Journal of Machine Learning Research 8 (2007) 103-130 Submitted 3/05; Revised 7/06; Published 1/07

Minimax Regret Classifier for Imprecise Class Distributions

Rocío Alaiz-Rodríguez ROCIO.ALAIZ@UNILEON.ES
Dpto. de Ingeniería Eléctrica y de Sistemas y Automática
Campus de Vegazana, Universidad de León
24071 León, Spain

Alicia Guerrero-Curieses ALICIA.GUERRERO@URJC.ES
Dpto. de Teoría de la Señal y Comunicaciones
Campus de Fuenlabrada, Universidad Rey Juan Carlos
Camino del Molino s/n, 28943 Fuenlabrada-Madrid, Spain

Jesús Cid-Sueiro JCID@TSC.UC3M.ES
Dpto. de Tecnologías de las Comunicaciones
EPS, Universidad Carlos III de Madrid
Avda. de la Universidad, 30, 28919 Leganés-Madrid, Spain

Editor: Dale Schuurmans

Abstract

The design of a minimum risk classifier based on data usually stems from the stationarity assumption that the conditions during training and test are the same: the misclassification costs assumed during training must be in agreement with real costs, and the same statistical process must have generated both training and test data. Unfortunately, in real world applications, these assumptions may not hold. This paper deals with the problem of training a classifier when prior probabilities cannot be reliably induced from training data. Some strategies based on optimizing the worst possible case (conventional minimax) have been proposed previously in the literature, but they may achieve a robust classification at the expense of a severe performance degradation. In this paper we propose a *minimax regret* (*minimax deviation*) approach, that seeks to minimize the maximum deviation from the performance of the optimal risk classifier. A neural-based *minimax regret* classifier for general multi-class decision problems is presented. Experimental results show its robustness and the advantages in relation to other approaches.

Keywords: classification, imprecise class distribution, minimax regret, minimax deviation, neural networks

1. Introduction - Problem Motivation

In the general framework of learning from examples and specifically when dealing with uncertainty, the robustness of the decision machine becomes a key issue. Most machine learning algorithms are based on the assumption that the classifier will use data drawn from the same distribution as the training data set. Unfortunately, for most practical applications (such as remote sensing, direct marketing, fraud detection, information filtering, medical diagnosis or intrusion detection) the target class distribution may not be accurately known during learning: for example, because the cost of labelling data may be class-dependent or the prior probabilities are non-stationary. Therefore, the data used to design the classifier (within the Bayesian context (see VanTrees, 1968), the prior probabilities and the misclassification costs) may be non representative of the underlying real distributions.

If the ratio of training data corresponding to each class is not in agreement with real class distributions, designing Bayes decision rules based on prior probabilities estimated from these data will be suboptimal and can seriously affect the reliability and performance of the classifier.

A similar problem may arise if real misclassification costs are unknown during training. However, they are usually known by the end user, who can adapt the classifier decision rules to cost changes without re-training the classifier. For this reason, our attention in this paper is mainly focused on the problem of uncertainty in prior probabilities. Furthermore, being aware that class distribution is seldom known (at least totally) in real world applications, a robust approach (as opposite to adaptive) that prevents severe performance degradation appears to be convenient for these situations.

Besides other adaptive and robust approaches that address this problem (discussed in more detail in Section 2.2) it is important to highlight those that handle the problem of uncertainty in priors by following a robust minimax principle: minimize the maximum possible risk. Analytic foundations of minimax classification are widely considered in the literature (see VanTrees, 1968; Moon and Stirling, 2000; Duda et al., 2001, for instance) and a few algorithms to carry out minimax decisions have been proposed. From computationally expensive ones such as estimating probability density functions (Takimoto and Warmuth, 2000; Kim, 1996) or using methods from optimization (Polak, 1997) to simpler ones like neural network training algorithms (Guerrero-Curieses et al., 2004; Alaiz-Rodriguez et al., 2005).

Minimax classifiers may, however, be seen as over-conservative since its goal is to optimize the performance under the least favorable conditions. Consider, for instance, a direct marketing campaign application carried out in order to maximize profits. Since optimal decisions rely on the proportion of potential buyers and it is usually unknown in advance, our classification system should take into account this uncertainty. Nevertheless, following a pure minimax strategy can lead to solutions where minimizing the maximum loss implies considering there are no potential clients. If it is the case, this minimax approach does not seem to be suitable for this kind of situation.

In this imprecise class distribution scenario, it can be noticed that the classifier performance may be highly deviated from the optimal, that is, that of the classifier knowing actual priors. Minimizing this gap (that is, the maximum possible deviation with respect to the optimal classifier) is the focus of this paper. We seek for a system as robust as the conventional minimax approach but less pessimistic at the same time. We will refer to it as a *minimax deviation* (or *minimax regret*) classifier. In contrast to other robust and adaptive approaches, it can be used in general multiclass problems. Furthermore, as shown in Guerrero-Curieses et al. (2004), minimax approaches can be used in combination with the adaptive proposal by Saerens et al. (2002) to exploit its advantages.

This *minimax regret* approach has recently been applied in the context of parameter estimation (Eldar et al., 2004; Eldar and Merhav, 2004) and a similar competitive strategy has been used in the context of hypothesis testing (Feder and Merhav, 2002).

Under prior uncertainty, our solution provides an upper bound of the performance divergence from the optimal classifier. We propose a simple learning rate scaling algorithm in order to train a neural-based *minimax deviation* classifier. Although training can be based on minimizing any objective function, we have chosen objective functions that provide estimates of the posterior probabilities (see Cid-Sueiro and Figueiras-Vidal, 2001, for more details).

This paper is organized as follows: the next section provides an overview of the problem as well as some previous approaches to cope with it. Next, Section 3 states the fundamentals of minimax classification together with a deeper analysis of the *minimax regret* approach proposed in this paper. Section 4 presents a neural training algorithm to get a neural-based *minimax regret* classifier under complete uncertainty. Moreover, practical situations with partial uncertainty in priors are also discussed. A learning algorithm to solve them is provided in Section 5. In Section 6, some experimental results show that *minimax regret* classifiers outperform (in terms of maximum risk deviation) classifiers trained on re-balanced data sets and those with the originally assumed priors. Finally, the main conclusions are summarized in Section 7.

2. Problem Overview

Traditionally, supervised learning lies in the fact that training data and real data come from the same (although unknown) statistical model. In order to carefully analyze to what extend classifier performance depends on conditions such as class distribution or decision costs, learning and decision theory principles are briefly revisited. Next, some previous approaches to deal with environment imprecision are reviewed.

2.1 Learning and Making Optimal Decisions

Let $S = \{(\mathbf{x}^k, \mathbf{d}^k), k = 1, \ldots, K\}$ denote a set of labelled samples where $\mathbf{x}^k \in \mathbb{R}^N$ is an observation feature vector and $\mathbf{d}^k \in U_L = \{\mathbf{u}_0, \ldots, \mathbf{u}_{L-1}\}$ is the label vector. Class-i label $\mathbf{u}_i$ is a unit L-dimensional vector with components $u_{i,j} = \delta_{ij}$, with every component equal to 0, except the i-th component which is equal to 1.

We assume a learning process that estimates parameters $\mathbf{w}$ of a non-linear mapping $\mathbf{f}_\mathbf{w} : \mathbb{R}^N \rightarrow \mathcal{P}$ from the input space into probability space $\mathcal{P} = \{\mathbf{p} \in [0,1]^L | \sum_{i=0}^{L-1} p_i = 1\}$. The *soft* decision is given by $\mathbf{y}^k = \mathbf{f}_\mathbf{w}(\mathbf{x}^k) \in \mathcal{P}$ and the hard output of the classifier is denoted by $\widehat{\mathbf{d}}$. Note that $\mathbf{d}$ and $\widehat{\mathbf{d}}$ will be used to distinguish the actual class from the predicted one, respectively.

Several costs (or benefits) associated with each possible decision are also defined: c_{ij} denotes the cost of deciding in favor of class i when the true class is j. Negative values represent benefits (for instance, c_{ii}, which is the cost of correctly classifying a sample from class i could be negative in some practical cases).

In general cost-sensitive classification problems, either misclassification costs c_{ij} or c_{ii} costs can take different values for each class. Thus, there are many applications where classification errors lead to very different consequences (medical diagnosis, fault detection, credit risk analysis), what implies misclassification costs c_{ij} that may largely vary between them. In the same way, there are also many domains where correct decision costs (or benefits) c_{ii} do not take the same value. For instance, in targeted marketing applications (Zadrozny and Elkan, 2001), correctly identifying a buyer implies some benefit while correctly classifying a non buyer means no income. The same applies to medical diagnosis domains such as the gastric carcinoma problem studied in Güvenir et al. (2004). In this case, the benefit of correct classification also depends on the class: the benefit of correctly classifying an early stage tumor is higher than that of a later stage.

The expected risk (or loss) R is given by

$$R = \sum_{j=0}^{L-1}\sum_{i=0}^{L-1} c_{ij} P\{\widehat{\mathbf{d}} = \mathbf{u}_i | \mathbf{d} = \mathbf{u}_j\} P_j \quad , \tag{1}$$

where $P\{\widehat{\mathbf{d}} = \mathbf{u}_i | \mathbf{d} = \mathbf{u}_j\}$ with $i \neq j$ represent conditional error probabilities, and $P_j = P\{\mathbf{d} = \mathbf{u}_j\}$ is the prior probability of class $\mathbf{u}_j$.

Defining the conditional risk of misclassifying samples from class $\mathbf{u}_j$ as

$$R_j = \sum_{i=0}^{L-1} c_{ij} P\{\widehat{\mathbf{d}} = \mathbf{u}_i | \mathbf{d} = \mathbf{u}_j\} \ \ ,$$

we can express risk (1) as

$$R = \sum_{i=0}^{L-1} R_i P_i \ \ . \tag{2}$$

It is well-known that the Bayes decision rule for the minimum risk is given by

$$\widehat{\mathbf{d}} = \arg \min_{\mathbf{u}_i} \{ \sum_{j=0}^{L-1} c_{ij} P\{\mathbf{d} = \mathbf{u}_j | \mathbf{x}\}\} \ \ , \tag{3}$$

where $P\{\mathbf{d} = \mathbf{u}_i | \mathbf{x}\}$ is the a *posteriori* probability of class i given sample $\mathbf{x}$.

