Università degli Studi di Cagliari

Dottorato di Ricerca in

Ingegneria Elettronica e Informatica

XIV Ciclo

ADVANCED METHODS FOR PATTERN RECOGNITION WITH REJECT OPTION

Relatore

Prof. Ing. Fabio ROLI

Tesi di Dottorato di

Dott. Ing. Giorgio FUMERA

Coordinatore

Prof. Ing. Massimo VANZI

Novembre 2001

Contents

Chapter 1
Introduction

In statistical pattern recognition the reject option has been introduced to safeguard against excessive misclassifications, thus improving classification reliability. It consists in withholding to automatically classify an input pattern if a wrong classification is more likely than a correct one. Rejected patterns must then be handled in a different way, for instance they can be classified by a human operator, or by using a more complex classifier as proposed by Pudil et al (1992). Obviously, rejects have a cost, as well as misclassifications, due to their exceptional handling. This means that the reject option is useful in applications for which a misclassification is more costly than a reject. An example is the classification of medical images. Moreover, also patterns that would have been correctly classified can be rejected. Therefore a suitable trade-off between misclassifications and rejects depends on their relative costs. In the framework of the minimum risk theory, the optimal classification rule with reject option was defined by Chow (1957; 1970). In the simplest case in which the costs of misclassifications and of rejects do not depend on the classes, Chow's rule consists in rejecting an input pattern if the maximum of its a posteriori probabilities is lower than a predefined threshold, whose value depends on the classification costs. The maximum of the a posteriori probabilities can therefore be considered as the measure of classification reliability. This means that the optimality of Chow's rule, analogously to Bayes rule for classification without reject option, relies on the exact knowledge of the a posteriori probabilities. However it is well-known that in real applications the a posteriori probabilities are usually unknown, and can only be approximated by some kinds of classifiers, like neural networks.

From the above discussion it is evident that Chow's rule does not allow to reach the optimal error-reject trade-off when applied on estimates of the a posteriori probabilities. Moreover, some classifiers, like support vector machines, do not even provide approximations of the a posteriori probabilities. In this case the classification reliability must be estimated on the basis of the specific classifier used. However, as pointed out by Hansen et al. (1997) and by De Stefano et al. (2000), little attention was devoted in the literature to the problem of characterising the error-reject trade-off achievable by a real classifier, and of defining rejection rules targeted to specific classifiers. For classifiers which provide approximations of the a posteriori probabilities, Chow's rule is commonly used despite its non-optimality. Some works proposed different rejection rules for neural network classifiers (Le Cun et al., 1990; Cordella et al., 1995; De Stefano et al., 2000), and for multiple classifier systems (Foggia et al., 1999; Sansone et al., 2001). These rules are based on evaluating the classification reliability by using not only the highest value of the estimated a posteriori probabilities (as in Chow's rule), but also the other values. However the effectiveness of these rules with respect to Chow's rule was not theoretically proven. Also for classifiers which do not provide approximations of the a posteriori probabilities, the rejection

rules proposed in the literature are based on simple heuristics rather than theoretical bases. A quite surprising example are support vector machine (SVM) classifiers. This new kind of classifier has been recently introduced by V. Vapnik on the basis of the statistical learning theory (Vapnik, 1998), and exhibited interesting advantages over traditional classifiers. However the theoretical derivation of SVMs did not take into account the reject option. Moreover, few works in the literature considered the problem of introducing the reject option in SVMs. Only a simple heuristic rule is available at present, based on applying a reject threshold on the output of a trained SVM.

In this thesis we address the two main topics discussed above. We consider first the problem of the error-reject trade-off achievable by classifiers which provide approximations of the a posteriori probabilities (Chapters 4 and 5). In particular, in Chapter 4 we analyse the effects of estimation errors on the performance of Chow's rule. On the basis of this analysis, we propose a new rejection rule based on a different reject threshold for each class. We formally prove that this rule allows to achieve a better error-reject trade-off than Chow's rule, in presence of estimation errors on the a posteriori probabilities. In Chapter 5 we analyse how the effects of the estimation errors on the error-reject trade-off can be reduced by classifier combination. We focus on simple and widely used combining rules based on linearly combining classifiers in output space, namely, simple and weighted average. To this aim, we extend a theoretical framework proposed by Tumer and Ghosh (1999) for the simple average combining rule without reject option. Then, in Chapter 6 we address the problem of introducing the reject option in SVMs. As pointed out above, Chow's rule is not applicable to SVMs, since they do not provide approximations of the a posteriori probabilities. We propose a method for introducing the reject option, based on the approach followed by Vapnik to derive SVMs from statistical learning theory (Vapnik, 1998). In Chapter 7 we present experiments aimed at evaluating the effectiveness of the methods proposed in Chapters 4 and 6.

Chapter 2
Classification with reject option in statistical pattern recognition

In this Chapter we present the problem of classification with reject option in statistical pattern recognition. We first focus on the theoretical setting of this problem in the framework of the minimum risk theory. This leads to the definition of the optimal classification rule with reject option. We then discuss the implementation of the reject option in real classifiers.

2.1 The reject option in the framework of the minimum risk theory

In statistical pattern recognition a pattern is represented by a d-dimensional feature vector $\mathbf{x} = (x_1, x_2, ..., x_d) \in D \subseteq \Re^d$, where D is the *feature space*. Patterns are assumed to be random observations, independent and identically distributed according to a probability density function $p(\mathbf{x})$. In a supervised classification problem, each pattern belongs to one of c predefined classes $\omega_1, \omega_2, ..., \omega_c$ according to a conditional probability function $P(\omega \mid \mathbf{x})$, which is called *a posteriori probability*. The goal of a classifier is to construct a *decision rule* $f(\mathbf{x})$ to assign any given input pattern $\mathbf{x}$ to one of the c classes. The decision rule $f(\mathbf{x})$ subdivides the feature space D in c disjoint subsets $D_1, ..., D_c$, named *decision regions*, so that input patterns belonging to the i-th subset are assigned to the i-th class. The decision rule is constructed according to a given performance criterion, which depends on the particular application. In the framework of the minimum risk theory, the performance criterion is defined by means of a *loss function* $L(\mathbf{x},\omega,f(\mathbf{x}))$, which represents the loss due to classifying a given pattern $\mathbf{x}$, belonging to class ω, using the decision rule $f(\mathbf{x})$. The goal is to construct a decision rule which minimises the expected value of the loss with respect to $\mathbf{x}$ and ω (*expected risk*):

$$R\left(f(\mathbf{x})\right) = \sum_{i=1}^{c} \int_D L\left(\mathbf{x},\omega_i,\phi(\mathbf{x})\right) p(\mathbf{x},\omega_i)\, d\mathbf{x} \ .$$

Usually loss functions do not depend on the particular pattern $\mathbf{x}$, but only on its true class and on the class it is assigned to by $f(\mathbf{x})$. They are therefore defined by constants $L(\omega_i,\omega_j)$, $i, j = 1, ..., c$, which represent the loss incurred in deciding ω_j when the true class is ω_i. In this case, the decision rule which minimises the expected risk is the well known Bayes rule (Duda et al., 2001), which consists in assigning a pattern $\mathbf{x}$ to the class ω_i, $i = 1, ..., c$, for which the conditional risk

$$R(\omega_i \mid \mathbf{x}) = \sum_{j=1}^{c} L(\omega_j,\omega_i) P(\omega_j \mid \mathbf{x}) \qquad (1)$$

is minimum. The simplest loss function represents the case in which the costs of misclassifications and correct classifications do not depend on the classes:

$$L\left(\mathbf{x},\omega, f(\mathbf{x})\right) = \begin{cases} w_C, & \text{if } f = \omega \ , \\ w_E, & \text{if } f \neq \omega \ . \end{cases} \qquad (2)$$

Obviously, $w_C < w_E$. If $w_C = 0$, and $w_E = 1$, the expected risk $R(f(\mathbf{x}))$ is equal to the probability of misclassifying a pattern (error probability), which can be written as

$$P(error) = \sum_{i=1}^{c} \int_{D_i} \sum_{\substack{j=1 \\ j \neq i}}^{c} P(\omega_j | \mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^{c} \int_{D_i} \sum_{\substack{j=1 \\ j \neq i}}^{c} P(\mathbf{x} | \omega_j) p(\omega_j) d\mathbf{x} \ . \quad (3)$$

Obviously, the probability of correct classification is $P(correct) = 1 - P(error)$, and can be expressed as:

$$P(correct) = \sum_{i=1}^{c} \int_{D_i} P(\omega_i | \mathbf{x}) p(\mathbf{x}) d\mathbf{x} \ . \quad (4)$$

In this case the Bayes rule consists in assigning a pattern $\mathbf{x}$ to the class ω_i having the maximum a posteriori probability (MAP rule):

$$P(\omega_i | \mathbf{x}) = \max_{j=1,\ldots,c} P(\omega_j | \mathbf{x}) \ . \quad (5)$$

The error probability achieved by Bayes rule is named *Bayes error*. It is by definition the lower bound of the error probability achievable in a pattern recognition problem, for a given feature set.

To achieve a lower expected risk, or a lower error probability, than the ones given by Bayes rule, the so-called *reject option* must be used. The reject option consists in avoiding to automatically classifying patterns for which the classification is not sufficiently reliable. Rejected patterns must then be handled with a different procedure, for instance they can be manually classified. Obviously, rejects have a cost, as well as misclassifications, due to their exceptional handling. This implies that the reject option is useful only if the cost of rejects is lower than the cost of misclassifications. Moreover, it is not possible to turn all misclassifications into rejects, but also patterns that would have been correctly classified could be rejected (Chow, 1970). Therefore, when using the reject option, a suitable trade-off between errors and rejects must be found, depending on their relative costs. For instance, as pointed out by Cordella et al. (1995), in medical applications the cost of a misclassification could be very expensive, making a high reject rate acceptable to keep the misclassification rate as low as possible. Instead, in OCR applications in which the resulting text has to be successively edited by hand, a high misclassification rate can be acceptable. For instance, a practical scheme of a pattern recognition system with reject option was proposed by Pudil et al. (1992), for cases in which the rejection is not acceptable as a final result. They proposed a multistage classifier in which patterns rejected at any stage, but the highest one, are processed by the next stage. In this scheme, each stage utilises more informative, but more costly, measurements. This can be the case of a medical diagnosis application. At the final stage a decision is taken in any case, so eventually no rejects remain.

Classification with reject option has been formalised under the framework of the minimum risk theory by Chow (1957). The decision of a classifier to reject a pattern $\mathbf{x}$ can be represented by introducing a fictious class ω_0 which rejected patterns are assigned to. Accordingly, a classifier with reject option subdivides the feature space D into $c + 1$ disjoint subsets D_0, D_1, ..., D_c, where D_0 denotes the *reject region*. The loss function (2) can be modified to take into account the cost of

rejects, by introducing costants $L(\omega_i,\omega_0)$, $i = 1, ..., c$, which represent the cost of rejecting a pattern belonging to class ω_i. The classification rule which achieves the optimal error-reject trade-off is the one which minimises the expected risk. Chow (1957) proved that this rule consists in assigning a pattern $\mathbf{x}$ to the class ω_i, $i = 0, ..., c$, for which the conditional risk (1) is minimum. Note that Chow's rule is the analogous of Bayes rule for classification with reject option. The simplest loss function in classification with reject option (the analogous to (2)) is:

$$L\left(\mathbf{x},\omega,f(\mathbf{x})\right) = \begin{cases} w_C, & \text{if } f = \omega, \\ w_R, & \text{if } f = \omega_0, \\ w_E, & \text{if } f \neq \omega,\ f \neq \omega_0, \end{cases} \tag{6}$$

where w_R denotes the cost of a reject. Obviously, $w_C < w_R < w_E$. The corresponding expected risk can be written as

$$w_C P(correct) + w_R P(reject) + w_E P(error), \tag{7}$$

where $P(reject)$ denotes the probability that a pattern is rejected, and can be expressed as

$$P(reject) = \int_{D_0} p(\mathbf{x})\, d\mathbf{x}. \tag{8}$$

In this case Chow's rule consists in assigning a pattern $\mathbf{x}$ to the class ω_i having the maximum a posteriori probability, if it is higher than a reject threshold T:

$$\max_j P(\omega_j|\,\mathbf{x}) = P(\omega_i|\,\mathbf{x}) \geq T, \tag{9}$$

where

$$T = \frac{w_E - w_R}{w_E - w_C}. \tag{10}$$

The patten $\mathbf{x}$ is rejected if $P(\omega_i\,|\,\mathbf{x}) < T$ (Chow, 1970). Obviously $0 \leq T \leq 1$. Note that, since the minimum value of the maximum a posteriori probability is $1/c$, for $0 \leq T \leq 1/c$ no pattern is rejected.

Using Chow's rule, the error probability (3) and the reject probability (8) can be viewed as functions of the threshold T, and can be denoted respectively as $E(T)$ and $R(T)$. Chow (1970) proved that $E(T)$ is an increasing function of T, while $R(T)$ is a decreasing function of T. Since they are both monotonic functions, the error-reject trade-off of a classifier can be described by the functional relation between E and R, for varying values of T. This relation can be given in differential form:

$$\frac{dE}{dR} = T - 1 \leq 0. \tag{11}$$

This means that E and R describe a curve whose intial slope is -1 (for $T = 0$), while the final slope is 0 (for $T = 1$). From Eq. (11) it follows that:

$$\frac{d^2E}{dR^2} = \frac{dT}{dR} \geq 0,$$

that is, the optimal error-reject curve is always convex. In particular, when $R = 0$, E is equal to Bayes error. Furthermore, E decreases to 0 as R increases from 0 to 1. The behaviour of the optimal error-reject curve is shown in Fig. 1. As said above, the optimal value of the reject

threshold T, and terefore of E and R, depend on the classification costs (10). Chow (1970) also proved that the optimal error-reject curve represents the minimum achievable error probability for any value of the reject probability, and vice-versa. This can be viewed as an alternative definition of optimal error-reject trade-off, when the loss function (6) is considered.

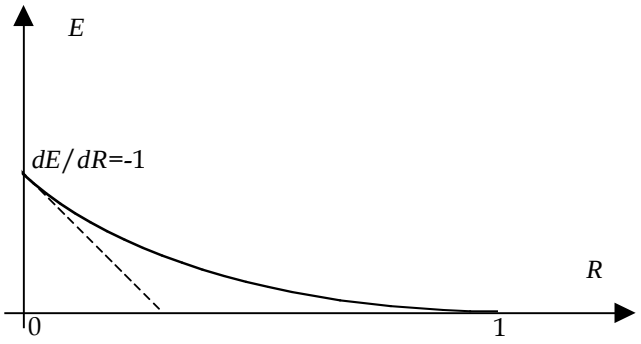


Fig. 1. The optimal error-reject curve.

From the above discussion, it is easy to see that in statistical pattern recognition the optimality of a classification rule relies on the exact knowledge of the a posteriori probabilities for the task at hand. However, in practical applications all the probability functions are usually unknown (Fukunaga, 1990). This implies that it is not possible to design an optimal classifier. In particular, this means that a real classifier will always achieve a greater error probability than Bayes error. Furthemore, both Bayes error and the actual error probability achieved by a real classifier are unknown, since they depend on the unknown probability functions (see Eq. (3)). Analogously, in classification with reject option, a real classifier can not achieve the optimal error-reject trade-off, and both the optimal and the actual error-reject curves are unknown. In this context, the reject option can not guarantee a lower error probability than Bayes error. Nonetheless, it is still useful to safeguard against excessive misclassifications, as pointed out by Chow (1970) and Pudil et al. (1992).

2.2 Class-selective rejection and distance rejection

The theoretical setting of classification with reject option described above is based on decision rules which either reject a pattern, or assign it to one of the predefined classes, with the aim of optimising the trade-off between errors and rejects. A different kind of decision rule, the so-called *class-selective* rejection, was considered by Ha (1997). Instead of simply rejecting a pattern which can not be reliably assigned to one of the c predefined classes, the pattern can be assigned to a non-empty subset of classes which most likely it belongs to. In this view, the pattern is rejected from the remaining classes. A class-selective decision rule $f(\mathbf{x})$ subdivides the feature space D in 2^c-1 decision regions, each one associated to one of the possible 2^c-1 non-empty subsets of the c classes. In this context, the optimality criterion was defined by Ha as the best trade-off between the error probability and the average number of selected classes. The loss function was defined as $L(\mathbf{x},\omega,f(\mathbf{x})) = L_m(\mathbf{x},\omega,f(\mathbf{x})) + L_n(f(\mathbf{x}))$, where L_m denotes the loss due to assigning a pattern belonging

to class ω to the subset of classes given by $f(\mathbf{x})$, and L_n denotes the loss due to having to deal with the number of these classes. Denoting with D_j, $j \in \{1,\ldots, 2^c\text{-}1\}$, the subset given by $f(\mathbf{x})$, the two parts of the loss function were defined as follows:

$$L_m\left(\mathbf{x}, \omega, f(\mathbf{x})\right) = \begin{cases} 0, & \text{if } \omega \in D_j, \\ w_E, & \text{if } \omega \notin D_j, \end{cases}$$

$$L_m\left(\mathbf{x}, \omega, f(\mathbf{x})\right) = w_n |D_j|,$$

where $|D_j|$ denotes the cardinality of the set D_j. The corresponding expected risk is:

$$w_E P(error) + w_n \bar{n},$$

where $P(error)$ is the probability that a pattern does not belong to any of the classes given by $\phi(\mathbf{x})$, and $\bar{n}$ denotes the average number of classes which a pattern is assigned to. Ha proved that the optimal decison rule with class-selective rejection consists in assigning a pattern $\mathbf{x}$ to all classes whose posterior probability exceeds a reject threshold $t = w_n / w_E$. If the maximum a posteriori probability is lower than t, the pattern $\mathbf{x}$ has to be assigned only to the corresponding class.

Horiuchi (1998) defined a different optimality criterion for class-selective rejection. The aim was to avoid cases in which the minimum distance between the a posteriori probabilities of selected and rejected classes is lower than the maximum distance between the a posteriori probabilities of selected classes. The optimality criterion was defined as the best trade-off between the number of selected classes and the maximum distance between the a posteriori probability of selected classes. The corresponding optimal decision rule consists in selecting all the classes for which the minimum distance between their a posteriori probabilities is lower than a threshold s, with $0 \leq s \leq 1$.

Dubuisson (1990) and Dubuisson and Masson (1993) proposed a rejection rule to deal with incomplete knowledge about classes. For instance, in applications like diagnostic problems the number of classes can be not known a priori, or it can be not possible to obtain training patterns from some classes. In these cases it could be desirable to reject patterns of unknown classes instead of assigning them to one of the known classes. Dubuisson argues that Chow's rule is not suitable to this kind of applications, since it can only solve *uncertainty* problems. The uncertainty arises when a pattern can be assigned to more than one of the known classes, but can not be reliably assigned to only one of them. For this reason the reject option dealt with by Chow's rule is called *ambiguity* rejection. To deal with the problem of incomplete knowledge about classes, Chow's rule was extended by rejecting also patterns which lie "far" from known classes. More precisely, a pattern $\mathbf{x}$ is distance-rejected if $p(\mathbf{x}) < C_d$, where $p(\mathbf{x})$ is the probability density function of the feature vector, with respect only to known classes, and C_d is the so-called distance reject threshold. This was called *distance* rejection, to distinguish it from ambiguity rejection. A modified version of this rule was proposed by Muzzolini et al. (1998). They used a different distance reject threshold for each class-conditional probability density function $p(\mathbf{x} \mid \omega_i)$, to balance the probability of distance rejection for patterns of different classes. We point out that, unlike Chow's rule and the class-selective rejection rule, the above distance rejection rules are not based on the definition of an optimality criterion.

In the rest of this work we will only consider the reject option as defined in paragraph 2.1.

2.3 Evaluation of classification reliability in real classifiers

Even if in practical applications the a posteriori probabilities are unknown, every type of classifier allows to evaluate a measure of the degree of certainty of the classification, as pointed out by Hansen et al. (1997). Rejection rules used in real classifiers are mainly based on obtaining an estimate of the a posteriori probabilities, to which Chow's rule is usually applied, despite its non-optimality. In this paragraph we review how the reject option is implemented in well-known classifiers.

Several classifiers provide approximations of the a posteriori probabilities in statistical sense. This justify the use of Chow's rule to implement the reject option, as far as the estimates are close to the true a posteriori probabilities (this point will be discussed in more detail in Chapters 4 and 5). For instance, under certain hypotheses, parametric classifiers approximate the class-conditional probability densities $p(\mathbf{x} \mid \omega)$ (Fukunaga, 1990). Estimates of the a posteriori probabilities can then be obtained using the well-known Bayes formula $P(\omega \mid \mathbf{x}) = p(\mathbf{x} \mid \omega) P(\omega) / p(\mathbf{x})$, where $p(\mathbf{x})$ is obtained as $\sum_{i=1}^{c} p(\mathbf{x} \mid \omega_i) P(\omega_i)$, and the class priors $P(\omega_i)$ are usually estimated as the fraction of training patterns belonging to each class. Also non-parametric classifiers like the k nearest neighbours (k-NN) classifier and neural networks, which are universal approximators, provide approximations of the a posteriori probabilities. For the k-NN classifier, let be k_i the number of patterns belonging to class ω_i, among the k nearest neighbours of a pattern $\mathbf{x}$ to be classified. It has been proven that, under certain hypotheses, the value of $P(\omega_i \mid \mathbf{x})$ is approximated by k_i / k (Bishop, 1995; Duda et. al., 2001). For neural networks, consider a c-class problem, and a multi-layer perceptron neural network with c output neurons whose activation values are in the range [0,1]. If the network is trained with the back-propagation algorithm, it has been shown that their outputs approximate the a posteriori probabilities in a mean square sense (Richard and Lippmann, 1991; Ruck et al., 1990).

In multiple classifier systems, estimates of the a posteriori probabilities are provided by combining rules based on Bayesian formalism (Xu et al, 1992; Huang and Suen, 1995), and on Dempster-Shafer evidence theory (Xu et al., 1992). More precisely, these rules provide a so-called *belief* value for each class. Beliefs are estimates of the probability that an input pattern belongs to a given class, given the class labels provided by each individual classifier. The corresponding classification rules with reject option are analogous to Chow's rule, since beliefs are treated as estimates of the a posteriori probabilities.

Some classifiers provide output values which can not be considered probability estimates, for instance because they are not in the range [0,1]. Whenever possible, to implement the reject option estimates of the a posteriori probabilities are computed. An example are distance classifiers, whose outputs are the distances d_i, $i = 1, \ldots, c$ of an input pattern from class centres. An input pattern is assigned to the class corresponding to the minimum d_i. In this case the probability that

a pattern belongs to any class can be intuitively related to the values of d_i, assuming that the higher is the distance of a pattern from a class centre, the lower is the probability that the pattern belongs to that class. Accordingly, a simple estimate of the a posteriori probabilities can be computed as $P(\omega_i|\,\mathbf{x}) = \frac{1/d_i}{\sum_{i=1}^{c} 1/d_i}$ (Xu et al, 1992), and Chow's rule can be applied to these estimates to implement the reject option. Another example are Support Vector Machines (SVMs), a pattern recognition technique recently introduced by Vapnik (1998). Basically, SVMs are two-class classifiers based on finding a linear class boundary (separating hyperplane) in a new feature space of higher dimension than the original one. The output of a SVM is the distance of the input pattern from the separating hyperplane. This value is not in the range [0,1] and has no relationship with the a posteriori probabilities. Nonetheless, it can be used as an indication of classification reliability, with the reasonable assumption that the higher the distance of a pattern from the class boundary, the more its classification can be considered reliable. A reject threshold can then be applied directly to the output of a SVM (Mukerjee et al., 1998). Methods for estimating the a posteriori probabilities from the output of SVMs have also been recently proposed, mainly aimed at introducing the reject option through Chow's rule (Kwok, 1999; Madevska-Bogdanova and Nikolic, 2000).

In some cases it is not possible to obtain meaningful estimates of the a posteriori probabilities, and only one measure of classification reliability associated to the winning class can be computed. For instance, this is the case of combining rules based on majority voting and weighted voting. For these rules the classification reliability can only be evaluated as the ratio between the number of votes received by the winning class, and the total amount of available votes. The reject option can only be implemented by applying a reject threshold on this measure (Xu et al., 1992; Battiti and Colla, 1994).

2.4 Rejection rules proposed in the literature

As said above, Chow's rule is commonly used for classifiers whose outputs approximate the a posteriori probabilities, or if meaningful estimates of the a posteriori probabilities can be obtained. Since Chow's rule is not optimal when it is not applied to the exact values of the a posteriori probabilities, alternative rejection rules have been proposed in the literature. Most of these rules are targeted to neural classifiers. Basically, these rules evaluate classification reliability using other parameters besides the value of the highest estimated a posteriori probability.

In a work on neural handwritten digit recognition, Le Cun et al. (1990) evaluated the classification reliability using three parameters. They considered the value of the most-active output unit $o_i(\mathbf{x}) = \max_{k=1,\ldots,c} o_k(\mathbf{x})$, the value of the second most-active unit $o_j(\mathbf{x}) = \max_{k \neq i} o_k(\mathbf{x})$, and the difference between these activity levels $o_i(\mathbf{x})$-$o_j(\mathbf{x})$. They applied three different reject thresholds on these parameters. An input pattern $\mathbf{x}$ was accepted and assigned to class ω_i under three

conditions: the value of $o_i(\mathbf{x})$ should by larger than a given threshold t_1 (analogously to Chow's rule), the value of $o_j(\mathbf{x})$ should be smaller than a given threshold t_2, and the difference $o_i(\mathbf{x})$ - $o_j(\mathbf{x})$ should be larger than a given threshold t_d. A similar rule, based only on the parameters $o_i(\mathbf{x})$ and $o_i(\mathbf{x})$-$o_j(\mathbf{x})$, was used by Battiti and Colla (1994) in experiments on neural networks combination. This last rule was used also by Cordella et al. (1995). In particular, they proposed a method to evaluate the two reject thresholds, based on maximising a performance function which takes into account the error and reject rates.

A specific rejection rule for binary classifiers was proposed by Tortorella (2000). He considered two-class classifiers which provide only one output value $o(\mathbf{x})$ in the range [0,1]. This is usually the case of neural networks, whose architecture for two-class problems contains one only output neuron. Denoting the two classes ω_1 and ω_2 respectively as *positive* and *negative*, two classification schemes without reject option can be used. A general scheme consists in assigning an input pattern $\mathbf{x}$ to class ω_1 or to class ω_2, depending on whether $o(\mathbf{x}) \geq t$ or $o(\mathbf{x}) < t$, where t is a given threshold. If the output $o(\mathbf{x})$ is considered an estimate of the a posteriori probability of class ω_1, the MAP rule is implemented by using a threshold value $t = 0.5$. The rejection rule proposed by Tortorella is based on extending the former classification scheme, using two reject thresholds. An input pattern $\mathbf{x}$ is assigned to class ω_1 if $o(\mathbf{x}) > t_1$, to class ω_2 if $o(\mathbf{x}) < t_2$, while it is rejected if $t_2 \leq o(\mathbf{x}) \leq t_1$. Obviously $t_2 \leq t_1$.

Methods for evaluating a single reliability parameter from the outputs of a classifier were also proposed. Foggia et al. (1999) and Sansone et al. (2001) considered multiple classifier systems based on the Bayesian combining rule, which provides estimates of the a posteriori probabilities. Denoting with π_1 the maximum of the estimated a posteriori probabilities, and with π_2 the second highest value, the two values $\psi_a = \pi_1$ and $\psi_b = 1 - \pi_2/\pi_1$ were used as indications of two typical situations which lead to unreliable classifications. The first situation is a diffuse disagreement between the individual classifiers, which can result in low values of the estimated a posteriori probablility of the winning class π_1. The second situation arises when individual classifiers part into groups, each agreeing on a different class. This would result in similar values of π_1 and π_2, and therefore in low values of ψ_b. Note that the values of π_1 and π_2 correspond to the parameter used in rejection rules for neural networks described above. Foggia et al. (1999) proposed to compute a unique reliability parameter $\psi \in [0,1]$ by combining the values of ψ_a and ψ_b using a suitable combining operator, so that higher values of ψ correspond to more reliable classifications. For instance, three possible choices are $\psi = \min\{\psi_a, \psi_b\}$, $\psi = \max\{\psi_a, \psi_b\}$, $\psi = (\psi_a + \psi_b)/2$. The rejection rule used is analogous to Chow's rule: an input pattern $\mathbf{x}$ is assigned to the class for which the estimated a posteriori probability is maximum, if the corresponding $\psi(\mathbf{x})$ is higher than a given reject threshold, otherwise the pattern is rejected. A similar method was proposed by De Stefano et al. (2000) for neural classifiers.

An approach based on computing a unique reliability parameter was also proposed by Vasconcelos et al. (1993) for neural networks, but in this case the reliability parameter was computed from the network inputs. This method consists in modifying the architecture of a multi-layer perceptron neural network, by introducing one *guard unit* for each class. Each guard unit is

fully connected with the input layer, and provides an additional network output to be used at the decision stage. The neural network is trained with the standard back-propagation algorithm, except for the guard units. The weight vector of each guard unit is computed as the mean of the feature vectors of training patterns belonging to the corresponding class. A pattern is then rejected if the output of the guard unit corresponding to the winning class is lower than a given threshold.

Note that all the above approaches are implicitely or esplicitely motivated by non-optimality of Chow's rule when applied to estimates of the a posteriori probabilities. However, the effectiveness of the proposed rejection rules over Chow's rule was not theoretically proven.

A different approach to the design of a classifier with reject option was proposed by Mizutani (1998). He pointed out that usually the classifier parameters and the parameters of the rejection rule (i.e., the reject thresholds) are set separately. In particular, the classifier parameters are set during the training phase without taking into account the reject option. Mizutani proposed a new training algorithm capable to setting both the classifier parameters and the parameters of the rejection rule, with the aim of simultaneously minimising the misclassification and reject rates. In particular, he considered classification rules based on discriminant functions with continuous values, and rejection rules based on reliability parameters evaluated from these values (like the rules described above). The proposed learning algorithm performed a minimisation of a weighted sum of misclassification and reject rates (analogous to the expected risk), evaluated as a function of the classifier parameters and of the reject thresholds.

Chapter 3
State of the art on open problems in classification with reject option

In this Chapter we point out two open problems in classification with reject option, and review the related state of the art. The first problem is the characterisation of the error-reject trade-off achievable by real classifiers, with respect to the optimal trade-off, both for individual classifiers and for multiple classifier systems. The second one consists in the introduction of the reject option in support vector machines. These problems will be the subject of the next Chapters.

