0.654 (0.065)
50	0.655 (0.501)	0.707 (0.065)	0.726 (0.061)	0.602 (0.090)
75	0.634 (0.080)	0.686 (0.067)	0.666 (0.057)	0.598 (0.068)
100	0.632 (0.076)	0.633 (0.060)	0.641 (0.064)	0.605 (0.042)

on all the employed data sets until the value of Q is lower than 50. When Q becomes greater than 50 RankBoost exhibits the best performance but MALC is still better than SVM and AUC-LPC (there is just one exception for the data set with Δ_μ equal to 0.5 and Q equal to 50 where SVM is better than MALC). The reason for the good performance of RankBoost in high dimension space can be found in its characteristics since it is a multistage approach that trains different classifiers on different samples in each round of its procedure (see subsec.3.1.1). On the contrary, the three linear classifiers have a different behavior on these data: if we look at tables 3.1-3.3, we can observe that MALC loses at least the 5% in AUC when passing from 30 to 100 features and a similar behavior is shown by SVM and AUC-LPC that, as reported by the same authors (Vapnik, 1998), (Tax *et al.*, 2006), suffer the high dimensionality of data. In our approach problems that occur with high dimensionality data are probably due to the greedy approach, i.e. to the propagation of the error in each step of the algorithm.

For correlated data the situation is more complicated to analyze since there is no clear dominance of one method above the others. In this case, RankBoost does not exhibit good performance in comparison with the other rankers probably due to the correlation among

Table 3.5: Results on the test set for a Gaussian data set with correlated class distributions, Δ_μ equal to 2 and variable number of features.

Q \ Classifiers	MALC	SVM	AUC-LPC	RankBoost
5	0.920 (0.013)	0.920 (0.039)	0.920 (0.037)	0.862 (0.035)
10	0.928 (0.022)	0.926 (0.042)	0.927 (0.039)	0.844 (0.087)
30	0.901 (0.072)	0.933 (0.027)	0.932 (0.027)	0.810 (0.027)
50	0.919 (0.021)	0.906 (0.017)	0.901 (0.024)	0.808 (0.075)
75	0.826 (0.070)	0.849 (0.041)	0.854 (0.042)	0.803 (0.082)
100	0.838 (0.075)	0.859 (0.037)	0.863 (0.040)	0.833 (0.068)

Table 3.6: Results on the test set for a Gaussian data set with correlated class distributions, Δ_μ equal to 3 and variable number of features.

Q \ Classifiers	MALC	SVM	AUC-LPC	RankBoost
5	0.980 (0.024)	0.979 (0.023)	0.979 (0.023)	0.843 (0.047)
10	0.985 (0.011)	0.984 (0.015)	0.985 (0.014)	0.862 (0.034)
30	0.975 (0.014)	0.975 (0.018)	0.971 (0.018)	0.835 (0.053)
50	0.972 (0.011)	0.969 (0.021)	0.972 (0.016)	0.844 (0.042)
75	0.979 (0.015)	0.968 (0.019)	0.970 (0.021)	0.825 (0.0686)
100	0.970 (0.055)	0.963 (0.022)	0.977 (0.019)	0.847 (0.061)

the weak classifiers that are used to perform the boosting approach. If we compare the linear rankers it is possible to highlight a dominance of MALC for low dimensional data, i.e. until Q is lower than 30. When the number of features grows, we have a different behavior according to the overlapping of the classes. When the classes are well separated (see table 3.6) MALC is the best classifier if we exclude one case ($Q = 100$ where AUC-LPC performs better). In table 3.5 for a medium overlapping we have a similar behavior (i.e. for $Q = 100$ and $Q = 75$ MALC is not the best ranker) except for $Q = 30$ where SVM and AUC-LPC perform better than MALC. When the data are almost completely overlapping (see table 3.4) the performance of MALC decrease quickly when Q grows and even for $Q = 50$ it exhibits lower values of the AUC than the other two linear classifiers.

In the end, from the analysis of artificial data we have shown that the proposed method performs very well for Gaussian data with low dimensionality (less than 50 features) both for correlated and uncorrelated data distributions, i.e. our ranker is admissible in comparison with well known methods in literature.

3.5.2 Experiments on Real Data Sets

In this subsection we propose another type of experiments based on real data sets to confirm the behavior of MALC shown on artificial data. However, there is no standard experimental set up to build a variety of classifiers and the probability that it occurs in a real-life experiment is a vacuous notion (Kuncheva & Whitaker, 2003). Then, we do not need to create a classifier for the purpose of finding out whether it is better or not of another one. Hence, our goal is not to find a new classifier that always performs better than all the other methods but to propose an admissible classifier. We can say that a classifier is admissible if no other classifier performs always equal or better, i.e. we demand that the proposed method is somewhere best.

To this aim, the proposed method has been tested on several data sets publicly available at the UCI Machine Learning Repository (Blake *et al.*, 1998). All of them have two classes[2] and a variable number of numerical input features. More details for each data sets are given in appendix A.

The results obtained for the four classifiers are reported in table 3.7 on 22 data sets. Each cell of the tables contains a value corresponding to the mean (and the standard deviation in parentheses) of the AUC relative to the performance of each classifier on each data set.

A first analysis can be done looking at the mean AUC values; in this case we can note that our algorithm performs better than the others in 12 (9 plus 3 ties) of the 22 considered data sets while SVM on 6 (4 plus 2 ties), RankBoost on 5 (4 plus 1 tie) and AUC-LPC on 2. Moreover, MALC exhibits the worst performance among the four classifiers just on the Diabetes data set and only in 3 cases (Glass2, Sonar and Wine1) has lower performance than two of the other methods.

To give more reliability to the comparison a statistical test has been performed. Statistics offers more powerful specialized procedures for testing the significance of differences between multiple means. In our situation, the most interesting is the Friedman test (Friedman, 1937), i.e. a non parametric equivalent of the well known ANOVA (Fisher, 1959). Friedman (1940) experimentally compared ANOVA and his test on 56 independent problems showing that the two methods mostly agree. The problem is that ANOVA is based on assumptions which are most probably violated when analyzing the performance of learning algorithms. First, ANOVA assumes that the samples are drawn from normal distributions and in general this is not guaranteed across a set of problems. However, even

[2]Three multiclass data sets (Glass, Waveform and Wine) have also been used. In this case, a One vs. All approach has been applied to select two of the classes from the multiclass data set (see Appendix A for more details).

Table 3.7: Results obtained in the experiments performed on real data sets.

Classifiers / Data Sets	MALC	SVM	AUC-LPC	RankBoost
Arrhythmia	0.783 (0.066)	**0.720** (**0.097**)	0.765 (0.095)	0.736 (0.070)
Biomed	0.968 (0.044)	0.958 (0.041)	0.961 (0.040)	**0.927** (**0.069**)
Breast	0.996 (0.005)	0.995* (0.006*)	0.994* (0.006*)	0.990* (0.008*)
Cancer_wpbc	0.762 (0.041)	0.780 (0.055)	0.762 (0.047)	0.741 (0.046)
Diabetes	0.811 (0.038)	0.826 (0.035)	0.821 (0.042)	0.834 (0.058)
Glass1	0.841 (0.084)	**0.840** (**0.093**)	**0.827** (**0.095**)	0.870 (0.056)
Glass2	0.660 (0.048)	0.627 (0.046)	0.714 (0.059)	0.748 (0.038)
Glass3	0.838 (0.045)	0.804 (0.054)	0.782 (0.082)	0.802 (0.031)
Glass4	0.943 (0.086)	0.940 (0.031)	0.925 (0.057)	0.937 (0.025)
Glass5	0.921 (0.029)	0.920 (0.033)	0.948 (0.023)	0.912 (0.058)
Heart	0.895 (0.057)	0.895 (0.057)	0.902 (0.060)	**0.542** (**0.058**)
Hepatitis	0.855 (0.048)	**0.784** (**0.074**)	**0.802** (**0.067**)	**0.790** (**0.053**)
Ionosphere	0.893 (0.087)	0.895 (0.056)	0.880 (0.070)	**0.701** (**0.120**)
Liver	0.720 (0.069)	0.714 (0.044)	0.718 (0.063)	0.720 (0.085)
Sonar	0.821 (0.046)	0.845 (0.042)	0.85 (0.050)	0.830 (0.030)
Thyroidsub	0.987 (0.013)	**0.973** (**0.017**)	0.984 (0.014)	0.998 (0.001)
Waveform1	0.937 (0.029)	0.937 (0.027)	**0.931** (**0.030**)	**0.921** (**0.027**)
Waveform2	0.940 (0.037)	0.922 (0.027)	**0.919** (**0.030**)	0.935 (0.031)
Waveform3	0.953 (0.036)	0.953 (0.037)	**0.948** (**0.036**)	**0.942** (**0.033**)
Wine1	0.996 (0.013)	1.000 (0.000)	**0.971** (**0.042**)	0.999 (0.004)
Wine2	0.994 (0.014)	0.993 (0.014)	**0.977** (**0.037**)	**0.990** (**0.017**)
Wine3	0.999 (0.005)	0.999 (0.005)	0.980 (0.063)	0.995 (0.010)

if distributions are not normal this is a minor problem and in many cases ANOVA is used unless the distributions were, for instance, clearly bimodal (Hamilton, 1990). The second and more important assumption is sphericity, a property similar to the homogeneity of variance which requires that the distributions have equal variance. Due to the nature of the learning algorithms and data sets this cannot be taken for granted. Therefore, ANOVA does not seem to be a suitable omnibus test for the study of learning problems.

In recent papers (Demšar, 2006) has been pointed out that the Friedman test even if it has theoretically less power than parametric ANOVA (when the ANOVA's assumptions are met) results to be more general than the ANOVA. This test ranks the algorithms separately: the best performing algorithm gets the rank of 1, the second best rank 2 and so on as shown in table 3.8. In case of ties average ranks are assigned. In our case, the null hypothesis for the Friedman test corresponds to a not statistically significant difference between the mean AUC of the employed methods. Therefore, when the null hypothesis is rejected there is a statistical difference among the classifiers. In our comparison the Friedman test has been performed with 3 (number of algorithms $-1 = 4 - 1$) and 36 ((number of algorithms -1)$*$(number of runs of the cross validation -1) $= (4-1)*(10-1)$) degrees of freedom (see appendix B).

In this case, we can proceed with a post-hoc test to find out which classifiers exhibits a statistically different behavior. Two different situations can be evaluated: to compare

Table 3.8: Comparison of the AUC obtained with the cross validation procedure on the employed methods using the Hepatitis data set. In parentheses we report the ranks that are used in the computation of the Friedman test and in the last rows the average rank obtained for each method to order the classifiers in the Holm procedure.

Cross Validation Runs	**Classifiers**			
	MALC	**SVM**	**AUC-LPC**	**RankBoost**
1	0.722 (3)	0.694 (4)	0.833 (1)	0.778 (2)
2	1.000 (1)	0.972 (2)	0.944 (3)	0.875 (4)
3	0.861 (1)	0.750 (3.5)	0.833 (2)	0.750 (3.5)
4	0.778 (1)	0.500 (4)	0.722 (2)	0.708 (3)
5	1.000 (1)	0.861 (3)	0.750 (4)	0.875 (2)
6	0.861 (3)	0.944 (1.5)	0.944 (1.5)	0.792 (4)
7	0.694 (2.5)	0.778 (1)	0.694 (2.5)	0.583 (4)
8	0.974 (1)	0.897 (2)	0.821 (3)	0.808 (4)
9	0.923 (1)	0.789 (4)	0.865 (2)	0.846 (3)
10	0.731 (2)	0.654 (3)	0.615 (4)	0.885 (1)
Average Rank	1.65	2.50	2.80	3.05

all the classifiers between each other or to compare all classifiers with a control method. However, the power of a post-hoc test is much greater in the second case when all classifiers are compared with a single method such as compare a newly proposed classifier with several existing methods. Thus, since we are testing if MALC gives better performance than the existing methods, we focus on the Holm's step-down procedure (Holm, 1979), that, in addition, does not make any additional assumptions about the hypotheses tested. Holm's procedure starts with the most significant rank r value. If r_1 is below $\alpha_{\text{ls}}/(k-1)$ (where α_{ls} is the level of significance of the test), the corresponding hypothesis is rejected and we are allowed to compare r_2 with $\alpha_{\text{ls}}/(k-2)$. If also the second hypothesis is rejected, the test proceeds with the third, and so on. As soon as a certain null hypothesis can not be rejected, all the remaining hypotheses are retained as well. More details on the employed tests can be found in appendix B.

As an example in table 3.8 we report the results obtained in terms of AUC on the Hepatitis data set on the 10 folds of the cross validation procedure. In parentheses we report the ranks among the classifiers that are used in the computation of the Friedman test. The average rank used to order the classifiers in the Holm procedure is reported in the last row of the table. In this example the Friedman test rejects the null hypothesis (i.e. there is statistical difference between some of the classifiers) and the Holm's test can be performed.

The obtained results are reported in table 3.7. A bold value in the table indicates that

the corresponding method on that data set has lower statistically significant performance than MALC according to the Holm procedure. If the value is underlined MALC exhibits lower performance compared to that method while if the value is in normal style it means that the corresponding method has undistinguishable performance from MALC. When the values in a row of the table are signed with an asterisk there is no statistical difference according to the Friedman test (i.e. the null hypothesis can not be rejected). All the tests (both the Friedman and the Holm test) have been performed with a level of significance equal to 0.05.

From these results we can see that there is a statistical difference among the employed methods according to the Friedman test in 14 of the considered data sets. In one case (i.e. Breast data set) the null hypothesis is rejected according to the Friedman test but the post-hoc fails to detect which classifiers are statistically different due to the lower power of the post-hoc with respect to Friedman (in such a case the only thing that we can say is that some of the algorithms differs but no other conclusions can be drawn). On the 14 data sets for which a statistical difference is found according to Friedman we can consider that only in two cases (Glass2 and Thyroidsub) MALC is worst than one of the other methods (in these cases RankBoost) while the proposed method results in four cases better than SVM and in seven cases better than RankBoost and AUC-LPC.

In conclusion, we have shown that also on real data the proposed approach can be profitably used to maximize the AUC on the analyzed problem. In fact, the reported results show the admissibility of the classifier on some of the considered data sets and therefore, our ranker is able to compete with other well known methods proposed in literature.

Chapter 4

Linear Combination of Classifiers via the AUC

In order to improve the classification performance, a well established technique is to combine more classifiers so as to take advantage of the strengths of the single classifiers and avoid their weaknesses. To this aim, a huge number of possible combination rules has been proposed up to now which generally try to decrease the classification error.

In this chapter, after a brief review of the characteristics of classifiers combination, we propose a method based on AUC maximization to achieve an optimal combination between already trained dichotomizers. In particular, in this work, we focus on the linear combination since it is the most frequently adopted in literature. Our problem consists in finding the optimal parameters to maximize the AUC of the resulting classification system. To this aim, an analysis of the dependence of the AUC on the weights has been performed and a method to find the optimal weights for two dichotomizers has been carried out. In order to accomplish an effective way to find α_{opt} and to extend the method to $K > 2$ dichotomizers, we introduce a new curve (the Difference Ratio Operating Characteristic curve) and discuss the problem of measuring the diversity among dichotomizers referred to their ranking capability. A greedy approach is proposed to extend the combination method to several classifiers.

4.1 Multiple Classifier Systems

The term *classifier fusion* or *multiple classifier system* usually refers to the combination of predictions from multiple classifiers to yield a single class prediction (Webb, 2002). The idea of combining classifiers is not a new one, but it has received increasing attention

in recent years. Early developed techniques focused on the combination of two-class discrimination rules (Devijver & Kittler, 1982) and also recursive partitioning methods (such as decision trees (Breiman *et al.*, 1984)) lead to the idea of defining different rules for different parts of a feature space. The terms "classifier selection" (Woods *et al.*, 1997) and "classifier choice" (Hand *et al.*, 2001) have been introduced for classification systems that attempt to predict the best classifier for a given region of the feature space. Recently, many experimental works have shown the improvement in performance that can be achieved by multiple classifiers in several applications (Kuncheva, 2005), (Oza *et al.*, 2005).

By combining classifiers we are aiming at a more accurate classification decision at the expense of increased complexity (Ho, 2002). In Dietterich (2001)three different reasons are presented to explain why a classifier ensemble should be better than a single classifier:

- Statistical: a statistical problem arises when the amount of training data available is too small compared to the size of the hypothesis space. Without sufficient data, a learning algorithm can find different hypotheses that all give the same accuracy on the training data. By constructing an ensemble out of all of these accurate classifiers, the algorithm can "average" their votes and reduce the risk of choosing the wrong classifier.

- Computational: many classifiers work by performing local search that may get stuck in local optima. An ensemble constructed by running the local search from many different starting points may provide a better approximation to the true unknown function than any of the individual classifiers.

- Representational: in most applications of machine learning and pattern recognition, the decision function f can not be represented by any of the possible hypotheses. By forming weighted sums of hypotheses, it may be possible to expand the space of representable functions.

A starting point for grouping ensemble methods can be sought in the ways of building the ensemble (Kuncheva, 2004). According to fig. 4.1 four different approaches aiming at building ensembles of diverse classifiers can be considered:

- Combination level: when different combiners are designed, i.e. when different ways of combining classifier decisions are chosen independently of the employed base classifiers (in this work we will focus on the design of this level of the ensemble).

- Classifier level: different classifiers can be used as base classifiers for the ensemble. The model of the classifiers is chosen according to interpretability of their decision

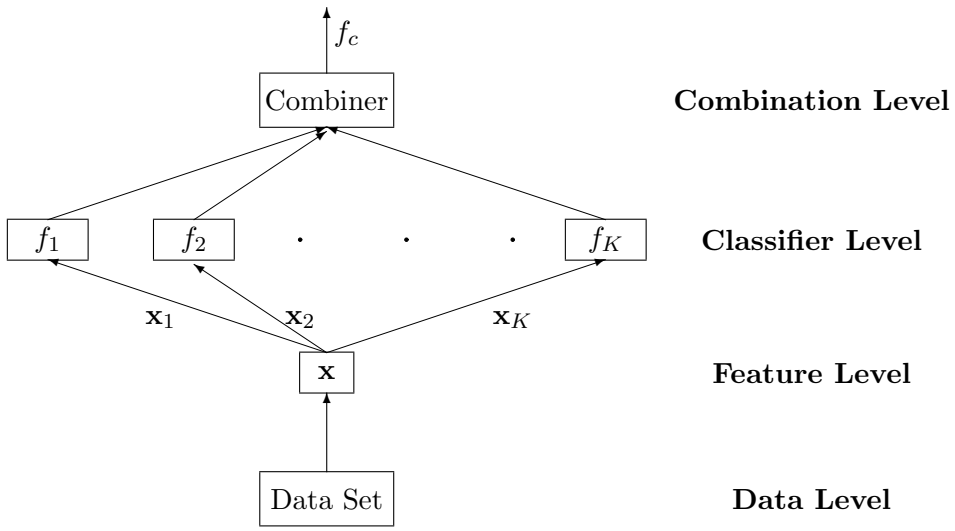


Figure 4.1: Combination system of classifiers evidencing the different levels that can be modified to build the ensemble.

process, implementability (the base classifiers are easy to implement with low computational complexity) and adaptability (the base classifiers has to be adaptable to different problems) (Webb, 2002).

- Feature level: when different feature subsets are used for the classifiers.

- Data level: if the data set can be modified so as each classifier is trained on a different data set (this approach is successfully in the bagging (Breiman, 1996) and boosting (Freund & Schapire, 1997) methods).

Another important characteristic of a combination system is the structure that can be:

- Serial: when the base classifiers are used sequentially with the output of one used by the next one in the sequence.

- Parallel: if all the results of the base classifiers are passed together to the combiner that takes the final decision.

- Hierarchical: when the classifiers are combined in a hierarchy with the output of the base classifiers used as inputs to a parent node.

4.2 Characteristics of a Combiner

The different ways of combining the outputs of K classifiers in an ensemble depend on which is the information that we obtain from the base classifiers. In Xu *et al.* (1992) three different classifier outputs (and so three different types of combiners) are defined:

- The abstract level: when each classifier produces a class label without any further information about the reliability of the predicted class (since any classifier is able to produce a label, this level is the most general).
- The rank level: when the output of each classifier is a subset of the set of the classes, with the alternatives ranked in order of reliability of being the correct label (Tubbs & Alltop, 1991), (Ho *et al.*, 1994).
- The measurement level: when each classifier produces a confidence degree on each class, i.e. an estimate (or a measure akin to an estimate) of the probability that a sample belongs to one of the classes.

In Kuncheva (2004) another level has been introduced:

- The oracle level: when the output of the classifier is just to know if the decision is wrong or correct without any knowledge on the class label that has to be assigned.

Another important characteristic of a combination rule is the choice of the training strategy of the combiner since we can have trainable and non trainable combiners. The choice is dependent on which part of the combining scheme we want to optimize, i.e. if we want to optimize the combiner alone or the base classifiers or both. According to this, a non trainable combiner, i.e. a fixed rule of combination (such as sum, product, maximum, majority etc.), can be used if we are sure that the base classifiers are not overtrained while if we use undertrained base classifiers a trainable combiner is preferred (Duin, 2002). An important aspect of the trainable combiner is the right choice of the training set. Duin (2002) suggests to choose a separate training set for the base classifiers and the combiner while in Dietrich *et al.* (2003) the second training set is chosen as partly overlapping of the first one used for the base classifiers. An alternative to these approaches is given by the stacked generalization proposed by Wolpert (1992) that improve the generalization in pattern classification.

Simple non trainable combiners calculate the support for each class ω_j using only the output of the classifiers involved in the combination for that class by:

$$f_{c,j}(\mathbf{x}) = F\left(f_{1,j}(\mathbf{x}) \dots f_{K,j}(\mathbf{x})\right), \tag{4.1}$$

where F is a combination function. The class label of $\mathbf{x}$ is found as the index of the maximum $f_{c,j}(\mathbf{x})$. The combination function F can be chosen in many different ways. The most popular choices are:

- Simple Average (SA) where F is the arithmetic average:

$$f_{\mathrm{SA},j}(\mathbf{x}) = \frac{1}{K}\sum_{i=1}^{K} f_{i,j}(\mathbf{x}). \tag{4.2}$$

- Minimum where F is the minimum function:

$$f_{\mathrm{MIN},j}(\mathbf{x}) = \min_i f_{i,j}(\mathbf{x}). \tag{4.3}$$

- Maximum where F is the maximum function:

$$f_{\mathrm{MAX},j}(\mathbf{x}) = \max_i f_{i,j}(\mathbf{x}). \tag{4.4}$$

- Median where F is the median function:

$$f_{\mathrm{MED},j}(\mathbf{x}) = \mathrm{median}_i f_{i,j}(\mathbf{x}). \tag{4.5}$$

- Trimmed Mean (competition jury): for a certain percentage pc trimmed mean the S degrees of support are sorted and pc percent of the values are dropped on each side. The overall support $f_{\mathrm{c,j}}(\mathbf{x})$ is found as the simple average of the remaining degrees of support.