The optimal decision rule depends on posterior probabilities and therefore, on the prior probabilities and the likelihood.

In theory, as long as posterior probabilities (or likelihood and prior probabilities) are known, the optimal decision in Eq. (3) can be expressed after a trivial manipulation as a function of the cost differences between the costs $(c_{ij} - c_{jj})$ (Duda et al., 2001). This is the reason why c_{jj} is usually assumed to be zero and the value of the cost difference is directly assigned to c_{ij}. When dealing with practical applications, however, some authors (Zadrozny and Elkan, 2001; Güvenir et al., 2004) have urged to use meaningful decision costs measured over a common baseline (and not necessarily taking $c_{jj} = 0$) in order to avoid mistakes that otherwise could be overlooked. For this reason and, what is more important, the uncertainty class distribution problem addressed in this paper, decision costs measured over a common baseline are considered. Furthermore, absolute values of decision costs are relevant to the design of classifiers under the minimax regret approach.

2.2 Related Work: Dealing with Cost and Prior Uncertainty

Most proposals to address uncertainty in priors fall into the categories of adaptive and robust solutions. While the aim of a robust solution is to avoid a classifier with very poor performance under any conditions, an adaptive system pursues to fit the classifier parameters using more incoming data or more precise information.

With an adaptive-oriented principle, Provost (2000) states that, once the classifier is trained under specific class distributions and cost assumptions (not necessarily the operating conditions), the selection of the optimal classifier for specific conditions is carried out by a correct placement of the decision thresholds. In the same way, the approaches in Kelly et al. (1999) and Kubat et al. (1998) consider that tuning the classifier parameters should be left to the end user, expecting that class distributions and misclassification costs will be precisely known then.

Some graphical methods based on the ROC curve have been proposed in Adams and Hand (1998) and Provost and Fawcett (2001) in order to compare the classifier performance under imprecise class distributions and/or misclassification costs. The ROC convex hull method presented in Provost and Fawcett (2001) (or the alternative representation proposed in Drummond and Holte (2000)) allows the user to select potentially optimal classifiers, providing a flexible way to select them when precise information about priors or costs is available. Under imprecision, some classifiers can be discarded but this does not necessarily provide a method to select the optimal classifier between the possible ones and fit its parameters. Furthermore, due to its graphical character, these methods are limited to binary classification problems.

Changes in prior probabilities have also been discussed by Saerens et al. (2002), who proposes a method based on re-estimating the prior probabilities of real data in an unsupervised way and subsequently adjusting the outputs of the classifier according to the new a *priori* probabilities. Obviously, the method requires enough unlabelled data being available for re-estimation.

As an alternative to adaptive schemes, several robust solutions have been proposed, as the resampling methods, especially in domains where imbalanced classes come out (Kubat and Matwin, 1997; Lawrence et al., 1998; Chawla et al., 2002; Barandela et al., 2003). Either by undersampling or oversampling, the common purpose is to balance artificially the training data set in order to get a uniform class distribution, which is supposed to be the least biased towards any class and, thus, the most robust against changes in class distributions.

The same approach is followed in cost sensitive domains, but with some subtle differences in practice. It is well known that class priors and decision costs are intrinsically related. For instance, different decision costs can be simulated by altering the priors and vice versa (see Ting, 2002, for instance). Thus, when a uniform distribution is desired in a cost sensitive domain, but working with cost insensitive decision machines, class priors are altered according to decision costs, what is commonly referred as *rebalancing*.

The manipulation of the training data distribution has been applied to cost-sensitive learning in two-class problems (Breiman et al., 1984) in the following way: basically, the class with higher misclassification cost (suppose *n* times the lowest misclassification cost) is represented with *n* times more examples than the other class. Besides random sampling strategies, other sampling-based *rebalancing* schemes have been proposed to accomplish this task, like those considering closeness to the boundaries between classes (Japkowicz and Stephen, 2002; Zhou and LiuJ, 2006) or the cost-proportionate rejection sampling presented in Zadrozny et al. (2003). Extending the formulation of this type of procedures to general multiclass problems with multiple (and possibly asymmetric) inter-class misclassification costs appears to be a nontrivial task (Zadrozny et al., 2003; Zhou and LiuJ, 2006), but some progress has been made recently with regard to this latter point (Abe et al., 2004). Note, also, that many (although not all) of these rebalancing strategies are usually implemented by oversampling and/or subsampling, that is, replicating examples (without adding any extra information) and/or deleting them (which implies information loss).

3. Robust Classifiers Under Prior Uncertainty: Minimax Classifiers

Prior probability uncertainty can be coped from a robust point of view following a minimax derived strategy. Minimax regret criterion is discussed in this section after presenting the conventional minimax criterion.

Although our approach extends to general multi-class problems and the discussion is carried out in that way, we will first illustrate, for the sake of clarity and simplicity, a binary situation.

3.1 Minimax Classifiers

As Eq. (3) shows, the minimum risk decisions depend on the misclassification costs, c_{ij}, and the posterior class probabilities and, thus, they depend on the prior probabilities, P_i. Different prior distributions (frequency for each class) give rise to different Bayes classifiers. Fig. 1 shows the Bayes risk curve, $R_B(P_1)$ versus class-1 prior probability for a binary classification problem.

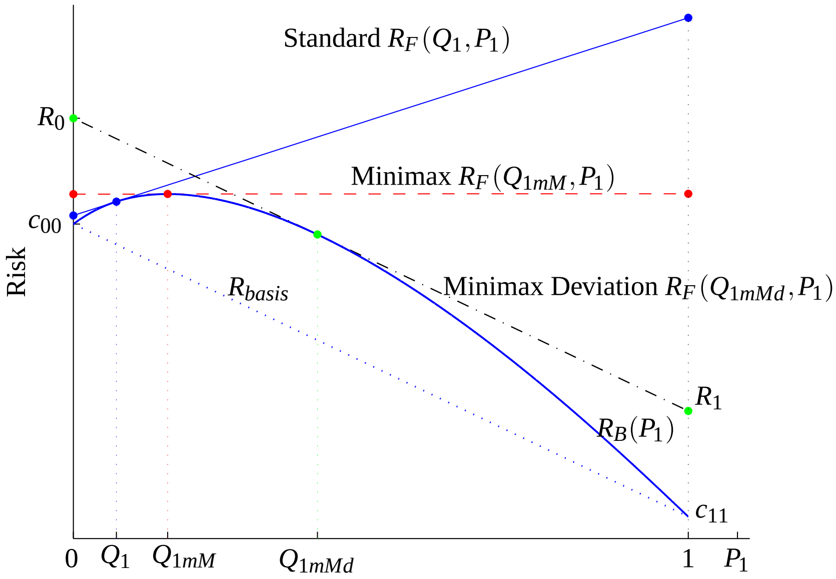

Figure 1: Risk vs. P_1. Minimum risk curve and performance under prior changes for the *standard*, *minimax* and *minimax deviation* classifier. $R_B(P_1)$ stands for the optimal Bayes Risk against P_1. $R_F(Q_1,P_1)$ denotes the Risk of a standard classifier (Fixed decision rule optimized for prior probabilities Q_1 estimated in the training phase) against P_1. $R_F(Q_{1mM},P_1)$ denotes the Risk of a *minimax* classifier (Fixed decision rule optimized for the minimax probabilities Q_{1mM}) against P_1. $R_F(Q_{1mMd},P_1)$ denotes the Risk of a *minimax deviation* classifier (Fixed decision rule optimized for the minimax deviation probabilities Q_{1mMd}) against P_1.

If the prior probability distribution is unknown when the classifier is designed, or this distribution changes with time or from one environment to other, the mismatch between training and test conditions can degrade significantly the classifier performance.

For instance, assume that $\mathbf{Q} = (Q_0, Q_1)$ is the vector with class-0 and class-1 prior probabilities estimated in the training phase, respectively, and let $R_B(Q_1)$ represent the minimum (Bayes) risk attainable by any decision rule for these priors. Note, that, according to Eq. (2), for a given classifier, the risk is a linear function of priors. Thus, risk $R_F(Q_1,P_1)$ associated to the (fixed) classifier optimized for $\mathbf{Q}$ changes linearly with actual prior probabilities P_1 and $P_0 = 1-P_1$, going from $(0,R_0)$ to $(1,R_1)$ (the continuous line in Fig. 1), where R_0 and R_1 refer to the class conditional risks for classes 0 and 1, respectively. Fig. 1 shows the impact of this change in priors and how performance deviates from optimal.

Also, it can be shown (see VanTrees, 1968, for instance) that the minimum risk curve obtained for each prior is convex and the risk function of a given classifier verifies $R_F(Q_1,P_1) \geq R_B(P_1)$ with a tangent point at $P_1 = Q_1$.

The dashed line in Fig. 1 shows the performance of the *minimax* classifier, which minimizes the maximum possible risk under the least favorable priors, thus providing the most robust solution, in the sense that performance becomes independent from priors. From Fig. 1, it becomes clear that the minimax classifier is optimal for prior probabilities $\mathbf{P} = \mathbf{Q_{mM}} = (Q_{0mM}, Q_{1mM})$ maximizing R_B. Thus, this strategy is equivalent to maximizing the minimum risk (Moon and Stirling, 2000; Duda et al., 2001). We will refer to them as the minimax probabilities.

Fig. 1 also makes clear that although a minimax classifier is a robust solution to address the imprecision in priors, it may become a somewhat pessimistic approach.

3.2 Minimax Deviation Classifiers

We propose an alternative classifier that, instead of minimizing the maximum risk, minimizes the maximum deviation (regret) from the optimal Bayes classifier. In the following, we will refer to it as the *minimax deviation* or *minimax regret* classifier.

A comparison between *minimax* and *minimax deviation* approaches is also shown in Fig. 1. This latter case corresponds to a classifier trained on prior probabilities $\mathbf{P} = \mathbf{Q_{mMd}}$ with performance as a function of priors given by a line (a plane or hyperplane for three or more classes, respectively) parallel to what we name, in the following, basis risk ($R_{basis} = c_{00}(1-P_1) + c_{11}P_1$).