3.1 The error-reject trade-off achievable by real classifiers

De Stefano et al. (2000) pointed out that the problem of classification with reject option has been tackled only occasionally in the literature. A more specific issue was previously raised by Hansen et al. (1997). They pointed out that little attention had been devoted in the literature to the problem of characterising the error-reject trade-off achievable by neural network classifiers. They gave a first contribution to this topic, by analysing the qualitative behaviour of the error-reject curve achievable by real classifiers, under the hypothesis of *effectively binary* classification problems. This analysis considered both individual classifiers and multiple classifier systems based on majority voting. However, besides their work, the above claim by Hansen et al. (1997) is still valid, and can be extended to every type of classifier. Indeed, no work in the literature analysed from a theoretical viewpoint the difference between the error-reject trade-off achievable by a real classifier and the optimal error-reject trade-off. In particular, for classifiers which provide approximations of the a posteriori probabilities, no work analysed how the estimation errors affect the performance of Chow's rule. Nonetheless, Chow's rule is commonly used despite its non-optimality. Note that a quantitative analysis of the effects of estimation errors on classifier performance was given by Tumer and Ghosh (1996a; 1996b). However, their analysis was limited to the case of classification without reject option.

Let us consider the rejection rules proposed in the literature, which were reviewed in paragraph 2.4. As already pointed out, these rules are motivated (often implicitely) by the fact that Chow's rule is not optimal when it is not applied on the exact values of the a posteriori probabilities. However, the effectiveness of these rules, compared to Chow's rule, was not proven from the theoretical point of view. Accordingly, these can be considered heuristic rules. Basically, most of these rules evaluate classification reliability as a function of the two highest estimated a posteriori probabilities. Hansen et al. (1997) suggested a justification for the use of other parameters, besides the maximum a posteriori probability, to evaluate classification reliability. They argued that using more parameters can provide independent measures of classification reliability in presence of estimation errors on the a posteriori probabilities, while it would be

obviously redundant if the a posteriori probabilities were exactly known. However, this was not theoretically proven.

A lack of theoretical analysis about the error-reject trade-off is to be pointed out also for multiple classifier systems (MCSs). MCSs are a pattern recognition technique which received increasing attention since the early work by Hansen and Salamon (1990), and are still a research topic of great interest (Kittler and Roli, 2000; 2001). MCSs are based on combining the outputs of an ensemble of individual classifiers. The rationale is that, if the ensemble of classifiers and the combining rule are properly designed, an MCS can exhibit better performances with respect to the individual classifiers. In the literature of MCSs several theoretical works analysed the hypotheses under which particular combining rules can improve the performances of the individual classifiers, and quantitatively evaluated the improvements. For instance, Lam and Suen (1997) analysed the majority-voting rule and explained some important aspects of its behaviour and performances. Tumer and Ghosh (1996a; 1999) developed a theoretical framework to quantify the performance improvements due to linearly combining classifiers in output space by simple averaging. They considered classifiers whose outputs approximate the a posteriori probabilities. In particular, their theoretical framework allows to understand the effects of estimation errors on the performance of individual classifiers and of a linear combination of classifiers. However, these works considered only classification without reject option. No theoretical work analysed the improvement of the error-reject trade-off due to combining classifiers. Only experimental works showed that combining individual classifiers can improve their error-reject trade-off, for instance Giacinto et al. (2000), Perrone and Cooper (1993), Lam and Suen (1995), Battiti and Colla (1994).

3.2 Reject option in support vector machines

Support vector machines (SVMs) are a technique recently introduced by V. Vapnik and co-workers (Vapnik, 1998), which encompasses problems like regression estimation, density estimation, and pattern recognition. It is based on Statistical Learning Theory, which was developed by Vapnik since the early 1960's. In the field of pattern recognition, SVMs exhibit significant advantages over traditional classifiers from the algorithmic point of view. SVMs have also proven to be effective in several applications, such as hand-written character recognition and image recognition (Vapnik, 1998; Cristianini and Shawe-Taylor, 2000).

Basically, a SVM is a two-class classifier based on the following idea. The original feature space D is projected into a new feature space D' having a higher dimension. In D' a linear decision surface (separating hyperplane) $\mathbf{w}\cdot\mathbf{x'} + b$ is constructed, trying to maximise the separation (margin) between the two classes. From statistical learning theory, it turns out that maximising the margin improves the generalisation capability of a classifier (Vapnik, 1998). For a given input pattern $\mathbf{x}$, a SVM provides the output $f(\mathbf{x}) = \mathbf{w}\cdot\mathbf{x'} + b$, where $\mathbf{x'}$ is the projection of $\mathbf{x}$ on D'. Denoting the class labels with {+1, -1}, the class label provided by a SVM is sign($f(\mathbf{x})$).

Note that $f(\mathbf{x})$ is a real number in the range $(-\infty,+\infty)$, and its absolute value is proportional to the distance between **x'** and the separating hyperplane. This value has no relationship with the a posteriori probabilities of the input pattern. Surprinsingly, despite the strong theoretical foundations of SVMs, and their successful application in several pattern recognition applications, the introduction of the reject option in the framework of statistical learning theory was not considered in the literature. Up to now, only heuristic rules were proposed.

A straightforward way to evaluate the classification reliability of a SVM is to consider the distance of an input pattern from the separating hyperplane as a certainty measure. Intuitively, the higher the distance of a pattern from the class boundary, the more its classification can be considered reliable. This approach was chosen by Mukherjee et al. (1998) for a problem of cancer classification. They observed that for patterns near to the separating hyperplane the classifier may not be confident enough of the class labels. Therefore they proposed to reject patterns whose distance from the separating hyperplane was below a given threshold value. This value was computed by introducing confidence levels based on SVMs output, estimated from training data.

Methods for estimating the a posteriori probabilities from the output of a SVM were also proposed. Hastie and Tibshirani (1996) defined a general classification strategy for multiclass problems, based on subdividing them into two-class problems, and on combining the resulting estimates of the a posteriori probabilities. Note that SVMs are two-class classifiers, and only heuristic extensions to multiclass problems have been proposed (for instance, in Vapnik (1998)). To apply their method to SVMs, Hastie and Tibshirani proposed to fit gaussians with equal variance to the class-conditional densities $p(f(\mathbf{x}) \mid \omega)$, where $f(\mathbf{x})$ is the SVM output. The corresponding estimate of the a posteriori probability $P(\omega \mid \mathbf{x})$ is a sigmoid whose slope depends on the variance of the gaussians. Platt (1999a) proposed a similar method, based on directly fitting a sigmoid after the output of a SVM:

$$P(\omega = +1 \mid \mathbf{x}) = \frac{1}{1+\exp(Af(\mathbf{x})+B)} \ .$$

The parameters A and B were evaluated using maximum likelyhood estimation from training data. Madevska-Bogdanova and Nikolic (2000) proposed to use a diffrent sigmoid to fit the output of a SVM:

$$P(\omega = +1 \mid \mathbf{x}) = \frac{1}{1+\exp(k(-d_x + d_{sv}))} = \frac{1}{1+\exp\left(k\left(\frac{1-|f(\mathbf{x})|}{\|\mathbf{w}\|}\right)\right)} \ ,$$

where d_{sv} denotes the margin of the separating hyperplane, and d_x denotes the absolute distance between the input pattern **x** and the hyperplane. The rationale is that, with the above choice, patterns inside the margin are given an estimated a posteriori probability less than 0.5. A more theoretically founded approach to estimate the a posteriori probabilities was proposed by Kwok (1999). He applied to SVMs the so-called evidence framework, which is a Bayesian framework proposed by MacKay (1992). Under this framework, the parameters of any learning machine (the weight vector **w** for SVMs), take a posterior distribution even after learning. They should then be

handled by marginalisation, that is, by integrating them out from the conditional distribution. The resultant marginalised output is called *moderated output*. By assuming a gaussian prior for the parameter $\mathbf{w}$, and a sigmoidal-like a posteriori probability distribution $P(\omega|\mathbf{x},\mathbf{w})$, Kwok (1999) showed that training a SVM can be regarded as finding the maximum a posteriori estimate of $\mathbf{w}$. By integrating $\mathbf{w}$ out of $P(\omega|\mathbf{x},\mathbf{w})$ he obtained estimates of the a posteriori probability $P(\omega|\mathbf{x})$ as a function of the output $f(\mathbf{x})$. Note that the functional form of the estimated a posteriori probabilities must be chosen a priori, and also in this case a sigmoidal function was proposed.

We point out that the estimates of the a posteriori probability $P(\omega|\mathbf{x})$ provided by all the above methods are sigmoidal functions of the output $f(\mathbf{x})$ of a SVM. All these estimates of $P(\omega|\mathbf{x})$ are then *monotonic* functions of $f(\mathbf{x})$. This means that applying Chow's rule on such estimates of $P(\omega|\mathbf{x})$ is equivalent to apply a reject threshold directly on the output $f(\mathbf{x})$ of a SVM, as proposed by Mukerjee et al. (1998). Therefore estimates of the a posteriori probabilities obtained using the above methods do not provide a different way to implement the reject option. Accordingly, we can say that at the state of the art the only rejection rule for SVMs is the heuristic rule consisting in applying a reject threshold on the output $f(\mathbf{x})$.

Chapter 4
A method for improving the error-reject trade-off of Chow's rule

In Chapter 3 we pointed out that no work in the literature analysed the performance of Chow's rule in presence of estimation errors on the a posteriori probabilities. Moreover, the effectiveness of alternative rejection rules proposed in the literature was not assessed from the theoretical point of view. In this Chapter we extend a theoretical framework proposed by Tumer and Ghosh (1996a; 1999), with the aim to analyse the effects of estimation errors on the error-reject trade-off achievable by Chow's rule. We then propose a new rejection rule, and prove that, in presence of estimation errors on the a posteriori probabilities, it allows to achieve a better error-reject trade-off than Chow's rule.

4.1 The effects of estimation errors on classifier performance

Tumer and Ghosh (1996a; 1999) developed an analytical framework to quantify the performance improvement due to combining classifiers in output space. They considered classifiers whose outputs approximate the a posteriori probabilities, and quantified the increment of the error probability over Bayes error, due to estimation errors. As already pointed out, their analysis was focused on classification without reject option. In the following we extend this framework to classification with reject option, with the aim to evaluate the error-reject trade-off achievable by Chow's rule in presence of estimation errors on the a posteriori probabilities.

The analytical framework by Tumer and Ghosh is based on the assumption that the decision boundaries provided by a reasonably well trained classifier are close to Bayesian decision boundaries. The effects of estimation errors on classifier performance can then be analysed around the decision boundaries. The estimate of the a posteriori probability of the i-th class provided by a classifier for a single-dimensional feature vector x can be expressed as:

$$\hat{p}_i(x) = p_i(x) + \varepsilon_i(x) \ , \tag{12}$$

where $p_i(x)$ is the true a posteriori probability and $\varepsilon_i(x)$ is the estimation error. The extension to the multi-dimensional case was discussed in Tumer (1996). The optimum boundary between classes i and j consists of points x^* such that $p_i(x^*) = p_j(x^*)$, where $p_i(x^*) = \max_k p_k(x^*)$, and $p_j(x^*) = \max_{k \neq i} p_k(x^*)$. The boundary x_b obtained using the estimated a posteriori probabilities may vary with respect to the optimal one, as shown in Fig. 2.

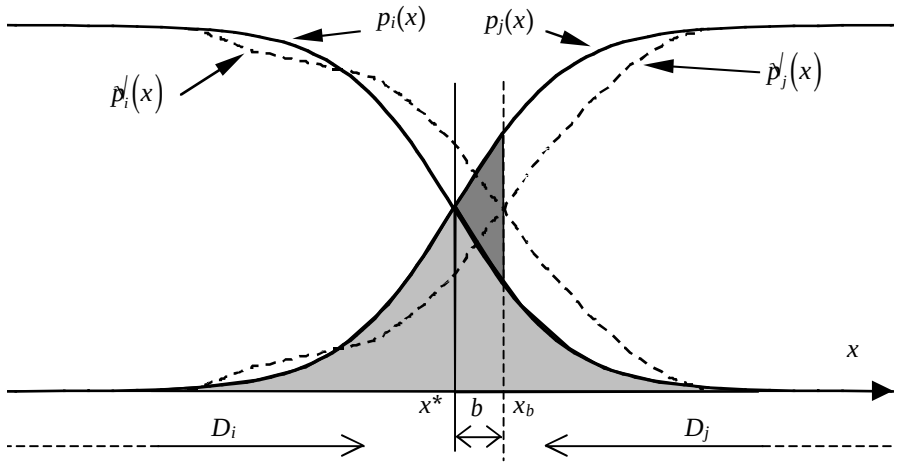


Fig. 2. The optimal and estimated a posteriori probabilities of classes i *and* j *are shown.*

Denoting with $b = x_b - x^*$ the offset between the two boundaries, the estimated one is characterised by:

$$\hat{p}_i(x^*+b) = \hat{p}_j(x^*+b) \ . \tag{13}$$

From Eqs. (12) and (13), one obtains:

$$p_i(x^*+b) + \varepsilon_i(x_b) = p_j(x^*+b) + \varepsilon_j(x_b) \ . \tag{14}$$

The error probability achieved using the estimated a posteriori probabilities is obviously greater than Bayes error. In Fig. 2 Bayes error is represented by lightly shaded regions, and the additional error is represented as a darkly shaded area. The additional error probability, denoted with $A(b)$, is given by:

$$A(b) = \int_{x^*}^{x^*+b} \left[p_i(x) - p_j(x) \right] p(x)\, dx \ , \tag{15}$$

where $p(x)$ is the probability density function of the feature vector x. To compute the above integral, a linear approximation of $p_k(x)$ around x^* was suggested by Tumer and Ghosh:

$$p_k(x^*+b) \cong p_k(x^*) + b p'_k(x^*) \ . \tag{16}$$

This approximation was justified by the fact that the a posteriori probabilities of the correct classes can be considered monotonically increasing (or decreasing) within a suitable chosen region about the optimum boundary. By substituting expression (16) in Eq. (15), it is possible to express the added error $A(b)$ as a function of the offset b. In turn, by making the same sostitution in Eq. (14), the offset b can be expressed as a function of the estimation errors $\varepsilon_i(x_b)$ and $\varepsilon_j(x_b)$. This allows to compute the expected value E_{add} of the added error $A(b)$ with respect to the estimation errors. It turns out that E_{add} can be expressed as a function of bias and variance of the estimation errors, under the hypothesis that these quantities do not vary along the decision boundary considered.

Let us now extend the above framework to classification with reject option. When using reject option, the classifier performance is expressed by the value of the expected risk (7) (related to the loss function (6)). The optimal value of the reject threshold in given by Eq. (11), for given values of the classification costs. Our aim is to evaluate the increment of the expected risk due to

applying Chow's rule on estimated a posteriori probabilities. Let us consider the same problem of Fig. 2, and any value of the reject threshold T, corresponding to given values of the classification costs w_C, w_R, and w_E. The corresponding optimal and estimated decision and rejection regions are shown in Fig. 3.

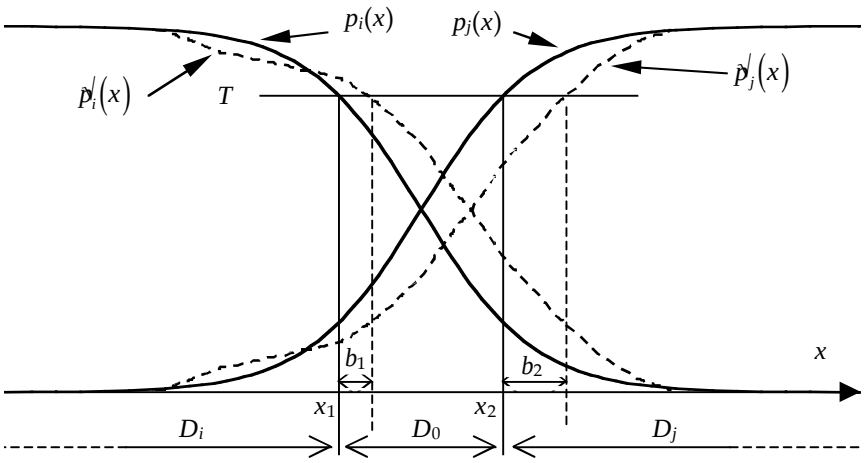


Fig. 3. The optimal and estimated decision and reject regions for classes i *and* j *are shown.*

The optimal boundaries between the decision regions of classes i and j, and the rejection region, are respectively the points x_1 and x_2 such that $p_i(x_1) = p_j(x_2) = T$. As shown in Fig. 3, the estimated boundaries differ from the optimal ones by offsets b_1 and b_2. From Fig. 3 it is easy to see that the differences between the optimal and the estimated regions consist in the intervals $[x_1,x_1+b_1]$ and $[x_2,x_2+b_2]$. In particular, patterns belonging to the interval $[x_1,x_1+b_1]$ are accepted and (correctly) assigned to class i instead of being rejected. Analogously, patterns belonging to the interval $[x_2,x_2+b_2]$ are rejected instead of being accepted and assigned to class j. The expected risk achieved using the estimated a posteriori probabilities is obviously greater than Bayes risk. Let us denote with ΔE the difference between the error probability achieved using the estimated a posteriori probabilities and the true ones, and with ΔR the difference between the reject probabilities. From Eq. (7) it follows that the difference between the actual expected risk, denoted with Δr, and Bayes risk, is:

$$\Delta r = (w_R - w_C)\Delta R + (w_E - w_C)\Delta E \ .$$

Since outside the intervals $[x_1,x_1+b_1]$ and $[x_2,x_2+b_2]$ the optimal and the estimated decision and rejection regions coincide, the values of ΔR and ΔE can be computed only on these intervals. In particular, the interval $[x_2,x_2+b_2]$ gives a positive contribution to ΔR, since the corresponding patterns are rejected instead of being classified. Analogously, the interval $[x_1,x_1+b_1]$ gives a negative contribution to ΔR. The value of ΔR can then be expressed as follows:

$$\Delta R = \int_{x_2}^{x_2+b_2} p(x)\,dx - \int_{x_1}^{x_1+b_1} p(x)\,dx \ .$$

Analogously, the value of ΔE can be expressed as:

$$\Delta E = \int_{x_1}^{x_1+b_1} \left[1 - p_i(x)\right] p(x)\,dx - \int_{x_2}^{x_2+b_2} \left[1 - p_j(x)\right] p(x)\,dx \ .$$

To compute the above integrals we make two approximations. We first approximate the values of $p(x)$ in the domains of integration $[x_1,x_1+b_1]$ and $[x_2,x_2+b_2]$, respectively with the constant terms $p(x_1)$ and $p(x_2)$. The expression of ΔR becomes:

$$\Delta R = p(x_2)b_2 - p(x_1)b_1 \ . \tag{17}$$

We then make the same linear approximation (16) suggested by Tumer and Ghosh for $p_i(x)$ and $p_j(x)$, respectively around x_1 and x_2:

$$\begin{aligned} p_i(x) &\cong p_i(x_1) + (x - x_1)p_i'(x_1) \ , \\ p_j(x) &\cong p_j(x_2) + (x - x_2)p_j'(x_2) \ . \end{aligned} \tag{18}$$

The expression of ΔE becomes:

$$\Delta E = \left[2 - 2p_i(x_1) - b_1 p_i'(x_1)\right] p(x_1)\frac{b_1}{2} - \left[2 - 2p_j(x_2) - b_2 p_j'(x_2)\right] p(x_2)\frac{b_2}{2} \ . \tag{19}$$

We can now express the values of the offsets b_1 and b_2, and therefore of ΔR and ΔE, as functions of the estimation errors. To this aim, note that the estimated posterior probabilities of classes i and j are equal to the reject threshold T in points x_1+b_1 and x_2+b_2 (see Fig. 3). Using Eq. (12) this can be written as follows:

$$p_i(x_1+b_1) + \varepsilon_i(x_1+b_1) = p_j(x_2+b_2) + \varepsilon_j(x_2+b_2) = T \ .$$

Applying the linear approximation (18) for $p_i(x)$ and $p_j(x)$, we obtain:

$$p_i(x_1) + b_1 p_i'(x_1) + \varepsilon_i(x_1 + b_1) = p_j(x_2) + b_2 p_j'(x_2) + \varepsilon_j(x_2 + b_2) = T \ .$$

Since $p_i(x_1) = p_j(x_2) = T$ (see Fig. 3), by subtracting T from the terms of the above expression, we obtain:

$$b_1 = -\frac{\varepsilon_i(x_1 + b_1)}{p_i'(x_1)} \ , \quad b_2 = -\frac{\varepsilon_j(x_2 + b_2)}{p_j'(x_2)} \ .$$

By substituting the above expressions of b_1 and b_2 in Eqs. (17) and (19), and taking into account that

$$p_i(x_1) = p_j(x_2) = T = \frac{w_E - w_R}{w_E - w_C} \ ,$$

we finally obtain:

$$\Delta r = a\varepsilon_i^2(x_1 + b_1) + b\varepsilon_j^2(x_2 + b_2) \ , \tag{20}$$

where a and b are constant terms:

$$a = -(w_E - w_C)\frac{p(x_1)}{2p_i'(x_1)} \ , \quad b = (w_E - w_C)\frac{p(x_2)}{2p_j'(x_2)} \ . \tag{21}$$

The value of the added risk Δr obtained above refers to specific values of the estimation errors. Let us consider its expected value:

$$r_{add} = E\{\Delta r\} = aE\{\varepsilon_i^2\} + bE\{\varepsilon_j^2\} \ .$$

Denoting the bias and the variance of the estimation error ε_i respectively with β_i and $\sigma_{\varepsilon_i}^2$, we obtain:

$$r_{add} = a\left(\sigma_{\varepsilon_i}^2 + \beta_i^2\right) + b\left(\sigma_{\varepsilon_j}^2 + \beta_j^2\right) . \quad (22)$$

The above equation shows that, under the hypotheses made, the expected value of the added risk of a classifier is proportional to the sum of bias squared and variance of the estimation errors. We point out that the above expression is similar to the one obtained by Tumer and Ghosh for the expected value of the added error of a classifier without reject option.

4.2 Class-related reject thresholds

4.2.1 The class-related reject thresholds rule

In the previous paragraph we derived an expression for the increment of the expected risk achieved by Chow's rule, with respect to Bayes risk, as a function of the estimation errors on the a posteriori probabilities. This expression shows how the error-reject trade-off achievable by Chow's rule is affected by estimation errors, under certain assumptions. In the following we propose a new rejection rule, and formally prove that it allows to obtain a better error-reject trade-off than Chow's rule (that is, a lower expected risk) in presence of estimation errors (Fumera et al., 2000a).

To introduce our rejection rule, let us consider again the effects of estimation errors on the class boundary of Fig. 3. Under the hypotheses made in paragraph 4.1, the boundaries of the rejection region obtained by applying Chow's rule on the estimated a posteriori probabilities are slightly shifted with respect to the optimal boundaries x_1 and x_2. This means that patterns of class j belonging to the interval $[x_2, x_2+b_2]$ are rejected, instead of being accepted, since their estimated a posteriori probability is lower than the reject threshold T. Note that these patterns could be accepted and correctly classified by using a *lower* value of T, despite the estimation errors on their a posteriori probabilities. Analogously, patterns of class i belonging to the interval $[x_1, x_1+b_1]$ are accepted instead of being rejected, since their estimated a posteriori probability is higher than T. These patterns could be rejected by using a *higher* value of T. This suggests that the effects of the estimation errors can be mitigated by using a different reject threshold for each class. A careful analysis of Fig. 3 shows that applying two different reject thresholds allows to achieve a better error-reject trade-off than the one achievable using Chow's rule. Indeed, the two different reject thresholds shown in Fig. 4 provide the optimal decision boundaries, and therefore a better error-reject trade-off than Chow's rule when applied on the estimated a posteriori probabilities.

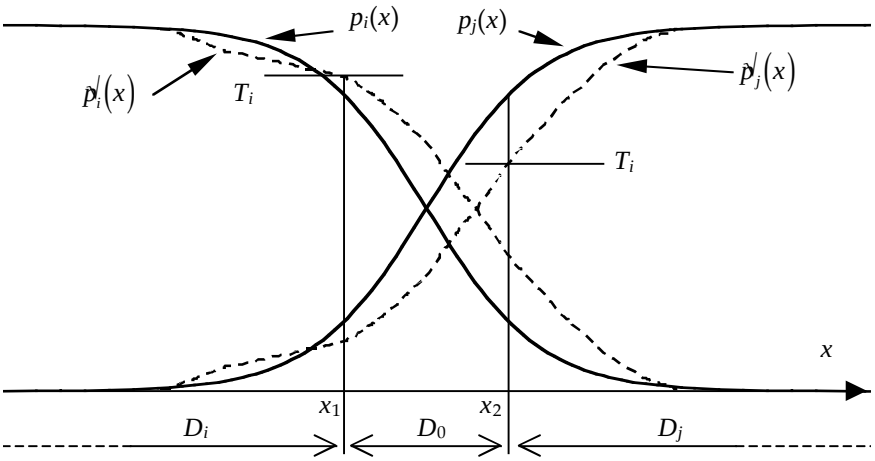


Fig. 4. Two different reject thresholds on the estimated a posteriori probabilities of classes i and j.

Let us now formally define this rejection rule, which we call *class-related reject thresholds* (CRT) rule. The CRT rule consists in assigning a pattern **x** to the class ω_i exhibiting the maximum (estimated) a posteriori probability, if it is higher than the corresponding class-related reject threshold T_i:

$$\max_{j=1,\ldots,c} \hat{P}(\omega_j | \mathbf{x}) = \hat{P}(\omega_i | \mathbf{x}) \geq T_i \ . \tag{23}$$

The pattern is otherwise rejected. Obviously $T_i \in [0,1]$, $i = 1,\ldots, c$. In the following paragraph we formally prove that the CRT rule allows to achieve a better error-reject trade-off than Chow's rule in presence of estimation errors on the a posteriori probabilities.

Note that the use of a different reject threshold for each class was previously proposed by Yau and Manry (1992) as a heuristic rule for different purposes. They pointed out that Chow's rule minimises the overall error probability, for any given reject probability. This means that if the different classes exhibit different a priori probabilities $P(\omega_i)$, the class-conditional error and reject probabilities $P(error \mid \omega_i)$ and $P(reject \mid \omega_i)$ may significantly differ. Using different reject thresholds for each class was then proposed to equalise these probabilities.

4.2.2 Proof

For our proof we use the definition of optimal error-reject trade-off given at the end of paragraph 2.1. The optimal error-reject trade off consists in minimising the expected risk (7) (when using the loss function (6)), and is equivalent to maximising the probability of correct classification for any value of the reject probability. This implies that a rejection rule provides a better error-reject trade-off than another rule, if it allows to achieve a higher or equal probability of correct classification for any value of the reject probability. Therefore we will compare the probability of correct classification achieved by the CRT and Chow's rules, for equal values of the reject probability. We will show that, for any value of the reject probability, values of the CRTs always exist such that the corresponding probability of correct classification is greater, or at least equal to the one achieved by Chow's rule.

Note first that the CRT rule can always achieve the same probability of correct classification than Chow's rule, in the trivial case in which the CRTs are all equal to Chow's threshold T, that is

$T_i = T$, $i = 1, ..., c$. We have therefore to prove that there exist conditions under which the CRT rule can provide a probability of correct classification *strictly* greater than that of Chow's rule.

Let us first provide some basic definitions. We will denote values related to the CRT rule with the apex *CRT*, and values related to Chow's rule with the apex *T*. Without reject option, the decision regions are defined using Bayes rule (5) on the estimated a posteriori probabilities. The decision regions can be expressed as:

$$D_i = \left\{ \mathbf{x} : \max_{k=1,K,c} \hat{P}(\omega_k | \mathbf{x}) = \hat{P}(\omega_i | \mathbf{x}) \right\}, \quad i = 1, K, c .$$

It is easy to see that using the CRT rule (23) and Chow's rule (9), the corresponding rejection regions can be expressed as the union of disjoint subsets D_{0i} of the decision regions D_i:

$$D_0^{CRT} = \left\{ \mathbf{x} \in D_i, \quad \hat{P}(\omega_i | \mathbf{x}) < T_i \right\}, \quad i = 1, K, c,$$
$$D_0^{T} = \left\{ \mathbf{x} \in D_i, \quad \hat{P}(\omega_i | \mathbf{x}) < T \right\}, \quad i = 1, K, c.$$

The rejection regions are then $D_0^{CRT} = \cup_{i=1}^{c} D_{0i}^{CRT}$, and $D_0^{T} = \cup_{i=1}^{c} D_{0i}^{T}$. It is also easy to see that the following relations hold:

$$\begin{aligned} &\text{if } T_i < T, \; D_0^{CRT} \subseteq D_{0i}^{T} , \\ &\text{if } T_i > T, \; D_0^{CRT} \supseteq D_{0i}^{T} , \\ &\text{if } T_i = T, \; D_0^{CRT} = D_{0i}^{T} . \end{aligned} \tag{24}$$

It follows that, when using the reject option, the decision regions can be expressed as $D_i^{CRT} = D_i - D_{0i}^{CRT}$ and $D_i^{T} = D_i - D_{0i}^{T}$, $i = 1, ..., c$. Note that the reject probabilities R^{CRT} and R^T can be viewed as functions of the corresponding reject thresholds T_i, $i = 1, ..., c$, and T.

Consider now any value of the reject threshold T used in Chow's rule, and the corresponding value of the reject probability R^T. Let us assume that the probability density function of the feature vector $p(\mathbf{x})$ and the a posteriori probability $P(\omega | \mathbf{x})$ are continuous and differentiable functions. This implies that R^{CRT} and R^T are continuous and differentiable functions of the reject thresholds. Moreover, this implies that infinite sets of CRTs $\{T_1,...,T_c\}$ exist, such that the corresponding reject probability R^{CRT} equals R^T, besides the trivial case $T_i = T$, $i = 1, ..., c$. To explain this point, consider that the reject probabilities R^{CRT} and R^T are obviously non-decreasing functions of the corresponding reject thresholds T_i, $i = 1, ..., c$, and T. Since they are continuous and differentiable functions, this can be written as:

$$\frac{dR^T}{dT} \geq 0 ,$$
$$\frac{\partial R^{CRT}}{\partial T_i} \geq 0 \quad i = 1,..., c.$$

Given a value of T, and the corresponding value of R^T, consider a set of CRTs such that $T_i = T$, $i = 1, ..., c$. Obviously $R^{CRT} = R^T$. From the above assumption, it is easy to see that by increasing the value of at least one of the above CRTs, and decreasing the value of at least another one, it is possible to find infinite sets of CRTs values such that the equation $R^{CRT} = R^T$ still holds. In other words, given a value of R^T, the equation $R^{CRT} = R^T$ has infinite solutions. Since $\partial R^{CRT} / \partial T_i \geq 0$, for

each solution at least one of the CRTs must be greater than T, and at least one must be lower than T.