- Product (F is the product):

$$f_{\mathrm{PROD},j}(\mathbf{x}) = \prod_{i=1}^{K} f_{i,j}(\mathbf{x}). \tag{4.6}$$

4.3 The Linear Combination of Classifiers

Of the various combining rules proposed in the literature, linear combiners are the most frequently used (Kittler *et al.*, 1998), (Tumer & Ghosh, 1999), (Kuncheva, 2002), (Tax *et al.*, 2000). A linear combination of classifiers is the linear combination of their outputs(see fig. 4.2). A weight α_i is assigned to each classifier and the decision of the combiner is taken

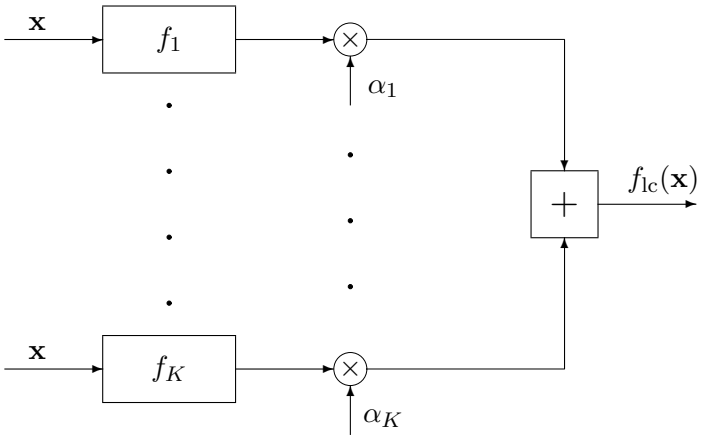


Figure 4.2: The linear combination rule for K classifiers.

according to:

$$f_{\text{lc}}(\mathbf{x}) = \sum_{i=1}^{K} \alpha_i f_i(\mathbf{x}). \tag{4.7}$$

In spite of their wide use and the success of linear combiners, many important issues related to their operation for pattern classification tasks have been developed. In Tumer & Ghosh (1996) an analysis of the decision boundaries for the linear combination of classifiers is proposed while Ueda (2000) describes a way to estimate the optimal weight for the combination of neural networks. A theoretical analysis has been proposed in Fumera & Roli (2005) where simple average (SA) and weighted average (WA) methods are studied. The former is a non trainable method discussed in the previous section while in the latter the weights are estimated on a training set proportionally to the performance of the base classifiers:

$$f_{\text{WA}}(\mathbf{x}) = \sum_{i=1}^{K} w_i f_i(\mathbf{x}). \tag{4.8}$$

In its simpler form with one non negative weight per classifier the evaluation of the weight is based on the estimated error rate of each base classifier:

$$w_i = \frac{Acc_i}{\sum_{i=1}^{K} Acc_i}. \tag{4.9}$$

Theoretically speaking, WA is always able to outperform SA but this is not guaranteed in practice where weights must be estimated from training data. In real applications, the

theoretical superiority of WA can be rapidly negated by weight estimations from small and noisy data sets to the extent that WA can actually perform worse than SA (Verikas *et al.*, 1999). Hence, Fumera & Roli (2005) conclude that it is not possible to show any clear experimental superiority over SA, in particular, for the simplest implementation of WA described before.

In literature the majority of the methods look at the estimation of the weight to maximize the accuracy of the combiner, while we are interested in AUC maximization. In literature, this topic has not received great attention: an approach is proposed in Su & Liu (1993) where the information carried by multiple classifiers is used for maximizing the TPR uniformly over the entire FPR range under the multivariate normal distribution model with proportional covariance matrices and, under these conditions, an estimate of the AUC of the combination is obtained. This work has been extended in Liu *et al.* (2005), where an alternative linear combination with higher TPR over a range of low FPR is derived. Another parametric approach based on the binormal model is proposed in Marrocco *et al.* (2005a). In this paper a method to estimate the ROC curve of the linear combination of two dichotomizers given the ROC curves of the single classifiers is derived. This represents a useful result to have an immediate preview of the performance of the system obtained by applying the combination without evaluating the outputs on the samples of the data set.

4.4 Linear Combination of Two Dichotomizers via AUC

The purpose of the method we are going to introduce is to construct a linear combination of dichotomizers aimed at maximizing the AUC of the resulting classification system. We focus first on the combination of two dichotomizers and then in the next sections we extend the method to $K > 2$ dichotomizers. On this topic some preliminaries have been proposed in Marrocco *et al.* (2005b), Marrocco *et al.* (2006b) and Marrocco *et al.* (2006a)

Let X be the set of samples as defined in sec. 1.1 and let us indicate the outputs of two dichotomizers f_1 and f_2 on positive and negative samples as:

$$x_i^1 = f_1(\mathbf{p}_i), \qquad y_j^1 = f_1(\mathbf{n}_j),$$
$$x_i^2 = f_2(\mathbf{p}_i), \qquad y_j^2 = f_2(\mathbf{n}_j).$$

The AUCs for the two dichotomizers evaluated according to the WMW statistic in eq.

(2.23) are:

$$AUC_1 = \frac{\sum_{i=1}^{m_P}\sum_{j=1}^{m_N} I(x_i^1, y_j^1)}{m_P m_N} \qquad AUC_2 = \frac{\sum_{i=1}^{m_P}\sum_{j=1}^{m_N} I(x_i^2, y_j^2)}{m_P m_N}. \tag{4.10}$$

Let us now consider a linear combination of f_1 and f_2. Without any loss of generality[1], the resulting classifier can be represented by:

$$f_{\text{lc}}(\mathbf{x}) = f_1(\mathbf{x}) + \alpha f_2(\mathbf{x}), \tag{4.11}$$

where α is the relative weight of f_2 with respect to f_1. The outputs of f_{lc} to $\mathbf{p}_i$ and $\mathbf{n}_j$ will be consequently:

$$\begin{aligned} \xi_i &= f_{\text{lc}}(\mathbf{p}_i) = x_i^1 + \alpha x_i^2, \\ \eta_j &= f_{\text{lc}}(\mathbf{n}_j) = y_j^1 + \alpha y_j^2. \end{aligned} \tag{4.12}$$

According to the WMW statistic, the AUC of f_{lc} is given by:

$$AUC_{\text{lc}} = \frac{\sum_{i=1}^{m_P}\sum_{j=1}^{m_N} I(\xi_i, \eta_j)}{m_P m_N}, \tag{4.13}$$

and depends on the value of the weight α. Therefore, the optimal choice for the weight is the value maximizing AUC_{lc}:

$$\alpha_{\text{opt}} = \arg\max_{\alpha} AUC_{\text{lc}}(\alpha). \tag{4.14}$$

To this aim, let us analyze the term $I(\xi_i, \eta_j)$ and study how it depends on the values of $I(x_i^1, y_j^1)$ and $I(x_i^2, y_j^2)$; for the following analysis we consider a tie as an error and thus we group together the cases for which $I(a, b) = 0.5$ and $I(a, b) = 0$. With this assumption, we can distinguish three cases:

- $I(x_i^1, y_j^1) = 1$ and $I(x_i^2, y_j^2) = 1$: in this case both the dichotomizers rank correctly the two samples and $I(\xi_i, \eta_j) = 1$ whatever the value of α.
- $I(x_i^1, y_j^1) = 0$ and $I(x_i^2, y_j^2) = 0$: in this case neither dichotomizer ranks correctly the samples and thus $I(\xi_i, \eta_j) = 0$ whatever the value of α.

[1] In general, a linear combination of two classifier is given by $\alpha_1 f_1 + \alpha_2 f_2$. However, any decision rule based on the comparison with a threshold t is equivalent to the decision rule which compares the output of the classifier f_{lc} with the threshold t/α_1.

- $I(x_i^1, y_j^1) \operatorname{xor} I(x_i^2, y_j^2) = 1$: only one dichotomizer ranks correctly the samples while the other one is wrong. In this case the value of $I(\xi_i, \eta_j)$ depends on the weight α.

According to this result, the set of all the pairs on which AUC_{lc} is evaluated can be split in four subsets $X_{12}, X_{\bar{1}2}, X_{1\bar{2}}, X_{\bar{1}\bar{2}}$, which are defined as:

$$X_{12} = \left\{(\mathbf{p}_i, \mathbf{n}_j) | I(x_i^1, y_j^1) = 1 \text{ and } I(x_i^2, y_j^2) = 1\right\}, \tag{4.15a}$$

$$X_{\bar{1}2} = \left\{(\mathbf{p}_i, \mathbf{n}_j) | I(x_i^1, y_j^1) = 0 \text{ and } I(x_i^2, y_j^2) = 1\right\}, \tag{4.15b}$$

$$X_{1\bar{2}} = \left\{(\mathbf{p}_i, \mathbf{n}_j) | I(x_i^1, y_j^1) = 1 \text{ and } I(x_i^2, y_j^2) = 0\right\}, \tag{4.15c}$$

$$X_{\bar{1}\bar{2}} = \left\{(\mathbf{p}_i, \mathbf{n}_j) | I(x_i^1, y_j^1) = 0 \text{ and } I(x_i^2, y_j^2) = 0\right\}. \tag{4.15d}$$

As a consequence the expression for AUC_{lc} in eq. (4.13) can be written as:

$$\begin{aligned} AUC_{\text{lc}} = & \frac{1}{m_P m_N} \left(\sum_{(\mathbf{p}_i, \mathbf{n}_j) \in X_{\bar{1}\bar{2}}} I(\xi_i, \eta_j) + \sum_{(\mathbf{p}_i, \mathbf{n}_j) \in X_{12}} I(\xi_i, \eta_j) \right. \\ & \left. + \sum_{(\mathbf{p}_i, \mathbf{n}_j) \in X_{1\bar{2}} \cup X_{\bar{1}2}} I(\xi_i, \eta_j) \right) = \frac{0 + \operatorname{card}(X_{12}) + \nu(\alpha)}{m_P m_N}. \end{aligned} \tag{4.16}$$

In other words, while the pairs on which both dichotomizers are wrong do not contribute to AUC_{lc} and the pairs correctly ranked by both the dichotomizers give a contribution independent of the value of α, the dependence of AUC_{lc} on α is limited to the set of pairs on which the dichotomizers disagree. Therefore, the larger the set $X_{1\bar{2}} \cup X_{\bar{1}2}$ (i.e. the higher the disagreement between f_1 and f_2), the higher the value of AUC_{lc} which, in principle, can be obtained. Taking into account eqs. (4.14) and (4.16) can be restated as:

$$\alpha_{\text{opt}} = \arg\max_{\alpha} \nu(\alpha). \tag{4.17}$$

In order to find the value of α_{opt} let us make explicit the dependence of $I(\xi_i, \eta_j)$ on α. To this aim, recall that the indicator function is not null only if $\xi_i > \eta_j$, i.e. if:

$$\left(x_i^1 - y_j^1\right) + \alpha \left(x_i^2 - y_j^2\right) > 0. \tag{4.18}$$

To simplify the following calculations, let us call *Score Difference Ratio (SDR)* the quantity:

$$SDR = -\frac{x_i^1 - y_j^1}{x_i^2 - y_j^2}, \tag{4.19}$$

and denote it with $\Gamma_2^1(\mathbf{p}_i, \mathbf{n}_j)$; for pairs $(\mathbf{p}_i, \mathbf{n}_j)$ belonging to $X_{1\bar{2}}$ or $X_{\bar{1}2}$ this value is positive because in both cases the differences have opposite signs. The condition (4.18) leads to different constraints on depending on which of the two sets $X_{1\bar{2}}, X_{\bar{1}2}$, we consider. In particular we obtain:

$$\alpha < \Gamma_2^1(\mathbf{p}_i, \mathbf{n}_j) \text{ if } (\mathbf{p}_i, \mathbf{n}_j) \in X_{1\bar{2}}, \tag{4.20a}$$

$$\alpha > \Gamma_2^1(\mathbf{p}_i, \mathbf{n}_j) \text{ if } (\mathbf{p}_i, \mathbf{n}_j) \in X_{\bar{1}2}. \tag{4.20b}$$

If such conditions are verified for each pair $(\mathbf{p}_i, \mathbf{n}_j) \in X_{1\bar{2}} \cup X_{\bar{1}2}$, we would obtain the maximum value allowable for $\nu(\alpha)$, i.e. $\text{card}(X_{1\bar{2}}) + \text{card}(X_{\bar{1}2})$. In this case, there would exist an α_{opt} such that:

$$\max_{(\mathbf{p}_i, \mathbf{n}_j) \in X_{\bar{1}2}} \Gamma_2^1(\mathbf{p}_i, \mathbf{n}_j) \leq \alpha_{\text{opt}} \leq \min_{(\mathbf{p}_i, \mathbf{n}_j) \in X_{1\bar{2}}} \Gamma_2^1(\mathbf{p}_i, \mathbf{n}_j), \tag{4.21}$$

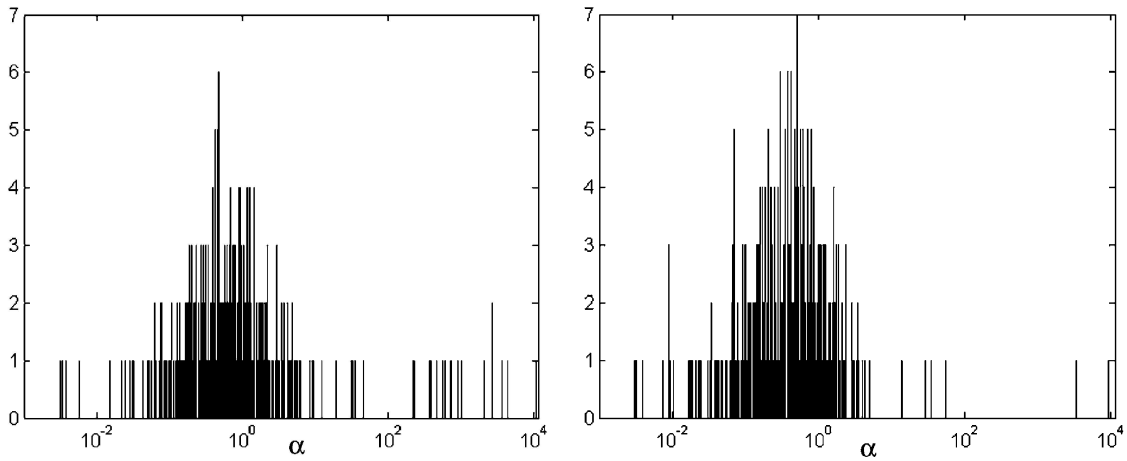
and the resulting AUC would be:

$$AUC_{\text{lc}} = \frac{\text{card}(X_{12}) + \text{card}(X_{1\bar{2}}) + \text{card}(X_{\bar{1}2})}{m_P m_N} = AUC_1 + AUC_2 - \frac{\text{card}(X_{12})}{m_P m_N},$$

where:

$$AUC_1 = \frac{\text{card}(X_{12}) + \text{card}(X_{1\bar{2}})}{m_P m_N}, \qquad AUC_2 = \frac{\text{card}(X_{12}) + \text{card}(X_{\bar{1}2})}{m_P m_N}.$$

However, the condition $\max_{(\mathbf{p}_i, \mathbf{n}_j) \in X_{\bar{1}2}} \Gamma_2^1(\mathbf{p}_i, \mathbf{n}_j) \leq \min_{(\mathbf{p}_i, \mathbf{n}_j) \in X_{1\bar{2}}} \Gamma_2^1(\mathbf{p}_i, \mathbf{n}_j)$ is verified only when the two dichotomizers are highly complementary. In particular, the term $\min_{(\mathbf{p}_i, \mathbf{n}_j) \in X_{1\bar{2}}} \Gamma_2^1(\mathbf{p}_i, \mathbf{n}_j)$ becomes high when the dichotomizer f_1 correctly ranks each pair $(\mathbf{p}_i, \mathbf{n}_j) \in X_{1\bar{2}}$ producing a high difference $|x_i^1 - y_j^1|$ between the outputs, while f_2, even though incorrectly ranking $(\mathbf{p}_i, \mathbf{n}_j)$, provides a low difference $|x_i^2 - y_j^2|$. This means that the errors made by f_2 can be recovered thanks to the good performance of f_1 on the same pairs. Conversely, a low value for the term $\max_{(\mathbf{p}_i, \mathbf{n}_j) \in X_{\bar{1}2}} \Gamma_2^1(\mathbf{p}_i, \mathbf{n}_j)$ is obtained when the dichotomizer f_2 correctly ranks each pair $(\mathbf{p}_i, \mathbf{n}_j) \in X_{\bar{1}2}$ with a high difference $|x_i^2 - y_j^2|$ between the outputs, while f_1 incorrectly ranks $(\mathbf{p}_i, \mathbf{n}_j)$, but with a low difference $|x_i^1 - y_j^1|$. In this case f_2 helps in recovering the erroneous rankings produced by f_1. When eq. (4.21) is verified, the value of α_{opt} allows eliminating all the errors made by both the dichotomizers, except for those on which f_1 and f_2 agree. Unfortunately, such condition is only rarely verified since the distributions of $\Gamma_2^1(\mathbf{p}_i, \mathbf{n}_j)$ evaluated on the two sets $X_{1\bar{2}}$ and $X_{\bar{1}2}$ are usually not separated.



(a) (b)

Figure 4.3: Example of the distributions of the ratio $\Gamma_{\bar{2}}^{1}(\mathbf{p}_i, \mathbf{n}_j)$ evaluated on the sets $X_{1\bar{2}}$ (a) and $X_{\bar{1}2}$ (b)

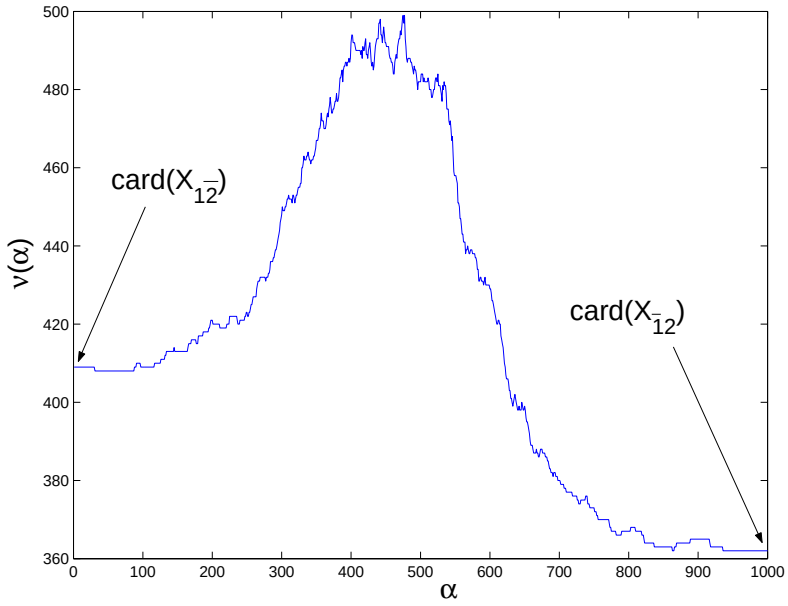


Figure 4.4: The trend of the function $\nu(\alpha) = F_{1\bar{2}}(\alpha) + F_{\bar{1}2}(\alpha)$ obtained by the two distributions shown in fig. 4.3. The points on which the combination reduces to one dichotomizer are shown.

As a consequence, α_{opt} has to be found by maximizing the number of the pairs satisfying eq. (4.20). To this aim, if we consider the cumulative functions:

$$F_{1\bar{2}} = \text{card}\left((\mathbf{p}_i, \mathbf{n}_j) \in X_{1\bar{2}} \middle| \Gamma_2^1(\mathbf{p}_i, \mathbf{n}_j) > \alpha \right), \tag{4.22a}$$

$$F_{\bar{1}2} = \text{card}\left((\mathbf{p}_i, \mathbf{n}_j) \in X_{\bar{1}2} \middle| \Gamma_2^1(\mathbf{p}_i, \mathbf{n}_j) < \alpha \right), \tag{4.22b}$$

the function to be maximized can be defined as:

$$\nu(\alpha) = F_{1\bar{2}}(\alpha) + F_{\bar{1}2}(\alpha), \tag{4.23}$$

and the optimal value of α is given by:

$$\alpha_{\text{opt}} = \arg\max_{\alpha} \left(F_{1\bar{2}}(\alpha) + F_{\bar{1}2}(\alpha) \right), \tag{4.24}$$

that can be easily found by means of a linear search.

An example of real distributions of the ratio $\Gamma_2^1(\mathbf{p}_i, \mathbf{n}_j)$ evaluated on the two sets $X_{1\bar{2}}$ and $X_{\bar{1}2}$ is shown in fig. 4.3, while fig. 4.4 shows the relative function $\nu(\alpha)$ as in eq. 4.23. If we consider what happens at the bounds of the range of α, it is possible to observe that, if $\alpha \to 0$, $\nu(\alpha) \to \text{card}(X_{1\bar{2}})$ and the combination reduces to the only dichotomizer f_1, while when $\alpha \to +\infty$, $\nu(\alpha) \to \text{card}(X_{\bar{1}2})$ and the combination reduces to the only dichotomizer f_2. These extreme points are shown in fig. 4.4: in this case, since the maximum of $\nu(\alpha)$ is higher than both the bound values, we have $AUC_{\text{lc}}(\alpha_{\text{opt}}) > AUC_1$ and $AUC_{\text{lc}}(\alpha_{\text{opt}}) > AUC_2$, i.e. the linear combination performs better than each of the two dichotomizers. As a concluding remark, it is worth noting that the method cannot be applied when $\text{card}(X_{1\bar{2}}) = 0$ or $\text{card}(X_{\bar{1}2}) = 0$. However, in this case the combination is not profitable since it does not give better results than the single dichotomizer. In fact, if e.g. $\text{card}(X_{\bar{1}2}) = 0$, there are no pairs incorrectly ranked by f_1 which are correctly ranked by f_2 and thus the combination is useless since it cannot recover any error made by f_1.

4.5 The DROC Curve

α_{opt} should be found by means of a linear search maximizing the number of the pairs satisfying eq. (4.20). However, it is possible to define another more effective method to evaluate the optimal weight. First of all, in order to simplify the notation in the following analysis, let us disregard the dependence on the particular samples in the SDRs and denote

with $\delta_r^{\bar{1}2}$ with $r = 1, \ldots, \text{card}(X_{\bar{1}2})$ the SDR value of the r-th pair contained in $X_{\bar{1}2}$) and with $\delta_s^{1\bar{2}}$ with $s = 1, \ldots, \text{card}(X_{1\bar{2}})$ the SDR value of the s-th pair contained in $X_{1\bar{2}}$.