Note that the maximum deviation (with respect to priors) of the classifier optimized for $\mathbf{Q}$ is given by

$$D(\mathbf{Q}) = \max_{P_1} \{R_F(Q_1, P_1) - R_B(P_1)\} = \max\{R_0 - c_{00}, R_1 - c_{11}\} \quad .$$

The inspection of Fig. 1 shows that the minimum of D (with respect to $\mathbf{Q}$) is achieved when

$$R_0 - c_{00} = R_1 - c_{11} \quad ,$$

which means that line $R_F(Q_1, P_1)$ is parallel to arc named R_{basis} in the figure and tangent to R_B at Q_{1mMd}. Therefore, the *minimax regret* classifier is also the Bayes solution with respect to the least favorable priors (Q_{0mMd}, Q_{1mMd}) (see Berger, 1985, for instance), which will be denoted as minimax deviation probabilities.

Now, we extend the formulation to a general L-class problem.

Definition 1 *Consider a L-class decision problem with costs $c_{ij}, 0 \le i, j < L$ and $c_{jj} \le c_{ij}$, and let $R_{\mathbf{w}}(\mathbf{P})$ be the risk of a decision machine with parameter vector $\mathbf{w}$ when prior class probabilities are given by $\mathbf{P} = (P_0, \ldots, P_{L-1})$. The deviation function is defined as*

$$D_{\mathbf{w}}(\mathbf{P}) = R_{\mathbf{w}}(\mathbf{P}) - R_B(\mathbf{P})$$

and the minimax deviation is defined as

$$D_{mMd} = \inf_{\mathbf{w}} \max_{\mathbf{P}} \{D_{\mathbf{w}}(\mathbf{P})\} \quad . \tag{4}$$

Note that the above definition assumes that the maximum exists. This is actually the case, since $D_{\mathbf{w}}(\mathbf{P})$ is a linear function over a compact set, $\mathcal{P}$. Note, also, that our definition includes the natural assumption that c_{jj} is never higher than c_{ij}, meaning that making a decision error is always less costly than taking the correct decision. This assumption is used in part of our theoretical analysis.

The algorithms proposed in this paper are based on the fact that the minimax deviation can be computed without knowing R_B

Theorem 2 *The minimax deviation is given by*

$$D_{mMd} = \inf_{\mathbf{w}} \max_{\mathbf{P}} \{\overline{D}_{\mathbf{w}}(\mathbf{P})\} \ ,$$

where

$$\overline{D}_{\mathbf{w}}(\mathbf{P}) = R_{\mathbf{w}}(\mathbf{P}) - R_{basis}(\mathbf{P}) \tag{5}$$

and

$$R_{basis}(\mathbf{P}) = \sum_{j=0}^{L-1} c_{jj} P_j \ \ . \tag{6}$$

Proof Note that, according to Eqs. (1) and (2), for any decision machine and any $\mathbf{u}_i \in \mathcal{U}_L$,

$$R(\mathbf{u}_j) = R_j = \sum_{i=0}^{L-1} c_{ij} P\{\widehat{\mathbf{d}} = \mathbf{u}_i | \mathbf{d} = \mathbf{u}_j\} \geq c_{jj} \ \ .$$

Since the bound is reached by the classifier deciding $\widehat{\mathbf{d}} = \mathbf{u}_j$ for any observation $\mathbf{x}$, we have $R_B(\mathbf{u}_j) = c_{jj}$. Therefore, using Eq. (6), we find that, for any $\mathbf{u} \in \mathcal{U}_L$,

$$R_B(\mathbf{u}) = R_{basis}(\mathbf{u})$$

and, thus,

$$D_{\mathbf{w}}(\mathbf{u}) = \overline{D}_{\mathbf{w}}(\mathbf{u}) \ \ .$$

Since Bayes minimum risk $R_B(\mathbf{P})$ is a convex function of priors and $R_{\mathbf{w}}(\mathbf{P})$ is linear, $D_{\mathbf{w}}(\mathbf{P})$ is concave and, thus, it is maximum at some of the vertices in $\mathcal{P}$ (i.e., at some $\mathbf{P} = \mathbf{u} \in \mathcal{U}_L$). Thus,

$$\max_{\mathbf{P}} \{D_{\mathbf{w}}(\mathbf{P})\} = \max_{\mathbf{u} \in \mathcal{U}_L} \{D_{\mathbf{w}}(\mathbf{u})\} \ \ . \tag{7}$$

Since the maximum difference between two hyperplanes defined over $\mathcal{P}$ is always at some vertex, we can conclude that

$$\max_{\mathbf{P}} \{\overline{D}_{\mathbf{w}}(\mathbf{P})\} = \max_{\mathbf{u} \in \mathcal{U}_L} \{\overline{D}_{\mathbf{w}}(\mathbf{u})\} = \max_{\mathbf{u} \in \mathcal{U}_L} \{D_{\mathbf{w}}(\mathbf{u})\} \ \ . \tag{8}$$

Combining Eqs. (4), (7) and (8), we get

$$D_{mMd} = \inf_{\mathbf{w}} \max_{\mathbf{P}} \{\overline{D}_{\mathbf{w}}(\mathbf{P})\} \ \ .$$

■

Note that R_{basis} represents the risk baseline of the ideal classifier with zero errors. Th. 2 shows that the minimax regret can be computed as the minimax deviation to this ideal classifier. Note, also, that if costs c_{ii} do not depend on i, Eq. (5) becomes equivalent (up to a constant) to the Bayes risk and the minimax regret classifier becomes equivalent to the minimax classifier .

Another important result for the algorithms proposed in this paper is that, under some conditions on the minimum risk, the minimum and maximum operators can be permuted. Although general results on the permutability of minimum and maximum operators can be found in the literature (see Polak, 1997, for instance), we provide here the proof for the specific case interesting to this paper.

Theorem 3 *Consider the minimum deviation function given by*

$$\overline{D}_{\min}(\mathbf{P}) = \inf_{\mathbf{w}}\{\overline{D}_{\mathbf{w}}(\mathbf{P})\} \quad , \tag{9}$$

where $\overline{D}_{\mathbf{w}}(\mathbf{P})$ is the normalized deviation function given by Eq. (5), and let $\mathbf{P}^*$ be the prior probability vector providing the maximum deviation,

$$\mathbf{P}^* = \arg\max_{\mathbf{P}}\left\{\overline{D}_{\min}(\mathbf{P})\right\} \quad . \tag{10}$$

If $\overline{D}_{\min}(\mathbf{P})$ is continuously differentiable at $\mathbf{P} = \mathbf{P}^*$, then the minimax deviation, D_{mMd}, defined by Eq. (4), is

$$D_{mMd} = \overline{D}_{\min}(\mathbf{P}^*) = \max_{\mathbf{P}}\inf_{\mathbf{w}}\left\{\overline{D}_{\mathbf{w}}(\mathbf{P})\right\} \quad . \tag{11}$$

Proof

For any classifier with parameter vector $\mathbf{w}$, we can write,

$$\max_{\mathbf{P}}\overline{D}_{\mathbf{w}}(\mathbf{P}) \geq \overline{D}_{\mathbf{w}}(\mathbf{P}^*) \geq \overline{D}_{\min}(\mathbf{P}^*)$$

and, thus,

$$\inf_{\mathbf{w}}\max_{\mathbf{P}}\overline{D}_{\mathbf{w}}(\mathbf{P}) \geq \overline{D}_{\min}(\mathbf{P}^*) \quad . \tag{12}$$

Therefore, $\overline{D}_{\min}(\mathbf{P}^*)$ is a lower bound of the minimax regret.

Now we prove that $\overline{D}_{\min}(\mathbf{P}^*)$ is also an upper bound. According to Eq. (9), for any $\varepsilon > 0$, there exists a parameter vector $\mathbf{w}_\varepsilon$ such that

$$\overline{D}_{\mathbf{w}_\varepsilon}(\mathbf{P}^*) \leq \overline{D}_{\min}(\mathbf{P}^*) + \varepsilon \quad . \tag{13}$$

By definition, for any $\mathbf{P}$, $\overline{D}_{\min}(\mathbf{P}) \leq \overline{D}_{\mathbf{w}_\varepsilon}(\mathbf{P})$. Therefore, using Eq. (13), we can write

$$\overline{D}_{\mathbf{w}_\varepsilon}(\mathbf{P}^*) - \overline{D}_{\mathbf{w}_\varepsilon}(\mathbf{P}) \leq \overline{D}_{\min}(\mathbf{P}^*) - \overline{D}_{\min}(\mathbf{P}) + \varepsilon \quad . \tag{14}$$

Since $\overline{D}_{\min}(\mathbf{P})$ is continuously differentiable and (according to Eq. (10)) maximum at $\mathbf{P}^*$, for any $\varepsilon' > 0$ there exists $\delta > 0$ such that, for any $\mathbf{P} \in \mathcal{P}$ with $\|\mathbf{P}^* - \mathbf{P}\| \leq \delta$ we have

$$\overline{D}_{\min}(\mathbf{P}^*) - \overline{D}_{\min}(\mathbf{P}) \leq \varepsilon'\|\mathbf{P}^* - \mathbf{P}\| \leq \varepsilon'\delta \quad . \tag{15}$$

Let $\mathbf{P}_\delta$ a prior such that $\|\mathbf{P}^* - \mathbf{P}_\delta\| = \delta$. Taking $\varepsilon = \varepsilon'\delta$ and combining Eqs. (14) and (15) we can write

$$\overline{D}_{\mathbf{w}_\varepsilon}(\mathbf{P}^*) - \overline{D}_{\mathbf{w}_\varepsilon}(\mathbf{P}_\delta) \leq 2\varepsilon'\delta \quad .$$

Since the above condition is verified for any $\varepsilon' > 0$ and any prior $\mathbf{P}_\delta$ at distance δ from $\mathbf{P}$, and taking into account that $\overline{D}_{\mathbf{w}_\varepsilon}(\mathbf{P})$ is a linear function of $\mathbf{P}$, we conclude that the maximum slope of $\overline{D}_{\mathbf{w}_\varepsilon}(\mathbf{P})$ is bounded by $2\varepsilon'$ and, thus, for any $\mathbf{P} \in \mathcal{P}$, we have

$$\overline{D}_{\mathbf{w}_\varepsilon}(\mathbf{P}) - \overline{D}_{\mathbf{w}_\varepsilon}(\mathbf{P}^*) \leq 2\varepsilon'\|\mathbf{P} - \mathbf{P}^*\| \leq 2\sqrt{2}\varepsilon' \quad ,$$