Let us now present the main assumption of this proof. Consider the boundaries of the rejection region obtained using Chow's rule. These boundaries are the sets of points belonging to regions D_i for which $\hat{P}(\omega_i|\,\mathbf{x}) = T,\ \ \mathbf{x} \in D_i,\ \ i = 1,\ldots,c.$ We hypothesise that two non-empty subsets of regions D_i exist, such that in a neighbourhood of the corresponding boundaries the estimation errors are strictly positive or strictly negative. Formally, we are assuming that two disjoint and non-empty subsets $P, Q \subset \{D_1,\ldots,D_c\}$ exist, such that the following relations hold:

$$\begin{aligned} &\varepsilon_i(\mathbf{x}) > \Delta\varepsilon_i > 0, \quad \text{for any } \mathbf{x} \in D_i \in Q, \text{ such that } T \le \hat{P}(\omega_i|\,\mathbf{x}) \le T + \Delta\varepsilon_i\ , \\ &\varepsilon_i(\mathbf{x}) < \Delta\varepsilon_i < 0, \quad \text{for any } \mathbf{x} \in D_i \in P, \text{ such that } T + \Delta\varepsilon_i \le \hat{P}(\omega_i|\,\mathbf{x}) \le T. \end{aligned} \tag{25}$$

Let us explain further the above relations. The terms $\Delta\varepsilon_i$ are constant values, and the expressions $T \le \hat{P}(\omega_i|\,\mathbf{x}) \le T + \Delta\varepsilon_i$ and $T + \Delta\varepsilon_i \le \hat{P}(\omega_i|\,\mathbf{x}) \le T$ define the neighbourhoods of the boundaries of the reject region.

Under the hypothesis that $p(\mathbf{x})$ and $P(\omega\,|\,\mathbf{x})$ are continuous and differentiable functions, we showed that infinite sets of CRTs $\{T_1,\ldots,T_c\}$ exist, such that the corresponding reject probability R^{CRT} equals R^T. According to Eq. (8), the equality between R^{CRT} and R^T can be written as:

$$\sum_{i=1}^{c} \int_{D_{0i}^{CRT}} p(\mathbf{x})\,d\mathbf{x} - \sum_{i=1}^{c} \int_{D_{0i}^{T}} p(\mathbf{x})\,d\mathbf{x} = 0\ , \tag{26}$$

where the two summands represent respectively R^{CRT} and R^T. Let us write the CRTs values as $T+\Delta T_1,\ldots,T+\Delta T_c$, where $\Delta T_1,\ldots,\Delta T_c$ are constant values for each given set. Among these sets, let us consider a set $\{T+\Delta T_1,\ldots,T+\Delta T_c\}$ satisfying the following relations:

$$\begin{aligned} &0 < \Delta T_i < \Delta\varepsilon_i, \text{ if } D_i \in Q\ , \\ &\Delta\varepsilon_i < \Delta T_i < 0, \text{ if } D_i \in P\ , \end{aligned} \tag{27}$$

where the terms $\Delta\varepsilon_i$ are the ones defined in (25).

Let us now compare the probability of correct classification of the CRT rule, using the above set of CRTs values, with that of Chow's rule. We denote these probabilities with C^{CRT} and C^T. According to Eq. (4), the difference C^{CRT} - C^T can be written as:

$$C^{CRT} - C^T = \sum_{i=1}^{c} \left[\int_{D_i - D_{0i}^{CRT}} P(\omega_i|\,\mathbf{x})\,p(\mathbf{x})\,d\mathbf{x} - \int_{D_i - D_{0i}^{T}} P(\omega_i|\,\mathbf{x})\,p(\mathbf{x})\,d\mathbf{x} \right]\ . \tag{28}$$

Note that for $D_i \in P$, we have $\Delta T_i < 0$ (see Eq. (27)), and then $D_{0i}^{CRT} \subseteq D_{0i}^{T}$ (from Eq. (24)). Accordingly, the corresponding term of the sum in Eq. (28) can be rewritten as:

$$\int_{D_{0i}^{T} - D_{0i}^{CRT}} P(\omega_i|\,\mathbf{x})\,p(\mathbf{x})\,d\mathbf{x}\ .$$

Analogously, if $i \in Q$, we have $\Delta T_i > 0$, and consequently $D_{0i}^{T} \subset D_{0i}^{CRT}$. The corresponding term of the sum in Eq. (28) can be rewritten as:

$$-\int_{D_{0i}^{CRT} - D_{0i}^{T}} P(\omega_i|\,\mathbf{x})\,p(\mathbf{x})\,d\mathbf{x}\ .$$

It is straightforward that, if $\Delta T_i = 0$, then $D_{0i}^{T} = D_{0i}^{CRT}$, and the corresponding term in Eq. (28) is null. Therefore, using Eq. (12), Eq. (28) can be rewritten as:

$$C^{CRT} - C^{T} = \sum_{D_i \in P} \int_{D_{0i}^{T} - D_{0i}^{CRT}} \left[\hat{P}(\omega_i | \mathbf{x}) - \varepsilon_i(\mathbf{x}) \right] p(\mathbf{x}) d\mathbf{x} - \sum_{D_i \in Q} \int_{D_{0i}^{CRT} - D_{0i}^{T}} \left[\hat{P}(\omega_i | \mathbf{x}) - \varepsilon_i(\mathbf{x}) \right] p(\mathbf{x}) d\mathbf{x} \ .$$

By substituting $\hat{P}(\omega_i | \mathbf{x})$ with $T+\Delta T_i$, and using inequalities (25) and (27), we obtain that the following inequality holds true:

$$\begin{aligned} C^{CRT} - C^{T} &> \sum_{D_i \in P} \int_{D_{0i}^{T} - D_{0i}^{CRT}} \left[T + \Delta T_i - \varepsilon_i(\mathbf{x}) \right] p(\mathbf{x}) d\mathbf{x} - \sum_{D_i \in Q} \int_{D_{0i}^{CRT} - D_{0i}^{T}} \left[T + \Delta T_i - \varepsilon_i(\mathbf{x}) \right] p(\mathbf{x}) d\mathbf{x} = \\ &= T \left[\sum_{D_i \in P} \int_{D_{0i}^{T} - D_{0i}^{CRT}} p(\mathbf{x}) d\mathbf{x} - \sum_{D_i \in Q} \int_{D_{0i}^{CRT} - D_{0i}^{T}} p(\mathbf{x}) d\mathbf{x} \right] + \\ &+ \sum_{D_i \in P} \int_{D_{0i}^{T} - D_{0i}^{CRT}} \left[\Delta T_i - \varepsilon_i(\mathbf{x}) \right] p(\mathbf{x}) d\mathbf{x} - \sum_{D_i \in Q} \int_{D_{0i}^{CRT} - D_{0i}^{T}} \left[\Delta T_i - \varepsilon_i(\mathbf{x}) \right] p(\mathbf{x}) d\mathbf{x} . \end{aligned} \tag{29}$$

In the above equation, the term $\left[\sum_{D_i \in P} \int_{D_{0i}^{T} - D_{0i}^{CRT}} p(\mathbf{x}) d\mathbf{x} - \sum_{D_i \in Q} \int_{D_{0i}^{CRT} - D_{0i}^{T}} p(\mathbf{x}) d\mathbf{x} \right]$ is equal to the difference between R^{CRT} and R^{T}, and is then null (see Eq. (26)). Therefore, inequality (29) can be rewritten as:

$$C^{CRT} - C^{T} > \sum_{D_i \in P} \int_{D_{0i}^{T} - D_{0i}^{CRT}} \left[\Delta T_i - \varepsilon_i(\mathbf{x}) \right] p(\mathbf{x}) d\mathbf{x} - \sum_{D_i \in Q} \int_{D_{0i}^{CRT} - D_{0i}^{T}} \left[\Delta T_i - \varepsilon_i(\mathbf{x}) \right] p(\mathbf{x}) d\mathbf{x} \ . \tag{30}$$

Finally, from inequalities (27) it turns out that the terms $[\Delta T_i - \varepsilon_i(\mathbf{x})]$ in Eq. (30) are positive for $D_i \in P$, and negative for $D_i \in Q$. The right hand side of Eq. (30) is therefore positive. It follows that

$$C^{CRT} - C^{T} > 0 \ .$$

This proves that, under the above hypothesis, the CRT rule allows to achieve a greater probability of correct classification than Chow's rule, for the same value of reject probability.

4.2.3 Discussion

In the previous paragraph we showed that the CRT rule allows to obtain a better error-reject trade-off than Chow's rule, when applied to estimates of the a posteriori probabilities. More precisely, we showed that, for any value of *P*(*reject*), a set of CRTs values always exists such that the corresponding *P*(*correct*) is higher or at least equal to that of Chow's rule. Since the same *P*(*correct*) of Chow's rule can always be obtained by trivially using CRTs values equal to Chow's threshold, we provided a *sufficient* condition under which the CRT rule achieves a *strictly* greater *P*(*correct*).

Let us first discuss the above condition. It concerns the estimation errors on the a posteriori probabilities, in a neighbourhood of the boundary of the rejection region of each class. It states that there must exist at least one class for which, in such neighbourhood, the estimation errors are strictly positive, and at least one class for which the estimation errors are strictly negative. It is easy to understand the meaning of this condition, by looking again at the example of Fig. 3. Patterns for which the estimation errors are strictly positive exhibit values of the estimated a posteriori probabilities higher than the true ones. Therefore, for a given value of the reject threshold, some of these patterns are accepted instead of being rejected. A higher value of the

reject threshold would allow to reject them. This is the case of patterns of class *i* belonging to the interval $[x_1,x_1+b_1]$ (see Figs. 3 and 4). Accordingly, note that in the above proof classes for which the estimation errors are strictly positive have CRTs values higher than Chow's threshold. Analogously, some of the patterns for which the estimation errors are strictly negative are rejected instead of being accepted. A lower value of the reject threshold would allow to accept them, as happens in Figs. 3 and 4 for patterns of class *j* belonging to the interval $[x_2,x_2+b_2]$. In the above proof, the CRTs values of classes with negative estimation errors are lower than Chow's threshold.

Let us now discuss two issues which are left open by the above proof. While the above proof shows that the CRT rule provides a better error-reject trade-off than Chow's rule, it does not allow to quantify the achievable performance improvements. Moreover, it does not allow to compute the CRT values which provide such improvements. These issues are clearly related to the practical usefulness of the CRT rule. Concerning the performance improvements, let us consider again the simple example of Fig. 4. In this case the CRT rule provides the maximum achievable improvement over Chow's rule, since it allows to obtain the optimal decision and reject regions. This means that the corresponding average improvement, in terms of the expected risk, is equal to the added risk of Chow's rule. As shown in paragraph 4.1, the added risk is proportional to the squared bias and to the variance of the estimation errors (see Eq. (22)). Obviously, in more complex problems the CRT rule will not be able to provide the optimal decision and reject regions. Nonetheless, we argue that also in the general case the achievable performance improvements still depend on the "amplitude" of the estimation errors, which can be characterised in terms of bias and variance. Consider now the second issue mentioned above, that is, how to find CRTs values which provide a higher $P(correct)$ than Chow's rule, for a given value of $P(reject)$. In practical applications the CRTs values must be estimated from validation data. This obviously applies as well to the threshold of Chow's rule. However, while Chow's rule always require to estimate only one parameter, the number of parameters to estimate in the CRT rule is equal to the number of classes. This means that for the CRT rule the estimation process could be critical in applications having a high number of classes. Besides the number of parameters, the estimation process could be critical even if there was one only set of CRTs values which provides a higher $P(correct)$ than Chow's rule, for any given value of $P(reject)$. However, this does not seem the case, at least from the theoretical viewpoint. Indeed, it is easy to see that if the probability distributions of the problem at hand are continuous and differentiable functions, then infinite sets of CRTs values exist, which provide a higher $P(correct)$ than Chow's rule. This can be shown as follows. Using the same notation of the previous paragraph, suppose that there exist a set of CRTs values for which $C^{CRT} > C^T$, and $R^{CRT} = R^T$, for a given value of R^T. As shown in the above proof, under the above assumption C^{CRT} and R^{CRT} are continous functions of the CRTs values. This implies that there exist a neighbourhood of the considered CRTs values, containing infinite CRTs values for which the relations $C^{CRT} > C^T$ and $R^{CRT} = R^T$ still hold.

In the following paragraph the problem of CRTs evaluation is discussed in more detail, and an algorithm for estimating the CRTs values is proposed. An experimental investigation of both issues discussed here will be given in Chapter 7.

4.3 Estimating the values of class-related reject thresholds

The optimal values of the CRTs can be defined as the ones which maximise the probability of correct classification, for a given value of the reject probability. Note that maximising $P(correct)$ is equivalent to maximising the classification accuracy, defined as $A = \frac{P(correct)}{1 - P(reject)}$. The accuracy and the reject probabilities can be viewed as functions of the CRTs values. Let us denote them with $A(T_1,\ldots,T_c)$ and $R(T_1,\ldots,T_c)$. The optimal CRTs values are then the solution of the following problem, where R denotes a given value of the reject probability:

$$\begin{aligned} &\text{maximise} \quad A(T_1,\ldots,T_c), \\ &\text{subject to } R(T_1,\ldots,T_c) = R. \end{aligned} \tag{31}$$

In practical applications the functions $A(T_1,\ldots,T_c)$ and $R(T_1,\ldots,T_c)$ are not known in analytical form, and can only be estimated from validation data. Estimates of the optimal CRTs values could then be obtained by solving the above problem, using the estimated values of accuracy and reject probability. However, note that the estimates of $A(T_1,\ldots,T_c)$ and $R(T_1,\ldots,T_c)$ obtained from a finite data set are discrete-valued functions of continuous variables. Standard optimisation techniques do not fit well this characteristic. On the other hand, it is difficult to approximate $A(T_1,\ldots,T_c)$ and $R(T_1,\ldots,T_c)$ using continuous functions. For this reasons, we developed a specific algorithm for solving problem (31) (Fumera et al., 2000b).

Note first that $R(T_1,\ldots,T_c)$ is a non-decreasing function of the CRTs values. Indeed, increasing the value of any of the CRTs can only increase the fraction of rejected patterns. Our algorithm is based on the experimental behaviour of $A(T_1,\ldots,T_c)$, which is very similar to that of $R(T_1,\ldots,T_N)$. Indeed we experimentally observed that $A(T_1,\ldots,T_c)$ is almost always a non-decreasing function of the CRTs values. Our algorithm exploits this characteristic by searching for the maximum of $A(T_1,\ldots,T_c)$ by iteratively increasing the values of the CRTs, until $R(T_1,\ldots,T_c)$ is lower than the given value R. The initial CRTs values can be chosen among the ones providing a null reject rate, for instance $T_i = 0$. To limit the complexity of the algorithm, only one of the CRTs is increased at each iteration. The increment is a multiple of a given discretisation step ΔT. The CRT to increase is chosen as the one whose increment provides the maximum value of $A(T_1,\ldots,T_c)$ in a neighbourhood of the current CRTs values. The neighbourhood is defined as the set of CRTs values obtained by incrementing each one of the CRTs, one at a time, by an amount $k \cdot \Delta T$, for all k between $-K$ and K, where K is a predefined value. If no CRTs values are found for which $A(T_1,\ldots,T_c)$ is greater than its current value, and $R(T_1,\ldots,T_c) \leq R$, the current CRTs values are returned as the solution. To mitigate the problem of local minima, the multistart technique can be used. In this case the initial CRTs values can be random values such that $R(T_1,\ldots,T_c) < R$.

Chapter 5

The error-reject trade-off of linearly combined multiple classifiers

In Chapter 4 we considered classifiers whose outputs provide approximations of the a posteriori probabilities, and analysed the effects of estimation errors on their error-reject trade-off. In this Chapter we consider again the same kind of classifiers, and show how their error-reject trade-off can be improved by classifier combination. As pointed out in Chapter 3, in the literature no theoretical work investigated the improvements of the error-reject trade-off achievable by combining classifiers. To this aim, we focus on a simple and widely used combining rule, based on the weighted average of classifiers in output space.

Our analysis is still based on the theoretical framework developed by Tumer and Ghosh, which was described in Chapter 4. We remind that this framework was originally developed for analysing the performance improvement achievable using the simple averaging combining rule, for classification without reject option. We already extended this framework to classification with reject option in Chapter 4. In this Chapter we further extend it to the weighted averaging combining rule (Fumera and Roli, 2001; Fumera et al., 2001).

5.1 Analysis of the error-reject trade-off

In Chapter 4 we evaluated the expected value of the added risk of an individual classifier, under the hypothesis that its decision boundaries are close to the optimal boundaries. Following the same approach, we now evaluate the expected value of the added risk of an ensemble of N classifiers, combined by weighted averaging in output space. The same notation introduced n Chapter 4 is used here. In the following we denote the quantities related to the k-th classifier with the superscript k.

The outputs of the combiner can be considered themselves estimates of the a posteriori probabilities, and can be expressed as follows:

$$\hat{p}_i^{ave}(x) = \sum_{k=1}^{N} w_k \hat{p}_i^k(x) = \sum_{k=1}^{N} w_k \left(p_i(x) + \varepsilon_i^k(x) \right) = p_i(x) + \bar{\varepsilon}_i(x) \ ,$$

where the w_j's are the coefficients of the linear combination, and

$$\bar{\varepsilon}_i(x) = \sum_{k=1}^{N} w_k \varepsilon_i^k(x) \tag{32}$$

is the estimation error of the combiner. In the following we consider normalised values of the coefficients:

$$\sum_{k=1}^{N} w_k = 1 \ , \quad w_k \geq 0, \quad k = 1, \ldots, N \ . \tag{33}$$

With reference to Fig. 3, we denote the offsets between the optimal and the estimated decision boundaries with b_1^{ave} and b_2^{ave}. By proceeding as described in Chapter 4, the following expression can be derived for the added risk of a linear combination of classifiers:

$$\Delta r^{ave} = a\bar{\varepsilon}_i^2\left(x_1 + b_1^{ave}\right) + b\bar{\varepsilon}_j^2\left(x_2 + b_2^{ave}\right) ,$$

where a and b are the same constant terms given in Eq. (21). Note that the above expression is analogous to the one obtained for the added risk of an individual classifier (Eq. (20)). The expected value of Δr^{ave} is:

$$r_{add}^{ave} = E\left\{\Delta r^{ave}\right\} = aE\left\{\bar{\varepsilon}_i^2\right\} + bE\left\{\bar{\varepsilon}_j^2\right\} . \tag{34}$$

From the above expression it is easy to see that the value of r_{add}^{ave} depends on bias and variance of the estimation errors, and on the correlation between them. Therefore, some assumptions about these parameters must be made to compute the value of r_{add}^{ave}, and to compare it with r_{add}. In the following paragraphs we analyse four cases, corresponding to different assumptions about bias and correlation. Using the same notation introduced in Chapter 4, the bias and the variance of the estimation errors on the i-th class are denoted respectively with β_i and $\sigma_{\varepsilon_i}^2$.

5.1.1 Unbiased and uncorrelated estimation errors

We first consider the simplest case of unbiased and uncorrelated estimation errors. We assume therefore $\beta_i = \beta_j = 0$. The expected value of the added risk of an individual classifier can be obtained from Eq. (22):

$$r_{add} = a\sigma_{\varepsilon_i}^2 + b\sigma_{\varepsilon_j}^2 .$$

For a linear combination of classifiers, from Eqs. (34) and (32) we obtain:

$$\begin{aligned} r_{add}^{ave} &= a\sigma_{\bar{\varepsilon}_i}^2 + b\sigma_{\bar{\varepsilon}_j}^2 = a\sum_{k=1}^{N} w_k^2\sigma_{\varepsilon_i^k}^2 + b\sum_{k=1}^{N} w_k^2\sigma_{\varepsilon_j^k}^2 = \\ &= \sum_{k=1}^{N} w_k^2 r_{add}^k . \end{aligned} \tag{35}$$

The above expression shows that r_{add}^{ave} can be expressed as a linear combination of the expected value of the added risk r_{add}^k of each individual classifier. It is easy to see that the weights which minimise r_{add}^{ave} are the following:

$$w_k = \left(\sum_{m=1}^{N}\frac{1}{r_{add}^m}\right)^{-1}\frac{1}{r_{add}^k} . \tag{36}$$

This means that, in the simplest case of unbiased and uncorrelated estimation errors, the optimal weights are inversely proportional to the added risk of each individual classifier. Accordingly, we can say that weighted averaging is required to compensate for different classifier performances. Instead, if all classifiers exhibit the same performances, from Eq. (36) it is easy to see that the optimal weights are $w_k = 1/N$, that is, simple averaging is the optimal combining rule. In this case, from Eq. (35) we obtain that simple averaging reduces the difference between the expected value of the added risk and Bayes risk by a factor N:

$$r_{ave}^{add} = \frac{1}{N} r_{add} \,. \tag{37}$$

Let us now evaluate the maximum achievable improvement of the error-reject trade-off with respect to the best individual classifier, in the general case. The value of r_{add}^{ave} corresponding to the optimal weights (36) is:

$$r_{add}^{ave} = \left(\sum_{k=1}^{N} \frac{1}{r_{add}^{k}} \right)^{-1} \,.$$

From this equation the following inequalities can be derived:

$$\frac{1}{N} \min_k r_{add}^k \le r_{ave}^{add} \le \min_k r_{add}^k \,.$$

This means first that weighted averaging always allows to achieve a better error-reject trade-off than the one of each individual classifier. Moreover, the difference between the expected value of the added risk and Bayes risk can be reduced up to a factor N with respect to the best individual classifier. According to Eq. (37), the maximum improvement can be achieved by combining classifiers exhibiting equal average performances.

Let us now consider what happens when using simple averaging for classifiers which do not exhibit the same performances. In this case from Eq. (36) we obtain:

$$r_{add}^{ave} = \frac{1}{N} \left(\frac{1}{N} \sum_{k=1}^{N} r_{add}^{k} \right) \,.$$

This means that in the general case simple averaging allows to improve by a factor N only the average value of the performances of the individual classifiers.

The main result of the above analysis is that, at least for the simplest case of unbiased and uncorrelated estimation errors, simple averaging is the optimal combining rule if the individual classifiers exhibit equal average performances. Weighted averaging allows instead to compensate for different classifier performances. We point out that this result formalises some conclusions drawn in the literature, for example by Tumer and Ghosh (1999), and generalises them to the case of classification with reject option.

5.1.2 Biased and uncorrelated estimation errors

Let us now consider biased estimation errors, that is, β_i, $\beta_j \neq 0$. The expected value of the added risk of an individual classifier was given in Eq. (22):

$$r_{add} = a\left(\sigma_{\varepsilon_i}^2 + \beta_i^2\right) + b\left(\sigma_{\varepsilon_j}^2 + \beta_j^2\right) \,. \tag{38}$$

Let us denote with $\bar{\beta}_i$ the bias of $\bar{\varepsilon}_i(x)$ (see Eq. (32)): $\bar{\beta}_i = \sum_{k=1}^{N} w_k \beta_i^k$. For a linear combination of classifiers, from Eq. (34) we obtain:

$$r_{add}^{ave} = a\left[\sigma_{\varepsilon_i}^2 + \left(\bar{\beta}_i\right)^2\right] + b\left[\sigma_{\varepsilon_j}^2 + \left(\bar{\beta}_j\right)^2\right] =$$
$$= a\sum_{k=1}^{N} w_k^2\left[\sigma_{\varepsilon_i^k}^2 + \left(\beta_i^k\right)^2\right] + 2a\sum_{m<n} w_m w_n \beta_i^m \beta_i^n + b\sum_{k=1}^{N} w_k^2\left[\sigma_{\varepsilon_j^k}^2 + \left(\beta_j^k\right)^2\right] + 2b\sum_{m<n} w_m w_n \beta_j^m \beta_j^n =$$
$$= \sum_{k=1}^{N} w_k^2 r_{add}^k + 2\sum_{m<n} w_m w_n\left(a\beta_i^m \beta_i^n + b\beta_j^m \beta_j^n\right).$$

In this case the optimal values of the weights w_k do not depend only on the performances of the individual classifiers, but also on the bias of the estimation errors. It can be shown that the optimal values of the weights, that is the values which minimise r_{add}^{ave}, can be obtained as the solution of a system of N linear equations (we remind the constraints (33) on the weights values). This system can be easily solved if the estimation errors of each individual classifier exhibit the same bias and the same variance. Note that this implies that the performances of the individual classifiers are equal. In this case we obtain $w_k = 1/N$, that is, simple averaging is the best combining rule. The corresponding expected value of the added risk is:

$$r_{add}^{ave} = \frac{1}{N}\left(a\sigma_{\varepsilon_i}^2 + b\sigma_{\varepsilon_j}^2\right) + \left(a\beta_i^2 + b\beta_j^2\right).$$

This means that for biased estimation errors simple averaging reduces by a factor N the component of r_{add} which depends on the variance of the estimation errors (see Eq. (38)). The component depending on the bias does not change.

In the general case in which the estimation errors of individual classifiers exhibit different values of bias and variance, different weights are required in order to achieve the minimum of r_{add}^{ave}.

5.1.3 Unbiased and correlated estimation errors

For an individual classifier the value of r_{add} is the same of the uncorrelated case:

$$r_{add} = a\sigma_{\varepsilon_i}^2 + b\sigma_{\varepsilon_j}^2 .$$

Taking into account the correlation between the estimation errors of different classifiers, for a linear combination of classifiers we obtain:

$$r_{add}^{ave} = a\sigma_{\varepsilon_i}^2 + b\sigma_{\varepsilon_j}^2 =$$
$$= a\sum_{k=1}^{N} w_k^2\sigma_{\varepsilon_i^k}^2 + b\sum_{k=1}^{N} w_k^2\sigma_{\varepsilon_j^k}^2 + 2a\sum_{m<n} w_m w_n \operatorname{cov}\left\{\varepsilon_i^m, \varepsilon_i^n\right\} + 2b\sum_{m<n} w_m w_n \operatorname{cov}\left\{\varepsilon_j^m, \varepsilon_j^n\right\} =$$
$$= \sum_{k=1}^{N} w_k^2 r_{add}^k + 2\sum_{m<n} w_m w_n\left(a\rho_i^{mn}\sigma_{\varepsilon_i^m}\sigma_{\varepsilon_i^n} + b\rho_j^{mn}\sigma_{\varepsilon_j^m}\sigma_{\varepsilon_j^n}\right).$$

In the above expression, the terms $\operatorname{cov}\left\{\varepsilon_i^m, \varepsilon_i^n\right\}$ and ρ_i^{mn} denote respectively the covariance and the correlation coefficient between the estimation errors of classifiers m and n on the i-th class. This expression is similar to the one obtained for the biased uncorrelated case in the previous paragraph. It shows that the optimal values of the weights w_k depend also on the correlation between the estimation errors of individual classifiers, besides their performances. The weights which minimise r_{add}^{ave} can be easily computed only if the estimation errors of each individual

classifier exhibit the same variance, and the same value of the correlation coefficient, that is $\rho_i^{mn} = \rho_i, \quad \forall m, n$. Under this hypothesis, the optimal weights correspond again to simple averaging, that is, $w_k = 1/N$. The value of r_{add}^{ave} is:

$$r_{add}^{ave} = \frac{1}{N}\left(a\sigma_{\varepsilon_i}^2 + b\sigma_{\varepsilon_j}^2\right) + \frac{N-1}{N}\left(a\rho_i\sigma_{\varepsilon_i}^2 + b\rho_j\sigma_{\varepsilon_j}^2\right) .$$

The above expression can be simplified by assuming that the correlation coefficients of classes *i* and *j* are equal, that is $\rho_i = \rho_j = \rho$. We obtain:

$$r_{add}^{ave} = \frac{1+\rho\left(N-1\right)}{N} r_{add} .$$

This expression points out the effects of the correlation between estimation errors. It shows that if the estimation errors of the individual classifiers are positively correlated, that is, $\rho > 0$, simple averaging improves the error-reject trade-off by a factor lower than N. An improvement by a factor equal to N is achieved if the estimation errors are uncorrelated ($\rho = 0$): this is the case analysed in paragraph 5.1.2. Finally, if the estimation errors are negatively correlated ($\rho < 0$), simple averaging improves the error-reject trade-off by a factor greater than N. We point out that this result formalises the advantage of negative correlation for the linear combination of classifiers. Other authors hypothesised that classifier combination can benefit from negative correlation between individual calssifiers. Some experimental results reported in the literature, for instance by Kuncheva and Duin (2000) for the majority rule, were in agreement with this assumption.

5.1.4 Biased and correlated estimation errors

The expected value of the added risk of an individual classifier is the same of the uncorrelated case:

$$r_{add} = a\left(\sigma_{\varepsilon_i}^2 + \beta_i^2\right) + b\left(\sigma_{\varepsilon_j}^2 + \beta_j^2\right) .$$

For a linear combination of classifiers we obtain:

$$\begin{aligned} r_{add}^{ave} &= a\left[\sigma_{\bar{\varepsilon}_i}^2 + \left(\bar{\beta}_i\right)^2\right] + b\left[\sigma_{\bar{\varepsilon}_j}^2 + \left(\bar{\beta}_j\right)^2\right] = \\ &= \sum_{k=1}^{N} w_k^2 r_{add}^k + 2\sum_{m<n} w_m w_n \left[a\left(\rho_i^{mn}\sigma_{\varepsilon_i^m}\sigma_{\varepsilon_i^n} + \beta_i^m\beta_i^n\right) + b\left(\rho_j^{mn}\sigma_{\varepsilon_j^m}\sigma_{\varepsilon_j^n} + \beta_j^m\beta_j^n\right)\right] . \end{aligned}$$

This expression generalises the ones obtained for unbiased and for uncorrelated errors. Also in this case it turns out that if the estimation errors of the individual classifiers exhibit the same bias, variance, and correlation, then the optimal weights are $w_k = 1/N$. Under the same hypothesis about the correlation coefficients made in the previous paragraph ($\rho_i = \rho_j = \rho$), we obtain:

$$\begin{aligned} r_{add}^{ave} &= \frac{1}{N}\left[a\left(\sigma_{\varepsilon_i}^2 + \beta_i^2\right) + b\left(\sigma_{\varepsilon_j}^2 + \beta_j^2\right)\right] + \frac{N-1}{N}\left[\rho\left(a\sigma_{\varepsilon_i}^2 + b\sigma_{\varepsilon_j}^2\right) + \left(a\beta_i^2 + b\beta_j^2\right)\right] = \\ &= \frac{1+\rho\left(N-1\right)}{N}\left(a\sigma_{\varepsilon_i}^2 + b\sigma_{\varepsilon_j}^2\right) + \left(a\beta_i^2 + b\beta_j^2\right) . \end{aligned}$$

This expression, like the ones found in the previous paragraphs, shows that in the general case simple averaging reduces only the component of r_{ave} depending on the variance of the estimation errors, by the same factor of the unbiased case. The component depending on the bias does not change. The same considerations made in the previous paragraph about positive and negative correlation apply here.