Now let us choose a value α for the weight of the linear combination; with such choice, in each set some pairs will be correctly ranked while others will not. Let us define the *Correctly Ranked Rate* on $X_{\bar{1}2}$, $CRR_{\bar{1}2}(\alpha)$ and the *Wrongly Ranked Rate* on $X_{\bar{1}2}$, $WRR_{\bar{1}2}(\alpha)$ as:

$$CRR_{\bar{1}2}(\alpha) = \frac{\text{card}\left(\left\{\delta_r^{\bar{1}2} < \alpha, r = 1 \ldots \text{card}(X_{\bar{1}2})\right\}\right)}{\text{card}(X_{\bar{1}2})}, \tag{4.25a}$$

$$WRR_{\bar{1}2}(\alpha) = \frac{\text{card}\left(\left\{\delta_r^{\bar{1}2} \geq \alpha, r = 1 \ldots \text{card}(X_{\bar{1}2})\right\}\right)}{\text{card}(X_{\bar{1}2})}. \tag{4.25b}$$

Both indices are in the range $[0, 1]$ and are not independent since $CRR_{\bar{1}2}(\alpha) + WRR_{\bar{1}2}(\alpha) = 1$. In a similar way, it is possible to evaluate the same indices on the set $X_{1\bar{2}}$:

$$CRR_{1\bar{2}}(\alpha) = \frac{\text{card}\left(\left\{\delta_s^{1\bar{2}} > \alpha, s = 1 \ldots \text{card}(X_{1\bar{2}})\right\}\right)}{\text{card}(X_{1\bar{2}})}, \tag{4.26a}$$

$$WRR_{1\bar{2}}(\alpha) = \frac{\text{card}\left(\left\{\delta_s^{1\bar{2}} \leq \alpha, s = 1 \ldots \text{card}(X_{1\bar{2}})\right\}\right)}{\text{card}(X_{1\bar{2}})}. \tag{4.26b}$$

Since for each set the indices are dependent on each other, it is sufficient to know only one index for each set in order to have the corresponding value for $\nu(\alpha)$. A possible choice could be to consider only $WRR_{\bar{1}2}(\alpha)$ and $CRR_{1\bar{2}}(\alpha)$ and to represent them as coordinates in a plane: in this way, the values produced by a particular α individuate a point in the unit square whose corners are the points $(0, 0), (1, 0), (0, 1)$ and $(1, 1)$ (see fig. 4.5).

When the value of the weight α varies between 0 and $+\infty$ the quantities $WRR_{\bar{1}2}(\alpha)$ and $CRR_{1\bar{2}}(\alpha)$ vary accordingly, thus drawing a curve running from $(1, 1)$ to $(0, 0)$. We call it *Difference Ratio Operating Characteristic* (*DROC*) curve.

Similarly to the ROC curve, the DROC curve has some noteworthy features:

- the extreme points of the curve represent the extreme configurations for the combination of f_1 and f_2: the point $(0, 0)$ is reached when $\alpha \to +\infty$ and the combination reduces to the only dichotomizer f_2, while $(1, 1)$ is obtained for $\alpha = 0$ where the combination reduces to the only dichotomizer f_1;

- if the distributions are perfectly separated, the curve passes through the point $(0, 1)$: in this case, there exists a value for α which verifies the condition in eq. (4.20).

Informally speaking, the closer the curve to the point (0,0), the more separated the distributions;

- if the distributions totally overlap, the curve turns into a diagonal line from the bottom left corner to the upper right corner;

- the DROC curve is not defined if $\text{card}(X_{1\bar{2}}) = 0$ or $\text{card}(X_{\bar{1}2}) = 0$. However, in this case the combination is not profitable since it does not give better results than the single dichotomizer. In fact, if e.g. $\text{card}(X_{\bar{1}2}) = 0$, there are no pairs incorrectly ranked by f_1 which are correctly ranked by f_2 and thus the combination is useless since it cannot recover any error made by f_1.

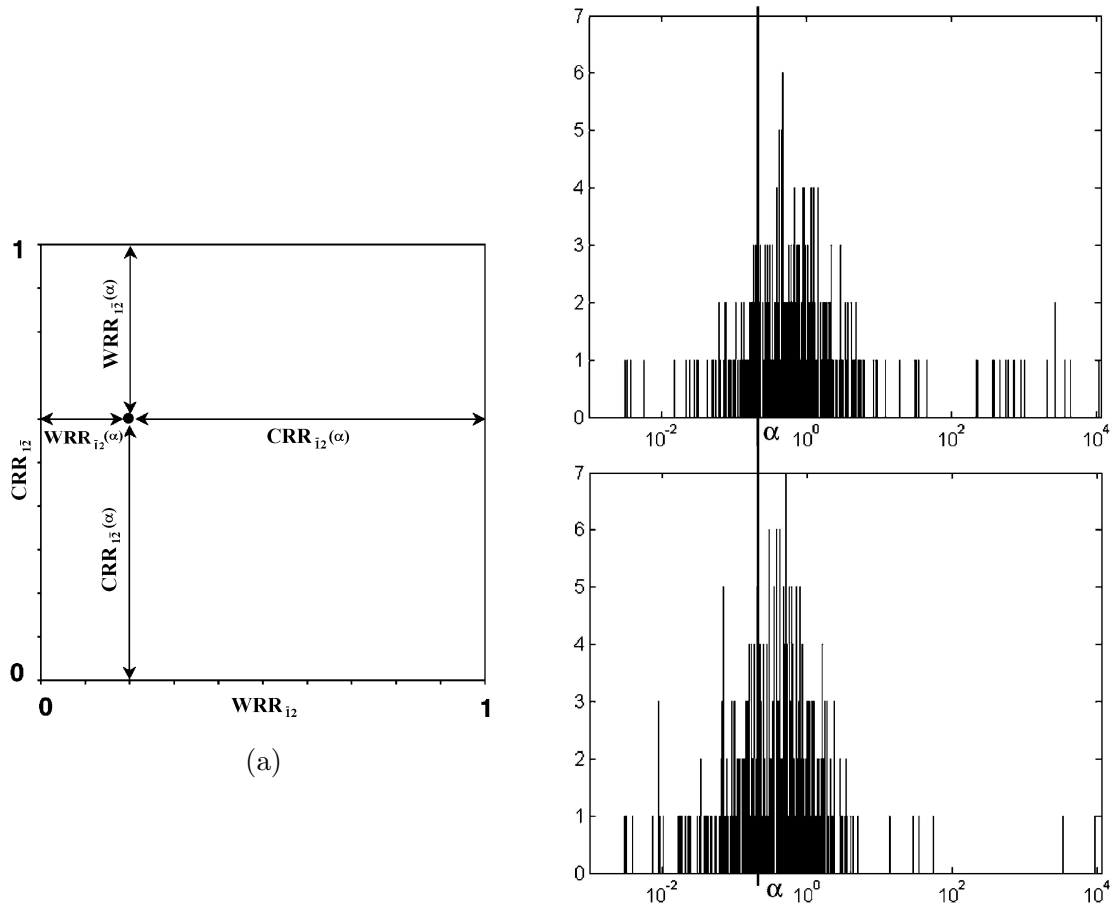
4.5.1 Generating the DROC Curve

The plot of the DROC curve can be drawn in many ways. For example, we could obtain T points of the DROC curve of a pair of dichotomizers by imposing T thresholds ranging from the smallest to the largest values obtained for the SDRs and evaluating the resulting CRR and WRR for each of the T thresholds. However, such kind of method is quite unsatisfactory because it is strictly dependent on the choice of T and when the discretization is too coarse (the T threshold values considered are few compared with the number of different values the SDRs assume) the approximation can be poor and misrepresent the actual plot. For this reason, we have chosen to generate the plots for the DROC curves by employing all the values exhibited by the SDRs as possible decision thresholds, thus obtaining a faithful plot. To this aim we have defined an efficient algorithm (see algorithm 4.1), derived by the one described in sec. 2.3.1 for plotting the ROC curve, with complexity $O(nlogn)$ in the number of the SDR values.

4.5.2 Finding α_{opt} by means of the DROC Curve

The DROC is not only a tool for visualizing how the difference ratio distributions are separated, but it can also be profitably used to select the optimal value of the weight α_{opt}. To this aim, let us point out that the quantity to be maximized in eq. (4.17) can be written as:

$$\nu(\alpha) = \text{card}(X_{1\bar{2}}) CRR_{1\bar{2}}(\alpha) + \text{card}(X_{\bar{1}2}) \left(1 - WRR_{\bar{1}2}(\alpha)\right). \tag{4.27}$$



(a)

(b)

Figure 4.5: The $(WRR_{\bar{1}2}, CRR_{1\bar{2}})$ plane (a) with the four indices $CRR_{\bar{1}2}(\alpha)$, $CRR_{1\bar{2}}(\alpha)$, $WRR_{\bar{1}2}(\alpha)$ and $WRR_{1\bar{2}}(\alpha)$ corresponding to the value of α shown on the histogram distributions in (b).

Algorithm 4.1 Efficient method to generate a DROC curve

Input: $m_P > 0$ and $m_N > 0$, the number of positive and negative samples; (x_i^1, y_j^1) and (x_i^2, y_j^2), the output of the two classifiers on the i-th positive sample and the j-th negative sample $\forall i = 1 \dots m_P, \forall j = 1 \dots m_N$.

Output: D, a list of DROC points.

```
build X_{1̄2} and X_{1 2̄}
δ^{1̄2} ← δ_r^{1̄2} with r = 1, ..., card(X_{1̄2})
δ^{1 2̄} ← δ_s^{1 2̄} with s = 1, ..., card(X_{1 2̄})
put the two distributions δ^{1̄2} and δ^{1 2̄} in the same vector τ
τ ← τ sorted by decreasing values
D ← [ ]
CR ← WR ← 0
τ_prev ← −∞
for h = 1 to length(τ) do
  if τ(h) ≠ τ_prev then
    put ( CR / card(X_{1 2̄}), WR / card(X_{1̄2}) ) onto D
    τ_prev ← τ(h)
  end if
  if τ(h) ∈ X_{1 2̄} then
    CR ← CR + 1
  else
    WR ← WR + 1
  end if
end for
put ( CR / card(X_{1 2̄}), WR / card(X_{1̄2}) ) onto D /*this corresponds to the point (1,1)*/
```

Two different points $\left(WRR'_{\bar{1}2}, CRR'_{1\bar{2}}\right)$ and $\left(WRR''_{\bar{1}2}, CRR''_{1\bar{2}}\right)$ give the same $\nu(\alpha)$ if:

$$\begin{aligned}\text{card}(X_{1\bar{2}})CRR'_{1\bar{2}} + \text{card}(X_{\bar{1}2})\left(1 - WRR'_{\bar{1}2}\right) \\ = \text{card}(X_{1\bar{2}})CRR''_{1\bar{2}} + \text{card}(X_{\bar{1}2})\left(1 - WRR''_{\bar{1}2}\right),\end{aligned}$$

that is if:

$$\frac{CRR'_{1\bar{2}} - CRR''_{1\bar{2}}}{WRR'_{\bar{1}2} - WRR''_{\bar{1}2}} = \frac{\text{card}(X_{\bar{1}2})}{\text{card}(X_{1\bar{2}})} = m. \tag{4.28}$$

This is equivalent to say that the two points lie on a line with slope m; obviously, all the points on the same line provides the same $\nu(\alpha)$ and thus it represents an *iso-performance* line. In other words, the level curves of the linear function in eq. (4.27) on the DROC plane are lines with slope m; the lines with the highest values for $\nu(\alpha)$ are

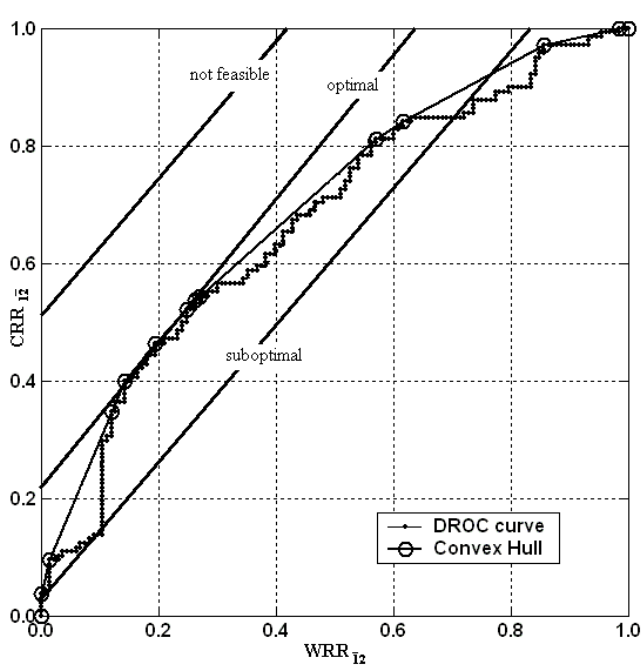


Figure 4.6: The DROC curve relative to the distributions in fig. 4.3 shown together with its convex hull and some iso-performance lines with slope m defined in eq. (4.28).

those with largest y-intercept. If the DROC curve is defined by means of a finite number of experimental points connected with straight lines, the optimal operating point is the one where a line with slope m touches the DROC curve. In particular, such point lies on the DROC *Convex Hull*, i.e. the smallest convex set containing the points of the DROC curve. This can be visually understood by looking at fig. 4.6 where an empirical DROC curve is shown together with its convex hull and some level lines with the same slope m and decreasing value for $\nu(\alpha)$. The line touching the DROC curve determines the optimal weight: in fact the line above, even though exhibits the highest value for $\nu(\alpha)$, does not determine any feasible point, while the line below intersects the DROC curve in at least two points, but at lowest values for $\nu(\alpha)$. Once the optimal point has been found, the optimal weight α_{opt} is consequently determined by reading the value of α related to that point.

This property is formally stated and proved in the following lemma 4.5.1. It is worth noting that a similar result holds for the convex hull of the ROC curve when the best classifier that minimizes the expected classification cost is searched. Such result is demonstrated in Provost & Fawcett (2001) and we strictly follow the proof provided in that paper formalizing it fir the DROC curve.

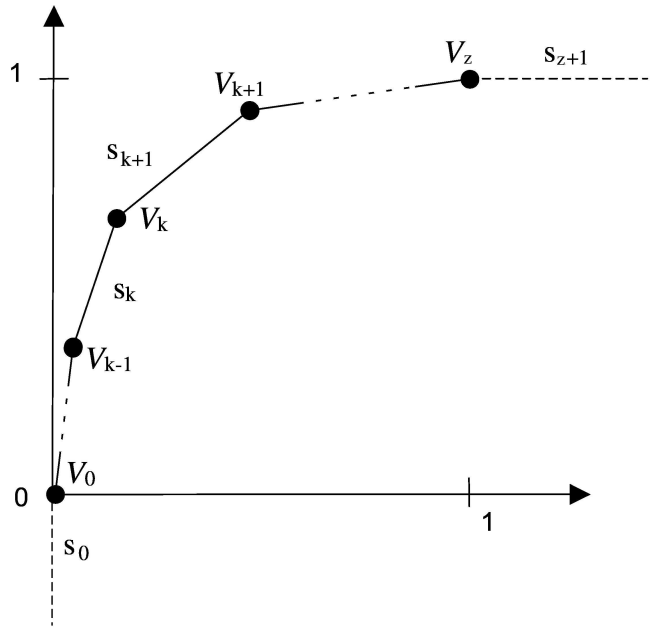


Figure 4.7: The notation used in the search of the optimal weight.

Lemma 4.5.1. *For a given DROC curve and a given ratio* $\text{card}(X_{\bar{1}2})/\,\text{card}(X_{1\bar{2}})$*, there exists a point on the DROC convex hull with maximum* $\nu(\alpha)$*. Thus, the value of* α *pertaining to that point is the optimum weight for the linear combination.*

Proof. Assume that for a given ratio $\text{card}(X_{\bar{1}2})/\,\text{card}(X_{1\bar{2}})$, the point M with the maximum $\nu(\alpha)$ is not on the DROC convex hull. Nevertheless, it must belong to the DROC curve otherwise it would not be a feasible point for the combination. M will be either above the convex hull or below the convex hull. If M is above, then the DROC convex hull does not enclose all the points of the DROC, but that is absurdum. If M is below the convex hull, then the line with slope $\text{card}(X_{\bar{1}2})/\,\text{card}(X_{1\bar{2}})$ containing M will intersect the DROC convex hull at least in one point M'. If M' is a vertex of the convex hull, then it will exhibit the same $\nu(\alpha)$ of M, which contradicts our initial assumption that the maximum $\nu(\alpha)$ is not provided by a point on the convex hull. If M' is not a vertex, then it will be lie on an edge of the convex hull with non null slope; in this case one of the vertices, let V denote it, of such edge will be also on a line parallel to the line $\overline{MM'}$, but with a higher y intercept. This means that the point V provides a larger value for $\nu(\alpha)$ than M, but this contradicts our initial assumption that the point with the maximum $\nu(\alpha)$ is not on the DROC convex hull. So we can conclude that the lemma must be true. □

Lemma 4.5.1 allows us to sensibly simplify the search for the optimal α: in fact, if an

estimate of the DROC curve on which the optimal weight has to be searched is available, we can limit the search only to the points coinciding with vertices of the DROC convex hull. On the contrary, a direct search on the points of the DROC would have a higher computational cost. Hence, the optimal thresholds can be found by means of a simple search on the slopes of the edges of the DROC convex hull. To this aim, let us call $V_0, V_1, \ldots, V_z$ the vertices of the DROC convex hull, with $V_0 \equiv (0,0)$ and $V_z \equiv (1,1)$ and let s_k be the slope of the edge joining the vertices V_{k-1} and V_k (see fig. 4.7); moreover, let us assume that $s_0 = +\infty$ and $s_{z+1} = 0$. For a given slope m, the vertex providing the searched weight is the vertex V_k such that $s_k > m > s_{k+1}$. However, it can exist an edge $\overline{V_{k-1}V_k}$ with slope $s_k = m$; in this case, the level curve and the edge are coincident and thus either of the vertices V_{k-1} and V_k can be chosen; the only difference is that the left vertex will have lower $CRR_{1\bar{2}}$ and $WRR_{\bar{1}2}$, while the right vertex will have higher $CRR_{\bar{1}2}$ and $WRR_{1\bar{2}}$, thus one can refer to the requirements of the application at hand to make the most appropriate choice.

4.5.3 The Area under the DROC Curve

As we have seen in sec. 4.5, the shape of the DROC curve can give some information about the degree of separation between the distributions of the SDRs. Informally, we can add that the *Area Under the DROC Curve (AUDC)* could be assumed as a concise index to measure such degree of separation: it ranges from 0.5 for distributions totally overlapped to 1.0 if the distributions are totally separated. In order to establish a rigorous relation between the AUDC and the degree of separation between the distributions, let us consider the two sets of the pairs coming from $X_{1\bar{2}} \times X_{\bar{1}2}$ which are correctly ranked by one dichotomizer but not by the other one, i.e.:

$$C_{1\bar{2}} = \left\{ (\rho, \sigma) \in X_{1\bar{2}} \times X_{\bar{1}2} | \delta^{1\bar{2}}(\rho) < \delta^{1\bar{2}}(\sigma) \text{ and } \delta^{\bar{1}2}(\rho) > \delta^{\bar{1}2}(\sigma) \right\}, \tag{4.29a}$$

$$C_{\bar{1}2} = \left\{ (\rho, \sigma) \in X_{1\bar{2}} \times X_{\bar{1}2} | \delta^{1\bar{2}}(\rho) > \delta^{1\bar{2}}(\sigma) \text{ and } \delta^{\bar{1}2}(\rho) < \delta^{\bar{1}2}(\sigma) \right\}. \tag{4.29b}$$

Moreover, let us define the probability $P_{1\bar{2}>\bar{1}2}$ that two pairs (ρ', σ') and (ρ'', σ''), randomly extracted from $C_{1\bar{2}}$ and $C_{\bar{1}2}$ respectively, have their SDRs ranked in decreasing order as:

$$P_{1\bar{2}>\bar{1}2} = Prob\left(\Gamma_2^1(\rho', \sigma') > \Gamma_2^1(\rho'', \sigma''), (\rho', \sigma') \in C_{1\bar{2}}, (\rho'', \sigma'') \in C_{\bar{1}2}\right). \tag{4.30}$$

The following theorem holds:

Theorem 4.5.2. *The area under the DROC curve (AUDC) evaluated on the sets* $X_{1\bar{2}}$

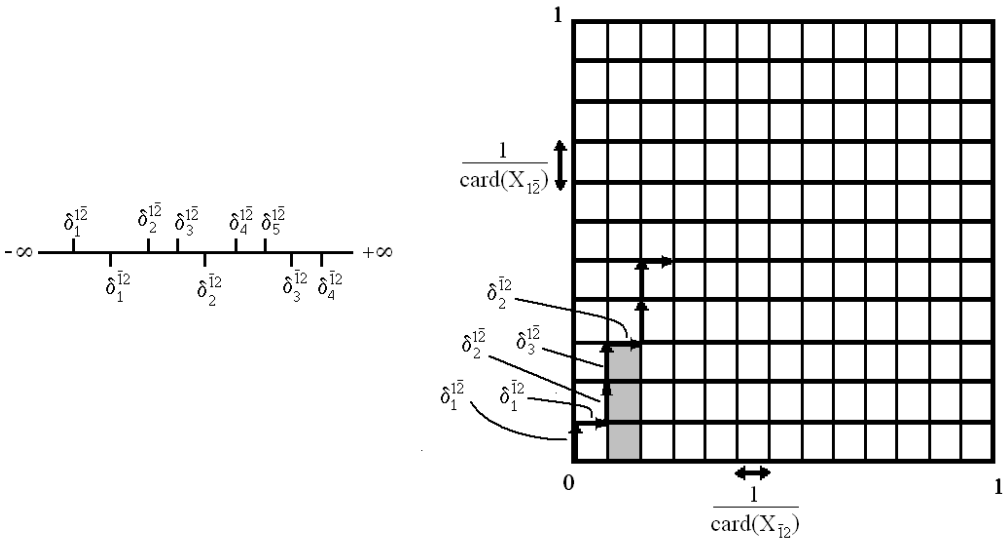


Figure 4.8: The notation used in the proof of the theorem 4.5.2 to evaluate the AUDC.

and $X_{\bar{1}2}$ is an unbiased estimate of the probability $P_{1\bar{2}>\bar{1}2}$ defined as in eq. (4.30).