(where we have used the fact that the maximum distance between two probability vectors is $\sqrt{2}$). Therefore, we can write

$$\max_{\mathbf{P}}\overline{D}_{\mathbf{w}_\varepsilon}(\mathbf{P}) \leq \overline{D}_{\mathbf{w}_\varepsilon}(\mathbf{P}^*) + 2\sqrt{2}\varepsilon'$$

and, thus,

$$\inf_{\mathbf{w}} \max_{\mathbf{P}} \overline{D}_{\mathbf{w}}(\mathbf{P}) \leq \overline{D}_{\mathbf{w}_\varepsilon}(\mathbf{P}^*) + 2\sqrt{2}\varepsilon' \ .$$

Finally, using Eq. (13) and taking into account that $\varepsilon = \varepsilon'\delta \leq \sqrt{2}\varepsilon'$ we get

$$\inf_{\mathbf{w}} \max_{\mathbf{P}} \overline{D}_{\mathbf{w}}(\mathbf{P}) \leq \overline{D}_{\min}(\mathbf{P}^*) + 3\sqrt{2}\varepsilon' \ . \tag{16}$$

Since the above is true for any $\varepsilon' > 0$ we conclude that $\overline{D}_{\min}(\mathbf{P}^*)$ is also an upper bound of $\overline{D}_{\mathbf{w}}$. Therefore, combining Eqs. (12) and (16), we conclude that

$$\inf_{\mathbf{w}} \max_{\mathbf{P}} \overline{D}_{\mathbf{w}}(\mathbf{P}) = \overline{D}_{\min}(\mathbf{P}^*) \ ,$$

which completes the proof. ■

Note that the deviation function needs to be neither differentiable nor a continuous function of **w** parameters.

If the minimum deviation function is not continuously differentiable at the minimax deviation probability, $\mathbf{P}^*$, the theorem cannot be applied. The reason is that, although there should exist at least one classifier providing the minimum deviation at $\mathbf{P} = \mathbf{P}^*$, it or they could not provide a constant deviation with respect to the prior probability. The situation can be illustrated with an example.

Let $x \in \mathbb{R}$ be given by $p(x|d=0) = 0.8N(x,\sigma) + 0.2N(x-2,\sigma)$ and $p(x|d=1) = 0.2N(x-1,\sigma) + 0.8N(x-3,\sigma)$, where $\sigma = 0.5$ and $N(x,\sigma) = (2\pi\sigma)^{-1/2}\exp(-x^2/(2\sigma^2))$, and consider the set Φ_λ of classifiers given by a single threshold over x and decision

$$\hat{d} = \begin{cases} 1 & \text{if } x \geq \lambda \\ 0 & \text{if } x < \lambda. \end{cases}$$

Fig. 2 shows the distribution of both classes over x, and Fig. 3 shows, as a function of priors, the minimum error probability (continuous line) that can be obtained using classifiers in Φ_λ. Note that decision costs $c_{00} = c_{11} = 0$ and $c_{01} = c_{10} = 1$ have been considered for this illustrative problem. An abrupt slope change is observed at the minimax deviation probability, for $P\{d=1\} = 1/2$. For this prior, there are two single threshold classifiers providing the minimum error probability, which are given by thresholds λ_1 and λ_2 in Fig. 2. However, as shown in Fig. 3 neither of them provides a risk that is constant in the prior. The minimax deviation classifier in Φ_λ, which has a threshold λ_0, does not attain minimum risk at the minimax deviation probability and, thus, cannot be obtained by using Eq. (11).

For this example, the desired robust classifier should have a deviation function given by the horizontal dotted line in Fig. 3. Fortunately, it can be obtained by combining the outputs of several classifiers. For instance, let $\hat{d_1}$ and $\hat{d_2}$ the decisions of classifiers given by thresholds λ_1 and λ_2, respectively. It is not difficult to see that the classifier selecting $\hat{d_1}$ and $\hat{d_2}$ at random (for each input sample x) provides a robust classifier.

This procedure can be extended to the multiclass-case: consider a set of L classifiers with parameters $\mathbf{w}_k$, $k = 0,\ldots,L-1$, and consider the classifier such that, for any input sample x, makes a decision equal to $\widehat{d_k}$ (i.e., the decision of classifier with parameters $\mathbf{w}_k$), with probability q_k. It is not difficult to show that the deviation function of this classifier is given by

$$\overline{D}(\mathbf{P}) = \sum_{j=0}^{L-1} P_j \left(\sum_{k=0}^{L-1} q_k \overline{D}_j(\mathbf{w}_k) \right) \ ,$$

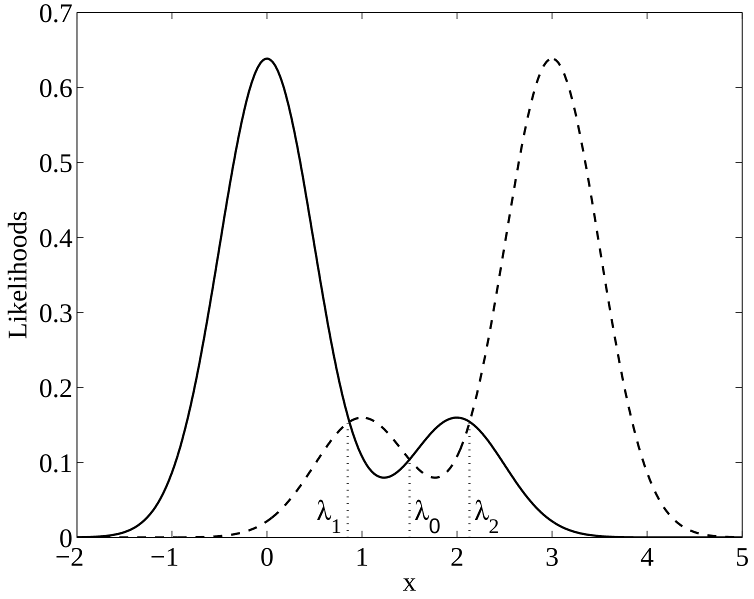

Figure 2: The conditional data distributions for the one-dimensional example discussed in the text. λ_1 and λ_2 are the thresholds providing the minimum risk at the minimax deviation probability. λ_0 provides the minimax deviation classifier.

where $\overline{D}_j(\mathbf{w}_k) = R_j(\mathbf{w}_k) - c_{jj}$. In order to get a constant deviation function, probabilities q_k should be chosen in such a way that

$$\sum_{k=0}^{L-1} q_k \overline{D}_j(\mathbf{w}_k) = D \ ,$$

where D is a constant. Solving these linear equations for q_k, $k = 0, \ldots, L-1$ (with the constraint $\sum_k q_k = 1$), the required probabilities can be found.

Note that, in order to build the non-deterministic classifier providing a constant deviation, a set of L independent classifiers that are optimal at the minimax deviation prior should be found. However, we go no further on the investigation of this special case for two main reasons:

- The situation does not seem to be common in practice. In our simulations, we have found that the maximum of the minimum risk deviation always provided a response which is approximately parallel to R_{basis}.
- In general, the abrupt change in the derivative may be a symptom that the classifier structure is not optimal for the data distribution. Instead of building a nondeterministic classifier, increasing the classifier complexity should be more efficient.

Although the least favorable prior providing the minimax deviation can be computed in closed form for some simple distributions, in general, it must be computed numerically. Moreover, we assume here that the data distribution is not known, and must be learned from examples. Thus,

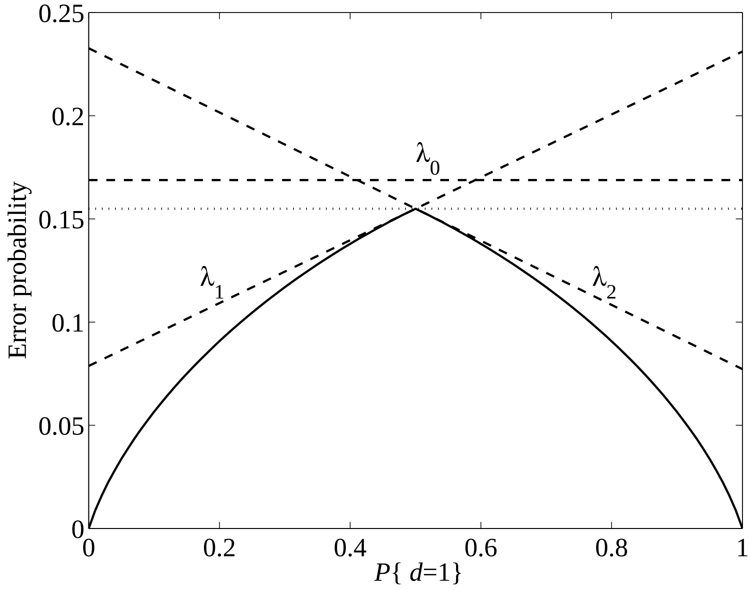

Figure 3: Error probabilities as a function of prior probability of class 1 for the example in Fig. 2. Thresholds λ_1 and λ_2 do not provide the minimax deviation classifier, which is obtained for threshold λ_0. However, the random combination of classifiers with thresholds λ_1 and λ_2 (dotted line) provides a robust classifier with deviation lower than that of λ_0.

we must incorporate the estimation of the least favorable prior in the learning process. Next, we propose a training algorithm in order to get a *minimax regret* classifier based on neural networks.

4. Neural Robust Classifiers Under Complete Uncertainty

Note that, if $\mathbf{Q}_{mMd}$ is the probability vector providing the maximum in Eq. (11), that is,

$$\mathbf{Q}_{mMd} = \arg\max_{\mathbf{P}} \left\{ \inf_{\mathbf{w}} \{\overline{D}_{\mathbf{w}}(\mathbf{P})\} \right\} \quad ,$$

then we can write

$$D_{mMd} = \inf_{\mathbf{w}} \{\overline{D}_{\mathbf{w}}(\mathbf{Q}_{mMd})\} \quad .$$

Therefore, the *minimax deviation* classifier can be estimated by training a classifier using prior in $\mathbf{Q}_{mMd}$. For this reason, $\mathbf{Q}_{mMd}$ will be called the *minimax deviation* prior (or least favorable prior). Our proposed algorithms are based on an iterative process of estimating parameters $\mathbf{w}$ based on an estimate of the minimax deviation prior, and re-estimating prior based on an estimate of network weights. This is shown in the following.