5.2 Discussion

The above analysis shows how linearly combining classifiers in output space can improve their error-reject trade-off. In particular, it points out the conditions under which simple averging or weighted averaging are required in order to optimise the error-reject trade-off of the combiner. The main result of the above analysis is that simple averaging is the optimal combining rule for classifiers whose estimation errors on the a posteriori probabilities exhibit equal values of bias, variance, and correlation. Note that this implies that the individual classifiers exhibit equal average performances. We can define these as balanced classifiers. Instead, weighted averaging provides the best error-reject trade-off for ensembles of imbalanced classifiers. As already pointed out in paragraph 5.1.1, reserchers agree that simple combining rules (like simple averaging) are best suited for problems where the individual classifiers have comparable average performances (Tumer and Ghosh, 1999). The results of our analysis formalise this conclusion, and extend it to classification with reject option. However, we point out that the definition of imbalanced classifiers given above is still a qualitative one. Further work is needed to obtain a quantitative definition of classifier imbalancing, which should be strongly related to the performance improvement achievable by weighted averaging over simple averaging. Such a definition would be useful in practical applications to decide if, for a given ensemble of classifiers, it is worth using weighted averaging. Note indeed that obtaining good estimates of the optimal weights could be difficult, besides than computationally expensive (Tumer and Ghosh, 1999). Moreover, experimental results reported in the literature, for instance by Ueda (2000), showed a quite small difference between the performances of simple and weighted averaging.

Chapter 6
A method for introducing the reject option in support vector machines

In this Chapter we propose a method to introduce the reject option in SVMs. Currently, the reject option in SVMs is implemented by using a heuristic rule, as pointed out in Chapter 3. In our opinion this is in contrast with the strong theoretical foundations of SVMs. In paragraph 6.1 we summarise the main results of statistical learning theory. In paragraph 6.2 we describe the theoretical derivation of SVMs. In paragraph 6.3 we show how the reject option can be introduced using the same theoretical derivation. This would allow to obtain the reject region, together with the decision regions, as a result of training a SVM. We then propose a formulation of the problem of training a SVM with reject option. Finally, in paragraph 6.4 we show how a simple and efficient algorithm proposed in the literature for standard SVMs can be modified to fit the characteristics of this problem.

6.1 Overview of statistical learning theory

Statistical learning theory was developed by V. Vapnik since the early 1960's (Vapnik, 1998). This theory deals with the general problem of function estimation from a collection of data, and encompasses problems such as pattern recognition, regression estimation, and density estimation. In the following we focus on the pattern recognition problem, whose setting is analogous to that of the minimum risk theory, which was presented in Chapter 2.

A classifier is viewed as a *learning machine* which can implement a set of decision functions $f(\mathbf{x},\alpha)$, $\alpha \in \Lambda$, whose output is a class label. The parameter α denotes one particular decision function of the set. For instance, if the classifier is a neural network of given structure, then α represents one particular set of connection weights. The problem is that of choosing from the given set of functions the one, denoted as $f(\mathbf{x},\alpha_0)$, which minimises the expected value $R(\alpha)$ of a loss function $L(\mathbf{x},\omega,f(\mathbf{x},\alpha))$:

$$R(\alpha) = \sum_{i=1}^{c} \int L\left(\mathbf{x},\omega_i,\ f\left(\mathbf{x},\alpha\right)\right) p\left(\mathbf{x},\omega_i\right) d\mathbf{x} \ .$$

The probability function $p(\mathbf{x},\omega) = p(\mathbf{x})P(\omega \mid \mathbf{x})$ is assumed to be unknown, and therefore $R(\alpha)$ itself is unknown. The decision function can only be chosen on the basis of l training samples:

$$(\mathbf{x}_1,\omega^1),\ (\mathbf{x}_2,\omega^2),\ ...,\ (\mathbf{x}_l,\omega^l) \ ,$$

which are assumed to be drawn randomly and independently from the (unknown) joint probability density function $p(\omega,\mathbf{x})$.

The induction principle usually used in pattern recognition is called *empirical risk minimisation* (ERM principle). It consists in choosing the function which minimises an approximation of $R(\alpha)$ constructed on the basis of the training set, called *empirical risk*:

$$R_{emp}(\alpha) = \frac{1}{l}\sum_{i=1}^{l} L\left(\omega^i,\, f(\mathbf{x}_i,\alpha)\right) .$$

Using the simplest loss function:

$$L(\mathbf{x},\omega,\alpha) = \begin{cases} 0, & \text{if } f(\mathbf{x},\alpha) = \omega , \\ 1, & \text{if } f(\mathbf{x},\alpha) \neq \omega , \end{cases} \tag{39}$$

the actual expected risk $R(\alpha)$ is equal to the error probability, and the empirical risk $R_{emp}(\alpha)$ is the misclassification rate on the training set. The function (39) is named *indicator* function, since it takes only the two values zero and one.

The analysis of the condition of consistency of the ERM principle leads to two main theoretical results. First, it turns out that the conditions of consistency are related to the concept of *capacity*, or *VC dimension* (Vapnik-Chervonenkis dimension) of a learning machine. The VC dimension of a classifier is a measure of its *complexity*, that is, of the capability of fitting a given training set with its decision rules $f(\mathbf{x},\alpha)$. In particular the VC dimension depends on the set $f(\mathbf{x},\alpha)$ and on the loss function used. For indicator functions (39), the VC dimension is defined as the maximum number h of vectors $(\mathbf{x}_1,\omega^1)$, $(\mathbf{x}_2,\omega^2)$, ... , $(\mathbf{x}_h,\omega^h)$ which can be separated in all 2^h possible ways (*shattered*) using the set of functions $L(\mathbf{x},\omega,\alpha)$. If for any n there exists a set of n vectors that can be shattered by the set, then the VC dimension is equal to infinity. Note that any indicator function separates a set of vectors into two subsets: the subset of vectors for which the function takes value zero, and the subset for which it takes value one.

The second of the main theoretical results of statistical learning theory is the definition of an upper bound for the expected risk achieved by any function $f(\mathbf{x},\alpha)$, which depends on the VC dimension. Using bounded loss functions $0 \le L(\mathbf{x},\omega,\alpha) \le B$, the following inequality holds true for any function $f(\mathbf{x},\alpha)$, with probability at least $1 - \eta$:

$$R(\alpha) \le R_{emp}(\alpha) + \frac{B\varepsilon}{2}\left(1 + \sqrt{1 + \frac{4R_{emp}(\alpha)}{B\varepsilon}}\right) , \tag{40}$$

where

$$\varepsilon = 4\frac{h\left(\ln\frac{2l}{h} + 1\right) - \ln\eta}{l} ,$$

and h denotes the VC dimension of the set of decision functions. Note that for indicator loss functions, $B = 1$. The above inequality states that the actual risk achieved using any function $f(\mathbf{x},\alpha)$ is upper bounded by the sum of the empirical risk achieved by $f(\mathbf{x},\alpha)$, and of a term depending on the VC dimension h of the set of functions. It turns out that when the ratio l/h is large, the second summand on the right hand side becomes small. The actual risk $R(\alpha)$ is then close to the value of the empirical risk $R_{emp}(\alpha)$. This means that, if the size of the training set is large enough with respect to the VC dimension of the classifier used, minimising the empirical risk guarantees to minimise the actual risk, that is, to achieve a good generalisation capability. Instead, if l/h is small, a small $R_{emp}(\alpha)$ does not guarantee a small value of $R(\alpha)$. Therefore the

ERM principle is suitable only for dealing with large sample size, with respect to the VC dimension of the chosen classifier. Note that for "large" values of l/h Vapnik means $l/h > 20$.

Inequality (40) shows that the ERM principle does not guarantee to minimise the actual risk. This result leads to the definition of a new induction principle, which is called the principle of *structural risk minimisation* (SRM). Basically, this principle is based on controlling the generalisation ability of a learning machine by reaching a trade-off between its complexity and the *empirical risk*, that is, the performance achievable on training data. Indeed, to obtain a small value of the upper bound on the actual risk $R(\alpha)$, the two terms in the right-hand side of inequality (40) should be simultaneously minimised. However, these two goals are conflicting, since decreasing the VC dimension means decreasing the complexity of the classifier, and this can result in higher values of the empirical risk, that is, a lower performance on training data. Therefore a trade-off must be found between the performance achievable on the training set and the VC dimension of the classifier. The SRM principle is based on defining a *structure* on the set S of loss functions $L(\mathbf{x},\omega,\alpha)$, so that S is composed of nested subsets

$$S_1 \subset S_2 \subset \ldots \subset S_n \ldots$$

having finite VC dimension h_k. Since the above structure is composed by nested subsets, from the definition of VC dimension it follows that:

$$h_1 \leq h_2 \leq \ldots \leq h_n \ldots .$$

For the same reason, denoting with $R_{emp}(\alpha_k)$ the minimum value of the empirical risk achievable using functions of S_k, it turns out that:

$$R_{emp}(\alpha_1) \geq R_{emp}(\alpha_2) \geq \ldots \geq R_{emp}(\alpha_n) \ldots$$

This means that lower values of the VC dimension correspond to higher values of the achievable empirical risk, as pointed out above. In terms of the right-hand side of inequality (40), this means that for decreasing values of the first term (the empirical risk), the second term (which depends on the VC dimension) increases. Therefore the SRM principle suggests to choose, for a given training set, the subset S_k and the particular function α_k from S_k for which the upper bound (40) is minimum.

The practical application of the SRM principle leads to the definition of support vector machines, which is described in the following paragraph.

6.2 Theoretical derivation of support vector machines

6.2.1 The optimal separating hyperplane

The application of the SRM principle to the problem of pattern recognition was studied by Vapnik for the simplest case of a two-class problem with indicator loss function (39). Note that the reject option was not considered. Vapnik considered also the simplest type of classifier, consisting of a set of linear decision functions:

$$f(\mathbf{x},\alpha) = \text{sign}(\mathbf{w}\cdot\mathbf{x} + b),\ \mathbf{x},\ \mathbf{w} \in \Re^d,\ b \in \Re\ , \tag{41}$$

where the parameter α denotes the pair ($\mathbf{w}$, b), and the class labels are {+1,-1}. Vapnik showed that the VC dimension of the set of all linear decision functions in a feature space of dimension d is d+1. To apply the SRM principle, a structure on the corresponding set of indicator loss functions must be defined. That is, subsets of the set (41) of linear decision functions having a VC dimension lower than d+1 must be found. This can be achieved by exploiting a result related to the concept of *margin* of a hyperplane.

Given a set $X = (\mathbf{x}_1,..., \mathbf{x}_l)$ of l vectors in R^d, the margin ρ of an hyperplane $\mathbf{w}\cdot\mathbf{x} + b$ is defined as the minimum distance between the vectors of X and the hyperplane:

$$\rho = \min_{\mathbf{x}_i \in X} \frac{|\mathbf{w}\cdot\mathbf{x}_i + b|}{\|\mathbf{w}\|} . \tag{42}$$

Let R be the radius of the smallest sphere containing X. Vapnik showed that the VC dimension h of the set of hyperplanes which separate the set X with margin at least ρ, is bounded by the inequality

$$h \le \min\left(\left\lfloor \frac{R^2}{\rho^2} \right\rfloor, d \right) + 1 , \tag{43}$$

where $\lfloor a \rfloor$ denotes the integer part of a.

This result can be exploited to apply the SRM princilpe in the case of a linearly separable training set X, by considering the set of hyperplanes which separate X without errors (that is, with a null empirical risk). These are called *separating hyperlpanes*. A structure S on the set of separating hyperplanes can indeed be constructed exploiting inequality (43). The subsets S_k consist of the set of separating hyperplanes which subdivide the training set X with minimum margin ρ_k such that:

$$\begin{aligned} &\rho_k^2 > \frac{R^2}{k+1}, \quad k = 1,\ldots,d-1, \\ &\rho_k^2 > 0, \quad k = d. \end{aligned} \tag{44}$$

From inequality (43), it turns out that the VC dimensions h_k of subsets S_k has the following upper bound:

$$h_k \le k + 1 .$$

Note that the number of subsets S_k can be lower than d, depending on the maximum margin between patterns of different classes of X. Since all the members of the subsets S_k have a null empirical risk, applying the SRM principle means choosing one of the separating hyperplanes which belong to the subset with minimum upper bound on the VC dimension (40). This can be achieved without computing the margin values of inequalities (44), by simply finding the separating hyperplane with maximum margin. This hyperplane obviously belongs to the subset with minimum upper bound on the VC dimension, and is called *optimal separating hyperplane* (OSH).

Vapnik showed that finding the OSH is appealing from the algorithmic point of view, for two main reasons. First, he showed that the OSH is unique. Secondly, he showed that the OSH can be found by solving a simple optimisation problem, for which efficient techniques from optimisation

theory can be used. Indeed any separating hyperplane $\mathbf{w}\cdot\mathbf{x}+b$ satisfies the constraints $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 0$, $i = 1, \ldots, l$, where $y_i = \{+1,-1\}$ is the class label of the training pattern $\mathbf{x}_i$. The margin of the hyperplane is given by Eq. (42). By rescaling the equation of the hyperplane so that $\min_i |\mathbf{w}\cdot\mathbf{x}_i + b| = 1$, the above conditions become $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1$, $i = 1, \ldots, l$, while the margin becomes $\rho = 1/\|\mathbf{w}\|$. This allows to express the OSH as the solution of a *quadratic programme*, that is, an optimisation problem with quadratic objective function and linear constraints:

$$\begin{aligned} &\text{minimise} \quad \frac{1}{2}\|\mathbf{w}\|^2, \\ &\text{subject to} \quad y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1, \quad i = 1,\mathrm{K},l. \end{aligned} \tag{45}$$

In optimisation theory, problem (45) is called *primal* problem. Vapnik showed that the main properties of the OSH can be emphasised by solving the *dual* problem associated to (45). The solution of the primal problem coincides with the saddle point of the corresponding Lagrange function:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i \left[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 \right],$$

where $\alpha_i \geq 0$ are the Lagrange multipliers. To find the saddle point, the Lagrange function must be minimised over the primal variables $\mathbf{w}$ and b, and maximised over the α_i. By imposing stationarity with respect to $\mathbf{w}$ and b:

$$\begin{aligned} \frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{l} y_i \alpha_i \mathbf{x}_i = 0, \\ \frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} &= \sum_{i=1}^{l} y_i \alpha_i = 0, \end{aligned} \tag{46}$$

and substituting the relations obtained in the Lagrange function, it turns out that the saddle point of the Lagrange function is the solution of the following problem:

$$\begin{aligned} &\text{maximise} \quad \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j), \\ &\text{subject to} \quad \alpha_i \geq 0, \quad i = 1,\mathrm{K},l, \\ &\qquad\qquad \sum_{i=1}^{l} y_i \alpha_i = 0. \end{aligned} \tag{47}$$

This is the dual problem, and the Lagrange multipliers α_i are called dual variables. From optimisation theory it follows that necessary and sufficient conditions for values of $\mathbf{w}$, b and α_i to be solutions of the primal (45) and dual (47) problems are the so-called Karush-Kuhn-Tucker (KKT) conditions:

$$\alpha_i \left[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 \right] = 0, \quad i = 1, \ldots, l.$$

Given the α_i which solve the dual problem (47), from Eq. (46) it follows that the value of $\mathbf{w}$ which solve the primal problem is:

$$\mathbf{w} = \sum_{i=1}^{l} y_i \alpha_i \mathbf{x}_i. \tag{48}$$

This means that the weight vector of the OSH can be expressed as a linear combination of the training points. Furthermore, from the KKT conditions it turns out that the only non-zero α_i correspond to the points $\mathbf{x}_i$ for which $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) = 1$. It is easy to see that these are the closest points to the OSH: indeed their distance from the OSH corresponds to the margin $1/\|\mathbf{w}\|$. The OSH can then be expressed as a linear combination of the training points closest to it. These points are therefore called *support vectors*. The value of b can be obtained from the KKT condition associated to any of the support vectors (for which $\alpha_i \neq 0$), by solving the corresponding equation $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) = 1$. The equation of the OSH can then be rewritten as:

$$\sum_{i \in SV} y_i \alpha_i (\mathbf{x}_i \cdot \mathbf{x}) + b ,$$

where SV denotes the set of support vectors. The example of Fig. 5 shows the OSH and the support vectors for a problem with a two-dimensional feature vector.

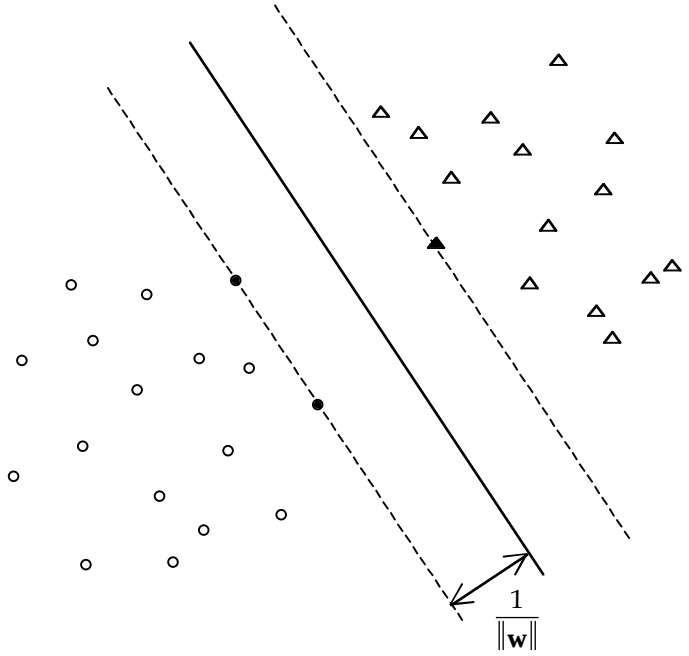


Fig. 5. The optimal separating hyperplane for the two classes of circles and traingles is shown as a solid line. The dashed lines represent the margin. The support vectors are shown as filled circles and triangles.

6.2.2 The optimal separating hyperplane for the non-separable case

The OSH as defined above is of no practical use, since it is based on the assumption that the training set is linearly separable. This condition does not hold in many practical applications, and in any case it can be difficult to verify if it holds. Note that for a non-linearly separable training set the constraints of problem (45) can not simultaneously hold true, and therefore this problem has no solution. To generalise the concept of OSH to the non-separable case Vapnik proposed to introduce non-negative variables $\xi_1,\ldots, \xi_l$, to allow the constraints to be violated: $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1-\xi_i$ (Cortes and Vapnik, 1995). The objective function of problem (45) was consequently modified to $\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i$, where C is a given positive value. As pointed out by

Vapnik (1998), the summand in the new objective function is an upper bound for the number of training errors. The primal problem for the non-linearly separable case is then the following:

$$\begin{aligned} &\text{minimise} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i \,, \\ &\text{subject to } y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) \geq 1-\xi_i, \quad i = 1,\ldots,l \,, \\ &\qquad\qquad \xi_i \geq 0, \quad i = 1,\ldots,l \,. \end{aligned} \tag{49}$$

The corresponding dual problem is similar to the one of the linearly separable case (47). The only difference is one additional constraint on the dual variables:

$$\begin{aligned} &\text{maximise} \quad \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right) , \\ &\text{subject to } 0 \leq \alpha_i \leq C, \quad i = 1,\ldots,l \,, \\ &\qquad\qquad \sum_{i=1}^{l} y_i \alpha_i = 0 \,. \end{aligned} \tag{50}$$

The KKT conditions are now:

$$\begin{aligned} &\alpha_i\left[y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) - 1 + \xi_i\right], \quad i = 1,\ldots,l \,, \\ &\xi_i\left(\alpha_i - C\right) = 0, \quad i = 1,\ldots,l \,. \end{aligned} \tag{51}$$

The value of $\mathbf{w}$ is given also in this case by Eq. (48). The support vectors are again defined as the points $\mathbf{x}_i$ corresponding to non-zero α_i. Pontil and Verri (1998) pointed out that from the KKT conditions it follows that the support vectors consist of all points misclassified by the OSH, and of the ones correctly classified whose distance from the OSH is less than the margin $1/\|\mathbf{w}\|$. For training points correctly classified which lie outside the margin (that is, their distance from the OSH is greater than $1/\|\mathbf{w}\|$), it turns out that $\xi_i = 0$. For the other points, the term ξ_i is proportional to the amount by which they fail to reach a margin at least equal to $1/\|\mathbf{w}\|$ from the correct side of the OSH. In particular, for misclassified points $\xi_i \geq 1$ (this explains why the summand on the objective function of problem (49) is an upper bound for the number of training errors). Pontil and Verri (1998) pointed out that this objective function represents a trade-off between the margin $1/\|\mathbf{w}\|$ and the number of training errors. It was also shown that minimising this objective function can be viewed as minimising an upper bound on the actual error probability, analogous to the upper bound on the actual risk (40) (Cristianini and Shawe-Taylor, 2000). An example of the generalised OSH is shown in Fig. 6.

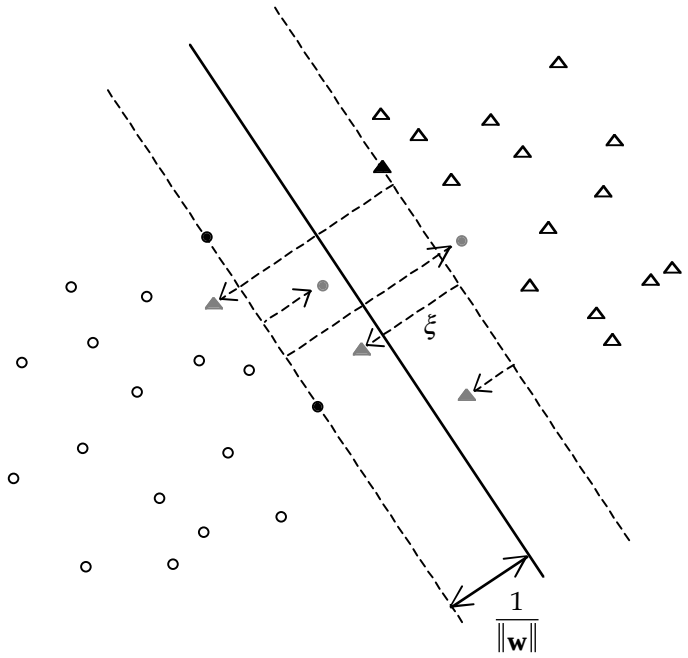


Fig. 6. The optimal separating hyperplane for a non-linearly separable training set. The support vectors are shown as filled circles and triangles (in black the margin vectors, in gray the other ones). The slack variables for non-margin support vectors (shown in grey) are also shown.

6.2.3 Support Vector Machines

The complexity of many real pattern recognition problems requires non-linear decision functions. However the results described in the previous paragraph are valid only for linear decision functions. Nonetheless, Vapnik (1998) showed that the technique of the OSH can be easily applied to the case of non-linear decision functions. Basically, the idea is to map the original feature space D into a high-dimensional space D' through a non-linear mapping chosen a priori, and to construct the OSH in D'. The decision surface in D corresponding to the OSH in D' is then non-linear. There is still a problem: finding the OSH in high-dimensional spaces can be computationally infeasible. However, this problem is overcomed thanks to fact that the expressions of the dual problem (50) and of the weight vector $\mathbf{w}$ (48) depend only on the inner product between vectors of D'. It is then possible to avoid computing explicitely the mapping from D to D' exploiting the fact that there exist functions $K(\mathbf{x},\mathbf{y})$, $\mathbf{x}, \mathbf{y} \in D$, which represent the inner product between the images of $\mathbf{x}$, $\mathbf{y}$ in D'. These functions are characterised by the Mercer theorem (Vapnik, 1998). This theorem states that any continuous symmetric function $K(\mathbf{x},\mathbf{y})$ in $L_2(C)$, $\mathbf{x}, \mathbf{y} \in D \subseteq R^d$, satisfying

$$\int_C \int_C K(\mathbf{x},\mathbf{y}) g(\mathbf{x}) g(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0 \quad \forall g \in L_2(C),$$

represents the inner product $\phi(\mathbf{x}) \cdot \phi(\mathbf{y})$ in a feature space D' defined by an *unknown* mapping ϕ (C being a compact subset of $\Re^d$). This means that the dual problem can be solved in D' simply substituting $K(\mathbf{x}_i,\mathbf{x}_j)$ to the inner products $(\mathbf{x}_i \cdot \mathbf{x}_j)$. The expression of the separating surface in D, corresponding to the OSH in D', becomes then:

$$\sum_{i=1}^{l} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \,. \tag{52}$$

Therefore one can choose the desired form of the decision surface (52) among the functions which satisfy Mercer conditions, and solve the dual problem using the values $K(\mathbf{x}_i,\mathbf{x}_j)$. Functions which satisfy Mercer conditions are called *kernel functions*. Well-known kernels are polynomials of degree n:

$$K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x} \cdot \mathbf{x}_i + 1)^n \,,$$

and radial basis kernels:

$$K_\gamma(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\gamma |\mathbf{x} \cdot \mathbf{x}_i|^2\right) \,.$$

The classifier based on the concept of OSH and on kernel functions is called *support vector machine*.

6.2.4 Algorithms for training support vector machines

Training a SVM consists in choosing a kernel function and the value of the parameter C, and solving the optimisation problem:

$$\begin{aligned} &\text{maximise} \quad \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i \cdot \mathbf{x}_j) \,, \\ &\text{subject to} \, 0 \le \alpha_i \le C, \quad i = 1, \ldots, l \,, \\ &\sum_{i=1}^{l} y_i \alpha_i = 0 \,, \end{aligned}$$

which consists in maximising a concave quadratic form under linear constraints. This problem has several interesting properties. Since the dual objective function is concave, it has no local maxima, and therefore its solution is unique. The solution is sparse, since the only non-zero α_i are the ones corresponding to the support vectors, which are tipically a small subset of the training patterns. Moreover, the implicit mapping into high-dimensional feature spaces makes the complexity of this problem independent on the dimensionality of the feature space. In principle, standard optimisation techniques could be used to solve this problem. However, these techniques require the entire kernel matrix $K(\mathbf{x}_i,\mathbf{x}_j)$ to be stored in memory. This is infeasbile for training sets of more than few hundred patterns, since the kernel matrix requires a memory space which grows quadratically with the size of the training set. For this reason several specific techniques have been developed for training SVMs (Cristianini and Shawe-Taylor, 2000). These techniques are mainly based on two heuristics, known as *chunking* and *decomposition*, which avoid to store the entire kernel matrix in memory, while exploiting the characteristics of the dual problem described above. In particular, the decomposition technique is based on optimising the dual objective function by iteratively acting on only a fixed size subset of the dual variables α_i, while keeping constant the others. A natural stopping criterion for these optimisation algorithms are the KKT conditions, since they are necessary and sufficient to characterise the solution of the primal and dual problems (Cristianini and Shawe-Taylor, 2000).

A simple and efficient algorithm is the Sequential Minimal Optimisation (SMO) algorithm (Platt, 1999b). It uses the decomposition method, by acting on the minimum number of variables at each iteration. Due to the constraint $\sum_{i=1}^{l} y_i \alpha_i = 0$, the minimum number of variables is two. The reason of this choice is that the resulting optimisation problem can be solved analytically. Basically, SMO iteratively chooses a pair of dual variables α_i, α_j using a heuristic, and analytically finds the maximum of the dual objective function with respect to these variables, by keeping constant the other ones. The maximum is found under the constraints of the dual problem $0 \le \alpha_i, \alpha_j \le C$, and $\sum_{i=1}^{l} y_i \alpha_i = 0$. This strategy guarantees that the current values of the dual variables are always a feasible solution of the dual problem. In this case the KKT conditions (51) are necessary and sufficient to characterise the solution of the primal and dual problems. These conditions are therefore used as a stopping criterion for the algorithm. Taking into account that the dual function is concave, the above strategy also guarantees the convergence of the algorithm if a proper selection of the pairs α_i, α_j is made at each iteration.

To speed up the convergence, two heuristics are used to choose the pairs α_i, α_j. The first variable of the pair is chosen in the outer loop of the algorithm, among the variables which violate the KKT conditions. The second variable is chosen using the second heuristic. If at least one pair of variables is updated in the outer loop, the next outer loop is made only on the non-bound variables, that is, on the ones such that $0 < \alpha_i < C$. This is another heuristic, which aims to increase the chances to find KKT violations. The outer loop is repeated until no variable violates the KKT conditions (the last outer loop is always a complete loop). In this case the algorithm terminates (as explained above, the KKT conditions are used as stopping criterion). The second heuristic consists in choosing the second variable α_j of the pair so that their updating causes a large change, which should result in a large increase of the dual objective function. If this choice does not provide a significant change, SMO looks first for each non-bound α_j, and then through the entire training set.