Proof. With reference to the algorithm 4.1 in sec. 4.5.1, let us analyze how the DROC curve is built. After the SDRs are sorted in decreasing order, the sequence is traversed from the highest value to the lowest one: for each SDR belonging to $X_{\bar{1}2}$, the curve moves on the DROC plane $1/\operatorname{card}(X_{\bar{1}2})$ to the right, while it moves $1/\operatorname{card}(X_{1\bar{2}})$ upward for each SDR coming from $X_{1\bar{2}}$. From fig. 4.8 it is possible to see that the generic $\delta_k^{\bar{1}2}$ coming from $X_{\bar{1}2}$ contributes to the AUDC with a rectangle having area

$$\hat{A} = \frac{1}{\operatorname{card}(X_{\bar{1}2})} \frac{1}{\operatorname{card}(X_{1\bar{2}})} \mathcal{N}_k, \tag{4.31}$$

where $\mathcal{N}_k$ is the number of SDRs $\delta_h^{1\bar{2}}$ greater than $\delta_k^{\bar{1}2}$. Therefore, we have:

$$AUDC = \frac{1}{\operatorname{card}(X_{\bar{1}2}) \operatorname{card}(X_{1\bar{2}})} \sum_{k=1}^{\operatorname{card}(X_{\bar{1}2})} \mathcal{N}_k. \tag{4.32}$$

Since $\mathcal{N}_k$ can be defined as:

$$\mathcal{N}_k = \sum_{h=1}^{\operatorname{card}(X_{1\bar{2}})} I\left(\delta_h^{1\bar{2}}, \delta_k^{\bar{1}2}\right), \tag{4.33}$$

we finally have:

$$AUDC = W_{X_{1\bar{2}}>X_{\bar{1}2}} = \frac{1}{\text{card}(X_{\bar{1}2})\,\text{card}(X_{1\bar{2}})} \sum_{k=1}^{\text{card}(X_{\bar{1}2})} \sum_{h=1}^{\text{card}(X_{1\bar{2}})} I\left(\delta_h^{1\bar{2}}, \delta_k^{\bar{1}2}\right), \tag{4.34}$$

where $W_{X_{1\bar{2}}>X_{\bar{1}2}}$ is the WMW statistic applied on the SDR distributions over $X_{1\bar{2}}$ and $X_{\bar{1}2}$. Let us now calculate the expectation of $W_{X_{1\bar{2}}>X_{\bar{1}2}}$:

$$E\left[W_{X_{1\bar{2}}>X_{\bar{1}2}}\right] = \frac{1}{\text{card}(X_{\bar{1}2})\,\text{card}(X_{1\bar{2}})} \sum_{k=1}^{\text{card}(X_{\bar{1}2})} \sum_{h=1}^{\text{card}(X_{1\bar{2}})} E\left[I\left(\delta_h^{1\bar{2}}, \delta_k^{\bar{1}2}\right)\right]. \tag{4.35}$$

If we recall the definition of the indicator function we obtain:

$$E\left[I\left(\delta_h^{1\bar{2}}, \delta_k^{\bar{1}2}\right)\right] = 1 \cdot P_{1\bar{2}>\bar{1}2} + 0 \cdot (1 - P_{1\bar{2}>\bar{1}2}) = P_{1\bar{2}>\bar{1}2}, \tag{4.36}$$

and thus:

$$E\left[W_{X_{1\bar{2}}>X_{\bar{1}2}}\right] = \frac{1}{\text{card}(X_{\bar{1}2})\,\text{card}(X_{1\bar{2}})} \sum_{k=1}^{\text{card}(X_{\bar{1}2})} \sum_{h=1}^{\text{card}(X_{1\bar{2}})} P_{1\bar{2}>\bar{1}2} = P_{1\bar{2}>\bar{1}2}. \tag{4.37}$$

Therefore, $W_{X_{1\bar{2}}>X_{\bar{1}2}}$ is an unbiased estimator of the probability $P_{1\bar{2}>\bar{1}2}$. Since $AUDC = W_{X_{1\bar{2}}>X_{\bar{1}2}}$, the theorem is proved. □

4.6 Measuring the Ranking Diversity

A recently emerging issue in classifier combination is to evaluate to which extent two classifiers are different. In Kuncheva (2005) it is claimed that the diversity among the classifiers to be combined is a "vital requirement for the success of the ensemble" since it is possible to improve the performance of the base classifiers only if they make errors on different objects. However, the relation between diversity and quality in classifier ensembles is very ambiguous for two main reasons: first, the diversity can be measured in many ways (Kuncheva & Whitaker, 2003) and no one of the possible different measures seems to be definitely better than the others[2]; the second reason is the lack of a definitive connection between the measures and the improvement of the accuracy which makes it difficult to decide how to employ the diversity for the design of the classifier ensemble.

[2]In conclusions of Kuncheva & Whitaker (2003) the authors suggest to use a particular measure of diversity among the ten described mainly for three reasons: ease of interpretation, formally demonstrable relationship between the value of the measure and the limits of majority voting combination, ease of computation.

Moreover, another issue to be taken into account is that the diversity measures based on accuracy are actually related to the particular operating point chosen for the classifier. As an example, consider a classifier providing an estimate of the a posteriori probability that an object belongs to a certain class. Depending upon the particular application, the different kinds of errors that the classifier could incur yield different costs and the decision about the class is taken by comparing the a posteriori probability with a threshold related to such costs. For this reason, whichever the assumed diversity index, its value will depend on the particular threshold adopted. When the costs are not known (or they are dynamically varying) the operating point is not univocally determined and thus the value of the diversity measure is consequently indefinite.

In this context, we try to extend the concept of diversity measure to a ranker, i.e. to a learning algorithm which is able to provide for each object a numerical value estimating the confidence degree about the membership to a particular class. In fact, in our case it is not possible to use the accuracy-based diversity of the output of two classifiers since we are interested at the diversity of the score differences between outputs of samples belonging to positive and negative classes for the maximization of the AUC.

This means that the diversity indices proposed in literature are not directly applicable. However, we can still define a diversity index between the dichotomizers on the basis of the results obtained in sec. 4.4. In particular, from eq. (4.13) we can extract of the diversity between the two dichotomizers, where the diversity we are looking at is not related to the classification capability but to the ranking capability of the dichotomizer. For this reason, we denote the maximum of $\nu(\alpha)$ as a ranking disagreement (RD) index between two generic classifiers f_h and f_k:

$$RD_{hk} = \frac{\text{card}(X_{\bar{h}k}) + \text{card}(X_{h\bar{k}})}{m_P m_N}. \tag{4.38}$$

Such index is 0 when the two dichotomizers rank in the same way all the pairs (p_i, n_j) while it is 1 when all the pairs are ranked in a different way by the dichotomizers. The expression of the ranking disagreement index is similar to the disagreement measure for classifiers proposed by Skalak (1996) (see also Kuncheva & Whitaker (2003)), but, in that case, the index refers to the number of samples incorrectly classified by f_h and correctly classified by f_k (and vice versa). In other words, the RD_{hk} index can be thought as the counterpart of the Skalak index for the rankers.

The ranking disagreement index has some noteworthy features. Since we have found that RD_{hk} is the upper bound for the improvement of the AUC, the ranking disagreement index is directly related to the performance improvement in terms of AUC attainable

from the combination of the dichotomizers, at least for the linear combination scheme. This represents an important difference point with respect to the accuracy based diversity indices, for which there is not any similar clear relationship. Moreover, the RD index is independent of any operating point since it looks at the dichotomizer as a ranker.

However, this is a quite loose relation since the actual improvement depends on how are separated the two distributions of the difference ratio evaluated on $X_{h\bar{k}}$ and $X_{\bar{h}k}$: if they are completely separated, the total improvement is equivalent to the RD index while it is null if the distributions totally overlap. In order to have a tighter bound of the improvement of the AUC for the linear combination of dichotomizers, let us take into account that the degree of separation between the distributions can be provided by the probability $P_{h\bar{k}>\bar{h}k}$ or, better, by the area under the relative DROC, $AUDC_{hk}$. In this way, a more accurate estimate of the attainable improvement $R\Delta_{hk}$ will be given by:

$$R\Delta_{hk} = RD_{hk} AUDC_{hk}. \tag{4.39}$$

4.7 A Greedy Approach for the Combination of Several Classifiers

Once we have introduced the DROC curve and a ranking diversity measure, we can now extend to K classifiers the approach proposed in section 4.4.

To this aim, let us now consider the linear combination of K classifiers:

$$f_{\text{lc}} = \alpha_1 f_1 + \alpha_2 f_2 + \ldots + \alpha_K f_K = \sum_{i=1}^{K} \alpha_i f_i. \tag{4.40}$$

In order to find the optimal weight vector:

$$\boldsymbol{\alpha}_{\text{opt}} = (\alpha_1 \ldots \alpha_K), \tag{4.41}$$

that maximizes the AUC associated with f_{lc}, the algorithm proposed in sec. 4.4 cannot be generalized to $K > 2$ dichotomizers in such a way that the maximization of the resulting AUC is computationally feasible.

Therefore, to avoid an high computational cost we adopt a suboptimal algorithm that approximates the solution using a greedy approach. Rather than considering every possible combination in its entirety, we iteratively find the optimal weight of the linear combination of two dichotomizers so as to evaluate all the combination weights in $K-1$ steps, producing in each step the largest immediate gain, i.e. the largest AUC.

Table 4.1: An example of diversity tables for the combination of four classifiers in the first step (a) and the second step (b) of the greedy approach

	f_1	f_2	f_3	f_4
f_1	0.0	0.3	0.2	0.5
f_2	0.3	0.0	0.6	0.2
f_3	0.2	0.6	0.0	0.4
f_4	0.5	0.2	0.4	0.0

(a)

	f_{lc_1}	f_1	f_4
f_{lc_1}	0.0	0.2	0.3
f_1	0.2	0.0	0.4
f_4	0.3	0.4	0.0

(b)

In this context an important role is played by the order of combination, i.e. to correctly choose which pair of classifiers should be combined in each step. From the previous sections we know that the greater the diversity among the classifiers to be combined the greater the improvement to the performance of the base classifiers which could be gained (see eq. (4.21) and the following discussion in sec. 4.4). Therefore, we choose to combine in each step the pair that exhibits the maximum disagreement coefficient in terms of ranking.

Once the weight has been computed, the two dichotomizers are replaced by their combination and thus the dichotomizers to be combined decrease from K to $K-1$. At this point, we have to evaluate the disagreement between the new classifier and the other classifiers. It is worth noting that, for this step, it is not necessary to compute the output of the combined classifier, since its score differences (SD) can be directly evaluated as the weighted sum (with the same weight estimated for the combination) of the score differences of the combined classifiers. To this aim, let us consider the r-th step of the algorithm where the pair (h,k) has been combined, we can evaluate the SD of the combined classifier as:

$$\begin{aligned} SD_{\mathrm{lc}_r} &= x_i^{\mathrm{lc}_r} - y_j^{\mathrm{lc}_r} = \left(x_i^h + \alpha_{\mathrm{lc}_r} x_i^k\right) - \left(y_j^h + \alpha_{\mathrm{lc}_r} y_j^k\right) \\ &= \left(x_i^h - y_j^h\right) + \alpha_{\mathrm{lc}_r}\left(x_j^k - y_j^k\right) = SD_h + \alpha_{\mathrm{lc}_r} SD_k. \end{aligned} \tag{4.42}$$

These steps are repeated until all the dichotomizers have been combined: in each iteration it is chosen the pair of dichotomizers with the highest disagreement coefficient. It is worth noting that applying the greedy approach one of the weights of the vector is always equal to 1 (say α_p) because step by step we evaluate only one weight. This means that the final weight vector is defined up to a normalizing constant. However, since the decision rule is based on a comparison with a threshold τ, this is equivalent to a decision rule where the comparison is made with a threshold τ/α_p (see footnote 1 in sec. 4.4).

As an example let us consider the combination of four classifiers f_1, f_2, f_3, f_4 and let us

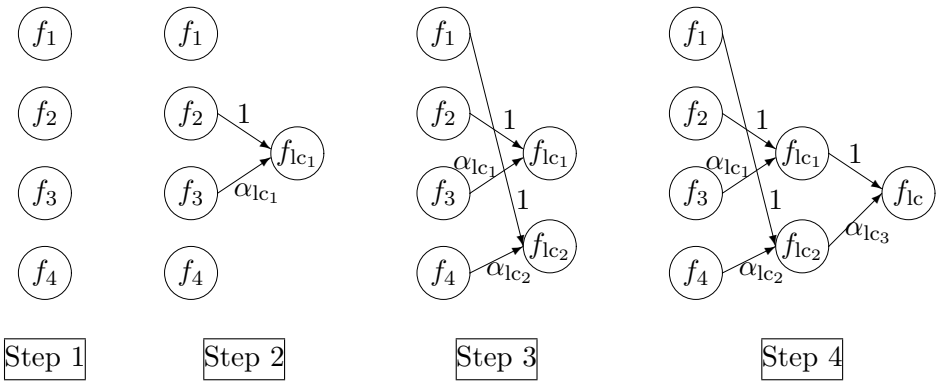


Figure 4.9: The construction of the combination tree along the various steps of the algorithm according the diversity tables in table 4.1.

consider the diversity table reported in table 4.1.(a). At the first step we combine f_2 and f_3 (since they have the highest diversity index) and obtain a classifier $f_{\mathrm{lc}_1} = f_2 + \alpha_{\mathrm{lc}_1} f_3$. In the second step we have to consider the updated diversity table reported in table 4.1.(b) and we obtain a new classifier $f_{\mathrm{lc}_2} = f_1 + \alpha_{\mathrm{lc}_2} f_4$. Finally, we combine f_{lc_1} and f_{lc_2} obtaining the final classifier:

$$f_{\mathrm{lc}} = f_{\mathrm{lc}_1} + \alpha_{\mathrm{lc}_3} f_{\mathrm{lc}_2} = f_2 + \alpha_{\mathrm{lc}_1} f_3 + \alpha_{\mathrm{lc}_3} (f_1 + \alpha_{\mathrm{lc}_2} f_4) = \alpha_{\mathrm{lc}_3} f_1 + f_2 + \alpha_{\mathrm{lc}_1} f_3 + \alpha_{\mathrm{lc}_3} \alpha_{\mathrm{lc}_2} f_4,$$

and the final weight vector:

$$\boldsymbol{\alpha}_{\mathrm{opt}} = \begin{pmatrix} \alpha_{\mathrm{lc}_3} & 1 & \alpha_{\mathrm{lc}_1} & \alpha_{\mathrm{lc}_2} \alpha_{\mathrm{lc}_3} \end{pmatrix}.$$

In order to recover the weight for each of the classifiers to be combined, a *combination tree* is built during the evaluation of the $\boldsymbol{\alpha}_{\mathrm{opt}}$. The original classifiers constitute the leaves of the tree and, each time a pair of classifiers is combined, a parent node is added which is connected to the nodes associated to the two combined classifiers. The edges are labelled with the weights assigned to each classifier. In fig. 4.9 an example of the construction of the combination tree is reported.

At the end of the computation, the weight of each classifier can be easily recovered by traversing the tree from the leaf up to the root and multiplying all the values found on the edges.

To better explain the proposed algorithm a pseudo code of the method is reported in algorithm 4.2.

Algorithm 4.2 A method for the application of the greedy approach in the combination rule

Input: K, the number of classifiers to be combined; $m_P > 0$ and $m_N > 0$, the number of positive and negative samples; (x_i^h, y_j^h), the output of the h-th classifiers on the i-th positive sample and the j-th negative sample $\forall h = 1 \dots K, \forall i = 1 \dots m_P, \forall j = 1 \dots m_N$.

Output: α, the weight vector of the linear combination.

$SD_h^{(0)} \leftarrow (\; x_1^h - y_1^h \quad x_1^h - y_2^h \quad \dots \quad x_1^h - y_{m_N}^h \quad x_2^h - y_1^h \quad \dots \quad x_{m_P}^h - y_{m_N}^h \;)^T$ /*the SD for each classifier at step 0*/

$S^{(0)} \leftarrow SD_h^{(0)}$

for $h = 1$ to $K - 1$ **do**

  for $k = h + 1$ to K **do**

    build $X_{\bar{h}k}$ and $X_{h\bar{k}}$

    $\delta^{\bar{h}k} \leftarrow \delta_r^{\bar{h}k}$ with $r = 1, \dots, \text{card}(X_{\bar{h}k})$

    $\delta^{h\bar{k}} \leftarrow \delta_s^{h\bar{k}}$ with $s = 1, \dots, \text{card}(X_{h\bar{k}})$

    $R\Delta_{h,k}^{(0)} \leftarrow \frac{\text{card}(X_{\bar{h}k}) + \text{card}(X_{h\bar{k}})}{m_P m_N} AUDC(\delta^{h\bar{k}}, \delta^{\bar{h}k})$ /*evaluate the diversity matrix at step 0*/

  end for

end for

for $m = 1$ to $K - 1$ **do**

  $(u, v) \leftarrow \arg\max_{h,k} R\Delta_{h,k}^{(m-1)}$ /*find the pair of classifiers with the highest diversity*/

  evaluate the α_{opt} on $DROC((\delta^{u\bar{v}}, \delta^{\bar{u}v})$

  update the combination tree

  put $S_u^{(m-1)} + \alpha_{opt} S_v^{(m-1)}$ in the 1^{st} column of $S^{(m)}$ eliminating the u-th and the v-th column of $S^{(m-1)}$

  $n \leftarrow 2$

  while ($m < K - 1$ and $n < K - m$) **do**

    build $X_{\bar{1}n}$ and $X_{1\bar{n}}$

    $\delta^{\bar{1}n} \leftarrow \delta_r^{\bar{1}n}$ with $r = 1, \dots, \text{card}(X_{\bar{1}n})$

    $\delta^{1\bar{n}} \leftarrow \delta_s^{1\bar{n}}$ with $s = 1, \dots, \text{card}(X_{1\bar{n}})$

    $R\Delta_{1,n} \leftarrow \frac{\text{card}(X_{\bar{1}n}) + \text{card}(X_{1\bar{n}})}{m_P m_N} AUDC(\delta^{1\bar{n}}, \delta^{\bar{1}n})$

    $n \leftarrow n + 1$

  end while

  update the diversity matrix, i.e. put $R\Delta_{1,n}$ in the 1^{st} row of the matrix $R\Delta^{(m)}$ eliminating the row and the column corresponding to the u-th and the v-th classifier

end for

evaluate α by multiplying the values on the edges of the tree

4.8 Experiments and Discussion

In this section we give some results on the quality of the performance of the proposed method in order to show its effectiveness. To this aim, two different comparisons have been performed. The former consists in an evaluation of the reliability of the weight search, i.e. we want to show that our algorithm is able to maximize the AUC, while in the second subsection we want to put in evidence the behavior of our method with respect to other combining rules described in literature.

In order to evaluate the performance of the proposed method, it has been tested on several data sets publicly available at the UCI Machine Learning Repository (Blake *et al.*, 1998). All of them have two classes and a variable number of numerical input features. The features were previously scaled so as to have zero mean and unit standard deviation.

To avoid any bias in the comparison, 10 runs of a multiple hold out procedure (Duda *et al.*, 2001) have been performed on all data sets. In each run, the data set has been parted into three sets: two training sets, one to train the classifier and one to estimate the optimal weight (i.e. to train the combiner and a test set to assess the reliability of the proposed method. More details for each data set are given in appendix A.

It is worth recalling that the comparison has been performed in terms of AUC since we are aiming at the maximization of the ranking quality of the combination rule and not at the evaluation of the error rate (or other measures depending on a threshold value). Hence, in our experiments only the value of the AUC has been evaluated using the WMW statistic according to eq. (2.23).

4.8.1 Validation of the Estimated Weight Vector

The first part of our experiments focuses on the analysis of the weight vector estimated by the proposed method. To this aim, the employed base dichotomizers are SVM and Multi-Layer Perceptrons (MLP) (see appendix 3.1.1 for the characteristic of the classifiers). The SV-based classifiers have been implemented by means of SVM^{light} tool (Joachims, 1999) while for the MLPs we have employed the NODElib library (Flake & Pearlmuter, 2000). Three different kernels have been used for the SVMs while for the MLPs we have considered three classifiers with different numbers of units in the hidden layer, all trained for 10,000 epochs using the back propagation algorithm with a learning rate of 0.01. The characteristics of the nine employed dichotomizers are reported in table 4.2 with the relative acronyms used in the following tables. Our experiments focus on the validation of the proposed method for two classifiers. To this aim, five data sets (Breast, CMC, Diabetes, German and Heart) have been used.

Table 4.2: Acronyms of the classifiers used in the experiments.

Type of Classifier	Type of Kernel or Number of Hidden Nodes	Acronym
SVM	Linear	SL
SVM	Polynomial of degree 2	SP2
SVM	Polynomial of degree 3	SP3
SVM	RBF with r = 1	SR1
SVM	RBF with r = 2	SR2
SVM	RBF with r = 5	SR5
MLP	2	M2
MLP	4	M4
MLP	6	M6

Let us analyze the behavior of our rule for two classifiers. In the performed experiments all the 15 combinations which can be accomplished with the employed dichotomizers have been considered. For each combination, we have evaluated the weight α_{opt} by means of the proposed method on the training set of the combiner and then the achieved AUC through the WMW statistic on both this set and the test set.

For the sake of comparison, we have also considered another method which trivially chooses the weight maximizing the AUC through an exhaustive search. In particular, this method considers the set of values for α varying in the range $[0, 50]$ with a step of 0.01; for each of them, the outputs of the two classifiers on the second training set are combined according to eq. (4.11) and the relative AUC is computed through the WMW statistic. Finally the value of α corresponding to the maximum AUC is picked out. The aim here is not to provide another algorithm to construct the optimal combination, but to obtain a reliable estimate of the weight maximizing the AUC on the training set of the combiner, which can be compared with the α_{opt} provided by the proposed method[3].

In this way, for each data set, the hold out procedure provides 10 AUC values for each method. This allows us to employ the Wilcoxon rank-sum test (Wilcoxon, 1945), (Walpole *et al.*, 1998) (see appendix B for more details), so as to verify if the differences in the means of the two populations are statistically significant. All the results were provided with a significance level equal to 0.05.

Let us firstly analyze the results obtained on the training set of the combiner of the five employed data sets which are reported in tables 4.3-4.7. Each entry of the tables contains the mean (and the standard deviation in parentheses) of the AUC values obtained in the 10 runs of the hold out procedure. A bold value means that such value is significantly better than the other one. If the compared methods have undistinguishable means the

[3]It is worth noting that the involved computational complexity of the exhaustive search is very high: $O(N_p m_P m_N)$ where N_p is the number of points considered for α.

Table 4.3: Results on the training set of the combiner for Breast data set.

	DROC	Exh. Search
SL-M2	0.995 (0.003)	0.995 (0.003)
SL-M4	0.995 (0.003)	**0.996** (**0.003**)
SL-M6	0.995 (0.004)	**0.996** (**0.004**)
SP2-M2	0.995 (0.004)	**0.995** (**0.004**)
SP2-M4	0.995 (0.004)	0.996 (0.003)
SP2-M6	0.995 (0.004)	0.996 (0.004)
SR1-M2	0.986 (0.009)	0.986 (0.009)
SR1-M4	0.990 (0.005)	**0.990** (**0.005**)
SR1-M6	0.987 (0.009)	**0.987** (**0.009**)
SL-SP2	0.996 (0.003)	0.996 (0.003)
SL-SR1	0.995 (0.004)	0.995 (0.003)
SP2-SR1	0.996 (0.004)	0.996 (0.004)
M2-M4	0.980 (0.016)	0.980 (0.016)
M2-M6	0.980 (0.017)	0.980 (0.017)
M4-M6	0.973 (0.027)	**0.976** (**0.027**)

Table 4.4: Results on the training set of the combiner for CMC data set.