4.1 Updating Network Weights

Learning is based on minimizing some empirical estimate of the overall error function

$$E\{C(\mathbf{y},\mathbf{d})\} = \sum_{i=0}^{L-1} P\{\mathbf{d}=\mathbf{u}_i\} E\{C(\mathbf{y},\mathbf{d})|\mathbf{d}=\mathbf{u}_i\} = \sum_{i=0}^{L-1} P_i C_i \ ,$$

where $C(\mathbf{y},\mathbf{d})$ may be any error function and C_i is the expected conditional error for class-i.

Selecting the appropriate error function (see Cid-Sueiro and Figueiras-Vidal, 2001, for instance), learning rules can be designed providing *a posteriori* probability estimates ($y_i \approx P\{\mathbf{d} = \mathbf{u}_i|\mathbf{x}\}$, where y_i is the soft decision) and, thus, according to Eq. (3), the hard decision minimizing the risk can be approximated by

$$\widehat{\mathbf{d}} = \arg\min_i \{ \sum_{j=0}^{L-1} c_{ij} y_j \} \ .$$

The overall empirical error function (cost function) used in learning for priors $\widehat{\mathbf{P}} = (\widehat{P}_0, \ldots, \widehat{P}_{L-1})$ may be written as

$$\begin{aligned} \widehat{C} &= \sum_{i=0}^{L-1} \widehat{P}_i \widehat{C}_i = \sum_{i=0}^{L-1} \widehat{P}_i \frac{1}{K_i} \sum_{k=1}^{K} d_i^k \widehat{C}(\mathbf{y}^k,\mathbf{d}^k), \\ &= \frac{1}{K}\left[\sum_{i=0}^{L-1}\left(\frac{\widehat{P}_i}{K_i/K}\sum_{k=1}^{K} d_i^k C(\mathbf{y}^k,\mathbf{d}^k)\right)\right], \\ &= \frac{1}{K}\sum_{k=1}^{K}\left[\sum_{i=0}^{L-1}\frac{\widehat{P}_i}{\widehat{P}_i^{(0)}} d_i^k \widehat{C}(\mathbf{y}^k,\mathbf{d}^k)\right] \ , \end{aligned} \tag{17}$$

where $\widehat{P}_i^{(0)} = K_i/K$ is an initial estimate of class-i prior based on class frequencies in the training set and $\widehat{P}_i$ is the current prior estimate.

Minimizing error function (17) by means of a stochastic gradient descent learning rule leads to update the network weights at k-th iteration as

$$\begin{aligned} \mathbf{w}^{(k+1)} &= \mathbf{w}^{(k)} - \mu\Big(\sum_{i=0}^{L-1}\frac{\widehat{P}_i^{(n)}}{\widehat{P}_i^{(0)}} d_i^k \nabla_{\mathbf{w}} C(\mathbf{y}^k,\mathbf{d}^k)\Big), \\ &= \mathbf{w}^{(k)} - \Big(\sum_{i=0}^{L-1}\mu_i^{(n)} d_i^k\Big)\nabla_{\mathbf{w}} C(\mathbf{y}^k,\mathbf{d}^k) \ , \end{aligned} \tag{18}$$

where

$$\mu_i^{(n)} = \mu\frac{\widehat{P}_i^{(n)}}{\widehat{P}_i^{(0)}} \tag{19}$$

is a learning step scaled by the prior ratio. Note that d_i selects the appropriate $\mu_i^{(n)}$ according to the pattern class membership. The classifier is trained without altering the original training data set class distribution $\widehat{P}_i^{(0)}$ and therefore, without missing or duplicating information.

4.2 Updating Prior Probabilities

Eq. (11) shows that the learning process should maximize (5) with respect to the prior probabilities. The estimate of (5) can be computed as

$$\widehat{\widehat{D}}_{\mathbf{w}}(\mathbf{P}) = \widehat{R}_{\mathbf{w}}(\mathbf{P}) - R_{basis}(\mathbf{P}) \ , \tag{20}$$

where

$$\widehat{R}_{\mathbf{w}}(\mathbf{P}) = \sum_{j=0}^{L-1} \widehat{R}_j P_j \tag{21}$$

is the overall Bayes risk estimate and

$$\widehat{R}_j = \frac{1}{N_j} \sum_{i=0}^{L-1} c_{ij} N_{ij} \tag{22}$$

is the class-j conditional risk estimate where N_j is the number of class $\mathbf{u}_j$ patterns in the training phase and N_{ij} is the number of samples from class $\mathbf{u}_j$ assigned to $\mathbf{u}_i$.

In order to derive a learning rule to find an estimate $\widehat{P}_i$ satisfying constraints $\sum_{i=0}^{L-1} \widehat{P}_i = 1$ and $0 \leq \widehat{P}_i \leq 1$, we will use auxiliary variables B_i such that

$$\widehat{P}_i = \frac{\exp(B_i)}{\sum_{j=0}^{L-1} \exp(B_j)} \ . \tag{23}$$

We maximize $\widehat{\widehat{D}}_{\mathbf{w}}$ with respect to B_i. Applying the chain rule,

$$\frac{\partial \widehat{\widehat{D}}_{\mathbf{w}}}{\partial B_i} = \sum_{j=0}^{L-1} \frac{\partial \widehat{\widehat{D}}_{\mathbf{w}}}{\partial \widehat{P}_j} \frac{\partial \widehat{P}_j}{\partial B_i} \ ,$$

and using Eqs. (20), (21) and (23), we get

$$\begin{aligned}
\frac{\partial \widehat{\widehat{D}}_{\mathbf{w}}}{\partial B_i} &= \sum_{j=0}^{L-1} (\widehat{R}_j - c_{jj}) \widehat{P}_i (\delta_{ij} - \widehat{P}_j), \\
&= \widehat{P}_i \left(\widehat{R}_i - c_{ii} - \sum_{j=0}^{L-1} (\widehat{R}_j \widehat{P}_j) + \sum_{j=0}^{L-1} (c_{jj} \widehat{P}_j) \right), \\
&= \widehat{P}_i \left(\left(\widehat{R}_i - c_{ii} \right) - \left(\widehat{R}_{\mathbf{w}} - \widehat{R}_{basis} \right) \right), \\
&= \widehat{P}_i \widehat{R}_{di} \ ,
\end{aligned}$$

where

$$\widehat{R}_{di} = (\widehat{R}_i - c_{ii}) - (\widehat{R}_{\mathbf{w}} - \widehat{R}_{basis}) \ .$$

The learning rule for auxiliary variable B_i is

$$\begin{aligned}
B_i{}^{(n+1)} &= B_i^{(n)} + \rho \frac{\partial \widehat{D}_{\mathbf{w}}}{\partial B_i}, \\
&= B_i^{(n)} + \rho \widehat{P}_i^{(n)} \widehat{R}_{di}^{(n)} \ ,
\end{aligned} \tag{24}$$

where parameter $\rho > 0$ controls the rate of convergence. Using Eq. (23) and Eq. (24), the updated learning rule for $\widehat{P}_i$ is

$$\begin{aligned}\widehat{P}_i^{(n+1)} &= \frac{\exp(B_i^{(n)})\exp\left(\rho\widehat{P}_i^{(n)}\widehat{R}_{di}^{(n)}\right)}{\sum_{j=0}^{L-1}\left[\exp\left(B_j^{(n)}\right)\exp\left(\rho\widehat{P}_j^{(n)}\widehat{R}_{dj}^{(n)}\right)\right]}, \\ &= \frac{\widehat{P}_i^{(n)}\exp\left(\rho\widehat{P}_i^{(n)}\widehat{R}_{di}^{(n)}\right)}{\sum_{j=0}^{L-1}\left[\widehat{P}_j^{(n)}\exp\left(\rho\widehat{P}_j^{(n)}\widehat{R}_{dj}^{(n)}\right)\right]} \quad . \end{aligned} \tag{25}$$

4.3 Training Algorithm for a Minimax Deviation Classifier

In the previous section, both the network weights updating rule (18) and the prior probability update rule (25) have been derived. The algorithm resulting from the combination is shown as follows:

```
for $n = 0$ to $N_{iterations} - 1$ do
  for $k = 1$ to $K$ do
    $\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \left(\sum_{i=0}^{L-1}\mu_i^{(n)}d_i^k\right)\nabla_{\mathbf{w}}C(\mathbf{y}^k,\mathbf{d}^k)$
  end for
  Estimate $\widehat{R}^{(n)}$, $\widehat{R}_i^{(n)}$, $i = 0,\dots,L-1$, according to (21) and (22)
  Update minimax probability $\widehat{P}_i^{(n+1)}$, $i = 0,\dots,L-1$ according to (25) and compute $\mu_i^{(n+1)}$ with (19)
end for
```

5. Robust Classifiers Under Partial Uncertainty

Although in many practical situations prior probabilities may not be specified with precision, they can be partially known. In this section we discuss how partial information about priors can be used to improve the classifier performance in relation to a complete uncertainty situation.

From now on, let us consider that lower (or upper) bounds of the priors are known based on previous experience. We will denote the lower and upper bounds of class-i prior probability as P_{il} and P_{iu}, respectively.

In order to illustrate this situation consider a binary classification problem where probability lower bounds P_{0l} and P_{1l} are known. That is, $P_1 \in [P_{1l}, 1 - P_{0l}]$ where this interval represents the uncertainty region. Let us denote by $\Gamma = \{\mathbf{P} : 0 \leq P_i \leq 1, \sum_{i=0}^{L-1} P_i = 1, P_i \geq P_{il}\}$ the probability region satisfying the imposed constraints. In the following, we will refer to Γ as the *uncertainty region*.

Now, the aim is to design a classifier that minimizes the maximum regret from the minimum risk only inside the uncertainty region. This is depicted in Fig. 4(a), which shows that reducing the uncertainty in priors allows to reduce deviation from the optimal classifier. This minimax regret approach for the uncertainty region Γ is often called Γ-minimax regret. As discussed before, the minimax deviation solution gives a Bayes solution with respect some priors denoted in the partial uncertainty case as $\mathbf{Q}_{mMd}^{\Gamma}$ in Fig. 4(a), which is the least favorable distribution according to the regret criterion.