The maximum of the dual objective function with respect to two variables α_i and α_j can be found analytically as follows. Note first that the constraint $\sum_{i=1}^{l} y_i \alpha_i = 0$ implies that α_i and α_j must lie on the line of equation

$$y_i \alpha_i + y_j \alpha_j = \text{constant} = y_i \alpha_i^{old} + y_j \alpha_j^{old} .$$

This allows to find the new value of just one variable, say α_i, and use it to find the new value of α_j from the above equation. The other constraint $0 \le \alpha_i, \alpha_j \le C$, together with the one above, implies the following constraint for α_i:

$$U < \alpha_i < V , \tag{53}$$

where $U = \max\left\{0, \alpha_i^{old} - \alpha_j^{old}\right\}$, $V = \min\left\{C, C - \alpha_j^{old} + \alpha_i^{old}\right\}$, if $y_i \neq y_j$, and $U = \max\left\{0, \alpha_i^{old} + \alpha_j^{old} - C\right\}$, $V = \min\left\{C, \alpha_i^{old} + \alpha_i^{old}\right\}$, if $y_i = y_j$ (Platt, 1999b). The unconstrained maximum of the dual objective function with respect to α_i is achieved by first computing:

$$\alpha_i^{unc} = \alpha_i^{old} + \frac{y_i\left(\sum_{k=1}^{l}\alpha_k y_k K(\mathbf{x}_k,\mathbf{x}_j) - y_j - \sum_{k=1}^{l}\alpha_k y_k K(\mathbf{x}_k,\mathbf{x}_i) + y_i\right)}{K(\mathbf{x}_i,\mathbf{x}_i) + K(\mathbf{x}_j,\mathbf{x}_j) - 2k(\mathbf{x}_i,\mathbf{x}_j)}, \tag{54}$$

and then, taking into account constraint (53):

$$\alpha_i = \begin{cases} V, & \text{if } \alpha_i^{unc} > V, \\ \alpha_i^{unc}, & \text{if } U \le \alpha_i^{unc} \le V, \\ U, & \text{if } \alpha_i^{unc} < U. \end{cases}$$

Note that to find a good variable α_j with few computation in the second heuristic, SMO chooses the α_j for which the absolute value of the term between round brackets at the numerator of Eq. (54) is maximum. Note also that to evaluate the KKT conditions in the outer loop, the value of *b* should be known in advance. SMO uses a value obtained by *imposing* the KKT conditions for the two dual variables which are modified at each iteration. It was shown that this value converges to the optimal value of *b* as the values of the objective function converges to its maximum. A basic scheme of SMO is given below.

```
alpha[]: vector of dual variables

main routine
  initialise alpha array to zero
  initialise b to zero
  numChanged = 0
  examineAll = 1
  while (numChanged > 0 || examineAll == 1)
  {
    numChanged = 0
    if (examineAll)
      loop i over all training examples
        numChanged += examineExample(i)
    else
      loop over examples whose alpha is not 0 and not C
        numChanged += examineExample(i)
    if (examineAll == 1)
      examineAll = 0
    else if (numChanged == 0)
      examineAll = 1
  }

procedure examineExample(i)
  if example i violates the KKT conditions
  {
    if (number of non-zero and non-C alpha > 1)
    {
      j = result of second choice heuristic
      if takeStep(i,j)
        return 1
    }
    loop j over non-zero and non-C alpha, starting at a random point
      if takeStep(i,j)
        return 1
    loop j over all training examples, starting at a random point
      if takeStep(i,j)
```

```
        return 1
  }
  return 0
endprocedure

procedure takeStep(i,j)
  if (i == j) return 0
  compute the value of alpha[i] which maximises the dual objective function,
    with respect to alpha[i] and alpha[j], under the dual constraints
  if the change of alpha[i] and alpha[j] is below a predefined tolerance,
    return 0
  update alpha[i] and alpha[j]
  update the threshold b
  return 1
endprocedure
```

6.3 Introducing the reject option in support vector machines

6.3.1 Problem formulation

In Chapter 3 we showed that the reject option in SVMs is currently implemented using a heuristic rule. Indeed, as shown in paragraph 6.2, the SVMs have been derived from statistical learning theory without taking into account the reject option. However, we point out that the reject option can be introduced using the same theoretical derivation. To this aim, consider again inequality (40), which represents an upper bound for the expected risk of a given classifier. This inequality involves the empirical risk and the VC dimension. Both these terms depend on the loss function $L(\mathbf{x},\omega,\alpha)$. Note that the definitions of empirical risk and of VC dimension, as well as inequality (40), hold for *any* bounded function $L(\mathbf{x},\omega,\alpha)$ (Vapnik, 1998; 1999). This means that they also hold for loss functions like the ones described in Chapter 2, in the case of classification with reject option. Therefore the SRM principle, which is based on inequality (40), can be applied also for classification with reject option. The key point is to define a set of decision functions $f(\mathbf{x},\alpha)$ whose output can be also the reject decision, besides one of the class labels. In the following we propose a generalisation of the concept of optimal separating hyperplane to classification with reject option.

Following Vapnik's approach to the derivation of the OSH, which has been described in paragraph 6.2, we consider a two-class problem, and the simplest loss function. As pointed out in Chapter 2, in classification with reject option the simplest loss function is:

$$L(\mathbf{x},\omega,\alpha) = \begin{cases} 0, & \text{if } f(\mathbf{x},\alpha) = \omega, \\ w_R, & \text{if } f(\mathbf{x},\alpha) = 0, \\ 1, & \text{if } f(\mathbf{x},\alpha) \neq \omega \text{ and } f(\mathbf{x},\alpha) \neq 0, \end{cases} \tag{55}$$

where $0 \leq w_R \leq 1$, and the output of $f(\mathbf{x},\alpha)$ corresponding to the reject region is denoted with 0. Now a set of decision functions $f(\mathbf{x},\alpha)$ must be chosen. Vapnik focused on the simplest ones, namely, linear decision functions. In our case, the simplest way to deal with the reject decision

using linear functions is to consider *pairs* of parallel hyperplanes, such that patterns lying between them are rejected. Formally, let us write the expressions of a pair of parallel hyperplanes as:

$$\mathbf{w}\cdot\mathbf{x}+b\pm\varepsilon, \quad \mathbf{w}\in\Re^d, \quad b,\varepsilon\in\Re, \quad \varepsilon\geq 0\,, \tag{56}$$

where α repersents the parameters $\mathbf{w}$, b, ε (see Fig. 7).

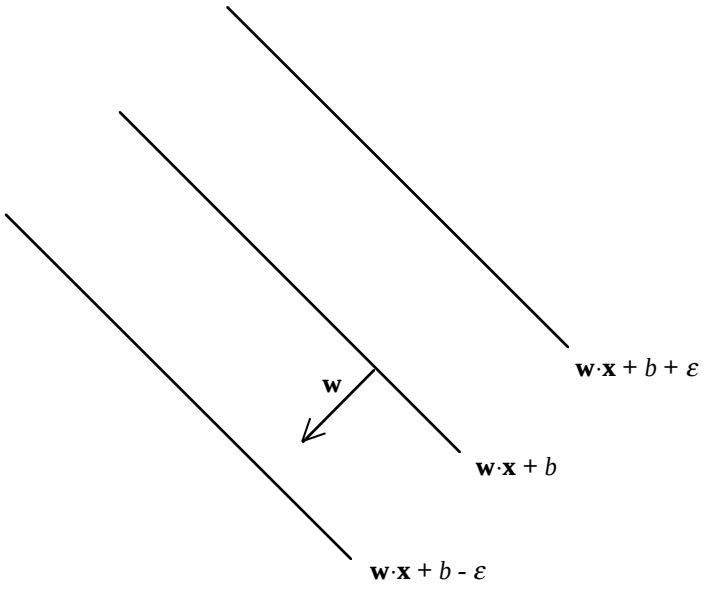


Fig. 7. Two parallel hyperplanes $\mathbf{w}\cdot\mathbf{x}$ + b ± ε *are shown. The reject region is bounded by the two hyperplanes.*

By denoting the class labels with $y \in \{+1,-1\}$, the decision function is the following:

$$\begin{aligned} f(\mathbf{x},\alpha) &= +1, \text{ if } \mathbf{w}\cdot\mathbf{x} + b \geq \varepsilon\,, \\ f(\mathbf{x},\alpha) &= -1, \text{ if } \mathbf{w}\cdot\mathbf{x} + b \leq -\varepsilon\,, \\ f(\mathbf{x},\alpha) &= 0, \text{ if } -\varepsilon < \mathbf{w}\cdot\mathbf{x} + b < \varepsilon\,. \end{aligned} \tag{57}$$

At this point, following again Vapnik's approach, to apply the SRM principle it would be first necessary to compute the VC dimension h of the set of loss functions (55) defined on decision functions (57). Then, subsets of the decison functions (57) should be found, having a VC dimension less than h. This should lead to the definition of a problem, perhaps similar to (49), whose solution is a pair of hyperplanes (56) which we could call *Optimal Separating Hyperplanes with Rejection* (OSHR). Note that the solution of this problem should depend on the parameter w_R of the loss function (i.e., the cost of a rejection). However these steps are beyond the scope of this work. Therefore we proceed by making a working hypothesis. Our hypothesis is that, for the set of loss functions (55) with decision functions (57), the VC dimension depends again on the margin of the pair of hyperplanes $\mathbf{w}\cdot\mathbf{x}+b\pm\varepsilon$, defined as $1/\|\mathbf{w}\|$. We also assume that the rejection region must always lie inside the margin. The trade-off between the VC dimension and the empirical risk can then be expressed through a functional similar to the one introduced by Vapnik for the non-linearly separable case (49). The OSHR is therefore solution of the follwing problem:

$$\begin{aligned} &\text{minimise} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} h(\xi_i, \varepsilon), \\ &\text{subject to} \; y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i \quad i = 1,\ldots,l, \\ &\qquad \xi_i \geq 0 \quad i = 1,\ldots,l, \\ &\qquad 0 \leq \varepsilon \leq 1, \end{aligned} \tag{58}$$

where C is a given positive constant. The function $h(\xi_i,\varepsilon)$ must be an approximation of the classification cost of the i-th training pattern, according to the loss function (55). It must therefore depend on the amplitude of the rejection region, defined by ε. We point out that, for the same reason, $h(\xi_i,\varepsilon)$ must depend also on the parameter w_R. Note that the constraint $0 \leq \varepsilon \leq 1$ enforces the rejection region to be inside the margin.

The problem is now to find a function $h(\xi_i,\varepsilon)$ so that the summand in the objective function of problem (58) is a suitable approximation of the empirical risk, that is, of the error-reject trade-off, according to the loss function (57). It would be desirable to deal with a convex function, since this would lead to a simpler optimisation problem from the computational viewpoint, as pointed out in Sect. 6.2. Let us consider the original definition of the OSH for the non-linearly separable case defined by Vapnik, (1998). The OSH was originally defined as the solution of the following problem:

$$\begin{aligned} &\text{minimise} \quad \sum_{i=1}^{l} h(\xi_i), \\ &\text{subject to} \; y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1,\ldots,l, \\ &\qquad \xi_i \geq 0, \quad i = 1,\ldots,l, \\ &\qquad \|\mathbf{w}\|^2 \leq A^2, \end{aligned}$$

where A is a predefined constant. To minimise the number of training errors, the function $h(\xi_i)$ was defined as $h(\xi_i)=0$ if $\xi_i=0$, and $h(\xi_i)=1$ if $\xi_i>0$. Note that this definition implies that correctly classified patterns which lie inside the margin, characterised by $0 < \xi_i < 1$, are considered as misclassified patterns, since the corresponding value of $h(\xi_i)$ is 1. However, since the above problem is NP-complete, Vapnik proposed to approximate it by using the function $h(\xi_i) = \xi_i$ (note that the ξ_i are constrained to be non-negative).

Proceeding by analogy, taking into account the decision rule (57) and the constraints of problem (58), in our case the function $h(\xi_i,\varepsilon)$ should be the following:

$$\begin{aligned} &h(\xi_i,\varepsilon) = 0, \quad \text{if } \xi_i = 0, \\ &h(\xi_i,\varepsilon) = w_C, \quad \text{if } 0 < \xi_i \leq 1-\varepsilon, \\ &h(\xi_i,\varepsilon) = w_R, \quad \text{if } 1-\varepsilon < \xi_i \leq 1+\varepsilon, \\ &h(\xi_i,\varepsilon) = 1, \quad \text{if } \xi_i > 1+\varepsilon, \end{aligned} \tag{59}$$

where w_C is a constant value such that $0 < w_C < w_R$. The role of w_C is to avoid that patterns correctly classified which lie outside the rejection region but inside the margin, are given a null cost. This is analogous to what happens in Vapnik's formulation, as explained above. The behaviour of function (59) is shown in Fig. 8.

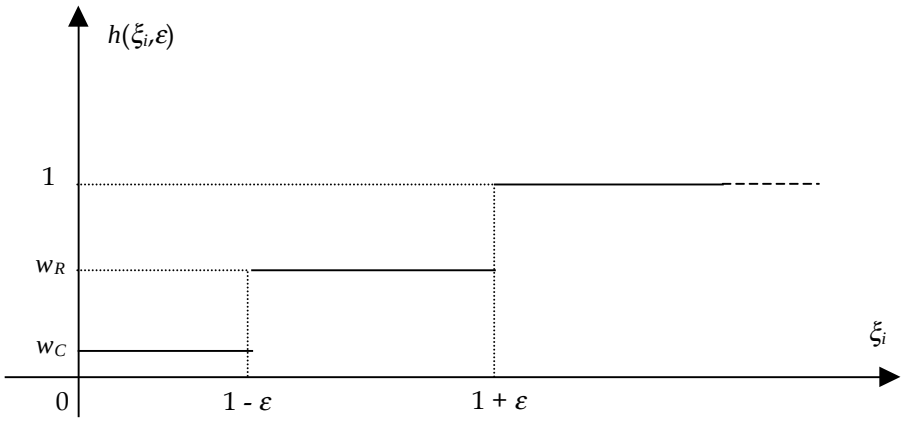


Fig. 8. Behaviour of the function $h(\xi_i,\varepsilon)$, representing the error-reject trade-off.

Unfortunately, a convex function is not suitable to approximate function (59). An approximation which does not lead to a trivial solution of problem (58), and is relatively simple from the computational viewpoint, can be the following:

$$h(\xi_i,\varepsilon) = \frac{1}{10}\xi_i^2 + 2w_C\left(\frac{1}{1+e^{-\alpha\xi_i}} - \frac{1}{2}\right) + \frac{w_R - w_C}{1+e^{-\alpha(\xi_i - 1 + \varepsilon)}} + \frac{1 - w_R}{1+e^{-\alpha(\xi_i - 1 - \varepsilon)}} \ .$$

To obtain a good approximation of function (59), a suitable value for the parameter α can be 100. The behaviour of this function, for α = 100, w_C = 0.1, and ε = 0.5, is shown in Fig. 9. Note that using this function the constraints $\xi_i \geq 0$ are not necessary. In the following we consider w_C = 0.1. This implies that the cost of a rejection is $0.1 \leq w_R \leq 1$.

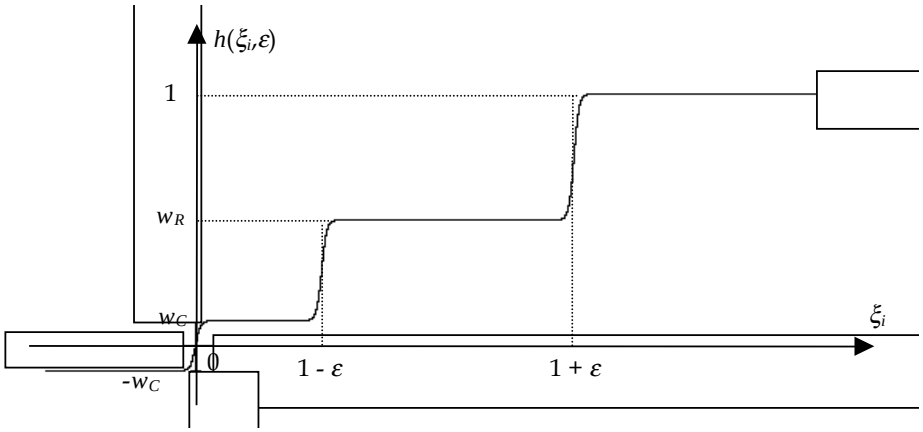


Fig. 9. Approximation of the function of Fig. 8, obtained using sigmoidal functions.

6.3.2 Primal and dual problems

Using the above formulation, the OSHR is the solution of the following (primal) problem:

$$\begin{aligned} &\text{minimise} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\left[\frac{1}{10}\xi_i^2 + 2w_C\left(\frac{1}{1+e^{-\alpha\xi_i}} - \frac{1}{2}\right) + \frac{w_R - w_C}{1+e^{-\alpha(\xi_i - 1 + \varepsilon)}} + \frac{1 - w_R}{1+e^{-\alpha(\xi_i - 1 - \varepsilon)}}\right], \\ &\text{subject to} \quad y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1,\ldots,l, \\ &\qquad\qquad 0 \leq \varepsilon \leq 1. \end{aligned}$$

The associated Lagrange function is:

$$L\left(\mathbf{w}, b, \xi_i, \varepsilon; \alpha_i\right) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} h\left(\xi_i, \varepsilon\right) - \sum_{i=1}^{l} \alpha_i \left[y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) - 1 + \xi_i \right] .$$

We chose not to include the constraints $0 \le \varepsilon \le 1$ in the Lagrange function. They must therefore be taken into account when minimising the Lagrange function with respect to the primal variables (Bazaraa, 1992). This choice is convenient from the algorithmic point of view, as explained below. With respect to the primal variables, the Lagrange function is the sum of a convex function of $\mathbf{w}$ and b, and a non-convex function of ξ_i and ε:

$$L\left(\mathbf{w}, b, \xi_i, \varepsilon; \alpha_i\right) = L_1\left(\mathbf{w}, b; \alpha_i\right) + L_2\left(\xi_i, \varepsilon; \alpha_i\right) ,$$

where

$$L_1\left(\mathbf{w}, b; \alpha_i\right) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i \left[y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) - 1 \right] ,$$

$$L_2\left(\xi_i, \varepsilon; \alpha_i\right) = C\sum_{i=1}^{l} \left[\left(\frac{1}{10}\xi_i^2 - \frac{\alpha_i}{C}\xi_i \right) + 2w_C \left(\frac{1}{1+e^{-\alpha\xi_i}} - \frac{1}{2} \right) + \frac{w_R - w_C}{1+e^{-\alpha\left(\xi_i - 1 + \varepsilon\right)}} + \frac{1 - w_R}{1+e^{-\alpha\left(\xi_i - 1 - \varepsilon\right)}} \right] . \qquad (60)$$

The minimum of L can therefore be found as the sum of the minima of L_1 and L_2, which can be computed independently. We remind that the minimum of L_2 must be found under the constraints $0 \le \varepsilon \le 1$. Since L_1 is convex, its minimum can be found by imposing stationarity:

$$\begin{aligned} &\frac{\partial L_1\left(\mathbf{w}, b, \alpha_i\right)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i = 0, \quad \mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i , \\ &\frac{\partial L_1\left(\mathbf{w}, b, \alpha_i\right)}{\partial b} = -\sum_{i=1}^{l} \alpha_i y_i = 0 . \end{aligned} \qquad (61)$$

Note that these are the same relations found for standard SVMs (46). The minimum of L_2 can not be found analytically, due to the complexity of this function. The behaviour of any of the terms of the summand in L_2 is shown in Fig. 10, for the same values of the parameters as in Fig. 9: α = 100, w_C = 0.1, and ε = 0.5.

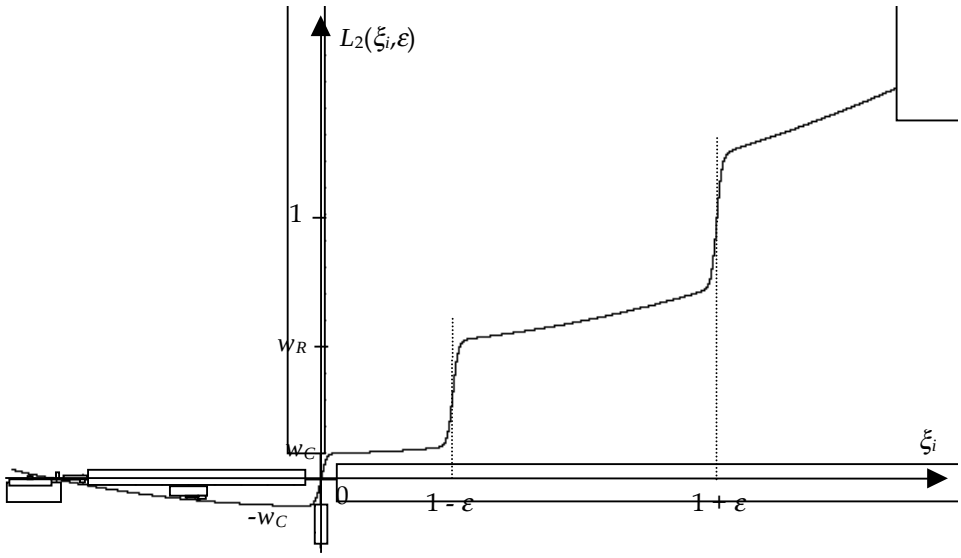


Fig. 10. Behaviour of any term of the summand in function $L_2(\xi_i,\varepsilon)$.

The problem of minimising L_2 can be simplified by exploiting the fact that, for a fixed value of ε, each term of the summation depends only on the corresponding ξ_i. In the next paragraph we describe a method to minimise L_2. It is now easy to see that the dual problem associated to problem (58) is the following:

$$\begin{aligned} &\text{maximise} \quad \theta(\alpha_1, \mathrm{K}, \alpha_l), \\ &\text{subject to} \sum_{i=1}^{l} y_i \alpha_i = 0, \\ &\qquad \alpha_i \geq 0, \quad i = 1, \mathrm{K}, l, \end{aligned} \tag{62}$$

where

$$\begin{aligned} \theta(\alpha_1, \mathrm{K}, \alpha_l) = &\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) + \\ &+ C \min_{\xi_i, 0 \leq \varepsilon \leq 1} \sum_{i=1}^{l} \left[\left(\frac{1}{10} \xi_i^2 - \frac{\alpha_i}{C} \xi_i \right) + 2 w_C \left(\frac{1}{1 + e^{-\alpha \xi_i}} - \frac{1}{2} \right) + \frac{w_R - w_C}{1 + e^{-\alpha (\xi_i - 1 + \varepsilon)}} + \frac{1 - w_R}{1 + e^{-\alpha (\xi_i - 1 - \varepsilon)}} \right]. \end{aligned} \tag{63}$$

Problem (62) is very similar to the standard formulation of the dual problem for SVMs (50). In particular, Eq. (61) shows that the weight vector of the pair of parallel hyperplanes is a linear combination of training points, as for standard SVMs. Therefore training points whose corresponding dual variable is zero can be called support vectors. Note also that in problem (62) the training points appear only in the form of inner products: this allows to deal with non-linear decision surfaces as standard SVMs. However, the main drawback of problem (62) is that the dual objective function $\theta(\alpha_1,\ldots,\alpha_l)$ is not known in analytical form. It can only be evaluated for given values of $\alpha_1,\ldots,\alpha_l$ by first solving the constrained minimisation problem in (63). Moreover, the KKT conditions are not necessary and sufficient to characterise the solution of the primal and dual problems. The only property of problem (62) which can be exploited is the fact that the dual objective function is concave, like the Lagrangian dual objective function of any optimisation problem with continuous primal objective function (Bazaraa, 1992). In the following paragraph we propose an algorithm for solving the dual problem (62).

6.4 An algorithm for finding the OSHR

To train a SVM with reject option by solving problem (62) it is possible to use an algorithm similar to the SMO algorithm, which was described in paragraph 6.2.4. Problem (62) is reported below.

$$\begin{aligned} &\text{maximise} \quad \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) + \\ &\qquad + C \min_{\xi_i, 0 \leq \varepsilon \leq 1} \sum_{i=1}^{l} \left[\left(\frac{1}{10} \xi_i^2 - \frac{\alpha_i}{C} \xi_i \right) + 2 w_C \left(\frac{1}{1 + e^{-\alpha \xi_i}} - \frac{1}{2} \right) + \frac{w_R - w_C}{1 + e^{-\alpha (\xi_i - 1 + \varepsilon)}} + \frac{1 - w_R}{1 + e^{-\alpha (\xi_i - 1 - \varepsilon)}} \right], \\ &\text{subject to} \sum_{i=1}^{l} y_i \alpha_i = 0, \\ &\qquad \alpha_i \geq 0, \quad i = 1, \mathrm{K}, l. \end{aligned} \tag{64}$$

As already pointed out, the main drawbacks of this optimisation problem are that the dual objective function $\theta(\alpha_1,\ldots,\alpha_l)$ is not known in analytical form, and the KKT conditions are not necessary and sufficient to characterise its solution. In the following we show how the SMO algorithm can be modified to deal with problem (64). We first show how the dual objective function can be evaluated, for given values of $\alpha_1,\ldots,\alpha_l$, by solving the constrained minimisation problem in (64). We then show how to find the maximum of the dual objective function with respect to a given pair of variables α_i, α_j. We finally tackle the problem of finding a heuristic to choose a suitable pair of variables α_i, α_j at each step of the algorithm, and of finding a stopping criteria.

6.4.1 Evaluation of the dual objective function

The dual objective function $\theta(\alpha_1,\ldots,\alpha_l)$ can be evaluated by first solving the following constrained minimisation problem:

$$\begin{aligned}&\text{minimise} \quad C\sum_{i=1}^{l}\left[\left(\frac{1}{10}\xi_i^2-\frac{\alpha_i}{C}\xi_i\right)+2w_C\left(\frac{1}{1+e^{-\alpha\xi_i}}-\frac{1}{2}\right)+\frac{w_R-w_C}{1+e^{-\alpha\left(\xi_i-1+\varepsilon\right)}}+\frac{1-w_R}{1+e^{-\alpha\left(\xi_i-1-\varepsilon\right)}}\right],\\&\text{subject to}\; 0\le\varepsilon\le 1.\end{aligned}$$

Let us denote the terms of the above summation with $h(\xi_i,\varepsilon)$. In paragraph 6.3.2 we pointed out that the above minimisation problem can be solved by exploiting the fact that, for a fixed value of ε, each term $h(\xi_i,\varepsilon)$ is independent from the other ones. The minimum of the summation can then be found as the sum of the minimum of each $h(\xi_i,\varepsilon)$. To find the minimum of $h(\xi_i,\varepsilon)$ for a fixed value of ε, we propose to approximate the three sigmoidal functions of its objective function with step functions (formally, this is achieved for $\alpha\to+\infty$). Denoting a step function as follows:

$$I_{x_0}(x)=\begin{cases}0, & \text{if } x\le x_0,\\ 1, & \text{if } x>x_0,\end{cases}$$

the above optimisation problem becomes:

$$\begin{aligned}&\text{minimise} \quad C\sum_{i=1}^{l}\left[\left(\frac{1}{10}\xi_i^2-\frac{\alpha_i}{C}\xi_i\right)+2w_C\left(I_0\left(\xi_i\right)-\frac{1}{2}\right)+\left(w_R-w_C\right)I_{1-\varepsilon}\left(\xi_i\right)+\left(1-w_R\right)I_{1+\varepsilon}\left(\xi_i\right)\right],\\&\text{subject to}\; 0\le\varepsilon\le 1.\end{aligned}\qquad(65)$$

Let us denote the i-th term of the above summation with $h'(\xi_i,\varepsilon)$. Consider now a fixed value of ε, in the interval [0,1]. The function $h'(\xi_i,\varepsilon)$ is the sum of a convex parabola ($0.1\alpha_i^2$ - $(\alpha_i/C)\xi_i$), and of three step functions. It is easy to see that the minimum of $h'(\xi_i,\varepsilon)$ can only be achieved at one of the discontinuity points of the step functions, or at the minimum of the parabola. The corresponding values of ξ_i are respectively: ξ_i=0, ξ_i = 1-ε, ξ_i = 1+ε, $\xi_i=5\alpha_i/C$. Furthermore, for varying values of ε, the minimum of $h'(\xi_i,\varepsilon)$ is either constant (if it is achieved at ξ_i=0 or at $\xi_i=5\alpha_i/C$), or moves along the parabola (if it is achieved at ξ_i = 1-ε or at ξ_i = 1+ε). For any $h'(\xi_i,\varepsilon)$ it is then possible to subdivide the interval [0,1] of ε values into a finite number n_i+1 of adjacent intervals, such that the analytical expression of the minimum of $h'(\xi_i,\varepsilon)$ in each interval is either a constant or a polynomial of degree two (which is function of ε). Reminding that $0\le\varepsilon\le 1$, let us denote the extremes of the sequence of intervals related to $h'(\xi_i,\varepsilon)$ as:

$$I_{\varepsilon_i} = \left\{ \varepsilon_{i,0}, \varepsilon_{i1}, \varepsilon_{i2}, \ldots, \varepsilon_{i,n_i}, \varepsilon_{i,n_i+1} \right\} ,$$

where

$$\varepsilon_{i,0} = 0 < \varepsilon_{i1} < \varepsilon_{i2} < \ldots < \varepsilon_{i,n_i} < 1 = \varepsilon_{i,n_i+1} .$$

The minimum of $h'(\xi_i,\varepsilon)$ in the k-th interval can then be written as:

$$\min_{\xi_i,\ \varepsilon_{ik} \le \varepsilon < \varepsilon_{i,k+1}} h'(\xi_i, \varepsilon) = a_{ik}\varepsilon^2 + b_{ik}\varepsilon + c_{ik}, \quad k = 0, \ldots, n_i . \tag{66}$$

Obviously, if the minimum is a constant term, then $a_{ik} = b_{ik} = 0$. The coefficients a_{ik}, b_{ik} and c_{ik} can be obtained as a function of the parameters of $h'(\xi_i,\varepsilon)$, that is, α_i / C, w_C and w_R (see (65)). For any value of ε, the minimum of the summation of the $h'(\xi_i,\varepsilon)$ is equal to the sum of the minimum of each $h'(\xi_i,\varepsilon)$. Consider now the finite sequence of intervals, denoted with I_ε, obtained by superimposing the sequences of each $h'(\xi_i,\varepsilon)$. More precisely, the extremes of I_ε are:

$$I_\varepsilon = \bigcup_{i=1}^{l} I_{\varepsilon_i} .$$

From what said above, it follows that in any interval of the sequence I_ε the expression of the minimum of the summation of the $h'(\xi_i,\varepsilon)$ is a constant term or a polynomial of degree two. These expressions can be obtained as the sum of the corresponding terms (66). The global minimum of the objective function of problem (65) can then be analytically obtained by analysing the sequence of such expressions for the intervals I_ε. Note that this also allows to compute the values of ε and of the ξ_i's which solve problem (65).

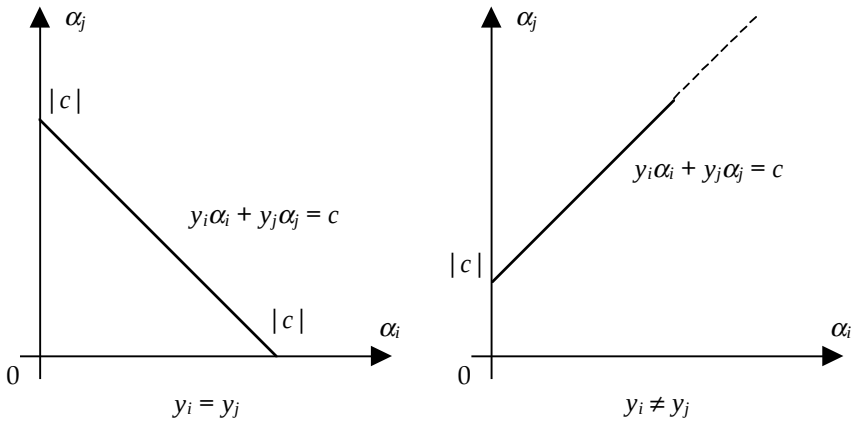
We point out that evaluating the dual objective function of problem (64) clearly requires a more time-consuming computation than the one needed for the training problem of standard SVMs.

6.4.2 Maximising the dual objective function with respect to a pair of dual variables

Let us now consider the problem of finding the maximum of $\theta(\alpha_1,\ldots,\alpha_l)$ with respect to a pair of dual variables α_i, α_j. We remind that in our case the constraints are α_i, $\alpha_j \ge 0$, and $\sum_{i=1}^{l} y_i\alpha_i = 0$. The second constraint implies that $y_i\alpha_i + y_j\alpha_j = \text{constan}$. We denote with c this constant term, related to the current values of the dual variables:

$$c = y_i\alpha_i^{old} + y_j\alpha_j^{old} . \tag{67}$$

Consider first the case $y_i = y_j$. It is easy to see that the above constraints cause the two variables to lie on a diagonal line segment, as shown in Fig. 11(a).