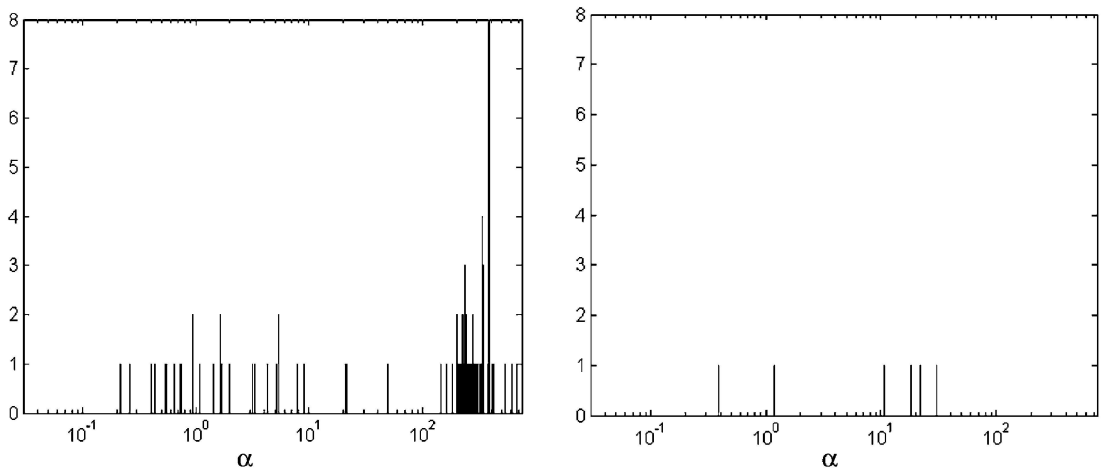
	DROC	Exh. Search
SL-M2	0.757 (0.037)	0.757 (0.037)
SL-M4	0.753 (0.030)	0.753 (0.030)
SL-M6	0.746 (0.038)	**0.746** (**0.038**)
SP2-M2	0.765 (0.035)	0.765 (0.035)
SP2-M4	0.763 (0.032)	**0.763** (**0.032**)
SP2-M6	0.764 (0.032)	0.764 (0.032)
SR1-M2	0.756 (0.034)	0.755 (0.034)
SR1-M4	0.746 (0.027)	0.747 (0.027)
SR1-M6	0.742 (0.033)	**0.742** (**0.034**)
SL-SP2	0.761 (0.037)	0.761 (0.037)
SL-SR1	0.743 (0.033)	**0.743** (**0.033**)
SP2-SR1	0.757 (0.036)	0.757 (0.036)
M2-M4	0.758 (0.029)	0.758 (0.029)
M2-M6	0.758 (0.035)	0.758 (0.035)
M4-M6	0.750 (0.028)	0.750 (0.028)

values are in normal style.

From these results we can see the good performance of the proposed method since the AUC values obtained are quite indistinguishable from those provided by the exhaustive search and thus the evaluated weight is actually able to maximize the AUC of the resulting classifier.

It is worth noting that, in the case of Breast data set, we have frequently obtained an extreme value for α_{opt} which excludes one dichotomizer from the combination. This is due to the very good performance reached by the best single dichotomizer which leads to two possible situations: one of the sets $X_{\bar{1}2}$ or $X_{1\bar{2}}$ is empty (i.e. the samples erroneously classified by a classifier are not correctly classified by the other) or there is a very low number of samples in one of the two sets. In the former case, one of the two dichotomizers is useless (as explained in sec. 4.4) because it cannot recover any error made by the other classifier. In the latter case, the distributions of the SDR $\Gamma_2^1(\mathbf{p}_i, \mathbf{n}_j)$ evaluated on the two sets $X_{\bar{1}2}$ or $X_{1\bar{2}}$ can be very imbalanced as shown in fig. 4.10, where the distributions for the linear combination of an SL and an M4 on the Breast data set are reported. In this case, each value of α greater than zero leads to a lower value for $F_{\bar{1}2}(\alpha) + F_{1\bar{2}}(\alpha)$ because the minimum value of α which allows some errors of f_1 to be recovered produces a higher number of errors of f_2 which can be no longer recovered. This can be clearly seen in fig. 4.11 where is shown that in this case $F_{\bar{1}2}(\alpha) + F_{1\bar{2}}(\alpha)$ is a monotonically decreasing function whose maximum is reached for $\alpha = 0$, i.e. when the combination reduces to the dichotomizer f_1.

Let us now analyze the results obtained on the test sets in terms of the AUC calculated



(a) (b)

Figure 4.10: The distributions of the SDRs $\Gamma_2^1(\mathbf{p}_i, \mathbf{n}_j)$ evaluated on $X_{1\bar{2}}$ (a) and $X_{\bar{1}2}$ (b) for the linear combination of an SL with an M4 on Breast data set.

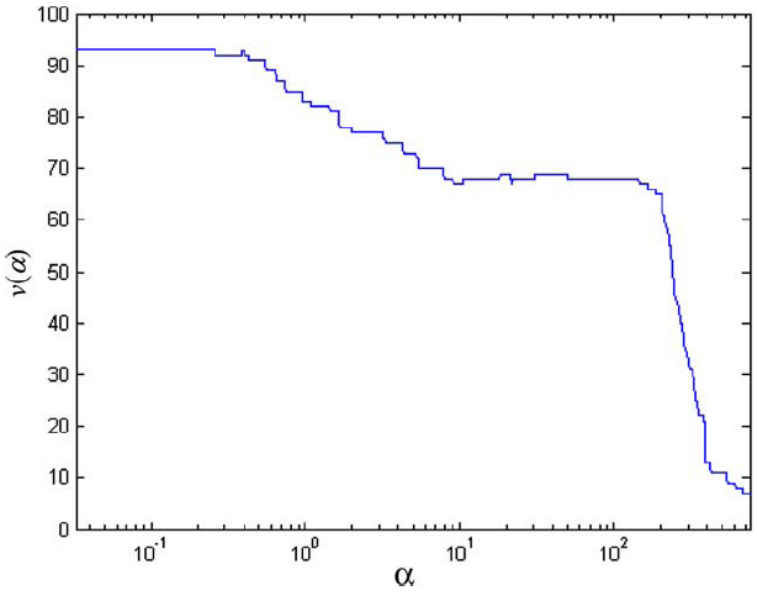


Figure 4.11: The trend of the function $\nu(\alpha) = F_{1\bar{2}}(\alpha) + F_{\bar{1}2}(\alpha)$ obtained by the two distributions shown in fig. 4.10 for the linear combination of an SL with an M4 on Breast data set.

Table 4.5: Results on the training set of the combiner for Diabetes data set.

	DROC	Exh. Search
SL-M2	0.838 (0.029)	0.838 (0.029)
SL-M4	0.841 (0.028)	0.841 (0.028)
SL-M6	0.838 (0.027)	0.838 (0.027)
SP2-M2	0.833 (0.029)	0.832 (0.030)
SP2-M4	0.831 (0.034)	0.832 (0.034)
SP2-M6	0.832 (0.032)	0.832 (0.032)
SR1-M2	0.809 (0.031)	0.809 (0.031)
SR1-M4	0.803 (0.037)	0.803 (0.037)
SR1-M6	0.795 (0.028)	0.795 (0.028)
SL-SP2	0.842 (0.029)	**0.843** (**0.029**)
SL-SR1	0.846 (0.027)	**0.847** (**0.027**)
SP2-SR1	0.833 (0.041)	**0.833** (**0.041**)
M2-M4	0.815 (0.031)	0.815 (0.030)
M2-M6	0.807 (0.033)	0.806 (0.032)
M4-M6	0.798 (0.034)	**0.798** (**0.034**)

Table 4.6: Results on the training set of the combiner for German data set.

	DROC	Exh. Search
SL-M2	0.803 (0.040)	0.803 (0.040)
SL-M4	0.801 (0.038)	0.802 (0.037)
SL-M6	0.808 (0.043)	0.808 (0.043)
SP2-M2	0.791 (0.038)	0.791 (0.038)
SP2-M4	0.782 (0.034)	**0.782** (**0.034**)
SP2-M6	0.789 (0.048)	0.789 (0.048)
SR1-M2	0.764 (0.029)	0.764 (0.029)
SR1-M4	**0.740** (**0.017**)	0.738 (0.019)
SR1-M6	0.742 (0.050)	0.741 (0.050)
SL-SP2	0.809 (0.040)	**0.810** (**0.040**)
SL-SR1	0.803 (0.038)	0.803 (0.038)
SP2-SR1	0.771 (0.043)	0.771 (0.043)
M2-M4	0.762 (0.029)	**0.762** (**0.029**)
M2-M6	0.776 (0.040)	**0.777** (**0.040**)
M4-M6	0.764 (0.037)	0.764 (0.037)

Table 4.7: Results on the training set of the combiner for Heart data set.

	DROC	Exh. Search
SL-M2	0.921 (0.022)	0.922 (0.022)
SL-M4	0.915 (0.025)	**0.916** (**0.025**)
SL-M6	0.923 (0.023)	0.924 (0.024)
SP2-M2	0.895 (0.037)	**0.896** (**0.037**)
SP2-M4	0.887 (0.033)	**0.888** (**0.033**)
SP2-M6	0.893 (0.038)	0.893 (0.037)
SR1-M2	0.886 (0.051)	**0.887** (**0.049**)
SR1-M4	0.871 (0.042)	0.872 (0.041)
SR1-M6	**0.869** (**0.057**)	0.867 (0.058)
SL-SP2	0.912 (0.026)	0.912 (0.026)
SL-SR1	0.915 (0.023)	**0.916** (**0.023**)
SP2-SR1	0.874 (0.037)	**0.875** (**0.037**)
M2-M4	0.895 (0.047)	0.895 (0.047)
M2-M6	0.884 (0.059)	0.885 (0.059)
M4-M6	0.883 (0.045)	**0.884** (**0.045**)

Table 4.8: Results on the test set for Breast data set.

	DROC	Exh. Search
SL-M2	0.988 (0.024)	0.988 (0.024)
SL-M4	0.991 (0.011)	**0.992** (**0.009**)
SL-M6	0.980 (0.049)	**0.980** (**0.049**)
SP2-M2	0.988 (0.023)	0.990 (0.023)
SP2-M4	0.995 (0.004)	0.995 (0.004)
SP2-M6	0.981 (0.049)	0.981 (0.049)
SR1-M2	0.975 (0.025)	**0.977** (**0.025**)
SR1-M4	0.982 (0.010)	0.982 (0.010)
SR1-M6	0.970 (0.046)	0.970 (0.045)
SL-SP2	**0.995** (**0.004**)	0.995 (0.005)
SL-SR1	0.994 (0.005)	0.994 (0.005)
SP2-SR1	0.995 (0.004)	0.993 (0.007)
M2-M4	0.960 (0.046)	0.960 (0.046)
M2-M6	0.950 (0.051)	0.950 (0.051)
M4-M6	0.950 (0.055)	0.950 (0.055)

Table 4.9: Results on the test set for CMC data set.

	DROC	Exh. Search
SL-M2	0.756 (0.029)	0.755 (0.030)
SL-M4	0.735 (0.033)	0.736 (0.033)
SL-M6	0.745 (0.034)	0.745 (0.035)
SP2-M2	0.752 (0.024)	**0.753** (**0.025**)
SP2-M4	0.746 (0.027)	0.746 (0.027)
SP2-M6	0.750 (0.035)	0.751 (0.036)
SR1-M2	0.757 (0.021)	0.756 (0.021)
SR1-M4	0.732 (0.022)	0.733 (0.022)
SR1-M6	0.740 (0.027)	0.740 (0.027)
SL-SP2	0.758 (0.032)	0.758 (0.031)
SL-SR1	0.737 (0.034)	0.737 (0.034)
SP2-SR1	0.753 (0.029)	0.753 (0.029)
M2-M4	0.746 (0.023)	0.746 (0.023)
M2-M6	0.756 (0.025)	0.755 (0.025)
M4-M6	0.756 (0.025)	0.755 (0.025)

Table 4.10: Results on the test set for Diabetes data set.

	DROC	Exh. Search
SL-M2	0.837 (0.036)	0.836 (0.036)
SL-M4	0.831 (0.026)	0.831 (0.027)
SL-M6	0.834 (0.036)	0.835 (0.033)
SP2-M2	0.831 (0.032)	0.831 (0.031)
SP2-M4	0.826 (0.031)	0.827 (0.031)
SP2-M6	0.831 (0.031)	0.831 (0.031)
SR1-M2	0.796 (0.032)	0.797 (0.032)
SR1-M4	0.796 (0.021)	0.793 (0.022)
SR1-M6	0.773 (0.033)	0.773 (0.033)
SL-SP2	0.835 (0.031)	0.835 (0.031)
SL-SR1	0.829 (0.024)	0.829 (0.025)
SP2-SR1	0.825 (0.022)	0.826 (0.022)
M2-M4	0.811 (0.037)	0.812 (0.037)
M2-M6	0.798 (0.045)	0.798 (0.045)
M4-M6	0.790 (0.045)	0.790 (0.046)

combining the two dichotomizers with the weights estimated on the training set of the combiner. The results reported in tables 4.8-4.12 are structured in the same way as before. Even in this case the proposed method provides practically the same results as the exhaustive search, thus proving that α_{opt} is a good estimate of the optimal combination weight also on the test sets.

4.8.2 Comparison with Other Combination Methods

Since we have shown that the proposed method has satisfactory results in estimating the weight for the combination of two classifiers, let us proceed with the analysis of the greedy approach to combine K classifiers. In this case, the exhaustive search has too high complexity to be performed; therefore, we focus on the comparison of the proposed combination rule with other common rules present in literature.

Since in these experiments we want to put in evidence the good behavior of the combination method we employ low correlated classifiers. In fact, weakening the individual classifiers appears to be an excellent ensemble building strategy, unequivocally demonstrated by AdaBoost in Freund & Schapire (1997).

However, using the classification models proposed in the previous experiments it is not possible to build an ensemble of $K > 2$ classifiers sufficiently different. To this aim, a linear classifier based on a random evaluation of the weights for the linear combination of the features has been considered. The only constrain on the model of this classifier is that an AUC greater than 0.5 has to be guaranteed. This classifier (that we called AUC-

Table 4.11: Results on the test set for German data set.

	DROC	Exh. Search
SL-M2	0.793 (0.045)	0.793 (0.046)
SL-M4	0.790 (0.041)	0.790 (0.041)
SL-M6	**0.790** (**0.042**)	0.789 (0.042)
SP2-M2	0.755 (0.049)	0.755 (0.049)
SP2-M4	0.720 (0.038)	0.719 (0.042)
SP2-M6	0.705 (0.049)	0.711 (0.043)
SR1-M2	0.764 (0.029)	0.764 (0.029)
SR1-M4	0.740 (0.017)	0.738 (0.019)
SR1-M6	0.742 (0.050)	0.741 (0.050)
SL-SP2	0.796 (0.045)	0.796 (0.045)
SL-SR1	0.788 (0.044)	0.790 (0.042)
SP2-SR1	0.744 (0.049)	0.746 (0.049)
M2-M4	0.750 (0.040)	0.751 (0.038)
M2-M6	**0.753** (**0.057**)	0.753 (0.058)
M4-M6	0.731 (0.042)	0.732 (0.041)

Table 4.12: Results on the test set for Heart data set.

	DROC	Exh. Search
SL-M2	0.899 (0.028)	0.900 (0.027)
SL-M4	0.902 (0.025)	0.901 (0.026)
SL-M6	0.888 (0.051)	**0.891** (**0.049**)
SP2-M2	0.877 (0.033)	0.876 (0.032)
SP2-M4	0.874 (0.045)	0.874 (0.047)
SP2-M6	0.846 (0.052)	0.847 (0.053)
SR1-M2	0.869 (0.056)	0.870 (0.055)
SR1-M4	0.827 (0.070)	0.835 (0.060)
SR1-M6	0.806 (0.075)	0.808 (0.074)
SL-SP2	0.901 (0.027)	0.902 (0.028)
SL-SR1	0.900 (0.030)	0.901 (0.027)
SP2-SR1	0.839 (0.045)	0.850 (0.055)
M2-M4	0.871 (0.044)	0.867 (0.050)
M2-M6	**0.856** (**0.065**)	0.850 (0.066)
M4-M6	0.835 (0.058)	0.836 (0.060)

Algorithm 4.3 A method to generate AUC-based random linear classifiers (ARLC)

Input: $X^{tr} = \{x_1 \dots x_Q\}$, a Q-dimensional training set;
Output: $\mathbf{w}$, the parameters of the ARLC;
$\mathbf{w} \leftarrow rand_Q$ /*initialize $\mathbf{w}$ with a Q-dimensional random vector*/
$\mathbf{w} \leftarrow \frac{\mathbf{w}}{\sum_{i=1}^{Q} w_i}$ /*normalize the weight vector*/
$f_{ARLC} = \sum_{i=1}^{Q} x_i w_i$
evaluate the AUC of f_{ARLC}
if $AUC < 0.5$ **then**
  $\mathbf{w} = -\mathbf{w}$
end if

based Random Linear Classifiers (ARLC), see the pseudo code in algorithm 4.3) is not a a dichotomizer with outstanding performance but let us easily build a certain number of classifiers with low correlation. For the sake of comparison in our experiments also MLP were used to employ a classifier well known in literature. Also in this case the MLP were built with a low complexity to guarantee a low correlation among the base classifiers. In particular, MLPs with 100 nodes in the hidden layer, all trained for 300 epochs using the back propagation algorithm with a learning rate of 0.01 were implemented.

To avoid any bias in the comparison several classifiers have been generated and 30 random combinations of them have been realized varying the number of base classifiers from 2 to 7 for both the ARLC and the MLP. In this comparison twelve data sets based on real data have been employed (see appendix A for more details).

Different linear and non linear combination rules both trainable and non trainable

have been employed. In particular, 7 combination rules (all presented in sect. 4.2 and 4.3) have been employed: maximum (MAX in the following tables), minimum (MIN), median (MED), SA, WA (with the weights chosen as in eq. (4.9) but related to the AUC of the base classifier and not to its accuracy), trimmed mean (TRIM) and product (PROD). For the sake of comparison the performance obtained by the best base classifier (BEST in the following tables) employed in each combination have been reported together with the results obtained using a a bound constrained global optimization algorithm, called *Multilevel Coordinate Search* (MCS, see appendix C for more details)(Huyer & Neumaier, 1999).

This algorithm is based on a multilevel coordinate search that balances global and local search; the local search is done via sequential quadratic programming and it is not exhaustive. Beyond its computational complexity lower than the exhaustive search (that in this case should have been performed with a K-dimensional grid approach), MCS has been used since it does not require any differentiation of the objective function and, as a consequence, it is possible to perform the maximization avoiding the use of an approximation of the WMW statistic. The aim here is not to provide another algorithm to construct the optimal combination, but to obtain a reliable estimate of the weight vector maximizing the AUC (even though with a computationally expensive algorithm) to compare the proposed method.

For each combination rule the mean and the standard deviation of the AUC on the 10 multiple hold out procedure has been evaluated and to assess a statistically significant difference between the employed rules the Friedman test (Friedman, 1937) on each data set has been performed with 9 (number of algorithms $-1 = 10 - 1$) and 261 ((number of algorithms -1) $*$ (number of performed combination -1) $= (10 - 1) * (30 - 1)$) degrees of freedom (see appendix B for more details) and a level of significance equal to 0.05 (see sec. 3.5.2 for more explanations).

In all the performed experiments, for every data set and every number of employed classifiers in the combination, the null hypothesis (i.e. no statistical difference between the employed combination rules) of the Friedman test has been rejected. Therefore, a post-hoc test has been applied. As in the experiments of the previous chapter also in this case we do not want to make pairwise comparisons between the different methods but to test if the DROC method is better than the existing ones Therefore, the Holm's step-down procedure (Holm, 1979) can be used to find which combination rule exhibits a statistically different behavior from the DROC approach.

The obtained results are reported in tables 4.13-4.18 for the ARLC and in tables 4.19-4.24 for the MLPs. Each table corresponds to the combination of a certain number of

classifiers. In this case, it is not correct to compare the AUC values of the combination, since the classifiers used in each combination are not the same. The proper way to make such comparison is to consider the mean rank on the thirty considered combination for each method on each data set. Therefore, each cell of the table contains the mean rank obtained by the corresponding combination rule on the relative data set. A bold value in the table indicates that the corresponding method on that data set has lower statistically significant performance with respect to the proposed approach according to the Holm procedure. If the value is underlined the DROC rule exhibits lower performance compared to that method while if the value is in normal style it means that the corresponding method has undistinguishable performance from our method. Also the Holm test has been performed with a level of significance equal to 0.05.

Let us firstly analyze the obtained results without considering the MCS algorithm. If we consider the ARLC (in tables 4.13-4.18) the DROC rule performs statistically better than all the other methods in the majority of cases for every number of employed classifiers. There are only few exceptions on Diabetes data set. In this case the minimum rule exhibits equal performance than the DROC method in almost all cases and better performance when using six or seven classifiers. Moreover, for seven classifiers also the WA, the SA and the best base classifiers has the same performance than our method.

Using the MLPs the results (see tables 4.19-4.24) are slightly different but our method always exhibits better performance in almost the all cases. When the number of classifiers varies between three and six DROC is better in all cases; for two classifiers the WA is better on Balance data set while the performance are almost equivalent among all methods for Breast data set that is a very simple problem (i.e., the performance of the base classifiers are very high). For seven classifiers WA is again better in Hayes and Diabetes data sets (and equal on Australian) while the median and the trimmed mean are the best methods on Breast data set.

Finally, we can assess that our method performs better of well known combination rule on all the employed data sets for the considered classifiers.

Let us now focus on the MCS rule. In the comparison with this method the Holm test assesses that the DROC method performs worse than the MCS in almost all cases (except for Liver data set) of the ARLC combination and in all cases when using the MLP. However, as said before, MCS is not a real combination method since, due to its high computational complexity, it is not applicable in a real case.

To this aim, let us analyze the computational complexity of this algorithm: the complexity of the global search depends on the parameters of the algorithm (dimensionality of the problem (i.e. K in our case), number of the boxes in which the space is partitioned

and number of iteration of the search in each box (see appendix C for more details)) while the local search is quadratic in the optimization problem that depends from the dimensionality of the problem (equal to the number of classifiers K in our case). Moreover, since for each run of the optimization we have to evaluate the WMW statistic (that has a quadratic complexity in the number of samples), the estimation of the combination weight using MCS is dependent on the square of the number of samples multiplied for the number of classifiers and the product of the parameters described before.