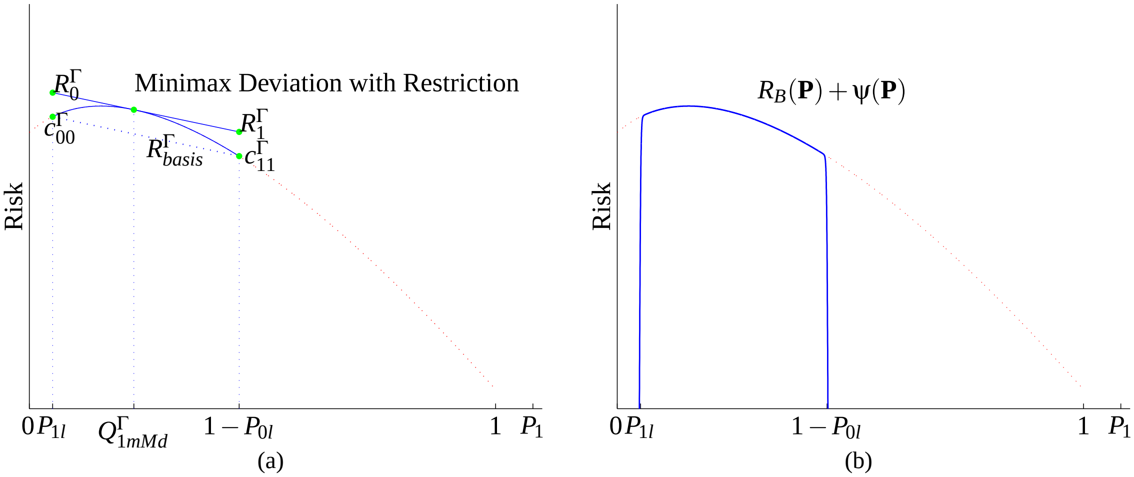

Figure 4: Minimax deviation classifier under partial uncertainty of prior probabilities: (a)Γ-*minMaxDev* Classifier. (b) Modified cost function defined as $R_B(\mathbf{P})+\psi(\mathbf{P})$.

In contrast to the minimax regret criterion, note that a classical minimax classifier for the considered uncertainty region would minimize the worst-case risk. It would be a Bayes solution for the prior where the minimum risk reaches its maximum and it could be denoted as $\mathbf{Q}_{mM}^\Gamma$.

Notice, also, that these solutions will be the same if the risk for the vertex of Γ take the same value ($c_{ii}^\Gamma = k$).

5.1 Neural Robust Classifiers Under Partial Uncertainty

Minimax search can be formulated as maximizing (with respect to priors) the minimum (with respect to network parameters) of deviation function (5), as described in previous section, but subject to some constraints

$$\begin{array}{ll} \arg\max_{\mathbf{P}} \inf_{\mathbf{w}} & \{D_{\mathbf{w}}^\Gamma(\mathbf{P})\}\ , \\ s.t. & P_i \geq P_{il},\ i=0,\ldots,L-1 \end{array}$$

where $D_{\mathbf{w}}^\Gamma = R_{\mathbf{w}}^\Gamma - R_{basis}^\Gamma$. When uncertainty is global, this hyperplane is defined by the risk in the L extreme cases with $P_i = \delta_{ik}$, that is, by the corresponding c_{ii}. However, with partial knowledge of the prior probabilities, this hyperplane becomes defined by the risk in L points which are the vertex given by the restrictions and with associated risk denoted by c_{jj}^Γ.

Defining

$$l(P_i) = \frac{1}{1+\exp^{-\tau(P_i-P_{il})}}\ , \tag{26}$$

where τ controls the hardness of this restriction, the minimax problem can be re-formulated as

$$\begin{array}{ll} \arg\max_{\mathbf{P}} \inf_{\mathbf{w}} & \{D_{\mathbf{w}}^\Gamma(\mathbf{P})\} \\ s.t. & l(P_i) \geq 1/2,\ i=0,\ldots,L-1. \end{array}$$

Thus, this constrained optimization problem can be solved as a non-constrained problem by considering an auxiliary function that incorporates the restriction as a barrier function

$$\arg\max_{\mathbf{P}}\ \inf_{\mathbf{w}}\ \{D^{\Gamma}_{\mathbf{w}}(\mathbf{P})+A\psi(\mathbf{P})\}\ ,$$

where $\psi(P_i)=\log(l(P_i))$ and the constant A determines the contribution of the barrier function.

Fig. 4(b) shows the new risk function corresponding to the binary case previously depicted in Fig. 4(a). Note that, it is the sum of the original $R_B(\mathbf{P})$ and the barrier function $\psi(\mathbf{P})$.

As in Section 4.1, in order to derive the network weight learning rule, we need to compute

$$\begin{aligned}\frac{\partial\widehat{\psi}}{\partial B_i} &= \sum_{j=0}^{L-1}\frac{\partial\widehat{\psi}}{\partial\widehat{P}_j}\frac{\partial\widehat{P}_j}{\partial B_i},\\ &= \tau\widehat{P}_i\sum_{k=0}^{L-1}\left(1-l(\widehat{P}_k)\right)(\delta_{ik}-\widehat{P}_k),\\ &= \tau\widehat{P}_i\widehat{\psi}_{di}\ ,\end{aligned}$$

where $\widehat{\psi}_{di}=\sum_{k=0}^{L-1}(1-l(\widehat{P}_k))(\delta_{ik}-\widehat{P}_k)$

As τ increases, the constraints become harder around the specified bound.

The update learning rule for the auxiliary variable B_i at cycle n is

$$B_i^{(n+1)} = B_i{}^{(n)}+\rho\widehat{P}_i^{(n)}\widehat{R}_{di}^{\Gamma(n)}+\rho A\tau\widehat{P}_i^{(n)}\widehat{\psi}_{di}^{(n)}\ .$$

And therefore, using (23), the update learning rule for P_i is

$$\widehat{P}_i^{(n+1)} = \frac{\widehat{P}_i^{(n)}\exp\left(\rho\widehat{P}_i^{(n)}\widehat{R}_{di}^{\Gamma(n)}\right)\exp\left(\rho A\tau\widehat{P}_i^{(n)}\widehat{\psi}_{di}^{(n)}\right)}{\sum_{j=0}^{L-1}\left\{\widehat{P}_j^{(n)}\exp\left(\rho\widehat{P}_j^{(n)}\widehat{R}_{dj}^{\Gamma(n)}\right)\exp\left(\rho A\tau\widehat{P}_j^{(n)}\widehat{\psi}_{dj}^{(n)}\right)\right\}}\ .$$

Note that if the upper bound is known instead of the lower bound, $l(P_i)$ defined by (26) should be replaced by $u(P_i)=(1+\exp(\tau(P_i-P_{iu})))^{-1}$ at the previous formulation.

The minimax constrained optimization problem has been tackled by considering a new objective function defined by the sum of the original cost function and a barrier function. Studying the convexity of this new function becomes important from the fact that a stationary point of this risk curve is a global maximum.

Since the minimum risk curve ($R_B(P)$) is a convex function of the priors (see VanTrees, 1968, for details), if we verify the convexity of the barrier function, we can conclude that the function defined by the sum of both of them is also convex.

This barrier function is convex in $\mathcal{P}$ if the Hessian matrix $\mathbf{H}_R$ verifies $\mathbf{P}^{\mathbf{T}}\mathbf{H}_R\mathbf{P}\leq 0$

The Hessian matrix of the barrier function equals to a diagonal matrix $\mathbf{D}_{\mathbf{r}}=diag(\mathbf{r})$ with all negative diagonal entries $r_i=A\tau^2(-l(P_i)(1-l(P_i)))$. As $l(P_i)\in[0,1]$ and therefore, $r_i\leq 0$, it is straightforward to see that

$$\begin{aligned}\mathbf{P}^{\mathbf{T}}\mathbf{H}_R\mathbf{P} &= \mathbf{P}^{\mathbf{T}}\mathbf{D}_{\mathbf{r}}\mathbf{P},\\ &= \sum_{i=0}^{L-1}P_i^2 r_i\leq 0\ .\end{aligned}$$

Since the barrier function is convex, the new objective function (defined by the sum of two convex functions) is also convex.

5.2 Extension to Other Learning Algorithms

The learning algorithm proposed in this paper is intended to train a minimax deviation classifier based on neural networks with feedforward architecture. Actually, the learning algorithm we propose becomes a feasible solution for any learning process based on minimizing some empirical estimate of an overall cost (error) function.

However, it is also applicable to a general classifier provided it is trained (in an iterative process) for the estimated minimax deviation probabilities and the assumed decision costs. Specifically, in this paper, scaling the learning rate allows to simulate different class distributions and the hard decisions are made based on posterior probability estimates and decision costs. Furthermore, the neural learning phase carried out in one iteration can be re-used for the next one, what allows to reduce computational cost with respect to a complete optimization process on each iteration. Apart from the general approach of completely training a classifier on each iteration and in order to reduce its computational cost, specific solutions may be studied for different learning machines. Nonetheless, it seems not feasible to readily achieve this improvement for classifiers like SVMs, where support vectors for one solution may have nothing in common with the ones obtained in next iteration and thus, making necessary to re-train the classifier in each iteration.