(a) *(b)*

Fig. 11. Constraints on the pair of dual variables α_i, α_j.

The corresponding constraints are:

$$0 \le \alpha_i \le |d|, \quad \text{if } y_i = y_j \ .$$

Due to constraint (67), the dual objective function can be expressed only as a function of one of the two variables. Since it is a concave function, its maximum on the diagonal line segment can be found by using the golden section method (Bazaraa, 1992). Consider now the case $y_i \neq y_j$. It is easy to see that constraint (67) cause now the two variables to lie on a diagonal half-line, as shown in Fig. 11(b). The corresponding constraint is:

$$\alpha_i \ge |d|, \quad \text{if } y_i \neq y_j, \text{and} \, y_i c \ge 0 ,$$
$$\alpha_i \ge 0, \quad \text{if } y_i \neq y_j, \text{and} \, y_i c < 0 .$$

In this case to apply the golden section method it is first necessary to find a segment of the half line which contains the maximum of $\theta(\alpha_1,\ldots,\alpha_l)$. To this aim, it is possible to exploit an upper bound of $\theta(\alpha_1,\ldots,\alpha_l)$. This function can indeed be expressed as a sum of two terms θ_1 and θ_2:

$$\theta_1(\alpha_1, \mathrm{K}, \alpha_l) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) ,$$

$$\theta_2(\alpha_1, \mathrm{K}, \alpha_l) = \min_{\xi_i \,, 0 \le \varepsilon \le 1} C \sum_{i=1}^{l} \left[\left(\frac{1}{10} \xi_i^2 - \frac{\alpha_i}{C} \xi_i \right) + 2 w_C \left(\frac{1}{1 + e^{-\alpha \xi_i}} - \frac{1}{2} \right) + \frac{w_R - w_C}{1 + e^{-\alpha (\xi_i - 1 + \varepsilon)}} + \frac{1 - w_R}{1 + e^{-\alpha (\xi_i - 1 - \varepsilon)}} \right] .$$

The first term θ_1 is known in analytical form, while the second term θ_2 does not. Note that θ_2 coincides with the minimum of function L_2 (60). By using the approximations described above for L_2, it can be shown that θ_2 is upper bounded by the term $0.8C \cdot l$. Therefore, the dual objective function has the following upper bound, which can be computed in analytical form:

$$\theta(\alpha_1, \mathrm{K}, \alpha_l) \le \theta_1(\alpha_1, \mathrm{K}, \alpha_l) + 0.8C \cdot l \ . \tag{68}$$

It is now easy to see that the maximum of $\theta(\alpha_1,\ldots,\alpha_l)$ lies in the line segment whose extremes are the vertex of the half-line of Fig. 11(b), and the point of the half-line for which the value of the upper bound (68) equals the value of $\theta(\alpha_1,\ldots,\alpha_l)$ in the vertex of the half-line. Denoting with θ^* the value of $\theta(\alpha_1,\ldots,\alpha_l)$ in the vertex of the half-line, the second extreme can then be found by solving the equation $\theta(\alpha_1,\ldots,\alpha_l) + 0.8C \cdot l = \theta^*$. Note that this is a simple quadratic equation. The golden

section method can then be applied to this line segment to find the maximum of the dual objective function with respect to α_i and α_j.

6.4.3 Selection heuristics and stopping criterion

Let us now turn our attention to the problems of choosing a suitable pair of variables at each step of the algorithm, and of finding a proper stopping criteria. As pointed out above, these problems are complicated by the fact that the KKT conditions are not necessary and sufficient to characterise the optimal solutions of the primal and dual problems. For problem (64) only necessary conditions can be given. Consider first the approximation described in paragraph 6.4.1 for evaluating the dual objective function by solving problem (65). Let us denote the corresponding value of ξ_i, at the solution of problem (65), as $\bar{\xi}_i$. We showed that $\bar{\xi}_i$ can only assume one of the four values in $\{0,\ 1\text{-}\varepsilon,\ 1\text{+}\varepsilon,\ 5\alpha_i / C\}$. In particular, it can be shown that, if $\alpha_i / C < \sqrt{2} / 5$, then $\bar{\xi}_i = 0$. This implies that the corresponding primal constraint becomes $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1$. That is, the training point $\mathbf{x}_i$ must be correctly classified, and must lie outside the margin. Therefore, a necessary condition for the solution of the primal and dual problems is the following:

$$y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) \geq 1, \quad \text{if} \quad \frac{\alpha_i}{C} < \frac{\sqrt{2}}{5} \ . \tag{69}$$

Checking if this condition holds is quite easy, since it does not require to compute the value of the corresponding $\bar{\xi}_i$. Since condition (69) is only a necessary condition for characterising the optimal solutions of the primal and dual problems, it can not be used as stopping criterion. Nevertheless, it can be used as a heuristic for choosing the first dual variable α_i in the outer loop of the algorithm. In particular, in the outer loop training points satisfying condition (69) can be ignored. If the value of the corresponding dual variable were not the optimal one (we remind that condition (69) it is necessary but not sufficient), it could nevertheless be changed since this variable can be chosen by the second heuristic in the inner loop. After an outer loop in which at least one pair of dual variables has been updated, the next outer loop can be restricted only to non-zero variables, analogously to SMO. The last outer loop of the algorithm should always be a complete loop.

Note that checking condition (69) requires to estimate the value of b. Analogously to SMO, the value of b can be updated at each iteration by imposing that the primal constraints hold for the pair of dual variables α_i, α_j which have been updated during the last step. In the general case the primal constraints are:

$$y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) \geq 1 - \bar{\xi}_i, \quad \text{if} \quad \frac{\alpha_i}{C} \geq \frac{\sqrt{2}}{5} \ , \tag{70}$$

where the $\bar{\xi}_i$ can be obtained as explained in paragraph 6.4.1. If $\bar{\xi}_i = 0$, the above condition implies that the training point $\mathbf{x}_i$ must be correctly classified, and must lie outside the margin, as for condition (69). If $\bar{\xi}_i = 1\text{-}\varepsilon$, condition (70) implies that the corresponding pattern is correctly classified, and lies inside the margin but outside the rejection region. If $\bar{\xi}_i = 1\text{+}\varepsilon$, condition (70)

implies that the corresponding pattern is rejected. Finally, if $\bar{\xi}_i = 5\alpha_i / C$, it implies that the corresponding pattern is misclassified (if $5\alpha_i / C > 1+\varepsilon$), or is rejected (if $1-\varepsilon < 5\alpha_i / C < 1+\varepsilon$). By imposing that the primal constraints hold for the pair α_i, α_j:

$$\begin{aligned} y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) &\geq 1-\bar{\xi}_i \,, \\ y_j\left(\mathbf{w}\cdot\mathbf{x}_j + b\right) &\geq 1-\bar{\xi}_j \,, \end{aligned} \tag{71}$$

four cases can occur. The value of b can then be updated as follows.

if $b \geq b_i$ and $b \geq b_j$, or $b \leq b_i$ and $b \leq b_j$, then set b as $(b_i + b_j)/2$;

if $b \geq b_i$ and $b \leq b_j$, with $b_i \leq b_j$, then set b as $(b_i + b_j)/2$;

if $b \leq b_i$ and $b \geq b_j$, with $b_i \geq b_j$, then set b as $(b_i + b_j)/2$;

otherwise keep the previous value of b.

We denoted with b_i and b_j the constant terms obtained from inequalities (71).

The second dual variable α_j can be chosen as in SMO, such that updating on the pair α_i, α_j causes the largest change in both variables, which should result in a large increase of the dual objective function $\theta(\alpha_1,\ldots,\alpha_l)$. (Note that, due to the linear constraint (67), both variables always change by the same amount.) To avoid computing the exact value of $\theta(\alpha_1,\ldots,\alpha_l)$ (this would require to solve problem (65)), only the term $\theta_1(\alpha_1,\ldots,\alpha_l)$ can be maximised. Note that $\theta_1(\alpha_1,\ldots,\alpha_l)$ coincides with the objective function of the standard SVMs dual formulation (50). The unconstrained maximum of $\theta_1(\alpha_1,\ldots,\alpha_l)$ can then be found as shown in paragraph 6.2.4 for the SMO algorithm:

$$\alpha_i^{unc} = \alpha_i^{old} + \frac{y_i\left(\sum_{k=1}^{l}\alpha_k y_k K\left(\mathbf{x}_k,\mathbf{x}_j\right) - y_j - \sum_{k=1}^{l}\alpha_k y_k K\left(\mathbf{x}_k,\mathbf{x}_i\right) + y_i\right)}{K\left(\mathbf{x}_i,\mathbf{x}_i\right) + K\left(\mathbf{x}_j,\mathbf{x}_j\right) - 2k\left(\mathbf{x}_i,\mathbf{x}_j\right)} \,.$$

The α_j for which $\left|\alpha_i^{unc} - \alpha_i^{old}\right|$ is maximum is chosen. The exact value of α_i which maximises the dual objective function can then be computed as explained in paragraph 6.4.2. The corresponding value of α_j can be found from the constraint $y_i\alpha_i + y_j\alpha_j = c$. If this choice did not provide a significant change $\left|\alpha_i - \alpha_i^{old}\right|$, it would be possible to trying first each non-zero α_j, and, if necessary, all the other α_j's, as in SMO. However this would require too much computation, due to the complexity of finding the value of α_i which maximises the dual objective function. Therefore we chose to try at most one other α_j, that is, the one which provides the second highest change $\left|\alpha_i^{unc} - \alpha_i^{old}\right|$.

Finding a proper stopping criterion is the more difficult point. Monitoring the growth of the dual objective function is an unreliable criterion (Cristianini and Shawe-Taylor, 2000). Moreover, it is not possible to monitor the decrease of the feasibility gap (that is, the difference between the primal and dual objective functions): since the primal objective function is not convex, the gap does not vanishes at the solution (Bazaraa, 1992). Obviously the algorithm must stop if no pair of dual variables causing an increment of the objective dual function has been found in a complete outer loop. This criterion can be inefficient, since the convergence of the dual objective function

could be very slow near the solution. Nevertheless, we used this criterion for preliminary experiments, whose results are reported in Chapter 7. Further work is needed to find necessary and sufficient conditions to characterise the solution of problem (62), to be used as a more efficient stopping criterion.

6.4.4 Pseudocode of the algorithm

In the following we report the pseudocode of the algorithm described above. At this level of detail, the only differences with respect to SMO are the heuristics for choosing a pair of dual variables at each iteration, and the stopping criterion.

```
alpha[]: vector of dual variables
w: weight vector of the hyperplanes
x[]: training point matrix
eps: a predefined tolerance level

main routine
  initialise alpha array to zero
  initialise b to zero
  numChanged = 0
  examineAll = 1
  while (numChanged > 0 || examineAll == 1)
  {
    numChanged = 0
    if (examineAll)
      loop i over all training examples
        numChanged += examineExample(i)
    else
      loop over examples whose alpha is not 0
        numChanged += examineExample(i)
    if (examineAll == 1)
      examineAll = 0
    else if (numChanged == 0)
      examineAll = 1
  }

procedure examineExample(i)
  if (alpha[i]/C ≥ sqrt(2)/5-eps or y[i]*(w·x[i]+b) ≤ 1+eps)
  {
    j = first result of second choice heuristic
    if takeStep(i,j)
      return 1
    j = second result of second choice heuristic
    if takeStep(i,j)
      return 1
  }
  return 0
endprocedure

procedure takeStep(i,j)
  if (i == j) return 0
  compute the value of alpha[i] which maximises the dual objective function,
    with respect to alpha[i] and alpha[j], under the dual constraints
  if the change of alpha[i] and alpha[j] is below a predefined tolerance,
    return 0
```

```
  update alpha[i] and alpha[j]
  update the threshold b
  return 1
endprocedure
```

6.5 Discussion

The main purpose of this Chapter was to show how the reject option can be introduced in SVMs by following a theoretical derivation from statistical learning theory, analogous to Vapnik's derivation of standard SVMs. This leaded to the definition of a set of decision functions whose outputs include the reject decision. Unlike usual training algorithms, this implies that the reject region must be defined during the training phase, together with the decision regions. (Note that this approach is analogous to the one proposed by Mizutani (1998) for other types of classifiers.) Following Vapnik's approach, we considered as the simplest set of decision functions with reject option the set of pairs of parallel hyperplanes. Since computing the VC dimension of this set of functions was beyond the scope of this work, we made a working hypothesis about it. We then proposed an approximation of the empirical risk, which takes into account the error-reject trade-off. This allowed us to formulate the problem of training a SVM with reject option as an optimisation problem analogous to that of standard SVMs. Finally, we proposed an algorithm for solving this problem, derived from one of the algorithms for training standard SVMs.

Let us now compare our approach for designing a SVM with reject option with the rejection rule described in Chapter 3. We remind that this rule consists in rejecting patterns whose distance from the optimal separating hyperplane is less than a predefined threshold. We point out that also this approach leads to a rejection region delimited by a pair of parallel hyperplanes, in the feature space induced by the chosen kernel. However, the two approaches differ in the way the rejection region is obtained. Using the rejection rule described in Chapter 3, the two parallel hyperplanes delimiting the reject region are constrained to be parallel to the OSH, and equidistant from it, for any value of the reject rate. From a theoretical viewpoint, this approach is suitable for problems in which the contours of the optimal reject region (in the feature space induced by the chosen kernel) are pairs of hyperplanes which are always parallel and equidistant from the class boundary at a null reject probability (being the class boundary itself an hyperplane). We remind that, for a two-class problem, the class boundary at a null reject rate is defined by $P(\omega_1 \mid \mathbf{x})=P(\omega_2 \mid \mathbf{x})$, while the contours of the optimal rejection region are defined by $P(\omega_i \mid \mathbf{x})=T$, for $1/2 < T \leq 1$. For instance, it is easy to see that the optimal rejection region for a problem with two classes having gaussian distribution exhibits the above characteristics. Using our approach instead, the orientation and position of the parallel hyperplanes delimiting the reject region (besides their reciprocal distance) depend on the value of the cost parameter w_R. The orientation and position can therefore change for different values of w_R (that is, for different values of the reject rate). This means that our approach for obtaining the rejection region is more flexible, and is suitable also for problems in which the orientation and position of the contours of the optimal

reject region change for different values of T (provided that they are always parallel hyperplanes). As an example of such a problem we devised the following probability distribution:

$$P(\omega_1|\,\mathbf{x}) = \begin{cases} \frac{2}{\pi} arctg\frac{x_2}{x_1}, & \text{if } x_2 \geq x_1, \\ \frac{2}{\pi} arctg\frac{1-x_1}{1-x_2}, & \text{if } x_2 < x_1, \end{cases}$$

where x_1 and x_2 are the components of a two-dimensional feature vector, taking on values in the unit square [0,1]×[0,1]. It is easy to see that the optimal class boundary at a null reject probability is the diagonal line segment of the unit square, shown in Fig. 12 as a solid line. The contours of the rejection region, defined by $P(\omega_i|\,\mathbf{x})=T$, for $1/2< T \leq 1$, are pairs of parallel straight lines radiating symmetrically outwards from the origin (0;0) and from the point (1;1). These contours are shown as dashed lines in Fig. 12. It is evident that the orientation of these contours varies for different values of the reject threshold T, or, equivalently, of the reject probability. In this case the rejection rule based on a threshold on the distance from the OSH can not provide the optimal reject region, for any value of the reject rate, even if the OSH coincided with the optimal class boundary. The optimal rejection region can instead be always obtained by using our approach.

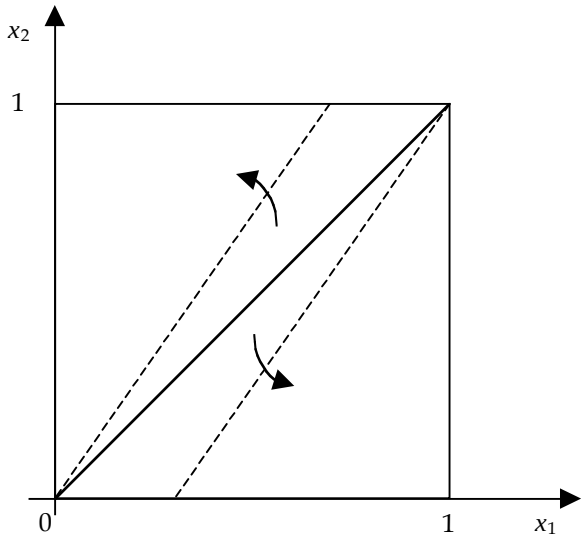


Fig. 12. The optimal class boundary (solid line) and the boundaries of the optimal rejection region (dashed lines), for the a posteriori probability distribution described above.

In the general case in which the optimal rejection region is not delimited by a pair of parallel hyperplanes, the greater flexibility of our approach allows in principle to better approximate it.

A drawback of our approach is its computational complexity. Indeed the optimisation problem resulting from our approach is intrinsically more complex than the one of standard SVMs, since the dual objective function is not available in analytical form. Moreover, our approach requires to train a different classifier for any value of the cost parameter w_R. The approach described in Chapter 3 requires instead to train one only classifier (a standard SVM

without rejection). Only the value of the reject threshold must then be computed for a given value of the cost parameter w_R. This problem will be further discussed in Chapter 7, where preliminary experimental results are reported.

We conclude by pointing out two possible developments of the approach presented in this Chapter. First, we remind that the formulation of the optimisation problem proposed in paragraph 6.3.1 is based on an assumption about the VC dimension of the set of pair of parallel hyperplanes. A more exact evaluation of the VC dimension could lead to a different formulation of the primal problem, perhaps simpler than the one proposed in paragraph 6.3.1. Secondly, it would be important to find necessary and sufficient conditions to characterise the solution of the primal and dual problems. Such conditions could be exploited as heuristics for speeding up the convergence of the algorithm, and as an efficient stopping criterion.

Chapter 7
Experiments

In this Chapter we present experiments related to the topics discussed in Chapters 4 and 6. The data sets used for the experiments are described in paragraph 7.1. In paragraph 7.2 we present experiments aimed at evaluating the performance of the CRT rejection rule, which was described in Chapter 4. In paragraph 7.3 we present preliminary results obtained using the technique proposed in Chapter 6 for introducing the reject option in support vector machines.

7.1 Data sets

7.1.1 The Feltwell data set

This data set consists of a set of multisensor remote-sensing images related to an agricultural area near the village of Feltwell (U.K.) (Serpico and Roli, 1995). The scene was acquired using an ATM scanner and a SAR sensor. From a section of 250×350 pixels of the image, the five numerically most representative classes were considered (see Table 1). Agricultural fields were then randomly subdivided into three disjoint sets: a training set of 5,124 pixels, a validation set of 582 pixels, and a test set of 5,238 pixels. Each pixel was characterised by fifteen features containing the brightness values in six optical bands and in nine radar channels. Table 1 shows the composition of the data set.

Classes	Training patterns	Validation patterns	Test patterns
Sugar beets	1,488	204	1,839
Stubble	1,070	137	1,234
Bare soil	341	56	499
Potatoes	1,411	88	796
Carrots	814	97	870
Total	5,124	582	5,238

Table 1. Composition of the Feltwell data set.

7.1.2 The Phoneme data set

This data set was taken from the University of California at Irvine machine learning database repository (`http://www.ics.uci.edu/~mlearn/MLRepository.html`). It was in use in the European ESPRIT 5516 project ROARS (Alinat, 1993), whose aim was the development and the implementation of a real time analytical system for French and Spanish phoneme recognition.

This data set consists of 5,404 patterns representing nasal and oral vowels (3,818 and 1,586 patterns respectively). Each pattern is characterised by five features, corresponding to the normalised amplitudes of the five first harmonics.

7.1.3 The Letter data set

This data set belongs to the UCI repository, as the Phoneme data set. It consists of 20,000 raster scan images of the 26 capital letters in the English alphabet, based on 20 different fonts. Each image was converted into 16 primitive numerical attributes (statistical moments and edge counts) which were then scaled to fit into a range of integer values from 0 through 15. The number of patterns of each class is reported in Table 2.

789 *A*	766 *B*	736 *C*	805 *D*	768 *E*	775 *F*	773 *G*
734 *H*	755 *I*	747 *J*	739 *K*	761 *L*	792 *M*	783 *N*
753 *O*	803 *P*	783 *Q*	758 *R*	748 *S*	796 *T*	813 *U*
764 *V*	752 *W*	787 *X*	786 *Y*	734 *Z*		

Table 2. Composition of the Letter data set.

7.2 Experiments on the CRT rejection rule

In Chapter 4 we formally proved that the CRT rule provides a better error-reject trade-off than Chow's rule, in presence of estimation errors on the a posteriori probabilities. In terms of the accuracy-rejection (A-R) curve, this means that the CRT rule can achieve a higher or equal accuracy than Chow's rule, for any value of the reject probability. Two main problems remained open, namely: quantitatively evaluating the performance improvement achievable by the CRT rule over Chow's rule, and obtaining a reliable estimate of the optimal CRTs values. These problems were tackled in the experiments presented below.

The experiments were conducted on the three data sets described above. For the purpose of our experiments, we included the 582 validation patterns of the Feltwell data set on the test set. The test pattern were therefore 5,820. For the Phoneme data set we randomly subdivided the 5,404 patterns into a training set and a test set of 2,707 patterns each, by keeping the original proportion between the two classes. The same was made for the Letter data set: half of the patterns of each class were randomly selected as training patterns, and the remaining were used as test patterns. Besides these data set, we used also an artificial data set, which is described in detail in the next paragraph.

For the Feltwell, Phoneme and Letter data sets, we used multi-layer perceptron (MLP) neural network classifiers with one hidden layer, trained with the back-propagation algorithm. The number of input and output units was always equal to the number of features and classes, respectively. For the Feltwell data set we used fifteen hidden neurons, and made 10 epochs during the classifiers training. For the phoneme data set the hidden layer contained thirty-six

neurons, and 400 epochs were made during the training. Fifteen hidden neurons were used for the Letter data set, and 200 epochs were made. The value of the learning rate was 0.01 for the three data sets. On the artificial data set no classifier was trained: the estimation errors on the a posteriori probabilities were instead simulated by generating random values, as explained in the next paragraph.

In order to investigate the above issues, three groups of experiments were made. The first group of experiments was aimed at evaluating the ideal performance of the CRT rule. To this aim, we computed the optimal CRTs values on the test set by exhaustive search, and compared the performances of the CRT and Chow's rules on the test set. The goal of the second group of experiments was to assess how the performance of the CRT rule is affected when the CRTs values are computed from a validation set. We considered again the ideal case, and computed the optimal CRTs values by exhaustive search on a validation set. The performances of the CRT and Chow's rules were again compared on the test set. The third group of experiments was aimed at evaluating the performance achievable by the CRT rule in a real setting in which the CRTs values must be *estimated* from the validation set. To this aim we used the simple algorithm described in paragraph 4.3.

In all experiments, the CRTs values were computed by maximising the classification accuracy (on the test set or on the validation set), for given values of the reject rates ranging from 0 to 30%. The reject threshold of Chow's rule was simply computed as the one which provided the desired reject rate. For each data set (except for the artificial one), ten different validation sets were randomly extracted from the original training set, by keeping the proportion between the different classes, and without replacement (that is, each of the original training patterns can appear at most once on each validation set). The remaining patterns were used for training ten different MLPs. All results are reported in terms of the average accuracy-reject (A-R) curve on the test set.

7.2.1 Results on the artificial data set

The first and the second group of experiments were first carried out using an artificial data set, in order to compare the ideal performances of the CRT and Chow's rules for given values of the amplitude of the estimation errors. We considered a two-class problem with a two-dimensional feature vector **x**. The two classes had equal priors $P(\omega_1) = P(\omega_2) = 1/2$, had a gaussian distribution with mean vectors respectively (-1.3;0) and (1.3;0), and the same covariance matrix $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. We generated a data set of one thousand patterns according to such probability distributions. We then simulated the effects of estimation errors on the a posteriori probability of each pattern, by adding random error values to the exact probability values. The error values were generated according to a gaussian distribution with zero mean. We considered five different values of the standard deviation, ranging from 0.1 to 0.5, with a step of 0.1.

Fig. 13 shows the A-R curves on the entire data set, obtained using CRTs values found by exahustive search on the entire data set (first group of experiments). A discretisation step of 0.001 was used.

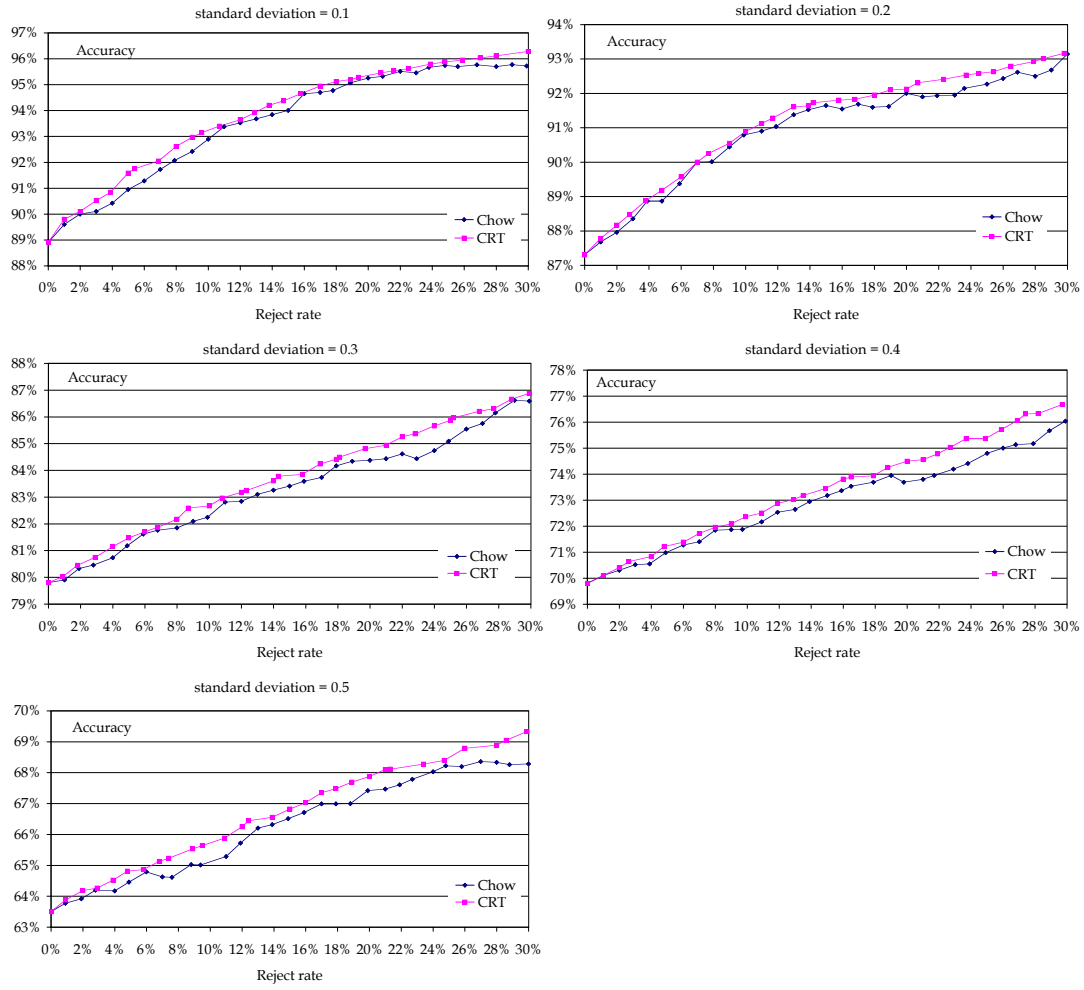
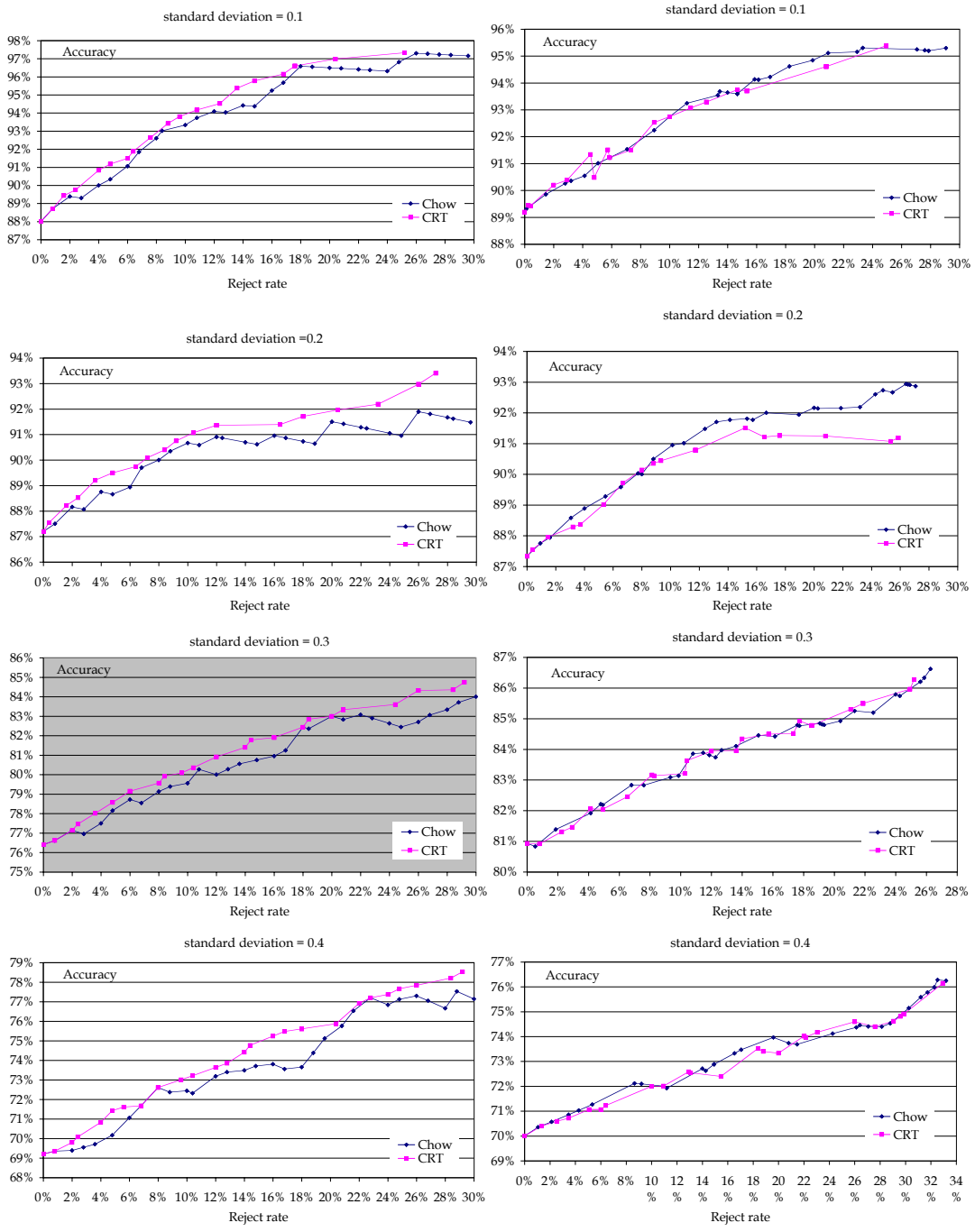


Fig. 13. A-R curves on the artificial data set, for five different values of the standard deviation of the estimation errors on the a posteriori probabilities.