Finally, let us make some reasoning about the computational complexity of the proposed method. The first step estimates the distributions of the SDRs $\delta^{\bar{1}2}$ and $\delta^{1\bar{2}}$: the complexity is $O(n^2)$ in the number of samples since it depends on the number of pairs (p_i, n_j) to be considered, that are $m_P \cdot m_N$. The second step is the evaluation of the DROC curve and its convex hull that is $O(nlogn)$ as shown using the algorithm 4.1 and, then, the evaluation of the optimal weight that is linear in the number of points of the convex hull. Therefore, for two classifiers the complexity is $O(n^2)$. When applying to K classifiers we have to compute the diversity matrix that is always dependent on the number of pairs (again a quadratic problem) and then perform $K-1$ times the previous procedure. At the end, the final computational complexity is K times $O(n^2)$. Since $K << L$ the proposed method has a quadratic computational complexity.

Hence, we can say that the complexity of the DROC approach is significantly lower than the MCS complexity that therefore, represents only a useful instrument for the comparison of our algorithm (it can be seen as an upper bound of the performance reachable by our method). Hence, we can conclude that MCS is not a suitable method in real applications and the proposed combiner exhibits better performance than other common rules in the maximization of the ranking for the combination of dichotomizers.

Table 4.13: The results in terms of mean rank obtained on all the data sets for thirty random combination of two ARLC.

Data Sets	Combination Rules									
	DROC	**MCS**	**WA**	**SA**	**MIN**	**MAX**	**MED**	**TRIM**	**PROD**	**BEST**
Australian	1.833	1.167	**3.367**	**5.367**	**8.300**	**7.467**	**5.367**	**5.367**	**10.000**	**6.767**
Balance	1.400	1.600	**4.500**	**4.500**	**7.867**	**7.400**	**4.500**	**4.500**	**10.000**	**8.733**
Breast	2.000	1.067	**5.283**	**5.483**	**8.033**	**6.167**	**5.483**	**5.483**	**10.000**	**6.000**
Heart	1.883	1.117	**3.800**	**5.617**	**8.300**	**5.217**	**5.617**	**5.617**	**10.000**	**7.833**
CMC	1.300	1.700	**3.900**	**6.100**	**8.967**	**3.800**	**6.100**	**6.100**	**10.000**	**7.033**
German	1.633	1.367	**4.100**	**6.500**	**5.433**	**8.433**	**6.500**	**6.500**	**10.000**	**4.533**
Hayes	2.000	1.000	**4.167**	**6.733**	**8.867**	**4.133**	**6.733**	**6.733**	**10.000**	**4.633**
Housing	1.933	1.067	**3.967**	**6.367**	**5.100**	**8.700**	**6.367**	**6.367**	**10.000**	**5.133**
Ionosphere	2.000	1.067	**4.767**	**6.667**	**9.000**	**3.067**	**6.667**	**6.667**	**10.000**	**5.100**
Liver	1.867	1.133	**4.033**	**5.400**	**6.200**	**8.767**	**5.400**	**5.400**	**9.400**	**7.400**
Diabetes	2.500	1.000	**3.733**	**5.900**	**4.033**	**8.800**	**5.900**	**5.900**	**10.000**	**7.233**
Sonar	2.000	1.000	**4.817**	**6.600**	**7.700**	**5.383**	**6.600**	**6.600**	**10.000**	**4.300**

Table 4.14: The results in terms of mean rank obtained on all the data sets for thirty random combination of three ARLC.

Data Sets	Combination Rules									
	DROC	MCS	WA	SA	MIN	MAX	MED	TRIM	PROD	BEST
Australian	1.967	1.033	**3.333**	**4.800**	**8.100**	**7.000**	**6.550**	**6.550**	**10.000**	**5.667**
Balance	1.933	1.067	**3.033**	**3.967**	**5.933**	**5.500**	**7.500**	**7.500**	**10.000**	**8.567**
Breast	2.000	1.000	**5.033**	**5.200**	**7.233**	**6.200**	**6.617**	**6.617**	**10.000**	**5.100**
Heart	2.000	1.000	**3.583**	**4.967**	**7.867**	**4.050**	**7.000**	**7.000**	**10.000**	**7.533**
CMC	1.700	1.300	**4.267**	**5.467**	**9.000**	**3.367**	**6.600**	**6.600**	**10.000**	**6.700**
German	1.850	1.150	**4.667**	**6.567**	**4.733**	**8.700**	**6.700**	**6.700**	**10.000**	**3.933**
Hayes	2.000	1.000	**4.300**	**7.267**	**8.833**	**4.800**	**5.917**	**5.917**	**10.000**	**4.967**
Housing	1.917	1.083	**3.900**	**5.967**	**4.867**	**8.867**	**6.967**	**6.967**	**10.000**	**4.467**
Ionosphere	1.950	1.117	**5.200**	**6.367**	**9.000**	**2.933**	**6.933**	**6.933**	**10.000**	**4.567**
Liver	1.600	1.400	**3.900**	**5.433**	**3.733**	**9.333**	**8.083**	**8.083**	**7.433**	**6.000**
Diabetes	2.733	1.000	**4.000**	**5.467**	3.233	**8.933**	**7.033**	**7.033**	**10.000**	**5.567**
Sonar	1.967	1.033	**4.983**	**6.267**	**7.683**	**5.033**	**7.350**	**7.350**	**10.000**	**3.333**

Table 4.15: The results in terms of mean rank obtained on all the data sets for thirty random combination of four ARLC.

Data Sets	Combination Rules									
	DROC	MCS	WA	SA	MIN	MAX	MED	TRIM	PROD	BEST
Australian	1.967	1.033	**3.433**	**5.167**	**8.500**	**7.600**	**6.317**	**6.317**	**10.000**	**4.667**
Balance	1.733	1.267	**3.133**	**3.900**	**7.833**	**6.967**	**5.633**	**5.633**	**10.000**	**8.900**
Breast	2.000	1.000	**4.950**	**5.083**	**8.100**	**6.267**	**6.333**	**6.333**	**10.000**	**4.933**
Heart	2.000	1.000	**3.717**	**5.250**	**8.500**	**3.567**	**6.567**	**6.567**	**10.000**	**7.833**
CMC	1.700	1.300	**4.133**	**5.433**	**9.000**	**3.000**	**6.567**	**6.567**	**10.000**	**7.300**
German	1.967	1.033	**4.967**	**7.100**	**4.533**	**9.000**	**6.333**	**6.333**	**10.000**	**3.733**
Hayes	1.917	1.083	**4.967**	**7.233**	**9.000**	**4.600**	**6.100**	**6.100**	**10.000**	**4.000**
Housing	1.933	1.067	**4.133**	**6.333**	**5.567**	**9.000**	**6.500**	**6.500**	**10.000**	**3.967**
Ionosphere	1.817	1.183	**5.067**	**6.300**	**9.000**	**3.000**	**7.283**	**7.283**	**10.000**	**4.067**
Liver	1.067	**1.933**	**3.933**	**5.067**	**3.867**	**9.967**	**7.033**	**7.033**	**8.267**	**6.833**
Diabetes	2.600	1.000	**3.933**	**5.400**	3.067	**9.000**	**6.900**	**6.900**	**10.000**	**6.200**
Sonar	1.967	1.033	**5.533**	**6.950**	**8.150**	**4.100**	**7.033**	**7.033**	**10.000**	**3.200**

Table 4.16: The results in terms of mean rank obtained on all the data sets for thirty random combination of five ARLC.

Data Sets	Combination Rules									
	DROC	MCS	WA	SA	MIN	MAX	MED	TRIM	PROD	BEST
Australian	2.000	1.000	**3.267**	**5.167**	**8.600**	**8.200**	**6.733**	**5.300**	**10.000**	**4.733**
Balance	1.667	1.333	**3.033**	**4.000**	**7.167**	**5.833**	**8.133**	**5.267**	**10.000**	**8.567**
Breast	1.967	1.033	**5.317**	**5.617**	**8.367**	**5.950**	**5.933**	**6.750**	**10.000**	**4.067**
Heart	1.850	1.150	**3.800**	**4.983**	**8.483**	**3.283**	**7.233**	**6.383**	**10.000**	**7.833**
CMC	1.800	1.200	**4.100**	**5.100**	**9.000**	**3.000**	**6.900**	**6.367**	**10.000**	**7.533**
German	2.000	1.000	**5.167**	**7.200**	**4.900**	**9.000**	**6.150**	**6.150**	**10.000**	**3.433**
Hayes	1.967	1.033	**5.417**	**7.733**	**9.000**	**4.100**	**5.700**	**6.350**	**10.000**	**3.700**
Housing	1.967	1.033	**4.300**	**6.567**	**5.700**	**9.000**	**6.833**	**6.467**	**10.000**	**3.133**
Ionosphere	1.767	1.233	**5.233**	**6.333**	**9.000**	**3.000**	**7.383**	**7.050**	**10.000**	**4.000**
Liver	1.000	**2.000**	**3.900**	**5.283**	**3.100**	**10.000**	**8.800**	**7.150**	**7.167**	**6.600**
Diabetes	2.700	1.000	**4.333**	**5.433**	2.367	**9.000**	**7.233**	**6.800**	**10.000**	**6.133**
Sonar	1.967	1.033	**5.533**	**6.983**	**8.433**	**4.033**	**6.867**	**6.950**	**10.000**	**3.200**

Table 4.17: The results in terms of mean rank obtained on all the data sets for thirty random combination of six ARLC.

Data Sets	Combination Rules									
	DROC	MCS	WA	SA	MIN	MAX	MED	TRIM	PROD	BEST
Australian	2.000	1.000	**3.133**	**5.167**	**8.650**	**8.317**	**6.267**	**4.800**	**10.000**	**5.667**
Balance	1.767	1.233	**3.167**	**3.833**	**7.900**	**6.867**	**6.233**	**5.000**	**10.000**	**9.000**
Breast	2.000	1.000	**5.600**	**5.717**	**8.733**	**5.983**	**6.033**	**6.167**	**10.000**	**3.767**
Heart	1.900	1.100	**3.900**	**5.133**	**8.767**	**3.100**	**7.067**	**6.333**	**10.000**	**7.700**
CMC	1.967	1.033	**4.000**	**5.267**	**9.000**	**3.000**	**6.767**	**6.333**	**10.000**	**7.633**
German	2.000	1.000	**5.100**	**7.600**	**4.600**	**9.000**	**5.833**	**6.733**	**10.000**	**3.133**
Hayes	1.950	1.050	**5.217**	**7.617**	**9.000**	**4.133**	**5.633**	**6.900**	**10.000**	**3.500**
Housing	2.000	1.000	**4.333**	**6.483**	**5.400**	**9.000**	**6.950**	**6.733**	**10.000**	**3.100**
Ionosphere	1.700	1.300	**5.067**	**6.133**	**9.000**	**3.000**	**7.467**	**7.333**	**10.000**	**4.000**
Liver	1.000	**2.000**	**4.133**	**5.733**	**3.000**	**10.000**	**7.533**	**7.400**	**7.333**	**6.867**
Diabetes	3.367	1.000	**4.000**	**5.367**	2.033	**9.000**	**7.267**	**6.833**	**10.000**	**6.133**
Sonar	1.933	1.067	**5.433**	**7.067**	**8.050**	**3.667**	**7.533**	**6.883**	**10.000**	**3.367**

Table 4.18: The results in terms of mean rank obtained on all the data sets for thirty random combination of seven ARLC.

Data Sets	Combination Rules									
	DROC	MCS	WA	SA	MIN	MAX	MED	TRIM	PROD	BEST
Australian	2.000	1.000	**3.100**	**5.000**	**8.700**	**8.300**	**6.283**	**4.283**	**10.000**	**6.333**
Balance	1.800	1.200	**3.133**	**3.867**	**7.267**	**6.200**	**7.733**	**5.000**	**10.000**	**8.800**
Breast	2.000	1.000	**6.083**	**6.417**	**9.000**	**5.317**	**5.783**	**6.000**	**10.000**	**3.400**
Heart	1.733	1.267	**4.000**	**5.000**	**8.867**	**3.000**	**7.333**	**6.167**	**10.000**	**7.633**
CMC	1.967	1.033	**4.000**	**5.067**	**9.000**	**3.000**	**7.233**	**6.300**	**10.000**	**7.400**
German	2.000	1.000	**4.833**	**7.467**	**4.367**	**9.000**	**6.433**	**6.867**	**10.000**	**3.033**
Hayes	1.933	1.067	**5.267**	**7.750**	**9.000**	**4.017**	**5.967**	**6.667**	**10.000**	**3.333**
Housing	2.000	1.000	**4.233**	**6.533**	**5.033**	**9.000**	**6.800**	**7.133**	**10.000**	**3.267**
Ionosphere	1.567	1.433	**5.200**	**6.400**	**9.000**	**3.000**	**6.933**	**7.467**	**10.000**	**4.000**
Liver	1.433	1.567	**4.033**	**5.600**	**3.000**	**10.000**	**8.667**	**7.100**	**7.267**	**6.333**
Diabetes	4.733	1.000	3.667	5.267	2.000	**9.000**	**7.667**	**6.900**	**10.000**	4.767
Sonar	1.933	1.067	**5.367**	**6.917**	**8.133**	**3.567**	**7.083**	**7.500**	**10.000**	**3.433**

Table 4.19: The results in terms of mean rank obtained on all the data sets for thirty random combination of two MLP classifiers.

Data Sets	Combination Rules									
	DROC	MCS	WA	SA	MIN	MAX	MED	TRIM	PROD	BEST
Australian	2.167	1.000	**2.833**	**5.133**	**7.333**	**8.033**	**5.133**	**5.133**	**10.000**	**8.233**
Balance	3.533	1.000	2.400	**4.800**	**7.700**	**8.067**	**4.800**	**4.800**	**10.000**	**7.900**
Breast	3.833	1.000	3.467	4.733	**8.433**	**5.900**	4.733	4.733	**10.000**	**8.167**
Heart	2.300	1.000	**3.000**	**4.967**	**7.000**	**8.367**	**4.967**	**4.967**	**10.000**	**8.433**
CMC	2.000	1.000	**3.133**	**5.333**	**8.567**	**6.900**	**5.333**	**5.333**	**10.000**	**7.400**
German	2.000	1.000	**3.067**	**5.133**	**7.433**	**8.367**	**5.133**	**5.133**	**10.000**	**7.733**
Hayes	2.433	1.000	**3.133**	**5.933**	**7.300**	**6.633**	**5.933**	**5.933**	**10.000**	**6.700**
Housing	2.167	1.000	**3.050**	**5.217**	**7.900**	**8.167**	**5.217**	**5.217**	**10.000**	**7.067**
Ionosphere	3.017	1.000	3.183	**5.033**	**8.900**	**6.267**	**5.033**	**5.033**	**10.000**	**7.533**
Liver	2.033	1.000	**3.633**	**5.867**	**7.733**	**7.367**	**5.867**	**5.867**	**9.967**	**5.667**
Diabetes	2.100	1.000	**2.933**	**5.067**	**6.967**	**8.700**	**5.067**	**5.067**	**10.000**	**8.100**
Sonar	2.100	1.000	**3.283**	**5.050**	**7.267**	**8.033**	**5.050**	**5.050**	**10.000**	**8.167**

Table 4.20: The results in terms of mean rank obtained on all the data sets for thirty random combination of three MLP classifiers.

Data Sets	Combination Rules									
	DROC	MCS	WA	SA	MIN	MAX	MED	TRIM	PROD	BEST
Australian	2.000	1.000	**3.000**	**4.200**	**7.967**	**8.300**	**5.500**	**5.500**	**10.000**	**7.533**
Balance	2.100	1.000	**3.067**	**5.100**	**7.967**	**8.367**	**4.967**	**4.967**	**10.000**	**7.467**
Breast	2.000	1.000	**3.967**	**4.767**	**8.667**	**6.467**	**4.967**	**4.967**	**10.000**	**8.200**
Heart	2.000	1.000	**3.067**	**4.133**	**6.800**	**8.233**	**5.733**	**5.733**	**10.000**	**8.300**
CMC	2.000	1.000	**3.433**	**5.133**	**8.933**	**7.233**	**5.100**	**5.100**	**10.000**	**7.067**
German	2.000	1.000	**3.333**	**4.867**	**7.233**	**9.000**	**5.100**	**5.100**	**10.000**	**7.367**
Hayes	2.200	1.000	**2.933**	**5.033**	**8.067**	**6.100**	**6.267**	**6.267**	**10.000**	**7.133**
Housing	2.000	1.000	**3.367**	**4.400**	**7.967**	**8.300**	**5.633**	**5.633**	**10.000**	**6.700**
Ionosphere	2.000	1.000	**3.300**	**4.367**	**8.933**	**6.600**	**5.867**	**5.867**	**10.000**	**7.067**
Liver	2.000	1.000	**3.500**	**5.467**	**8.333**	**7.833**	**5.733**	**5.733**	**10.000**	**5.400**
Diabetes	2.000	1.000	**3.000**	**4.133**	**6.167**	**8.867**	**5.867**	**5.867**	**10.000**	**8.100**
Sonar	2.000	1.000	**3.067**	**4.133**	**7.717**	**7.683**	**5.800**	**5.800**	**10.000**	**7.800**

Table 4.21: The results in terms of mean rank obtained on all the data sets for thirty random combination of four MLP classifiers.

Data Sets	Combination Rules									
	DROC	MCS	WA	SA	MIN	MAX	MED	TRIM	PROD	BEST
Australian	2.000	1.000	**3.067**	**4.367**	**7.700**	**8.400**	**5.600**	**5.133**	**10.000**	**7.733**
Balance	2.000	1.000	**3.567**	**5.933**	**8.233**	**8.267**	**4.650**	**3.850**	**10.000**	**7.500**
Breast	2.000	1.000	**4.367**	**5.400**	**8.933**	**6.833**	**4.617**	**3.883**	**10.000**	**7.967**
Heart	2.000	1.000	**3.133**	**4.267**	**7.033**	**8.633**	**5.300**	**5.300**	**10.000**	**8.333**
CMC	2.000	1.000	**3.200**	**5.367**	**8.967**	**7.400**	**4.733**	**4.733**	**10.000**	**7.600**
German	2.000	1.000	**3.933**	**5.733**	**7.533**	**8.967**	**4.200**	**4.200**	**10.000**	**7.433**
Hayes	2.000	1.000	**3.067**	**5.167**	**7.950**	**7.633**	**5.550**	**5.733**	**10.000**	**6.900**
Housing	2.000	1.000	**3.267**	**4.633**	**8.217**	**8.350**	**5.433**	**5.167**	**10.000**	**6.933**
Ionosphere	2.000	1.000	**3.383**	**4.700**	**9.000**	**6.800**	**5.167**	**5.350**	**10.000**	**7.600**
Liver	2.000	1.000	**3.300**	**6.100**	**8.567**	**8.400**	**5.033**	**4.967**	**10.000**	**5.633**
Diabetes	2.000	1.000	**3.000**	**4.267**	**6.900**	**8.967**	**5.150**	**5.683**	**10.000**	**8.033**
Sonar	2.000	1.000	**3.067**	**4.567**	**7.967**	**7.900**	**5.283**	**5.183**	**10.000**	**8.033**

Table 4.22: The results in terms of mean rank obtained on all the data sets for thirty random combination of five MLP classifiers.

Data Sets	Combination Rules									
	DROC	MCS	WA	SA	MIN	MAX	MED	TRIM	PROD	BEST
Australian	2.000	1.000	**3.133**	**4.633**	**7.467**	**8.600**	**6.000**	**4.333**	**10.000**	**7.833**
Balance	2.000	1.000	**3.300**	**5.967**	**8.233**	**8.633**	**4.167**	**4.567**	**10.000**	**7.133**
Breast	2.000	1.000	**4.817**	**5.650**	**9.000**	**7.033**	**3.967**	**3.567**	**10.000**	**7.967**
Heart	2.000	1.000	**3.250**	**4.333**	**6.900**	**8.633**	**5.883**	**4.633**	**10.000**	**8.367**
CMC	2.000	1.000	**3.033**	**5.300**	**9.000**	**7.667**	**5.233**	**4.433**	**10.000**	**7.333**
German	2.000	1.000	**3.700**	**5.433**	**7.700**	**9.000**	**5.100**	**3.767**	**10.000**	**7.300**
Hayes	2.000	1.000	**3.067**	**4.800**	**8.033**	**7.333**	**5.667**	**5.700**	**10.000**	**7.400**
Housing	2.000	1.000	**3.133**	**4.433**	**8.633**	**8.167**	**5.933**	**4.867**	**10.000**	**6.833**
Ionosphere	2.000	1.000	**3.200**	**4.533**	**8.967**	**7.167**	**5.833**	**4.767**	**10.000**	**7.533**
Liver	2.000	1.000	**3.250**	**6.167**	**8.600**	**8.400**	**5.083**	**5.233**	**10.000**	**5.267**
Diabetes	2.000	1.000	**3.000**	**4.067**	**6.967**	**8.967**	**6.000**	**4.967**	**10.000**	**8.033**
Sonar	2.000	1.000	**3.033**	**4.333**	**8.633**	**7.733**	**5.500**	**5.133**	**10.000**	**7.633**

Table 4.23: The results in terms of mean rank obtained on all the data sets for thirty random combination of six MLP classifiers.

Data Sets	Combination Rules									
	DROC	MCS	WA	SA	MIN	MAX	MED	TRIM	PROD	BEST
Australian	2.000	1.000	**3.333**	**5.100**	**7.400**	**8.767**	**5.767**	**3.800**	**10.000**	**7.833**
Balance	2.000	1.000	**3.767**	**6.000**	**8.233**	**8.767**	**3.333**	**4.900**	**10.000**	**7.000**
Breast	2.133	1.000	**4.867**	**5.867**	**9.000**	**7.000**	**3.200**	**3.933**	**10.000**	**8.000**
Heart	2.000	1.000	**3.267**	**4.483**	**7.000**	**8.867**	**5.583**	**4.667**	**10.000**	**8.133**
CMC	2.000	1.000	**3.267**	**5.900**	**9.000**	**7.867**	**4.233**	**4.600**	**10.000**	**7.133**
German	2.000	1.000	**3.833**	**5.900**	**7.900**	**9.000**	**4.367**	**3.900**	**10.000**	**7.100**
Hayes	2.300	1.000	2.700	**5.083**	**8.367**	**7.567**	**5.450**	**5.167**	**10.000**	**7.367**
Housing	2.000	1.000	**3.033**	**4.400**	**8.633**	**8.333**	**5.767**	**4.867**	**10.000**	**6.967**
Ionosphere	2.000	1.000	**3.300**	**4.900**	**9.000**	**7.300**	**5.367**	**4.600**	**10.000**	**7.533**
Liver	2.000	1.000	**3.300**	**6.500**	**8.633**	**8.367**	**4.600**	**5.500**	**10.000**	**5.100**
Diabetes	2.000	1.000	**3.000**	**4.100**	**7.000**	**8.933**	**5.933**	**4.967**	**10.000**	**8.067**
Sonar	2.000	1.000	**3.017**	**4.467**	**8.467**	**7.800**	**5.250**	**5.267**	**10.000**	**7.733**

Table 4.24: The results in terms of mean rank obtained on all the data sets for thirty random combination of seven MLP classifiers.