Another possible solution for any classifier that provides a posteriori probabilities estimates or any score that can be converted into probabilities (for details on calibration methods see Wei et al., 1999; Zadrozny and Elkan, 2002; Niculescu-Mizil and Caruana, 2005) is outlined here. In this case, an iterative procedure able to estimate the minimax deviation probabilities and consequently to adjust (without re-training) the outputs of the classifier could be studied. The general idea for this approach is as follows: first, the new minimax deviation prior probabilities are estimated according to (25) and then, posterior probabilities provided by the model are adjusted as follows (see Saerens et al., 2002, for more details)

$$P^{(k)}\{\mathbf{d}=\mathbf{u}_i|\mathbf{x}\} = \frac{\frac{P_i^{(k)}}{P_i^{(k-1)}} P^{(k-1)}\{\mathbf{d}=\mathbf{u}_i|\mathbf{x}\}}{\sum_{j=0}^{L-1} \frac{P_j^{(k)}}{P_j^{(k-1)}} P^{(k-1)}\{\mathbf{d}=\mathbf{u}_j|\mathbf{x}\}} \ . \tag{27}$$

The algorithm's main structure is summarized as

```
for $k = 1$ to $K$ do
    Estimate $\widehat{R}^{(k)}$, $\widehat{R}_i^{(k)}$, $i = 0, \ldots, L-1$, according to (21), (22) and decision costs $c_{ij}$
    Update minimax probability $\widehat{P}_i^{(k+1)}$ according to (25)
    Adjust classifier outputs according to (27)
end for
```

The effectiveness of this method relies on the accuracy of the initial *a posteriori* probability estimates. Studying in depth this approach and comparing different minimax deviation classifiers (decision trees, SVMs, RBF networks, feedforward networks and committee machines) together with different probability calibration methods appears as a challenging issue to be explored in future work.

6. Experimental Results

In this section, we first present the neural network architecture used in the experiments and illustrate the proposed *minimax deviation* strategy on an artificial data set. Then, we apply it to several real-world classification problems. Moreover, a comparison with other proposals such as the traditional *minimax* and the common *re-balancing* approach is carried out.

6.1 Softmax-based Network

Although our algorithms can be applied to any classifier architecture, we have chosen a neural network based on the softmax non-linearity with soft decisions given by

$$y_i = \sum_{j=1}^{M_i} y_{ij} \quad ,$$

with

$$y_{ij} = \frac{\exp(\mathbf{w}_{ij}^T \mathbf{x} + w_{ij0})}{\sum_{k=0}^{L-1} \sum_{l=1}^{M_k} \exp(\mathbf{w}_{kl}^T \mathbf{x} + w_{kl0})} \quad ,$$

where L stands for the number of classes, M_j the number of softmax outputs used to compute y_j and $\mathbf{w}_{ij}$ are weight vectors. We will refer to this network as a *Generalized Softmax Perceptron*(GSP).[1] A simple network with $M_j = 2$ is used in the experiments.

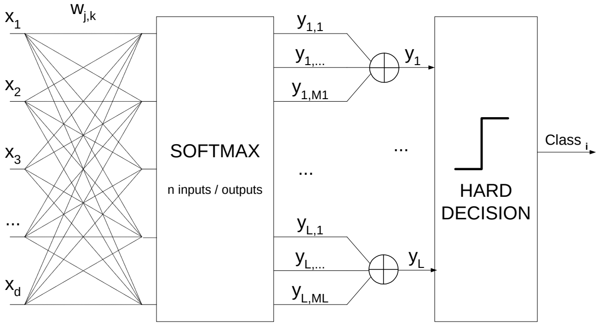

Figure 5: GSP(Generalized Softmax Perceptron) Network

Fig. 5 corresponds to the neural network architecture used to classify the samples represented by feature vector $\mathbf{x}$. Learning consists of estimating network parameters $\mathbf{w}$ by means of the stochastic gradient minimization of certain objective functions. In the experiments, we have considered the *Cross Entropy* objective function given by

$$CE(\mathbf{y}, \mathbf{d}) = -\sum_{i=1}^{L} d_i \log y_i \quad .$$

The stochastic gradient learning rule for the GSP network is given by Eq. (18). Learning step $\mu^{(k)}$ decreases according to $\mu^{(k)} = \frac{\mu^{(0)}}{1+k/\eta}$, where k is the iteration number, $\mu^{(0)}$ the initial learning rate and η a decay factor.

[1] Note that the GSP is similar to a two layer MLP with a single layer of weights and with coupled saturation function (softmax), instead of sigmoidal units.

The reason to illustrate this approach with a feedforward architecture is that, as mentioned in Section 5.2, it allows to exploit (in the iterative learning process) the partially optimized solution in current iteration for the next one. On the other hand, posterior probability estimation makes it possible to apply the adaptive strategy based on prior re-estimation proposed by Saerens to the minimax deviation classifier, as long as a data set representative of the operation conditions is available. Finally, the fact that intermediate outputs y_{ij} of the GSP can be interpreted as subclass probabilities may provide quite a natural way to cope with the unexplored problem of uncertainty in subclass distributions as already pointed out by Webb and Ting (2005). Nonetheless, both architecture and cost function issues are not the goal of this paper, but merely illustrative tools.

6.2 Artificial Data Set

To illustrate the *minimax regret* approach proposed in this paper both under complete and partial uncertainty, an artificial data set with two classes (class $\mathbf{u}_0$ and class $\mathbf{u}_1$) has been created. Data examples are drawn from the normal distribution $p(\mathbf{x}|\mathbf{d} = \mathbf{u}_i) = N(m_i, \sigma_i^2)$ with mean m_i and standard deviation σ_i. Mean values were set to $m_0 = 0$, $m_1 = 2$ and standard deviation to $\sigma_0 = \sigma_1 = \sqrt{2}$. A total of 4000 instances were generated with prior probabilities of class membership $P\{\mathbf{d} = \mathbf{u}_0\} = 0.93$ and $P\{\mathbf{d} = \mathbf{u}_1\} = 0.07$. The cost-benefit matrix $\begin{pmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{pmatrix}$ is given by $\begin{pmatrix} 2 & 5 \\ 4 & 0 \end{pmatrix}$.

Initial learning rate was set to $\mu^{(0)} = 0.3$, decay factor to $\eta = 2000$ and training was ended after 80 cycles. Classifier assessment was carried out by following 10-fold cross-validation.

Two classifiers were trained, to be called a *standard classifier* and a *minMaxDev classifier*. The former is built by considering that the estimated class prior information is precise and stationary and the latter is the approach proposed in this paper to cope with uncertainty in priors. Thus, for the *standard classifier*, its performance may deviate from the optimal risk in 3.39 when priors change from training to test conditions. However, a *minimax deviation classifier* reduces this worst-case difference from the optimal classifier to 0.77.

Now, we suppose that some information about priors is available (partial uncertainty). For instance, we consider that the lower bound for prior probabilities P_0 and P_1 are known and set to $P_{0l} = 0.55$ and $P_{1l} = 0.05$, respectively, so that the uncertainty region is $\Gamma = \{(P_0, P_1) | P_0 \in [0.55, 0.95], P_1 \in [0.05, 0.45]\}$.

A minimax deviation classifier can be derived for Γ (it will be called Γ-*minMaxDev classifier*).The narrower Γ is, the closer the minimax deviation classifier performance is to the optimal. For this particular case, under partially imprecise priors, the *standard classifier* may differ from optimal (in Γ) in 0.83, while the use of the simple *minMaxDev classifier* designed under total prior uncertainty conditions attains a maximum deviation of 0.53. However, the Γ-*minMaxDev classifier* only differs from optimal in 0.24. These data are reported in Table 1 where both, experimental and also theoretical results, are shown.

6.3 Real Databases

In this section we report experimental results obtained with several publicly available data sets. From the UCI repository (Blake and Merz, 1998) the following benchmarks: German Credits, Australian Credits, Insurance Company, DNA slice-junction, Page-blocks, Dermatology and Pen-digits.

	Classifier		
	Standard	minMaxDev	Γ-minMaxDev
	Th/Exp	Th/Exp	Th/Exp
Maximum deviation from optimal (complete uncertainty)	3.41/*3.39*	**0.72**/***0.77***	–
Maximum deviation from optimal in Γ (partial uncertainty)	0.85/*0.83*	0.50/*0.53*	**0.19**/***0.24***

Table 1: A comparison between the *standard* classifier (build under stationary prior assumptions), the minimax deviation classifier (*minMaxDev*) and the minimax deviation classifier under partial uncertainty (Γ-*minMaxDev*) for an artificial data set

Database	**# Classes**	**Class distribution**	**# Attributes**	**# Instances**
German Credits (**GCRE**)	2	[0.70 0.30]	8	1000
Australian Credits (**AUS**)	2	[0.32 0.68]	14	690
Munich Credits (**MCRE**)	2	[0.30 0.70]	20	1000
Insurance Company (**COIL**)	2	[0.94 0.06]	85	9822
DNA Slice-junction (**DNA**)	3	[0.24 0.24 0.52]	180	3186
Page-blocks (**PAG**)	5	[0.90 0.06 0.01 0.01 0.02]	10	5473
Dermatology (**DER**)	6	[0.31 0.16 0.20 0.13 0.14 0.06]	34	366
Pen-digits (**PEN**)	10	[0.104 0.104 0.104 0.096 0.104 0.096 0.096 0.104 0.096 0.096]	16	10992

Table 2: Experimental Data sets

Other public data set used is Munich Credits from the Dept. of Statistics at the University of Munich.[2]

Data set description is summarized in Table 2, and cost-benefit matrices are shown in Table 3. We have used the cost values that appear in Ikizler (2002) for those data sets in common. Otherwise, for lack of an expert analyst, the cost values have been chosen by hand.

[2] Data sets available at *http://www.stat.uni-muenchen.de/service/datenarchiv/welcome_e.html*.

Insurance Company	German, Australian, Munich Credits	DNA
$\begin{pmatrix} 0 & 0 \\ 1 & -17 \end{pmatrix}$	$\begin{pmatrix} -1 & 5 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} -1 & 2 & 3 \\ 2 & -1 & 3 \\ 2 & 2 & 0 \end{pmatrix}$
Page-Blocks	Dermatology	Pendigits
$\begin{pmatrix} -1 & 1 & 1 & 1 & 1 \\ 2 & 0 & 1 & 1 & 1 \\ 2 & 1 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 1 \\ 2 & 1 & 1 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} -4 & 3 & 3 & 2 & 2 & 2 \\ 2 & -3 & 3 & 2 & 1 & 3 \\ 3 & 3 & -8 & 4 & 4 & 5 \\ 4 & 5 & 5 & -10 & 5 & 2 \\ 3 & 1 & 4 & 3 & -6 & 3 \\ 4 & 5 & 5 & 4 & 5 & -10 \end{pmatrix}$	$c_{ij} = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{Otherwise} \end{cases}$

Table 3: Cost-Benefit matrices for the experimental Data sets

	Maximum Risk Deviation from the optimal classifier			
	Standard	Re-balanced	Minimax Deviation *minMaxDev*	Minimax *minMax*
GCRE	0.70	0.80	(0.55 **0.60**)	0.99
ACRE	1.00	1.00	(0.76 **0.86**)	1.00
MCRE	0.91	0.77	(0.54 **0.59**)	0.99
COIL	2.78	0.99	(0.87 **0.92**)	16.32
DNA	0.34	0.53	(**0.30** 0.27 0.25)	1.14
PAG	0.62	0.26	(0.13 0.13 **0.20** 0.16 0.16)	0.86
DER	1.03	1.28	(0.67 **0.78** 0.51 0.48 0.54 0.60)	7.62
PEN	0.061	0.059	(0.024 0.025 0.023 0.026 0.023 0.028 0.019 0.021 0.022 **0.029**)	**0.029**

Table 4: Classifier Performance evaluated as Maximum Risk Deviation from the optimal classifier for several real-world applications. Class-conditional risk deviations ($R_i - c_{ii}$) reported for the *minMaxDev* classifier.