In the above experiments the CRT rule provided always a better error-reject trade-off than Chow's rule. However, a quite small improvement can be observed. The maximum difference in accuracy, for equal values of the reject rate, was about 1%. Such difference was achieved only for large values of the estimation errors (standard deviation equal to 0.4 and 0.5), and only for small range of values of the reject rate. In particular, for small estimation errors (standard deviation equal to 0.1), the largest difference in accuracy was achieved for values of the reject rate lower than 10%. For larger estimation errors instead (standard deviation from 0.2 to 0.4), the difference in accuracy was significant for values of the reject rate greater than 20%. Only for the largest estimation errors the difference in accuracy was about constant over all values of the reject rate. From these results it seems that the advantage of the CRT rule over Chow's rule weakly depends on the amplitude of estimation errors on the a posteriori probabilities. A possible explanation of these fact is that in these experiments the estimation errors were generated according to a zero-mean probability density function. We showed in Chapter 4 that using different reject threshold

can be useful when the estimation errors are *uniformly* positive or negative around the boundaries of the rejection region. Intuitively, this condition could be more likely to occur when the probability density function of the estimation errors has a non-zero mean (that is, for biased estimation errors).

For the second group of experiments, we randomly extracted from the original data set the 25% of the patterns, and used them as validation set. The remaining patterns were used as test set. The CRTs values were computed by exhaustive search on the validation set, for the five values of the standard deviation of the estimation errors. The value of the discretisation step was again 0.001. Such CRTs values were then applied on the test set. In Fig. 14 we compare the A-R curves obtained on the validation set and on the test set.



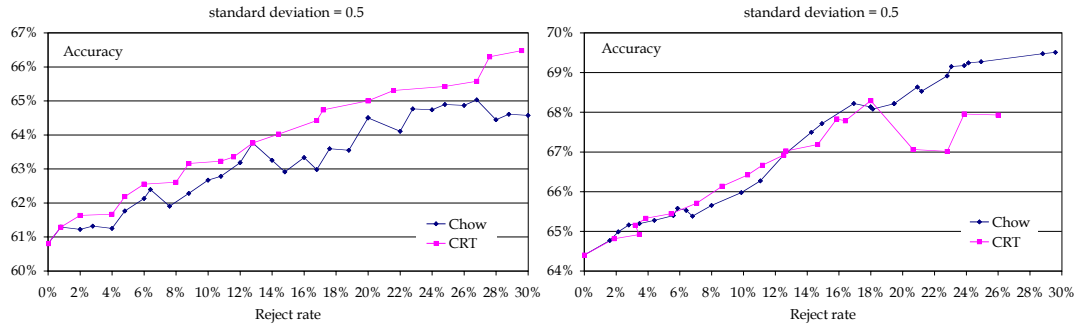


Fig. 14. A-R curves on the validation set (on the left) and on the test set (on the right), for five different values of the standard deviation of the estimation errors on the a posteriori probabilities.

The performance improvement of the CRT rule over Chow's rule on the validation set are slightly greater than the ones obtained on the entire data set (see Fig. 13). In this case the validation set should be considered quite representative of the test set, since it was randomly extracted from the original data set. However, the optimal CRTs values computed on the validation set did not improve the performance of Chow's rule on the test set. The performances of the two rules on the test set were quite similar. In particular, in two cases Chow's rule outperformed the CRT rule. This happened for values of the standard deviation of the estimation errors equal to 0.2 and 0.5, and for values of the reject rate greater than 10% and 18% respectively. These results seem to mean that it could be difficult to obtain good CRTs values from a validation set, at least if the maximum achievable advantage of the CRTs rule over Chow's rule is not large.

Let us now consider the last issue mentioned in paragraph 4.2.3, that is, whether there exists only one set of CRTs values which provides a greater classification accuracy than Chow's rule, for any given value of the reject rate. In paragraph 4.2.3 we answered this question from a mathematical viewpoint. We proved that, under the assumption that the probability density functions of the features are continuous with their first derivatives, infinite values of the CRTs can exist, which provide a higher accuracy than Chow's rule. From the practical viewpoint, we consider here the case where the CRTs are computed as discrete values. For instance, this is the case of the algorithm presented in paragraph 4.3. It is easy to see that this could decrease the chances of finding good CRTs values, even if they are in principle infinite, particularly if no exhaustive search is made. To investigate this point, we computed by exhaustive search the number of sets of CRTs values which provide a greater classification accuracy than Chow's rule, on the entire data set, for values of the reject rate between 0 and 30%. We used a discretisation step of 0.001 as above. Note that, since the number of classes was two, the total number of sets of CRTs values to be considered would be 1001^2. In practice, since values of the reject thresholds lower than 0.5 provide a null reject rate (for a two-class problem), the maximum number of sets of CRTs values to be considered was 501^2. The average number of CRTs sets over the considered values of the reject rate are reported in Table 3, for the five different values of the standard deviation of the estimation errors. Table 3 shows that a significant number of sets of CRTs values provide a greater accuracy than Chow's rule. This is particularly interesting because the maximum difference between the accuracy achievable with the two rules was quite small, as shown in Fig.

13. The only exception was the case of estimation errors of lowest amplitude, as one could expect. Note however that the number of sets of CRTs was not monotonically increasing with the amplitude of the estimation errors.

Standard deviation	0.1	0.2	0.3	0.4	0.5
No. of sets of CRTs values	71	215	181	217	153

Table 3. Average number of sets of CRTs values which provided a higher accuracy than that of Chow's rule, for the same value of the reject rate. Values of the reject rate ranging from 0 to 30% were considered. The value of the discretisation step was 0.001.

7.2.2 Results on Feltwell and Phoneme data set

In this paragraph we present the results of the three groups of experiments on the Feltwell and Phoneme data set. The Feltwell data set can be considered particularly significant for testing the CRT rule, for two reasons. First, it consists of five classes. This means that five parameters must be evaluated for the CRT rule, while Chow's rule always requires to evaluate one only parameter. Secondly, the training set and the test set of the Feltwell data set are really independent, as in most practical application, since they consists of pixels belonging to different parts of the remote-sensing image. These characteristics could make it critical to obtaining reliable estimates of the CRTs values from a validation set.

The average A-R curves on the test set obtained by computing on the CRTs values by exhaustive search on the same test set (first group of experiments) are shown in Fig. 15. In this case we used a discretisation step equal to 0.01, due to the large number of patterns of the test sets, and on the number of classes of the Feltwell data set.

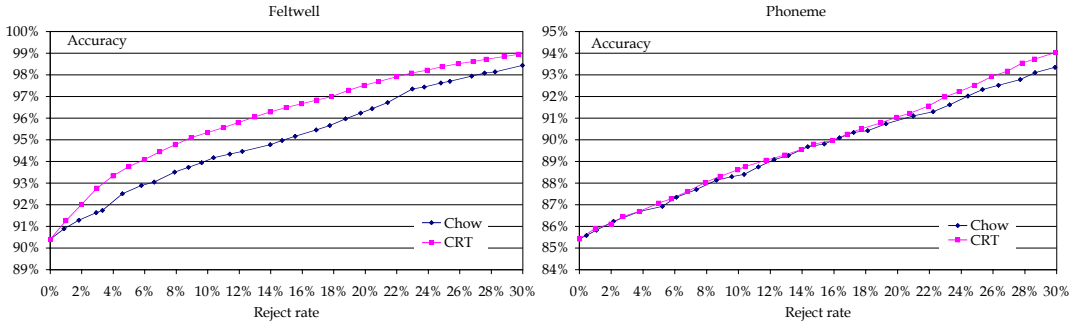


Fig. 15. Average test set A-R curves for Feltwell and Phoneme data sets, for the first group of experiments.

Fig. 15 shows that for the Feltwell data set the CRT rule provides a significantly better error-reject trade-off than Chow's rule. The difference in accuracy is quite constant over all values of the reject rate. The average difference is slightly greater than 1%, and reaches about 1.5% for values of the reject rate between 10% and 20%. The performances of the two rules on the Phoneme data set were instead very similar. A slight improvement was achieved by the CRT rule for values of the rejct rate greater than 20%.

For the second group of experiments we evaluated the performances of the two rules by computing the threshold values on a validation set. In order to analyse also the effects of the validation set size, we made these experiments with two different validation set sizes: we considered validation sets containing the 20% and the 50% of the patterns of the original training set. In this case the discretisation step used for computing the CRTs values by exahustive search on the validation set was 0.001, due to the lower number of patterns with respect to the first group of experiments. The average A-R curves on the test set are shown in Fig. 16.

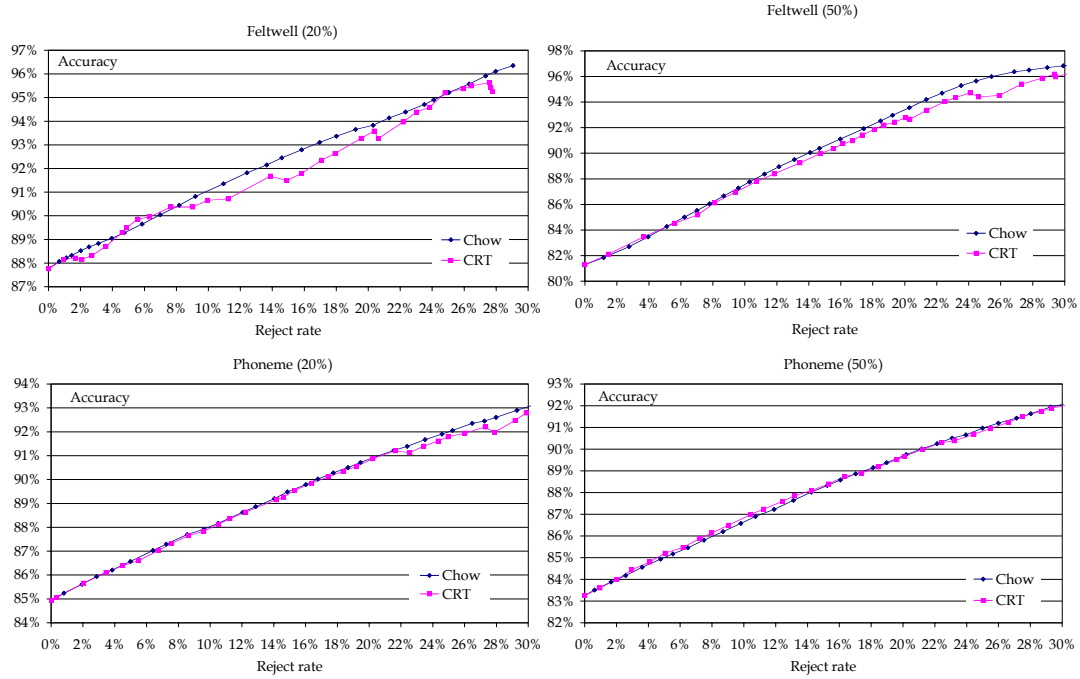


Fig. 16. Average test set A-R curves for Feltwell and Phoneme data sets. The CRTs values were computed from a validation set extracted from the original training set.

In this case, for the Feltwell data set the CRTs values computed from the validation set did not allow to outperform Chow'rule on the test set. As one can expect, the performance of the CRT rule was more affected when the CRTs values were computed on the smaller validation set. However, the performances achieved using the two validation sets were quite similar. On the Phoneme data set instead, the two rules performed similarly, as when computing the optimal CRTs values from the test set.

To better analyse the effects of the representativeness of the validation set with respect to the test set, we repeated the second group of experiments by extracting the validation set from the test set. Note that in this case we obtained ten different test sets, consisting of the patterns of the original test set which were not used as validation set. Ten different MLPs were trained on the original training set, which was not modified in these experiments. The corresponding A-R curves, reported in Fig. 17, are therefore the average curves with respect to the ten different test sets.

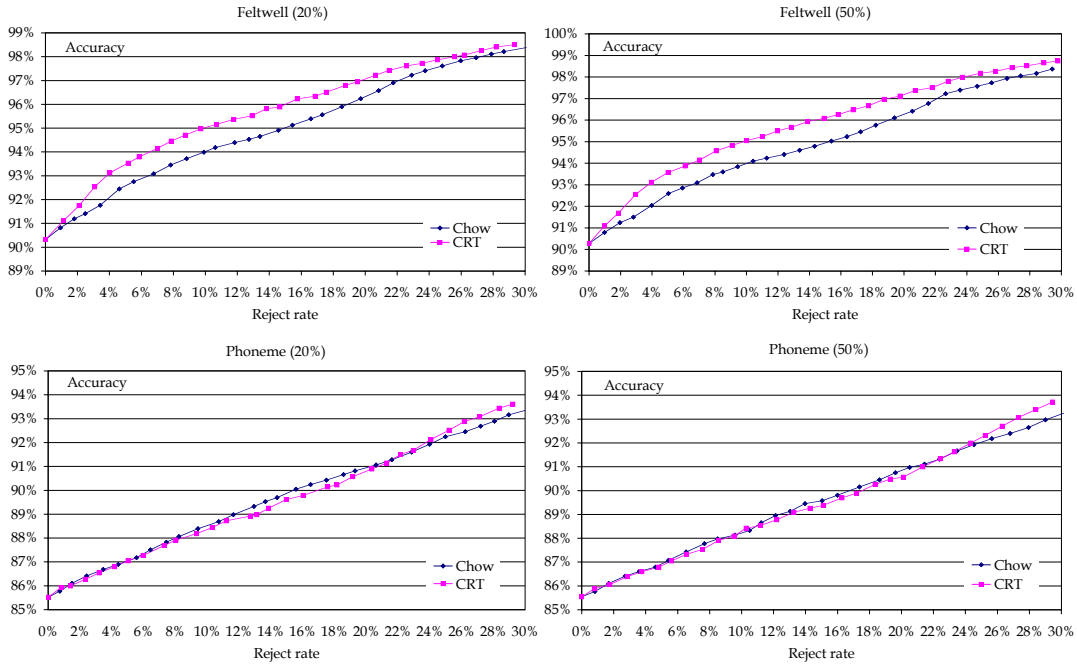
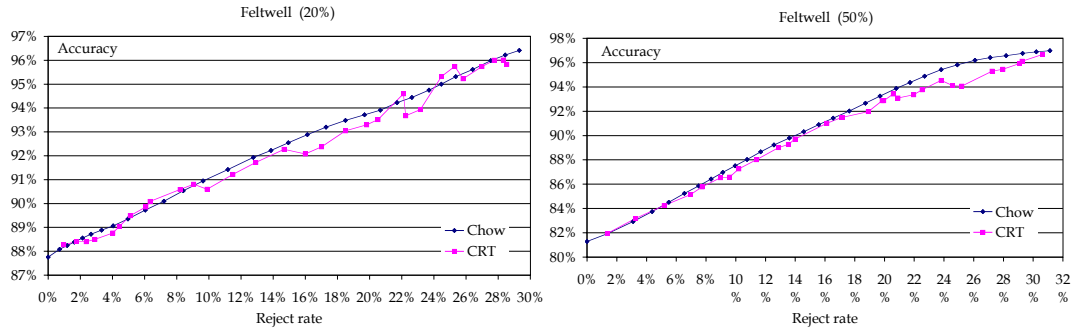


Fig. 17. Average test set A-R curves for the Feltwell and Phoneme data sets. The CRTs values were computed from a validation set extracted from the test set.

In this case it is interesting to note that on the Feltwell data set the CRT rule allowed to achieve a similar performance improvement over Chow's rule, as the one obtained by computing the optimal CRTs values on the test set. Moreover, there was no difference between the two different validation set sizes. An analogous result was obtained on the Phoneme data set, for which the performances of the CRT rule was similar to that of Chow's rule, as when computing the optimal CRTs values on the test set. This result is better than the one obtained for the artificial data set. It shows that it is possible to obtain reliable estimates of the optimal CRTs values from a validation set, if it is really representative of the test set. Moreover, this seems not to depend on the number of CRTs values to compute (that is, on the number of classes).

At this point, it was interesting to verify if the CRTs values can be estimated by avoiding an exhaustive search. This problem was tackled in the third group of experiments. In this case we repeated the second group of experiments described above, by computing the CRTs values using the algorithm described in paragraph 4.3. We used a discretisation step of 0.001, and a value of the parameter K equal to 200 (we remind that K is the number of threshold values considered at each iteration). For each value of the desired reject rate, the algorithm was run for twenty times, starting from random initial values of the CRTs. The results are reported in Figs. 18 and 19.



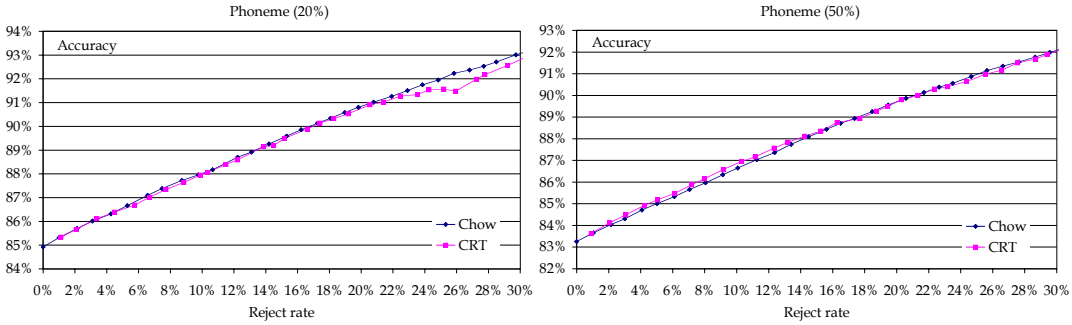


Fig. 18. Average A-R curves on the test set, for CRTs values estimated with the algorithm of paragraph 4.3 on a validation set extracted from the training set.

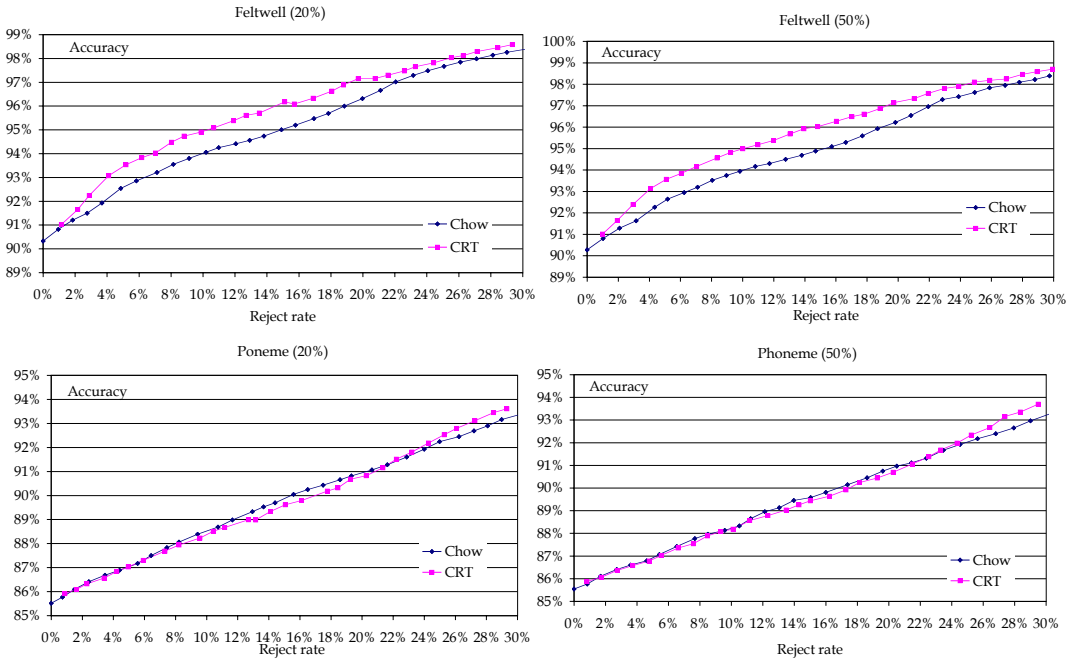


Fig. 19. Average A-R curves on the test set, for CRTs values estimated with the algorithm of paragraph 4.3 on the validation set extracted from the test set.

Let us compare the A-R curves of Figs. 18 and 19 with the corresponding curves of Figs. 16 and 17 respectively. It is possible to see that the behaviour of the CRT rule obtained by computing the CRTs values using our simple algorithm and by exhaustive search were very similar. This means that the problem of finding the CRTs values which maximise the classification accuracy, for a given value of the reject rate, can be reliably solved by using a simple algorithm which avoids an exhaustive search.

7.2.3 Results on Letter data set

In this paragraph we present results obtained for the first and second group of experiments on a large collection of two-class problems. The aim was to provide more significant conclusions about the CRT rule, from a statistical viewpoint. To this aim, we took every pair of classes of the Letter data set to be a two-class problem. In particular we considered only the non-linearly separable problems, since they are the most significant for testing the performance of a rejection rule. Of the

325 two-class problems, 193 were found to be non-linearly separable by Basu and Ho (1999). We present here the results for 42 of these problems.

From the first group of experiments (by computing the optimal CRTs values on the test set, by exhaustive search), it turned out that the maximum difference in accuracy between the CRT and Chow's rules, on the test set, reached for some problems 4%. The average difference in accuracy over all values of the reject rate, for any problem, was at least 1%. An example of the average test set A-R curves are reported in Fig. 20 for two two-class problems.

In the second group of experiments we computed the CRTs values from a validation set extracted from the original training set, by exhaustive search. We considered sizes of the validation set equal to 20% and 50% of the original training set. The results were quite interesting. First, we found that the average A-R curves on the test set always outperformed Chow's rule. More precisely, for 29 problems out of the 42 considered, the difference in accuracy on the test set between the CRTs and Chow's rules was lower than 1%. In the remaining 13 cases the difference in accuracy was between 1% and 2%. In Fig. 21 we reported the average A-R curves on the test set, obtained by computing the CRTs values from the validation set, for the same problems of Fig. 20. In particular, the first problem (classes B-K) is one example of cases in which the difference between the two rules was below 1%, while the other problem (classes D-O) is one example of cases in which the difference was between 1% and 2%. From these examples it is possible to see that the CRT rule uniformly outperformed Chow's rule on the test set, even if the CRTs values were computed from a relatively small validation set. By comparing the results of the first and second group of experiments, it turns out that even when the ideal advantage of the CRT rule over Chow's rule was quite low, estimating the CRTs values from a validation set did not affect considerably such advantage.

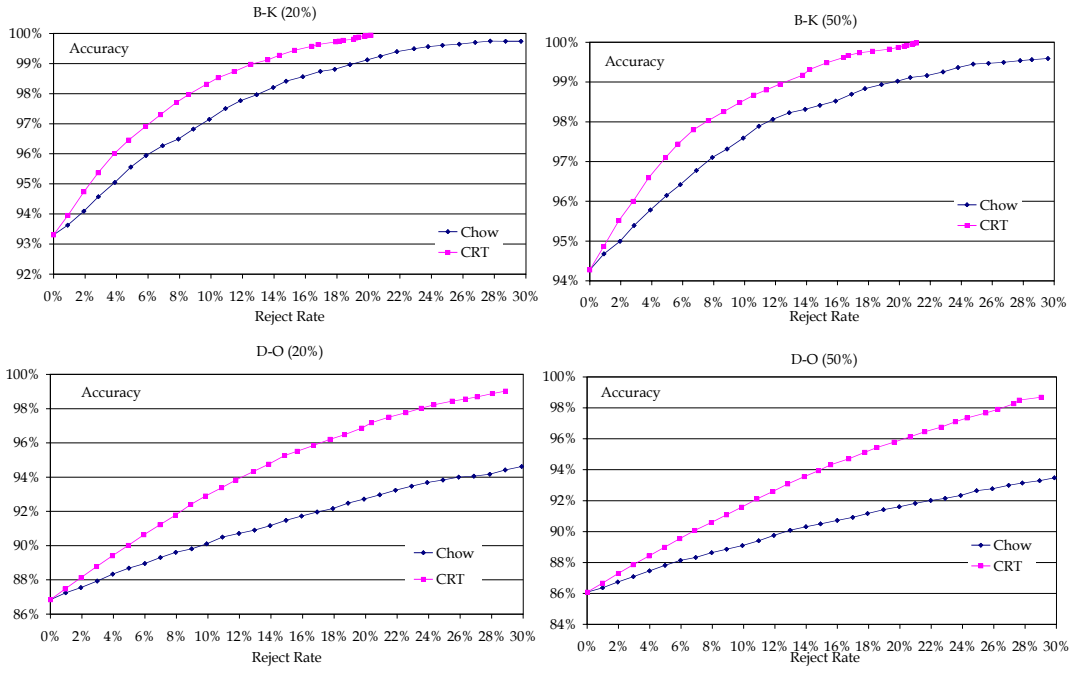


Fig. 20. Average test set A-R curves for problems B-K and D-O, for CRTs values computed on the test set.

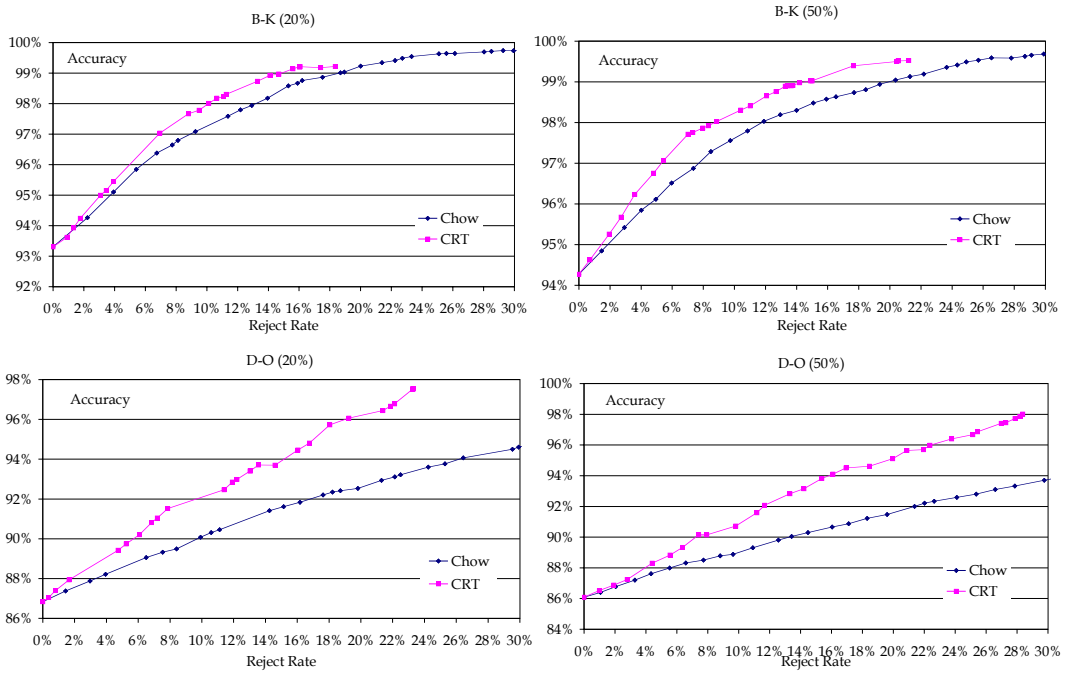


Fig. 21. Average test set A-R curves for problems B-K and D-O, for CRTs values computed on the validation set.

7.2.4 Conclusions

The results of the experimental analysis reported above allow to give a first answer to the two issues raised at the end of Chapter 4. Let us first consider the ideal improvement of the error-reject trade-off achievable by the CRT rule over Chow's rule. The first group of experiments on real data sets (Feltwell, Phoneme and Letter) showed an achievable improvement of classification accuracy up to 4% with respect to Chow's rule. In particular, the largest improvements were observed on some problem obtained from the Letter data set. On Feltwell data set the maximum improvement was about 1.5%, while on Phoneme data set no significant improvement was observed. Let us now consider the problem of obtaining reliable estimates of the optimal CRTs values. The second group of experiments showed that good CRTs values could be obtained from a validation set, provided that it is really representative of the problem at hand. This was pointed out in particular by experiments on Feltwell data set, where the ideal advantage of the CRT rule over Chow's rule was not large. In this case, good CRTs values were found only from a validation set extracted from the test set. Finally, the third group of experiments clearly pointed out that the CRTs values which maximise the classification accuracy on a given set of patterns can be reliably found without making an exhaustive search, by using a simple algorithm as the one described in Chapter 4.

From this experimental analysis we can conclude that the CRT rule can be useful for problems for which even accuracy improvements of 1% are significant, provided that a validation set representative of the problem at hand is available.

7.3 Experiments on the reject option in support vector machines

In this paragraph we present preliminary experimental results obtained by using the approach proposed in Chapter 6 to design SVMs with reject option. The aim of our experiments was to compare the error-reject trade-off achievable by our method with the one achievable by using the methods described in Chapter 3. To this aim, we considered a large collection of problems as in the experiments described above. Since SVMs are basically two-class classifiers, we used for these experiments the same non-linearly separable two-class problems obtained from the Letter data set, described in paragraph 7.2.3.

7.3.1 Setting of the experiments

As pointed out in Chapter 3, the rejection techniques proposed in the literature for SVMs are all equivalent to the follwing technique. A pattern $\mathbf{x}$ is rejected if its distance from the optimal separating hyperplane (OSH) is below a predefined threshold. The OSH is found by training a standard SVM without reject option. Since the output $f(\mathbf{x})$ of a SVM is proportional to the distance of the input pattern $\mathbf{x}$ from the OSH (in the feature space induced by the chosen kernel), this rejection rule can be restated as follows. A pattern $\mathbf{x}$ is rejected if:

$$|f(\mathbf{x})| < D ,$$

where D denotes the reject threshold. Otherwise the pattern is classified according to sign($f(\mathbf{x})$), as without reject option.