Data Sets	Combination Rules									
	DROC	MCS	WA	SA	MIN	MAX	MED	TRIM	PROD	BEST
Australian	2.600	1.000	2.750	**5.067**	**7.300**	**8.933**	**5.900**	**3.683**	**10.000**	**7.767**
Balance	2.000	1.000	**3.833**	**6.000**	**8.133**	**8.867**	**3.300**	**4.867**	**10.000**	**7.000**
Breast	3.367	1.000	**4.850**	**5.850**	**9.000**	**7.000**	2.433	3.500	**10.000**	**8.000**
Heart	2.117	1.000	**3.083**	**4.867**	**6.967**	**8.967**	**5.767**	**4.200**	**10.000**	**8.033**
CMC	2.233	1.000	**3.000**	**5.733**	**9.000**	**8.000**	**4.667**	**4.367**	**10.000**	**7.000**
German	2.367	1.000	**3.433**	**5.600**	**7.967**	**9.000**	**5.100**	**3.500**	**10.000**	**7.033**
Hayes	3.433	1.000	2.433	**4.700**	**8.767**	**7.267**	**5.200**	**5.067**	**10.000**	**7.133**
Housing	2.000	1.000	**3.000**	**4.400**	**8.533**	**8.467**	**6.100**	**4.667**	**10.000**	**6.833**
Ionosphere	2.300	1.000	**3.017**	**4.567**	**9.000**	**7.267**	**6.000**	**4.250**	**10.000**	**7.600**
Liver	2.300	1.000	**3.333**	**6.800**	**8.567**	**8.433**	**4.367**	**5.800**	**10.000**	**4.400**
Diabetes	2.100	1.000	**2.900**	**4.200**	**7.000**	**8.967**	**5.933**	**4.867**	**10.000**	**8.033**
Sonar	3.067	1.000	2.500	**4.050**	**8.800**	**7.633**	**5.550**	**4.833**	**10.000**	**7.567**

Chapter 5

Conclusions

In this thesis an application of the ROC methodology to construct classifiers and combination rule has been presented and in particular the AUC has been used as performance measure to be optimized in classification problems.

The ROC analysis has been introduced in the context of pattern recognition in which the aim is to build a rule to assign an object to one of a finite set of classes starting from known measurements of the features of the objects. In this work focusing on two class problems we considered a particular aspect of the classification. In particular, the difference between classification and ranking has been discussed and an analysis of the performance measure has been shown to assess the effectiveness of building a classification system using the AUC. In literature, AUC is known as a good measure to evaluate the classification performance since it is independent on the prior distributions of the classes and on misclassification costs. Recently, it has been also proposed as a measure of ranking. In fact, it has been shown that the AUC derived from an empirical ROC curve is equivalent to the WMW statistic that represents a statistical measure of ranking. Therefore, AUC can be used to build a good ranker.

Following this approach, in this thesis we have focused on the analysis of new techniques to improve the performance of a classification system in terms of AUC. In particular, two methods have been proposed to build a linear classifier and a combination rule for dichotomizers. In the former case a linear combination of features has been proposed and a greedy approach has been introduced to perform a pairwise combination estimating at each step the optimal weights in terms of ranking. In the latter a technique for the optimal linear combination between already trained dichotomizers has been investigated. Also in this case the dependence of the AUC on the weights of the combination has been analyzed and a method to find the optimal weight between two dichotomizers has been

shown. In particular, a new curve (the DROC curve) has been introduced to estimate the ranking separation of the dichotomizers output and a measure of ranking diversity has been proposed to better the performance of the combination rule. In this context we showed with extensive experiments that the proposed rules exhibit good performance in comparison with other well known methods of the literature using both artificial and real data. In particular, an important result has been obtained by our ranker in comparison with the RankBoost that is currently considered the best algorithm in literature.

The results obtained in this thesis are quite helpful in two different situations. Since AUC is independent on priors and misclassification costs it is often a more suitable measure than the classification error when dealing with medical detection problems or other screening applications where imbalanced class priors or misclassification costs are often present. The second aspect is the relation with ranking that becomes useful when the ordering is more important than the classification, e.g. when a ranking of customers in terms of their likelihoods of buying is needed or when the order of relevance of some documents in a database has to be estimated.

There are also numerous directions for future works. A first thing to be considered is to extend the approach to multiclass problems. When expressed in the form of the area under a curve, the AUC measure has no obvious generalization to multiple classes. However, when expressed in its equivalent probabilistic form, it has a straightforward generalization obtained by aggregation over all pairs of classes. Recently new papers focusing on the study of three dimensional ROC curves and on the meaning of the volume under this curve have been proposed and they can become useful in our context.

Another aspect that is marginal in this thesis but that can be usefully extended is the diversity problem applied to the combination approach in ranking problems. In fact, in literature the proposed diversity measures are all based on the accuracy of the classifier and the study and comparison of ranking diversity measures is missing. The recent increasing interest in building effective rankers clearly requires a deeper insight into this topic.

Finally, we would like to note that this work is part of a general research effort on learning algorithms on ranking problems. In the end, we can consider that if enough is known about the classes distributions and the costs of the problem the classical approach can be followed. However, if insufficient information is known about these characteristics then the ranking model can be more useful. In this case, we hope that our work will give further interest in problems which are becoming challenging in the pattern recognition community to explore new ways of designing learning algorithms.

Appendix A

Notes on Data Sets

This appendix is intended to present the principal characteristics of the data sets employed in the performed experiments. A brief description of each data set, both artificial and real one, is reported. The goal is both to render the work self-contained and to provide some useful notes for a deeper analysis of the behavior of the proposed methods. For the description of the real data the appendix is almost based on the notes found in the UCI website (Blake *et al.*, 1998) for each data set.

The appendix is organized as follows. The first section describes the artificial data sets employed in the experiments of chapter 3 while the second section focuses on the real data employed in both chapter 3 and 4. Each subsection describes a different data set and the title contains in parentheses the corresponding name used throughout the book. Finally, a table summarizing the principal characteristics of real data sets is reported.

A.1 Artificial Data Sets

To illustrate and compare the performance of the method proposed in chapter 3 several artificial data sets have been used in our experimental investigations. In particular, Gaussian data sets with different dimensionality have been employed since this is helpful to understand which factors affect the proposed technique. To this aim six different values for the dimensionality have been employed and the number of features has been chosen equal to 5, 10, 30, 50, 75 and 100. For each dimensionality different data sets have been generated to contemplate both correlated and uncorrelated data with different overlapping class distributions. All these data sets have been generated employing the PRTools (Duin, 2000) toolbox.

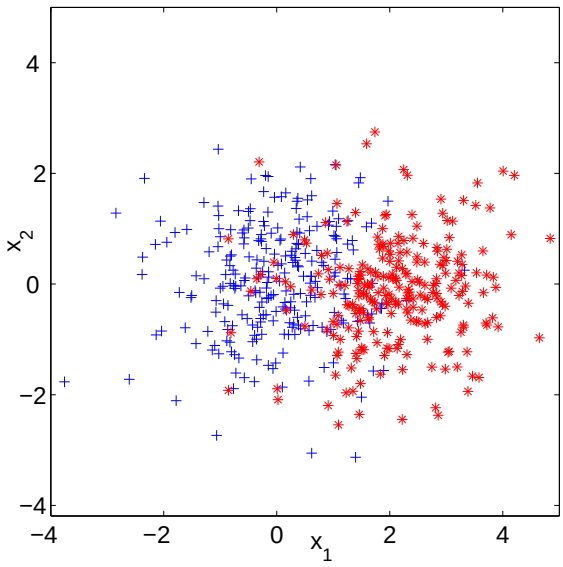


Figure A.1: Scatter plot of a two-dimensional projection of the 30-dimensional Gaussian spherical data.

A.1.1 Gaussian Spherical Data

The first set consists of two Gaussian spherical distributed data classes with equal covariance matrices. Each data set contains 500 samples with equal prior probabilities. The first class is Gaussian distributed with identity covariance matrix and zero mean. The covariance matrix of the second class is also an identity matrix while the mean has been chosen equal to 0.3, 0.5 and 1 so as to generate three different data sets (for each considered dimensionality) with different overlapping for the two distributions. A plot of a two dimensional projection of the features is presented in fig. A.1.

A.1.2 Gaussian Correlated Data

The second set is a correlated Gaussian data set consisting of two classes with equal covariances matrices. Each class consists of 250 samples. The mean of the first class is equal to zero for all features. The mean of the second class has been chosen equal to 1, 2 and 3 (so as to generate three different data sets for each considered dimensionality) for the first feature and equal to 0 for all the other features. The two covariance matrices are equal diagonal matrices with a variance of 40 for the second feature and a unit variance for all the other features. The data set is rotated in the subspace spanned by the first two

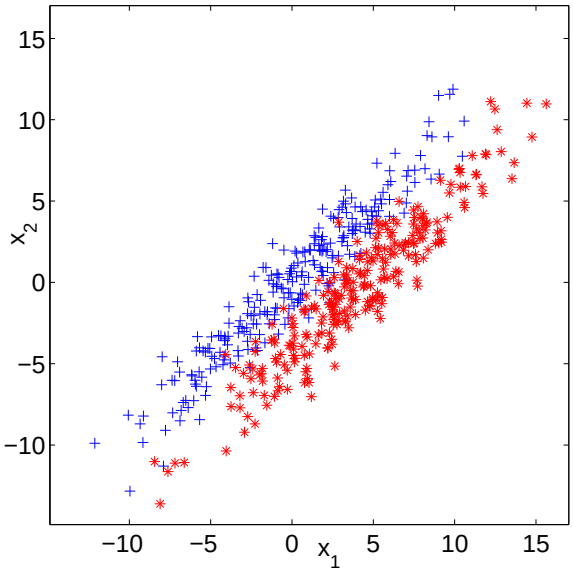


Figure A.2: Scatter plot of a two-dimensional projection of the 10-dimensional Gaussian correlated data.

features using a matrix equal to $\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$ that corresponds to a rotation of 45 degrees to construct a strong correlation. The first two features for this data set are reported in fig. A.2.

A.2 Real Data Sets

In our experiments also real data sets are needed to show that the proposed techniques may be effective when solving real world problems. To this aim several data sets from UCI Machine Learning Repository (Blake *et al.*, 1998) commonly used by researchers in literature have been employed to evaluate the performance of our approaches with respect to the state of the art. The principal characteristics of the data are reported in table A.1.

A.2.1 Cardiac Arrhythmia Database (Arrhythmia)

The aim of this data set is to determine the type of arrhythmia from the electrocardiogram recordings and to classify it in one of the 16 groups. In particular, in our case we have to distinguish between the presence and absence of a cardiac arrhythmia so reducing the problem to a binary one. This database contains 278 attributes, 206 of which are linear

Table A.1: Principal characteristics of the employed real data sets.

Data Sets	Number of Samples	Number of Features	% of Samples in the P Class	% of Samples in the N Class
Arrhythmia	420	278	43.57	56.43
Australian	690	14	44.49	55.51
Balance	625	4	54.01	45.99
Biomed	194	5	34.54	65.46
Breast	699	16	65.01	34.99
Cancer_wpbc	198	32	23.74	76.26
CMC	1473	9	42.70	57.30
Diabetes	768	8	65.10	34.90
German	1000	24	70.00	30.00
Glass1	214	9	66.36	33.64
Glass2	214	9	64.49	35.51
Glass3	214	9	92.06	7.94
Glass4	214	9	93.93	6.07
Glass5	214	9	86.45	13.55
Hayes	132	4	50.39	49.61
Heart	303	13	54.13	45.87
Hepatitis	155	19	20.65	79.35
Housing	506	12	49.21	50.79
Ionosphere	351	34	64.10	35.90
Liver	345	6	57.97	42.03
Sonar	260	60	53.37	46.63
Thyroidsub	3772	21	7.53	92.47
Waveform1	900	21	66.67	33.33
Waveform2	900	21	66.67	33.33
Waveform3	900	21	66.67	33.33
Wine1	178	13	66.85	33.15
Wine2	178	13	60.11	39.89
Wine3	178	13	73.03	26.97

valued and the rest are nominal. The number of instances are 452 but the entries with missing values are removed so reducing the samples to 420, 237 belonging to the positive class and 183 to the negative class.

A.2.2 Australian Credit Approval (Australian)

This data set concerns credit card applications. All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data. The data are interesting because there is a good mix of attributes, i.e. we have continuous, nominal with small numbers of values, and nominal with larger numbers of values attributes. In particular, we have 14 attributes, six of which are numerical and eight categorical. There are also few missing values that have been discarded. The number of instances are 690 (307 for the positive class and 383 for the negative class).

A.2.3 Balance Scale Weight and Distance Database (Balance)

This data set was generated to model psychological experimental results. Each example is classified as having the balance scale tip to the right, tip to the left, or be balanced. The attributes are the left weight, the left distance, the right weight, and the right distance all numeric. The samples are 625 (49 for the balanced class, 288 for the left and 288 for the right). In our experiments the balanced class is the positive one and the other two classes are joined together as negative class.

A.2.4 Biomed Data Set (Biomed)

The aim of this data set is to develop screening methods to identify carriers of a rare genetic disorder. Because the disease is rare, there are only a few carriers of the disease from whom data are available. The data consists of 194 objects with 5 features categorized in 2 classes (67 samples for the positive class and 127 for the negative class). Entries with missing values have been removed.

A.2.5 Wisconsin Breast Cancer Database (Breast)

This database is concerned to diagnose the presence of a breast cancer. It was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. Each instance has 9 attributes and one of 2 possible classes: benign or malignant. There are 16 instances that contain a single missing attribute value that have been removed. The objects are 699, 458 for the benign class and 241 for the malignant.

A.2.6 Wisconsin Prognostic Breast Cancer (Cancer_wpbc)

Each record represents follow-up data for one breast cancer case. These are consecutive patients seen by Dr. Wolberg since 1984, and include only those cases exhibiting invasive breast cancer and no evidence of distant metastases at the time of diagnosis. There are 198 instances (151 non recur (negative class), 47 recur (positive class)) with 32 real valued input features. The first 30 features are computed from a digitized image of a fine needle aspirate of a breast mass. They describe characteristics of the cell nuclei present in the image. The mean, standard error, and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features.

A.2.7 Contraceptive Method Choice (CMC)

This data set is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples are married women who were either not pregnant or do not know if they were at the time of interview. The problem is to predict the current contraceptive method choice (between no use, long-term methods, or short-term methods) of a woman based on her demographic and social-economic characteristics. The instances are 1473 with 9 features. In our experiments the classes for long-term and short-term methods are joined in a negative class with 844 samples.

A.2.8 Pima Indians Diabetes Database (Diabetes)

The database is from the National Institute of Diabetes and Digestive and Kidney Diseases. The diagnostic, binary-valued variable investigated is whether the patient shows signs of diabetes according to World Health Organization criteria (i.e. if the 2 hour post-load plasma glucose was at least 200 mg/dl at any survey examination or if found during routine medical care). In particular, all patients here are females at least 21 years old of Pima Indian heritage. The objects are 768 (500 for the negative class and 268 for the positive class) with 8 features.

A.2.9 German Credit Data (German)

The original data set has been provided by Prof. Hofmann and contains both categorical and symbolic attributes. In our experiments we used the modified data set provided by the Strathclyde University in which different categorical attributes have been coded as integer. The aim of the data set is to evaluate the reliability of a client for the grant of a bank loan. The instances are 1000 (700 of bad clients and 300 for good clients) with 24 numerical features.

A.2.10 Glass Identification Database (Glass)

The study of classification of types of glass is motivated by criminological investigation since at the crime scene, correctly identified glass left can be used as evidence. The instances are 214 with 9 continuously valued features and 6 different possible classes (building float, building non float, vehicle float, containers, headlamps and tableware). To reduce it to several binary problems a One versus All approach has been considered, i.e. in each binary data set one class has been chosen as positive and all the others as negative so as to obtain a number of data sets equal to the class number. In our case we considered just

five data sets since the class tableware has only the 4% of the samples. Therefore, in the experiments the 5 data sets, Glass1, Glass2, Glass3, Glass4 and Glass5 consider respectively the classes building float, building non float, vehicle float, containers and headlamps as positive class with 70, 17, 76, 13 and 29 samples.

A.2.11 Hayes-Roth & Hayes-Roth Database (Hayes)

This data set has been developed for the human subjects classification and recognition performance according to four features: hobby, age, education and marital status. All features are nominal and numerically converted. The possible categories are three but some of the instances can be classified in both the first two classes and therefore, these classes have been joined. The number of instances is 132 with equal priors.

A.2.12 Heart Disease Cleveland Database (Heart)

The database has been provided by the Cleveland Clinic Foundation. This database contains 75 attributes, but all published experiments refer to a subset of 13 of them. The goal is to diagnose the presence of heart disease in the patient that can be of 5 different types. Experiments with the Cleveland database have concentrated on simply attempting to distinguish presence from absence of disease. Hence, the objects are 303 with 13 features in 2 classes (164 samples for the positive and 139 for the negative class).

A.2.13 Hepatitis Domain (Hepatitis)

This database has been used to predict if a patient will live or die for a hepatitis disease. Nineteen attributes (13 symbolic and 6 continuous) are considered for the samples belonging to two possible classes ("live" or "die"). The instances are 155 with priors equal to 20.65% and 79.35% for the positive and negative class respectively. Missing values have been substituted with the mean of the corresponding feature.

A.2.14 Boston Housing Data (Housing)

This data set was taken from the StatLib library which is maintained at Carnegie Mellon University. The aim is to evaluate the housing values in suburbs of Boston. In particular, 12 continuous features have been measured to evaluate the distribution around a median value of owner-occupied houses of 1000 dollars. The instances are 506, 249 for values greater than 1000 dollars and 257 for values lower than the median.

A.2.15 Johns Hopkins University Ionosphere Database (Ionosphere)

This radar data was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not, i.e. their signals pass through the ionosphere. Received signals were processed using an autocorrelation function whose arguments are the time of a pulse and the pulse number (there were 17 pulse numbers for the Goose Bay system). The instances (that are 351, 225 of which belonging to the positive class) in this database are described by 2 attributes per pulse number (34 continuous features in total), corresponding to the complex values returned by the function resulting from the complex electromagnetic signal.

A.2.16 BUPA Liver Disorders (Liver)

This data set has been provided by the BUPA Medical Research Ltd. and it is used to diagnose a liver disorder. The features are six in total. The first 5 are all blood tests which are thought to be sensitive to liver disorders that might arise from excessive alcohol consumption while the last one is the number of drinks (drinks > 5 is some sort of a selector on this database). Each sample constitutes the record of a single male individual. The instances are 345 with priors equal to 57.97% and 42.03% for the positive and the negative class respectively.

A.2.17 Sonar: Mines versus Rocks (Sonar)

This data set has been used in the study of the classification of sonar signals using a neural network. The task is to train a network to discriminate between sonar signals bounced off a metal cylinder and those bounced off a roughly cylindrical rock. The data set contains 111 patterns obtained by bouncing sonar signals off a metal cylinder (positive class) at various angles and under various conditions and 97 patterns obtained from rocks (negative class) under similar conditions. The transmitted sonar signal is a frequency-modulated chirp, rising in frequency. The data set contains signals obtained from a variety of different aspect angles, spanning 90 degrees for the cylinder and 180 degrees for the rock. The instances are 260 described by 60 numerical features.

A.2.18 Thyroid Gland Data (Thyroidsub)

The scope of this data set is to try to predict whether a patient has or not a disfunction of the thyroid. In particular, three classes can be defined according to euthyroidism, hypothyroidism or hyperthyroidism. The diagnosis (the class label) is based on a complete medical record, including anamnesis and scan. In our experiments the class hypothyroidism has been chosen as target class (positive class) and the others as negative class. The instances are 3772 (284 for the positive class) with 21 features.

A.2.19 Waveform Database Generator (Waveform)

The database has been provided by Breiman *et al.* (1984) in his book on the decision trees. Three different classes of waves have been considered with 21 attributes with continuous values between 0 and 6, all of which include gaussian additive noise with zero mean and unit standard deviation. The instances are 5000 and the classes are equally distributed. Since we are facing with two class problems, a One versus All approach has been applied (see subsection A.2.10) on a subset of 900 instances so as to obtain three different data sets: Waveform1, Waveform2 and Waveform3 all with 600 samples for the positive class and 300 for the negative class.

A.2.20 Wine Recognition Data (Wine)

These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determines the quantities of 13 constituents found in each of the three types of wines. In a classification context, this is a well posed problem with "well behaved" class structures. Therefore, it is a good data set for first testing of a new classifier, but not very challenging. The instances are 178 with 59, 71 and 48 samples per class. Also in this case a One versus All approach (see subsection A.2.10) has been applied and three different data sets have been generated Wine1, Wine2 and Wine3 with respectively 119, 107 and 130 samples for the positive class.

Appendix B

Notes on Statistical Tests

In this appendix a brief introduction to the characteristics of the statistical tests employed in he performed experiments is presented. The goal is to provide useful information for the analysis of the performed comparison and to justify the use of such procedures. For the description of the statistical test, we strictly follow the analysis proposed in Demšar (2006) adapting it to the performed experiments.

The appendix is organized into two sections. The former describes the Wilcoxon rank sum test used in chapter 4 for the comparison of the two methods employed to estimate the weight of the combination of two dichotomizers. The latter focuses on the test used for the comparison of different rankers (in chapter 3) or different combination rules (in chapter 4), i.e. the Friedman test and its post hoc tests. Tables for the critical values of each described statistic can be found in any statistical book, e.g. Sheskin (2000).

B.1 The Wilcoxon Rank Sum Test

The Wilcoxon signed-ranks test (Wilcoxon, 1945) is a non parametric test which ranks the differences in performances of two classifiers for each data set, ignoring the signs, and compares the ranks for the positive and the negative differences.

Let d_i be the difference between the performance scores (i.e. AUC in our case) of the two methods on the i-th run of the n cross validation procedure and let us rank these differences according to their absolute values (average ranks are assigned in case of ties). Let R_P be the sum of ranks for the runs of the cross validation on which the second algorithm outperformed the first, and R_N the sum of ranks for the opposite. Since ranks of $d = 0$ are split evenly among the sums (if there is an odd number of them, one is

ignored), we get:

$$R_P = \sum_{d_i>0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i),$$

$$R_N = \sum_{d_i<0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i).$$

Let T be the smaller of the sums, $T = \min(R_P, R_N)$. For a larger number of data, the statistics:

$$z = \frac{T - \frac{1}{2}n(n+1)}{\sqrt{\frac{1}{24}n(n+1)(2n+1)}}$$

is distributed approximately normally.

The Wilcoxon signed ranks test assumes commensurability of differences, but only qualitatively: greater differences still count more but the absolute magnitudes are ignored. From the statistical point of view, the test is safer since it does not assume normal distributions. Moreover, the outliers (exceptionally good/bad performances on a few runs) have less effect on the this test.

B.2 The Friedman Test

The Friedman test (Friedman, 1937), (Friedman, 1940) is a non parametric test that ranks the algorithms separately for each set[1], the best performing algorithm getting the rank of 1, the second best rank 2, etc. In case of ties average ranks are assigned (see table 3.8 for more clarity).

Let r_i^j be the rank of the j-th of k algorithm on the i-th of n sets. The Friedman test compares the average ranks of algorithms:

$$R_j = \frac{1}{n}\sum_{i=1}^{n} r_i^j.$$

Under the null hypothesis, which states that all the algorithms are equivalent and so their

[1] in our case, a set can be a single run of a cross validation procedure (as in chapter 3) or one of the performed combinations (as in the experiment in chapter 4)

ranks R_j should be equal, the Friedman statistic:

$$\chi_F^2 = \frac{12n}{k(k+1)} \left(\sum_{j=1}^{k} R_j^2 - \frac{k(k+1)^2}{4} \right)$$

is distributed according to a χ^2 distribution with $k-1$ degrees of freedom when n and k are big enough.

Iman & Davenport (1980) showed that the Friedman χ_F^2 is undesirably conservative and derived a better statistic:

$$F_F = \frac{(n-1)\chi_F^2}{n(k-1) - \chi_F^2}$$

which is distributed according to the F-distribution with $k-1$ and $(k-1)(n-1)$ degrees of freedom.

If the null hypothesis is rejected, it is possible to proceed with a post hoc test. The Nemenyi test (Nemenyi, 1963) is used when all methods are compared to each other. The performance of two methods is significantly different if the corresponding average ranks differ by at least the critical difference:

$$CD = q_{\alpha_{\text{ls}}} \sqrt{\frac{k(k+1)}{6n}}$$

where critical values $q_{\alpha_{\text{ls}}}$ for a level of significance equal to α_{ls} are based on the Studentized range statistic divided by $\sqrt{2}$.

When all methods are compared with a control one, instead of the Nemenyi test we can use one of the general procedures to control the family-wise error in multiple hypothesis testing, such as the Bonferroni-Dunn (Dunn, 1961) correction or similar procedures. The test statistics for comparing the u-th and v-th method using these methods is:

$$z = (R_u - R_v) \Big/ \sqrt{\frac{k(k+1)}{6n}}.$$

The z value is used to find the corresponding probability from the table of normal distribution, which is then compared with an appropriate α_{ls}. The comparison between the Nemenyi and Dunn test shows that the power of the post hoc test is much greater when all classifiers are compared only to a control classifier and not between themselves. Therefore, it should not be made any pairwise comparisons when we only want to test whether a newly proposed method is better than the existing ones.

For a contrast from the single step Bonferroni-Dunn procedure, step up and step down

procedures sequentially test the hypotheses ordered by their significance. Let us denote the ordered r values by $r_1 \leq r_2 \leq \ldots \leq r_{k-1}$. The simplest of these procedures are presented in Holm (1979) and Hochberg (1988). They both compare each r_i with $\alpha_{\rm ls}/(k-i)$, but differ in the order of the tests. Holm step down procedure starts with the most significant r value. If r_1 is below $\alpha_{\rm ls}/(k-1)$ (where $\alpha_{\rm ls}$ is the level of significance of the test), the corresponding hypothesis is rejected and we are allowed to compare r_2 with $\alpha_{\rm ls}/(k-2)$. If also the second hypothesis is rejected, the test proceeds with the third, and so on. As soon as a certain null hypothesis can not be rejected, all the remaining hypotheses are retained as well. Hochberg step up procedure works in the opposite direction, comparing the largest r value with $\alpha_{\rm ls}$, the next largest with $\alpha_{\rm ls}/2$ and so on until it encounters a hypothesis it can reject. All hypotheses with smaller r values are then rejected as well.

Another procedure is the Hommel procedure (Hommel, 1988) that is more complicated to compute; first, we need to find the largest j for which $r_{n-j+k} > k\alpha_{\rm ls}/j \ \forall k = 1, \ldots, j$. If no such j exists, we can reject all hypotheses, otherwise we reject all for which $r_i \leq \alpha_{\rm ls}/j$.

Holm procedure is more powerful than the Bonferroni-Dunn and makes no additional assumptions about the hypotheses tested. In turn, Hochberg and Hommel methods reject more hypotheses than Holms, therefore, under some circumstances they may exceed the family-wise error. More details for these procedures can be found in Shaffer (1995) or in different statistical books, e.g. Walpole *et al.* (1998), Sheskin (2000).

Appendix C

The Multilevel Coordinate Search Algorithm

In chapter 4 for the comparison of the DROC method we evaluated the weights of ht elinear combination of dichotomizers through an optimization algorithm called Multilevel Coordinate Search (MCS). Following the description in the original paper (Huyer & Neumaier, 1999), we propose a brief introduction to the algorithm with the only goal of rendering our work self-contained.

Let us consider a bound constrained optimization problem:

$$\min f(x) s.t. x \in [u, v]$$

with finite or infinite bounds, where we indicate a rectangular box with:

$$[u, v] = \left\{ x \in \mathbb{R}^n \middle| u_i \leq x_i \leq v_i, i = 1, \ldots, n \right\},$$

with u and v being $n$dimensional vectors with components in $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ and $u_i < v_i$ for $i = 1, \ldots, n$, i.e. only points with finite components are regarded as elements of a box $[u, v]$ whereas its bounds can be infinite. If all bounds are infinite an unconstrained optimization problem is obtained.

In MCS, the minimizer is found by splitting the search space into smaller boxes. Each box contains a distinguished point, the so called base point, whose function value is known. The partitioning procedure is not uniform but parts where low function values are expected to be found are preferred. The algorithm combines global search (splitting boxes with large unexplored territory) and local search (splitting boxes with good function values). The key to balancing global and local search is the multilevel approach. As measure of the

number of times a box has been processed, a level $s \in \{0, 1, \ldots, s_{\max}\}$ is assigned to each box. Boxes with level $s_{\max}$ are considered too small for further splitting; a level $s = 0$ indicates that a box has already been split and can be ignored. Whenever a box of level s $(0 < s < s_{\max})$ is split, its level is set to zero, and its descendants get level $s + 1$ or $\min(s + 2, s_{\max})$.

After an initialization procedure, the algorithm proceeds by a series of sweeps through the levels starting with the boxes at the lowest levels in each sweep constitutes the global part of the algorithm, and at each level the box with lowest function value is selected, which forms the local part of the algorithm.

The split is made along a single coordinate in each step: the base points of the descendants of a box are chosen such that they differ from the base point of the parent box in (at most) one coordinate. To split a box a single new function evaluation is needed and to determine the splitting coordinate and the position of the split the information gained from already sampled points is used.

MCS without local search puts the base points and function values of boxes of level s max into the socalled shopping basket (containing "useful" points). MCS with local search tries to accelerate convergence of the algorithm by starting local searches from these points before putting them into the shopping basket. The local search is performed only if the base point of a new box of level $s_{\max}$ is not in the basin of attraction of a local minimizer in the shopping basket. The local search algorithm used in our implementation of MCS essentially consists of building a local quadratic model by triple searches, then defining a promising search direction by minimizing the quadratic model on a suitable box and finally making a line search along this direction.

For more details on the implementation of the algorithm and more particular on the theoretical aspects we remand again at Huyer & Neumaier (1999).

References

Ataman, K., Street, N. & Zhang, Y. (2006). Learning to rank by maximizing auc with linear programming. *Proceedings of the IEEE International Joint Conference on Neural Networks*, 123. 32

Bamber, J.A. (1975). The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. *Journal of Mathematical Psychology*, 387–415. 20

Bennett, K. & Mangasarian, O. (1992). Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, **1**, 23. 32

Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press. 3

Blake, C., Keogh, E. & Merz, C.J. (1998). Uci repository of machine learning databases. [www.ics.uci.edu/∼mlearn/MLRepository.html]. 48, 79, 95, 97

Bradley, A.P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 1145–1159. 23, 26

Brefeld, U. & Scheffer, T. (2005). Auc maximizing support vector learning. *Proceedings of the 22nd International Conference on Machine Learning - Workshop on ROC Analysis in Machine Learning*. 32

Breiman, L. (1996). Bagging predictors. *Machine Learning*, **26**, 123–140. 55

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984). *Classification and Regression Trees*. Wadsworth International. 32, 54, 103

Clearwater, S. & Stern, E. (1991). A rule-learning program in high energy physics event classification. *Computer Physics Communication*, 159–182. 9

Cortes, C. & Mohri, M. (2003). Auc optimization vs. error rate minimization. *Advances in Neural Information Processing Systems (NIPS 2003)*. 26, 27

REFERENCES

Cristianini, N. & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines*. Cambridge University Press. 42

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 1–30. 49, 105

Devijver, P.A. & Kittler, J. (1982). *Pattern Recognition, a Statistical Approach*. Prentice Hall. 54

Dietrich, C., Palm, G. & Schwenker, F. (2003). Decision templates for the classification of bioacoustic time series. *Information Fusion*, **4**, 101. 56

Dietterich, T.G. (2001). Ensemble methods in machine learning. *In J. Kittler and F. Roli, eds, Multiple Classifier Systems*, **LNCS 1857, Springer-Verlag**, 1–15. 54

Duda, R.O., Hart, P.E. & Stork, D.G. (2001). *Pattern Classification*. John Wiley & Sons, 2nd edn. 4, 44, 79

Duin, R.P.W. (2000). Prtools version 3.0, a matlab toolbox for pattern recognition. [http://www.prtools.org]. 44, 95

Duin, R.P.W. (2002). The combining classifier: to train or not to train?. *Proceedings of the 16th IEEE International Conference on Pattern Recognition*, 765. 56

Dunn, O.J. (1961). Multiple comparisons among means. *Journal of the American Statistical Association*, **56**, 52–64. 107

Egan, J.P. (1975). *Signal Detection Theory*. Series in Cognition and Perception, Academic Press. 11, 19

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, **27**, 861–874. 9, 11, 16, 17, 20

Fawcett, T. & Provost, F. (1996). Combining data mining and machine learning for effective user profiling. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, **42**, 8–13. 9

Fawcett, T. & Provost, F. (1997). Adaptive fraud detection. *Data Mining and Knowledge Discovery*, **1**, 291–316. 9

Ferri, C., Flach, P. & Hernandez-Orallo, J. (2002). Learning decision trees using the area under the roc curve. *Proceedings of 19th International Conference on Machine Learning*. 32

Fisher, R.A. (1959). *Statistical Methods and Scientific Inference*. Hafner Publishing Co., 2nd edn. 48

Flake, G.W. & Pearlmuter, B.A. (2000). Differentiating functions of the jacobian with respect to the weights. *In Solla et al., eds., Advances in Neural Information Processing Systems*, **12**. 79

Freund, Y. & Schapire, R.E. (1997). A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences*, **55**, 119. 30, 44, 55, 84

Freund, Y., Iyer, R., Schapire, R.E. & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, **4**, 933. 30, 42

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, **32**, 675–701. 48, 86, 106

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics*, **11**, 86–92. 48, 106

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. Academic Press, 2nd edn. 2

Fumera, G. & Roli, F. (2005). A theoretical and experimental analysis of linear combiners for multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**, 942. 58, 59

Garthwaite, P.H., Jolliffe, I.T. & Jones, B. (2002). *Statistical Inference*. Oxford University Press, 2nd edn. 15

Goddard, M.J. & Hinberg, I. (1990). Receiver operating characteristic (roc) curves and non-normal data: an empirical study. *Statistics in Medicine*, 325–337. 16, 18

Green, D. & Swets, J. (1966). *Signal Detection: Theory and Psychophysics*. John Wiley & Sons. 13

Hajian-Tilaki, K.O., Hanley, J.A., Joseph, L. & Collet, J.P. (1996). A comparison of parametric and nonparametric approaches to roc analysis of quantitative diagnostic tests. *Medical Decision Making*, 94–102. 19

HAMILTON, L.C. (1990). *Modern Data Analysis: a First Course in Applied Statistics*. Wadsworth. 49

HAND, D.J. & TILL, R.J. (2001). A simple generalization of the area under the roc curve to multiple class classification problems. *Machine Learning*, 171–186. 23

HAND, D.J., ADAMS, N.M. & KELLY, M.G. (2001). Multiple classifier systems based on interpretable linear classifiers. *In J. Kittler and F. Roli, eds, Multiple Classifier Systems*, **LNCS1857, Springer-Verlag**, 136–147. 54

HANLEY, J.A. (1988). The robustness of the "binormal" assumptions used in fitting roc curves. *Medical Decision Making*, 197–203. 19

HANLEY, J.A. & MCNEIL, B.J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 29–36. 24

HERSCHTAL, A. & RASKUTTI, B. (2004). Optimising area under the roc curve using gradient descent. *Proceedings of the 21st International Conference on Machine Learning*. 24, 30

HO, T.K. (2002). Multiple classifier combination: Lessons and the next steps. *In A. Kandel and H. Bunke, eds., Hybrid Methods in Pattern Recognition*, **World Scientific Publishing**, 171–198. 54

HO, T.K., HULL, J.J. & SRIHARI., S.N. (1994). Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **16**, 66–75. 56

HOCHBERG, Y. (1988). A sharper bonferroni procedure for multiple tests of significance. *Biometrika*, **75**, 800. 108

HOLM, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, **6**, 65–70. 50, 86, 108

HOMMEL, G. (1988). A stagewise rejective multiple test procedure based on a modified bonferroni test. *Biometrika*, **75**, 383. 108

HUANG, J. & LING, C.X. (2005). Using auc and accuracy in evaluating learning algorithms. *IEEE Transaction on Knowledge Data Engineering*, **17**, 299–310. 9, 23, 26

HUYER, W.A. & NEUMAIER, A. (1999). Global optimization by multilevel coordinate search. *Journal of Global Optimization*, **14**, 331. 86, 109, 110

Iman, R.L. & Davenport, J.M. (1980). Approximations of the critical region of the friedman statistic. *Communications in Statistics*, 571. 107

Joachims, T. (1999). Making large-scale svm learning practical. *In Schölkopf et al., eds., Advances in Kernel Methods-Support Vector Learning*, 169. 79

Kanungo, T. & Haralick, R.M. (1995). Receiver operating characteristic curves and optimal bayesian operating points. *Proceedings of the IEEE International Conference on Image Processing*, 256–259. 13

Kittler, J., Hatef, M., Duin, R.P.W. & Matas, J. (1998). On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, 226. 57

Kubat, M., Holte, R.C. & Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 195–215. 9

Kuncheva, L.I. (2002). A theoretical study on six classifier fusion strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, 281. 57

Kuncheva, L.I. (2004). *Combining Pattern Classifiers. Methods and Algorithms*. John Wiley & Sons. 54, 56

Kuncheva, L.I. (2005). Diversity in multiple classifier systems (editorial). *Information Fusion*, **6**, 3. 54, 73

Kuncheva, L.I. & Whitaker, C.J. (2003). Measures of diversity in classifier ensembles. *Machine Learning*, **51**, 181. 48, 73, 74

Kuncheva, L.I., Whitaker, C.J., Shipp, C.A. & Duin, R.P.W. (2000). Is independence good for combining classifiers?. *Proceedings of the 15th IEEE International Conference on Pattern Recognition*, 168. 42

Lehmann, E.L. & D'Abrera, H.J.M. (1975). *Nonparametrics. Statistical Methods based on Ranks*. McGraw Hill International Book Company. 26, 42

Ling, C. & Li, C. (1998). Data mining for direct marketing-specific problems and solutions. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 73–79. 27

Liu, A., Schisterman, E.F. & Zhu, Y. (2005). On linear combinations of biomarkers to improve diagnostic accuracy. *Statistics in Medicine*, **24**, 37. 59

REFERENCES

Mann, H.B. & Whitney, D.R. (1947). On a test whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statististic*, **18**, 50. 24

Marrocco, C., Molinara, M. & Tortorella, F. (2005a). Estimating the roc curve of linearly combined dichotomizers. *In F. Roli and S. Vitulano, eds, Image Analysis and Processing*, **LNCS3617, Springer-Verlag**, 778–785. 59

Marrocco, C., Molinara, M. & Tortorella, F. (2005b). Optimal linear combination of dichotomizers via auc. *Proceedings of the* 22^{nd} *International Conference on Machine Learning 2005-Workshop on ROC Analysis in Machine Learning*, 778–785. 59

Marrocco, C., Molinara, M. & Tortorella, F. (2006a). Auc-based linear combination of dichotomizers. *In D. Yeung et al., eds, Advances in Statistical, Structural and Syntactical Pattern Recognition*, **LNCS4109, Springer-Verlag**, 714–722. 59

Marrocco, C., Molinara, M. & Tortorella, F. (2006b). Exploiting auc for optimal linear combination of dichotomizers. *Pattern Recognition Letters*, **27**, 900–907. 59

Metz, C.E. (1986). Roc methodology in radiologic imaging. *Investigative Radiology*, 720–733. 11

Metz, C.E., Herman, B.A. & Shen, J.H. (1998). Maximum-likelihood estimation of receiver operating characteristic (roc) curves from continuously-distributed data. *Statistics in Medicine*, 1033–1053. 16, 19

Mukhopadhyay, N. (2000). *Probability and Statistical Inference*. Marcel Dekker Inc. 15

Nemenyi, P.B. (1963). *Distribution-free multiple comparisons*. Ph.D. thesis, Princeton University. 107

Neyman, J. & Pearson, E.S. (1933). The testing of statistical hypotheses in relation to probabilities a priori. *Proceedings of the Cambridge Philosophical Society*, **24**, 492–510. 15

Oza, N.C., Polikar, R., Kittler, J. & Roli, F. (2005). *Multiple Classifiers Systems*, vol. LNCS3541 of *Lecture Notes in Computer Science*. Springer-Verlag. 54

Pepe, M. (2003). *The Statistical Evaluation of Medical Tests for Classification and Prediction*. Oxford University Press. 24

PROVOST, F. & FAWCETT, T. (1998). Robust classification systems for imprecise environments. *Proceedings of the 15th American National Conference on Artificial Intelligence (AAAI98)*, 706–713. 9

PROVOST, F. & FAWCETT, T. (2001). Robust classification for imprecise environments. *Machine Learning*, **42**, 203–231. 10, 14, 23, 69

RAKOTOMAMONJY, A. (2004). Optimizing area under roc curve with svms. *Proceedings of the Workshop on ROC Analysis and Artificial Intelligence*. 32

RUDIN, C., CORTES, C., MOHRI, M. & SCHAPIRE, R. (2005). Margin-based ranking meets boosting in the middle. *Proceedings of 18th Annual Conference on Computational Learning Theory*. 31

SAITTA, L. & NERI, F. (1998). Learning in the "real world". *Machine Learning*, 133–163. 9

SHAFFER, J.P. (1995). Multiple hypothesis testing. *Annual Review of Psychology*, **46**, 561. 108

SHESKIN, D.J. (2000). *Handbook of parametric and nonparametric statistical procedures*. Chapman & Hall, CRC. 105, 108

SING, T. (2004). *Learning Localized Rule Mixtures by Maximizing the Area under the ROC Curve, with an Application to the Prediction of HIV-1 Coreceptor Usage*. Masters thesis, Max Planck Institute for Informatics. 24

SKALAK, D.B. (1996). The sources of increased accuracy for two proposed boosting algorithms. *Proceedings of the 13th American National Conference on Artificial Intelligence (AAAI96), Integrating Multiple Learned Models Workshop*. 74

SPACKMAN, K.A. (1989). Signal detection theory: Valuable tools for evaluating inductive learning. *Proceedings of the Sixth International Workshop on Machine Learning*, 160–163. 11

SU, J.Q. & LIU, J.S. (1993). Linear combinations of multiple diagnostic markers. *Journal of the American Statistical Association*, **88**, 1350. 59

SWETS, J.A. (1986). Form of empirical rocs in discrimination and diagnostic tasks: Implications for theory and measurement of performance. *Psychological Bulletin*, 181–198. 19

Swets, J.A. (1988). Measuring the accuracy of diagnostic systems. *Science*, 1285–1293. 9, 11

Swets, J.A., Tanner, W.P.J. & Birdsall, T.G. (1961). Decision processes in perception. *Psychological Review*, 301–340. 19

Swets, J.A., Dawes, R.M. & Monahan, J. (2000). Better decisions through science. *Scientific American*, 82–87. 11

Tax, D.J.M., Duin, R.P.W. & Arzhaeva, Y. (2006). Linear model combining by optimizing the area under the roc curve. *Proceedings of the 18th IEEE International Conference on Pattern Recognition*, 119. 32, 42, 46

Tax, D.M.J., van Breukelen, M., Duin, R.P.W. & Kittler, J. (2000). Combining multiple classifiers by averaging or by multiplying. *Pattern Recognition*, **33**, 1475. 57

Tubbs, J.D. & Alltop, W.O. (1991). Measures of confidence associated with combining classification rules. *IEEE Transactions on Systems, Man and Cybernetics*, **21**, 690–692. 56

Tumer, K. & Ghosh, J. (1996). Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognition*, **29**, 341. 58

Tumer, K. & Ghosh, J. (1999). Linear and order statistics combiners for pattern classification. *Combining Artificial Neural Nets, in A.J.C. Sharkey ed.*, 127. 57

Ueda, N. (2000). Optimal linear combination of neural networks for improving classification performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**, 207. 58

van der Heijden, F., Duin, R.P.W., de Ridder, D. & Tax, D.M.J. (2004). *Classification, Parameter Estimation and State Estimation - an Engineering Approach Using Matlab*. John Wiley & Sons. 44

Van Trees, H.L. (2001). *Detection, Estimation and Modulation Theory. Part I*. John Wiley & Sons. 13

Vapnik, V.N. (1998). *Statistical Learning Theory*. John Wiley & Sons. 32, 42, 46

Verikas, A., Lipnickas, A., Malmqvist, K., Bacauskiene, M. & Gelzinis, A. (1999). Soft combination of neural classifiers: A comparative study. *Pattern Recognition Letters*, **20**, 429. 59

Walpole, R.E., Myers, R.H. & Myers, S.L. (1998). *Probability and Statistics for Engineers and Scientists*. Prentice Hall Int., 6th edn. 80, 108

Webb, A. (2002). *Statistical Pattern Recognition*. John Wiley & Sons, 2nd edn. 2, 4, 14, 30, 53, 55

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics*, **1**, 80. 80, 105

Wolpert, D.H. (1992). Stacked generalization. *Neural Networks*, **5**, 241. 56

Woods, K., Kegelmeyer, W.P. & Bowyer, K. (1997). Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**, 405–410. 54

Xu, L., Krzyzak, A. & Suen, C.Y. (1992). Methods of combining multiple classifiers and their application to handwriting recognition. *IEEE Transactions on Systems, Man and Cybernetics*, **22**, 418–435. 56

Yan, L., Dodier, R., Mozer, M.C. & Wolniewicz, R. (2003). Optimizing classifier performance via the wilcoxon-mann-whitney statistics. *Proceedings of the Twentieth International Conference on Machine Learning*, 848–855. 24

Zweig, M.H. & Campbell, G.S. (1993). Receiver-operating characteristic plots: A fundamental evaluation tool in clinical medicine. *Clinical Chemistry*, **39**, 561–577. 14