To implement this method, we trained SVMs by using the software SVM^{light} implemented by Joachims (1999). This software is available at `http://svmlight.joachims.org`. For our experiments we used a simple linear kernel: this means that the OSH was constructed on the original feature space. The value of the regularisation parameter C was set automatically by SVM^{light}. Using a trained SVM, the values of the reject threshold were computed by minimising the expected risk (7) estimated from the training set, for different values of the classification costs. Note that minimising the expression of the expected risk (7) is equivalent to minimise the expression

$$P(reject) + W \cdot P(error) ,$$

where $W = \frac{w_E - w_C}{w_R - w_C}$. Since $w_C < w_R < w_E$, it follows that the cost parameter W belongs to the interval $[1,+\infty)$. Therefore the different points of the A-R curve (corresponding to different values of the reject threshold) were obtained by minimising the above expression for different values of W. The values of the reject thresholds were then used to classify the test set. Note that we computed the values of the reject thresholds on the training set, instead of using a validation set. The reason of this choice is explained below.

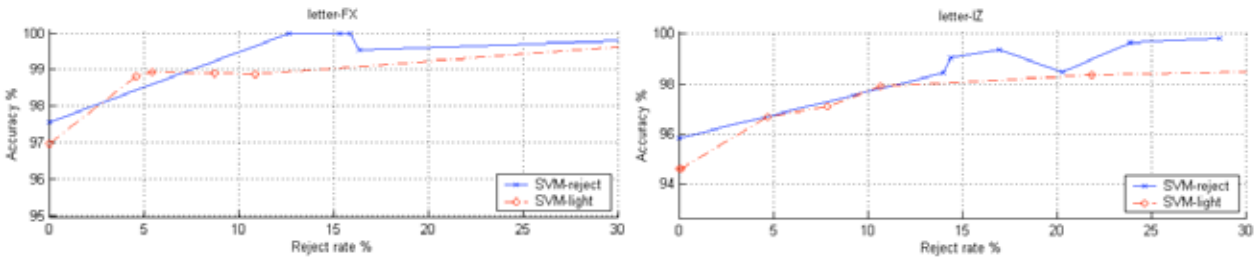
To implement our method, we used the training algorithm described in paragraph 6.4. We remind that our method consists in training a SVM-like classifier. The result of the training phase is not a separating hyperplane, but a pair of parallel hyperplanes which define the boundaries of the rejection region (in the feature space induced by the chosen kernel). This means that the

rejection region is obtained as a result of the training phase. The reason for which the reject threshold of the previous method was computed on the training set, instead of on a validation set, was to allow a correct comparison between the two methods. Let us call the classifier obtained using our method *SVM-reject* classifier. Training a SVM-reject classifier requires to choose, besides the kernel and the value of C, also the cost parameter w_R. As explained in Chapter 6, this parameter defines the cost of a rejection, relative to a misclassification cost equal to 1. To find a suitable value of C, we trained several SVM-reject classifiers on ten data set, for different values of C. We found that good values of the classification accuracy on the training set were achieved for a value of C equal to 0.1. This value was then used for all the experiments. The A-R curves for each data set were found by training a SVM-reject classifier for different values of the cost parameter w_R. More precisely, for each value of w_R we obtained from the training phase a pair of parallel hyperplanes. These hyperplanes were then used to classify the test set, according to the classification rule (57). Obvioulsy we used a linear kernel also for the SVM-reject classifier.

7.3.2 Results

In the following we present the A-R curves obtained on the test set for the first 29 two-class problems considered from the Letter data set, for values of the reject rate ranging from 0 to 30%. We denoted with *SVM-reject* the A-R curves obtained using our method, and with *SVM-light* the ones obtained using standard SVMs (trained with the SVM^{light} software). The results can be subdivided into two groups. For some problems the A-R curves obtained using the two methods intersect for one or more values of the reject rate. The difference in classification accuracy, for equal values of the reject rate, does not exceed 3%. In this cases it is not possible to say which of the two methods is best. This happens for 9 out of 29 of the considered problems, as shown in Fig. 22. Curiously, our method provides often a greater classification accuracy at a null reject rate. Let us now consider the other 20 problems. The corresponding test set A-R curves are shown in Fig. 23. For these problems our method provided a significant improvement of the accuracy-reject trade-off with respect to standard SVMs. The classification accuracy obtained by using our method were indeed greater than the one obtained with standard SVMs, for all values of the reject rate. The difference in accuracy was often between 2% and 4%. In two cases (for problems T-Y and X-Z), even accuracy improvements of about 7% were observed.

Note that the A-R curves obtained by using our method exhibited non-monotonic behaviour for some values of the reject rate. A possible explanation is that, for the corresponding values of the cost parameter w_R, the algorithm described in Chapter 6 allowed to find only a sub-optimal solution of the optimisation problem.



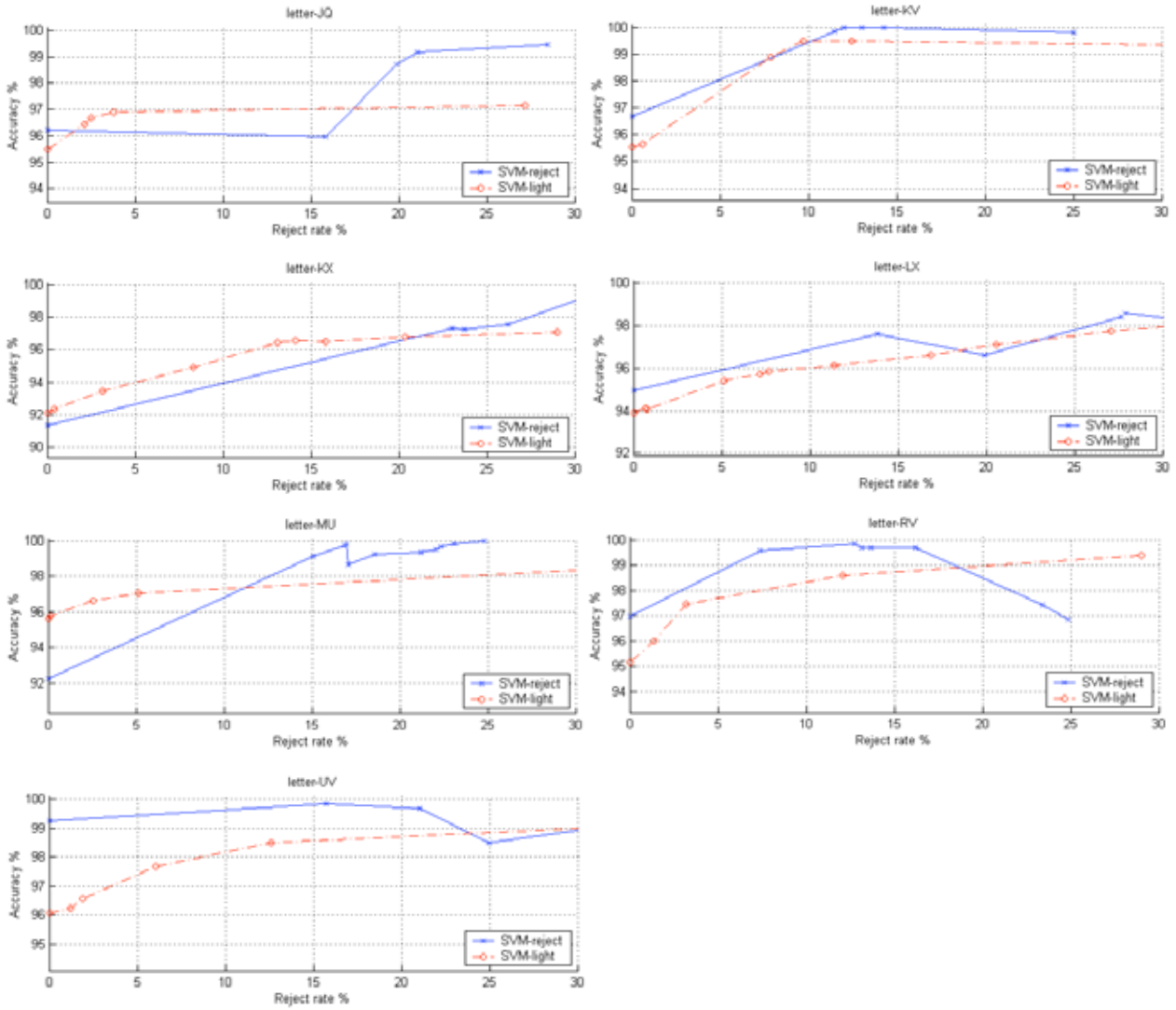
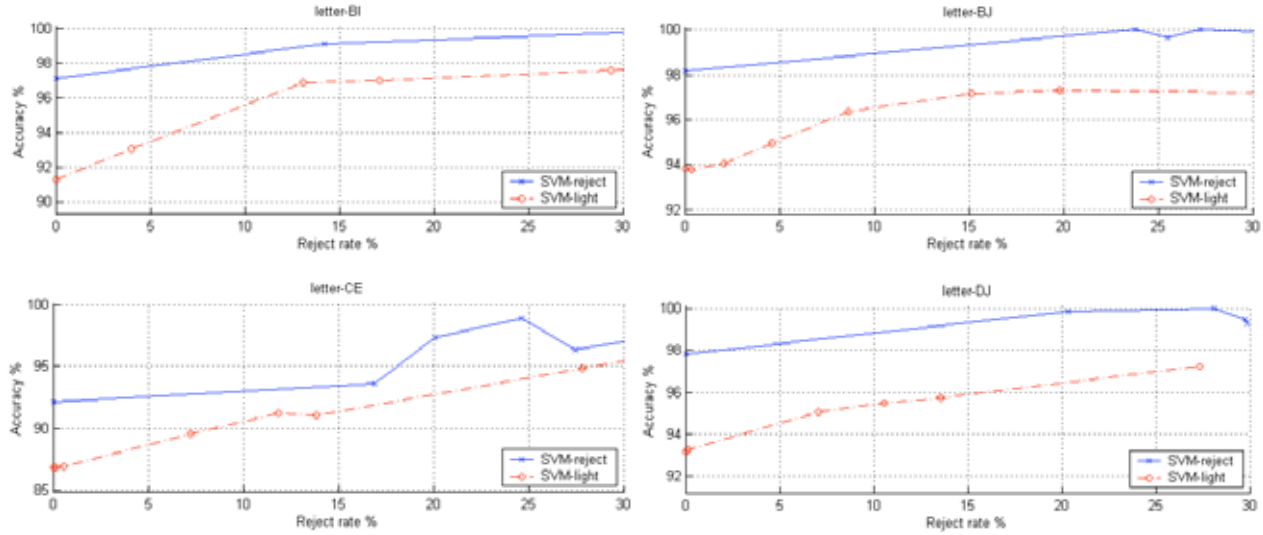
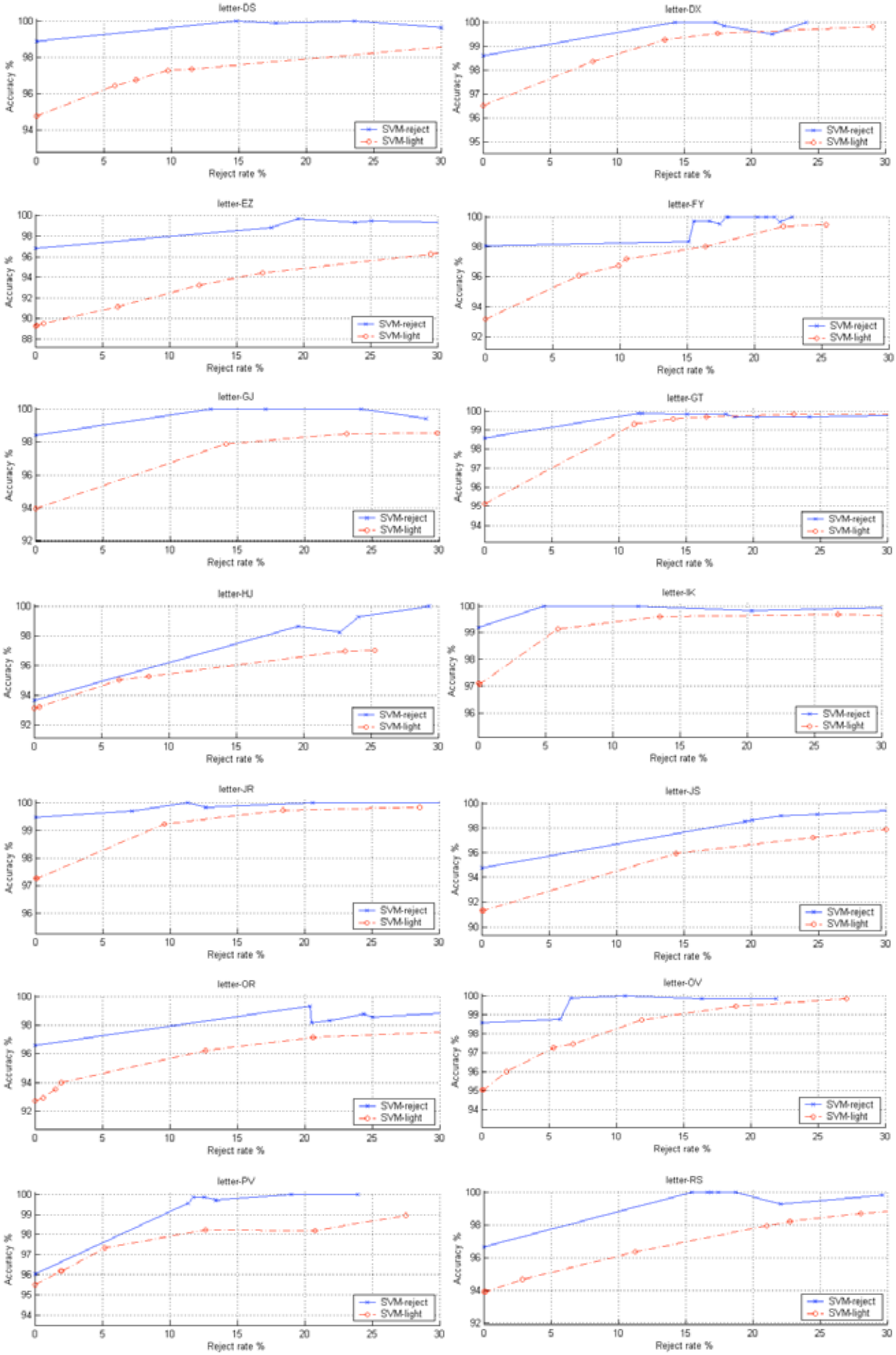


Fig. 22. Test set A-R curves for nine two-class problems for which neither of the two classification rules with reject option outperformed the other one.




letter-DS
letter-DX
letter-EZ
letter-FY
letter-GJ
letter-GT
letter-HJ
letter-IK
letter-JR
letter-JS
letter-OR
letter-OV
letter-PV
letter-RS
Accuracy %
Reject rate %
SVM-reject
SVM-light

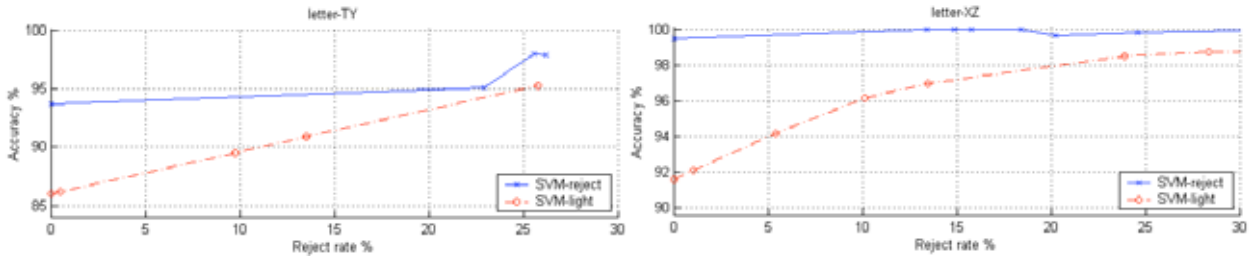


Fig. 23. Test set A-R curves for twenty two-class problems for which our method outperformed the rejection rule for standard SVMs proposed in the literature.

The above preliminary results show that our method can allow to effectively improve the accuracy-reject trade-off achievable by standard SVMs. As pointed out at the end of Chapter 6, both methods define the boundaries of the rejection region as a pair of parallel hyperplanes. However, our method does not constrain them to be always parallel to a given hyperplane, nor equidistant from it. The orientation and the position of the hyperplanes can instead vary for different values of the reject rate. The above results seem to prove that this greater flexibility can be exploited to improve the classification accuracy at any given reject rate.

We conclude by discussing the issues related to the complexity of the training algorithm we proposed to implement our method. As already pointed out, our algorithm does not guarantee to quickly converge to the optimal solution of the optimisation problem proposed in Chapter 6 to train SVMs with reject option. Indeed, the heuristics for choosing the pair of variables to update at each iteration and the stopping criterion, were based on conditions which are not necessary and sufficient to characterise the solution of that optimisation problem. In our experiments, we observed that the average training time required by our algorithm was greater of about one order of magnitude with respect to SVM^{light}. In particular, on problems obtained from the Letter data set, with training sets of about 400 patterns, our algorithm took less than five minutes, which was still an acceptable time. On the basis of the above results, we can say that it is worth devoting further work to obtain a simpler formulation of the optimisation problem proposed in Chapter 6, or to find a more efficient algorithm to solve that problem.

Chapter 8
Conclusions

The reject option is necessary in pattern recognition applications which require a high classification reliability. While the theoretical issues related to classification with reject option have been investigated in the literature since the works by Chow (1957; 1970), its practical implementation still presents remarkably open problems. In this thesis we focused on two open issues. The first one is related with the non-optimality of Chow's rule for classifiers which provide approximations of the a posteriori probabilities. The second one concerns the definition of a suitable rejection rule for classifiers which do not even provide approximations of the a posteriori probability, in particular for support vector machine classifiers. We proposed methods for implementing the reject option on these two kinds of classifiers, based on a theoretical analysis of the related problems.

Concerning the first issue, we pointed out that no work in the literature analysed how the estimation errors on the a posteriori probabilities affect the performance of Chow's rule. Therefore, the effectiveness of alternative rejection rules proposed in the literature was not theoretically proven. In this thesis we showed how the effects of the estimation errors can be reduced by using a rejection rule based on different reject thresholds for each class (CRT rule). In particular, we formally proved that the CRT rule provides a better error-reject trade-off than Chow's rule. A quantitative evaluation of the effectiveness of the CRT rule over Chow's rule was provided through experiments on real pattern recognitions problems. The results pointed out that on some applications the CRT rule can provide significant improvements of the classification accuracy with respect to Chow's rule. This means that, in principle, if good estimates of the optimal CRTs values could be obtained, using the CRT rule would always be preferable than using Chow's rule. Indeed, in the worst case in which the achievable improvement was negligible, the performance of the CRT rule would be equal to that of Chow's rule. As one can expect, we found that obtaining good estimates of the CRTs values relies on the availability of a validation set representative of the problem at hand. From a practical viewpoint one should also consider the computational cost required to find the optimal CRTs values relative to a given validation set. An exhaustive search can indeed be infeasible in problems with many classes. However, from our experiments it turned out that simple algorithms of negligible computational cost allow to obtain the same results achievable by an exhaustive search. Therefore, the main requirement for effectively exploiting the potential advantages of the CRT rule over Chow's rule is the availability of a representative validation set.

We also provided a preliminary theoretical analysis of the error-reject trade-off in multiple classifier systems, focusing on the simple average and weighted average combining rules. For these rules, we analysed how the effects of the estimation errors on the error-reject trade-off can be reduced by classifier combination. The main result was that simple averaging is the optimal linear

combination rule for “balanced” classifier ensembles, that is, for ensemble of classifiers whose estimation errors are identically distributed, with equal correlations. Note that this implies that the individual classifiers also exhibit equal average performances. Weighted averaging is instead required for imbalanced classifier ensembles. This result formalises some conclusions drawn in the literature, and extends them to classification with reject option. Further work is needed to formally define the concept of classifier imbalancing, so that it can be exploited to predict the improvement of the error-reject trade-off achievable by weighted averaging over simple averaging.

The second issue addressed in this thesis is related to the introduction of the reject option in classifiers which do not provide approximations of the a posteriori probabilities. In particular we considered support vector machine classifiers (SVMs). We pointed out that, despite the strong theoretical foundations of SVMs, the only rejection rule proposed in the literature is based on a simple heuristic, and consists in applying a reject threshold on the output of a trained SVM. We showed how SVM-like classifiers with reject option can be derived from statistical learning theory, by following an approach analogous to that of Vapnik. We then proposed a formulation of the training problem for such SVM-like classifiers with reject option, which consists in solving an optimisation problem similar to that of standard SVMs. We also proposed an algorithm to solve such an optimisation problem, based on one of the algorithms proposed in the literature for SVMs. The peculiarity of our method is that the rejection region in the feature space is obtained as a result of the training phase. This allows a greater flexibility in defining the rejection region, with respect to the rejection rule proposed in the literature. Preliminary experimental results showed that our method can allow to significantly improve the error-reject trade-off achievable by this rule. The main drawback is the computational complexity of the optimisation problem on which the training phase of a SVM-like classifier with reject option is based. A more efficient algorithm than the one proposed is needed to make our method applicable to classification problems with training sets of thousands of patterns or greater. Further work can also be devoted to find a formulation of the training problem leading to an optimisation problem of lower computational complexity.

Acknowledgements

The author whises to acknowledge the fundamental support of CRS4 (Center for Advanced Studies, Research and Development in Sardinia) to his work. This doctoral work would have been impossible without CRS4 grant.

References

Basu, M. and T.K. Ho (1999). The learning behavior of single neuron classifiers on linearly separable or nonseparable input. *Proc. of the 1999 Int. Joint Conference on Neural Networks*. Washington, DC.

Battiti, R. and A.M. Colla (1994). Democracy in neural nets: voting schemes for classification. *Neural Networks* 7, 691-707.

Bazaraa, M.S., H.D. Sherali and C.M. Shetty (1992). *Nonlinear Programming. Theory and Algorithms*. Wiley.

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

Chow, C.K. (1957). An Optimum Character Recognition System Using Decision Functions. *IRE Transactions on Electronic Computers* EC-6, 247-254.

Chow, C.K. (1970). On Optimum Recognition Error and Reject Tradeoff. *IEEE Transactions on Information Theory* IT-16, 41-46.

Cordella, L.P., C. De Stefano, F. Tortorella and M. Vento (1995). A method for improving classification reliability of multilayer perceptrons. *IEEE Transactions on Neural Networks* 6, 1140-1147.

Cortes, C. and V.N. Vapnik (1995). Support vector networks. *Machine Learning* 20, 1-25.

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

De Stefano, C., C. Sansone and M. Vento (2000). To reject or not to reject: that is the question - an answer in case of neural classifiers. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 30, 84-94.

Dubuisson, B. (1990). Decision with reject options. In: L. Torres, E. Masgrau and M.A. Lagunas, Eds., *Signal Processing V: Theories and Applications*, pp. 1715-1718. Elsevier Science, Amsterdam.

Dubuisson, B. and M. Masson (1993). A statistical decision rule with incomplete knowledge about classes. *Pattern Recognition* 26, 155-165.

Duda, R.O., P.E. Hart and D.G. Stork (2001). *Pattern Classification*. 2nd Edition, Wiley, New York.

Foggia, P., C. Sansone, F. Tortorella and M. Vento (1999). Multiclassification: reject criteria for the Bayesian combiner. *Pattern Recognition* 32, 1435-1447.

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

Fumera, G., F. Roli and G. Giacinto (2000a). Reject option with multiple thresholds. *Pattern Recognition* 33, 2099-2101.

Fumera, G., F. Roli and G. Giacinto (2000b). Multiple reject thresholds for improving classification reliability. *Proc. of Joint IAPR Int. Workshops on Syntactical and Structural Pattern Recognition and Statistical Pattern Recognition (SSSPR'2000)*, pp. 863-871. Springer LNCS.

Fumera, G. and F. Roli (2001). Error rejection in linearly combined multiple classifiers. *Proc. of the Second Int. Workshop on Multiple Classifier Systems (MCS 2001)*, pp. 329-338. Springer LNCS, Heidelberg.

Fumera, G., F. Roli and G. Vernazza (2001). A method for error rejection in multiple classifier systems. *Proc of 11th International Conference on Image Analysis and Processing (ICIAP 2001)*, pp. 454-458. IEEE Computer Society.

Giacinto, G., F. Roli and L. Bruzzone (2000). Combination of neural and statistical algorithms for supervised classification of remote-sensing images. *Pattern Recognition Letters* 21, 385-397.

Ha, T.M (1997). The optimum class-selective rejection rule. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 608-615.

Hansen, L.K., C. Liisberg and P. Salamon (1997). The error-reject trade-off. *Open Systems and Information Dynamics* 4, 159-184.

Hansen, L.K. and P. Salamon (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16, 993-1001.

Hastie, T. and R. Tibshirani (1996). Classification by pairwise coupling. Technical Report. Stanford University and University of Toronto.

Horiuchi, T. (1998). Class-selective rejection rule to minimise the maximum distance between selected classes. *Pattern Recognition* 31, 1579-1588.

Huang, Y.S. and C.Y. Suen (1995). A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17, 90-94.

Joachims, T. (1999). Making large-scale SVM learning practical. In: B. Schölkopf, C.J.C. Burges and A.J. Smola, Eds., *Advances in Kernel Methods - Support Vector Learning*, pp. 169-184. MIT Press.

Kittler, J. and F. Roli, Eds. (2000). *Proc. of the First International Workshop on Multiple Classifier Systems (MCS 2000)*. Springer LNCS, Heidelberg.

Kittler, J. and F. Roli, Eds. (2001). *Proc. of the Second International Workshop on Multiple Classifier Systems (MCS 2001)*. Springer LNCS, Heidelberg.

Lam., L. and C.Y. Suen (1995). Optimal combination of pattern classifiers. *Pattern Recognition Letters* 16, 945-954.

Lam., L. and C.Y. Suen (1997). Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Transactions on Systems, Man and Cybernetics - Part A* 27, 553-568.

Le Cun Y., B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard and L.D. Jakel (1990). Handwritten digit recognition with a back-propagation network. In D.S. Touretzsky, Ed., *Advances in Neural Information Processing Systems II*, pp. 396-404, Morgan Kaufman, San Mateo.

Kuncheva, L.I. and R.P.W. Duin (2000). Is independence good for combining classifiers? *Proc. of the 15th Int. Conference on Pattern Recognition*, Vol. 2, pp.168-171.

Kwok, J. T.-Y. (1999). Moderating the outputs of support vector machines. *IEEE Transactions on Neural Networks* 10, 1018-1031.

Madevska-Bogdanova, A. and D. Nikolic (2000). A new approach on modifying SVM outputs. *Proc. of the IEEE-INNS-ENNS Int. Joint Conference on Neural Networks*, Vol. 6, pp. 395-398.

MacKay, D.J.C. (1992). Bayesian interpolation. *Neural Computation* 4, 415-447.

Mizutani, H. (1998). Discriminative learning for minimum error and minimum reject classification. *Proc. of 14th International Conference on Pattern Recognition*, Vol. 1, pp. 136-140.

Mukherjee, S., P. Tamayo, D. Slonim, A. Verri, T. Golub, J.P. Mesirov and T. Poggio (1998). Support vector machine classification of microarray data. Technical report. Massachusetts Institute of Technology.

Muzzolini, R., Y.H. Yang and R. Pierson (1998). Classifier design with incomplete knowledge. *Pattern Recognition* 31, 345-369.

Perrone, M.P. and L.N. Cooper (1993). When networks disagree: ensemble methods of hybrid neural networks. In: R.J. Mammone, Ed., *Neural Networks for Speech and Image Processing*. Chapman-Hall.

Platt, J.C. (1999a). Probabilistic outputs for support vector machines and comparison to regularised likelihood methods. In: A.J. Smola, P. Bartlett, B. Schölkopf, D. Schuurmans, Eds., *Advances in Large Margin Classifiers*. MIT Press.

Platt, J.C. (1999b). Fast training of supprt vector machines using sequential minimal optimisation. In: B. Schölkopf, C.J.C. Burges and A.J. Smola, Eds., *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

Pontil, M. and A. Verri (1998). Properties of Support Vector Machines. *Neural Computation* 10, 955-974.

Pudil, P., J. Novovicova, S. Blaha and J. Kittler (1992). Multistage pattern recognition with reject option. *Proc. of 11th IAPR International Conference on Pattern Recognition Methodology and Systems*, 92-95.

Richard, M.D. and R.P. Lippmann (1991). Neural network classifiers estimate bayesian a posteriori probabilities. *Neural Computation* 3, 461-483.

Ruck, D.W., S.K. Rogers, M. Kabrisky, M. Oxley and B. Suter (1990). The multilayer perceptron as an approximation to a Bayes optimal discriminant function. *IEEE Transactions on Neural Networks* 4, 296-298.

Sansone, C., F. Tortorella and M. Vento (2001). A classification reliability driven reject rule for multi-expert systems. *Int. Journal of Pattern Recognition and Artificial Intelligence* 15, 885-904.

Serpico, S.B. and F. Roli (1995). Classification of multi-sensor remote-sensing images by structured neural networks. *IEEE Transactions on Geoscience and Remote Sensing* 33, 562-578.

Tortorella, F. (2000). An optimal reject rule for binary classifiers. *Proc. of Joint IAPR Int. Workshops on Syntactical and Structural Pattern Recognition (SSPR 2000) and Statistical Pattern Recognition (SPR 2000)*, Springer LNCS, 611-620.

Tumer, K. (1996). *Linear and Order Statistics Combiners for Reliable Pattern Classification*. PhD thesis, The University of Texas, Austin.

Tumer, K. and J. Ghosh (1996a). Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognition* 29, 341-348.

Tumer, K. and J. Ghosh (1996b). Error correlation and error reduction in ensemble classifiers. *Connection Science* 8, 385-404.

Tumer, K. and J. Ghosh (1999). Linear and order statistics combiners for pattern classification. In: A.J.C.. Sharkey, Ed., *Combining Artificial Neural Nets*. Springer, London, 127-161.

Ueda, N. (2000). Optimal linear combination of neural networks for improving classification performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 207-215.

Vapnik, V.N. (1995). *The nature of Statistical Learning Theory*. Springer-Verlag, New York.

Vapnik, V.N. (1998). *Statistical Learning Theory*. Wiley, New York.

Vapnik, V.N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks* 10, 988-999.

Vasconcelos, G.C., M.C. Fairhurst and D.L. Bisset (1993). Enhanced reliability of multilayer perceptron networks through controlled pattern rejection. *Electronic Letters* 29, 261-263.

Wan, E.A. (1990). Neural network classification: a Bayesian interpretation. *IEEE Transactions on Neural Networks* 1, 303-304.

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

Yau, H.C. and M.T. Manry (1992). Automatic determination of reject threshold in classifiers employing discriminant functions. *IEEE Transactions on Signal Processing* 40, 711-713.