Experimental results for these data sets are shown in the following sections. The robustness of different decision machines under complete uncertainty of prior probabilities is analyzed in Section 6.3.1. If uncertainty is only partial, a similar study and comparison with the previous approach (complete uncertainty) is carried out in Section 6.3.2.

6.3.1 CLASSIFIER ROBUSTNESS UNDER COMPLETE UNCERTAINTY

We now study how different neural-based classifiers cope with worst-case situations in prior probabilities. The maximum deviation from the optimal classifier (see Table 4) is reported for the proposed *minMaxDev* strategy as well as for other alternative approaches: the one based on the assumption of stationary priors (*standard*) and the common alternative of deriving the classifier from an equally distributed data set (*re-balanced*). A comparison with the traditional minimax strategy is also provided. Together with the previously mentioned value (maximum deviation or regret), deviation for the L class-conditional extreme cases ($R_i - c_{ii}$) is also reported for the *minMaxDev* classifier in Table 4. Results allow to verify that this solution is fairly close to the optimal one where deviation is not dependent on priors and thus, class-conditional deviations take the same value.

Although the balanced class distribution to train the classifier can be obtained by means of undersampling and/or oversampling, it is simulated by altering the learning rate used in the training phase according to (19) as $\mu_i = \mu \dfrac{1/L}{\widehat{P}_i^{(0)}}$, where $1/L$ represents the simulated probability, equal for all classes.

Results evidence that the assumption of stationary priors may lead to significant deviations from the optimal decision rule under “unexpected”, but rather realistic, prior changes. This deviation may reach up to three times more than the robust minimax deviation strategy. Thus, for classification problems like Page-blocks the maximum deviation from the optimal classifier is **0.62** for the

	Maximum Risk			
	Standard	Re-balanced	Minimax Deviation	Minimax
			minMaxDev	*minMax*
GCRE	0.70	0.15	0.60	**0.00**
ACRE	0.01	0.02	0.86	**-0.00**
MCRE	0.05	0.20	0.59	**0.00**
COIL	0.76	0.99	0.86	**0.02**
DNA	0.34	0.53	0.25	**0.13**
PAG	0.62	0.26	0.20	**0.10**
DER	-2.10	-1.68	-2.21	**-2.38**
PEN	0.061	0.059	**0.029**	**0.029**

Table 5: Classifier Performance measured as Maximum Risk for several real-world applications.

standard classifier while this reduces to **0.20** for the *minMaxDev* one. Likewise, for the Insurance company(COIL) application the maximum deviation for the *standard classifier* is **2.78** compared with **0.92** for the *minMaxDev* model. The remaining databases also show the same behavior as it is presented in Table 4.

On the other hand, the use of a classifier inferred from a re-balanced data set does not necessarily involve a decrease in the maximum deviation with respect to the *standard classifier*. In the same way, the traditional minimax classifier does not protect against prior changes in terms of maximum relative deviation from the minimum risk classifier.

However, if our criterion is more conservative and our aim is the minimization of the maximum possible risk (not the minimization of the deviation), the traditional minimax classifier represents the best option. It is shown in Table 5 where the maximum risk for the different classifiers is reported. Positive values in this table indicate a cost while negative values represent a benefit. For instance, for the Page-blocks application the minimax classifier assures a maximum risk of **0.10** while the *standard*, *re-balanced* and *minMaxDev* classifiers reach values of 0.62, 0.26 and 0.20, respectively. It can be noticed that for the Pen-digits data set, the minimax deviation and minimax approaches attain the same results. The reason is that, for this problem, the R_{basis} plane takes the same value (in this case, zero) in the probability space.

6.3.2 CLASSIFIER ROBUSTNESS UNDER PARTIAL UNCERTAINTY

Unlike the previous section, we consider now that partial information about the class priors is available. The aim is to find a classifier that behaves well for a delimited and realistic range of priors what constitutes an aid in reducing the maximum deviation from the optimal classifier. This situation can be treated as a constrained minimax regret strategy where the constraints represent any extra information about prior probability value.

Experimental results for several situations of partial prior uncertainty are presented in this section. We consider that lower bounds for the prior probabilities are available (see Table 6). In order to get the Γ-*minMaxDev classifier*, the risk for the different vertex of the uncertainty domain needs to be calculated. With them, the basis risk R^{Γ}_{basis} over which deviations are measured is derived.

	Lower bound for prior probabilities									
Data Set	P_{0l}	P_{1l}	P_{2l}	P_{3l}	P_{4l}	P_{5l}	P_{6l}	P_{7l}	P_{8l}	P_{9l}
GCRE	0.40	0.25								
ACRE	0.20	0.25								
MCRE	0.20	0.25								
COIL	0.15	0.03								
DNA	0.10	0.10	0.25							
PAG	0.22	0.02	0.00	0.01	0.02					
DER	0.1	0.20	0.10	0.10	0.10	0.02				
PEN	0.10	0.06	0.06	0.10	0.10	0.06	0.06	0.10	0.05	0.05

Table 6: Lower bounds for prior probabilities defining the uncertainty region, Γ region for the experimental data sets.

	Maximum Risk Deviation in the uncertainty region		
	Standard	Minimax Deviation	Minimax Deviation with restriction
		minMaxDev	Γ-*minMaxDev*
GCRE	0.24	0.19	(**0.10** 0.09)
ACRE	**0.03**	0.64	(**0.03** **0.03**)
MCRE	0.22	0.38	(**0.13** 0.10)
COIL	2.33	0.77	(**0.17** 0.11)
DNA	0.14	0.08	(**0.07** **0.07** 0.06)
PAG	0.37	0.15	(**0.10** 0.08 0.08 0.05 0.04)
DER	0.08	**0.05**	(0.03 0.03 0.04 0.02 **0.05** **0.05**)
PEN	0.013	0.007	(**0.003** **0.003** 0.001 0.000 0.001 0.001 0.000 0.001 **0.003** 0.001)

Table 7: Classifier Performance under partial knowledge of prior probabilities measured as Maximum Risk Deviation for several real-world applications. Class-conditional risk deviations $(R_i^\Gamma - c_{ii}^\Gamma)$ are reported for the Γ-*minMaxDev* classifier.

Maximum deviation from the optimal in Γ is reported for the Γ-*minMaxDev* classifier together with the *standard* and the *minMaxDev* ones. For instance, the *standard classifier* for the Page-blocks data set deviates from the optimal classifier, in the defined uncertainty region, up to 0.37, while when complete uncertainty is assumed the maximum deviation is equal to 0.62.

In the same way, reducing the uncertainty also means a reduction in the maximum deviation for *minMaxDev* classifier (trained without considering this partial knowledge). Thus, for Γ, this classifier assures a deviation bound of 0.15. However, taking into account this partial information to train a Γ-*minMaxDev* classifier allows to reduce the deviation for the worst-case conditions to 0.10. It can be seen the same behavior for the other databases in Table 7.

7. Conclusions

This work concerns the design of robust neural-based classifiers when the prior probabilities of the classes are partially or completely unknown, even by the end user.

This problem of uncertainty in the class priors is often ignored in supervised classification, even though it is a widespread situation in real world applications. As a result, the reliability of the inducted classifier can be greatly affected as previously shown by the experiments.

To tackle this problem, we have proposed a novel *minimax deviation* strategy with the goal to minimize the maximum deviation with respect to the optimal classifier.

A neural network training algorithm based on learning rate scaling has been developed. The experimental results show that this minimax deviation (*minMaxDev*) classifier protects against prior changes while other approaches like ignoring this uncertainty or use a balanced learning data set may result in large differences in performance with respect to the minimum risk classifier. Also, it has been shown that the conventional minimax classifier reduces the maximum possible risk following a conservative attitude but at the expense of large worst-case differences from the optimal classifier.

Furthermore, a constrained minimax deviation approach (Γ-*minMaxDev*) has been derived for those situations where uncertainty is only partial. This may be seen as a general approach with some particular cases: a) precise knowledge of prior probabilities and b) complete uncertainty about the priors. In a) the region of uncertainty collapses to a point and we have the Bayes' rule of minimum risk and in b) the pure minimax deviation strategy comes up. While the first one may be criticized for being quite unrealistic, the other may be seen rather pessimistic. The experimental results for this proposed intermediate situation show that the Γ-*minMaxDev* classifier allows to reduce the maximum deviation from the optimal and performs well over a range of prior probabilities.

Acknowledgments

The authors thank the four referees and the associate editor for their helpful comments. This work was partially supported by the project TEC2005-06766-C03-02 from the Spanish Ministry of Education and Science.

References

N. Abe, B. Zadrozny, and J. Langford. An iterative method for multi-class cost-sensitive learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 3–11, 2004.

N. M. Adams and D. J. Hand. Comparing classifiers when the misallocation costs are uncertain. *Pattern Recognition*, 32(7):1139–1147, March 1998.

R. Alaiz-Rodriguez, A. Guerrero-Curieses, and J. Cid-Sueiro. Minimax classifiers based on neural networks. *Pattern Recognition*, 38(1):29–39, January 2005.

R. Barandela, J. S. Sanchez, V. García, and E. Rangel. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36(3):849–851, March 2003.

J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, second edition, 1985.

C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998. URL `http://www.ics.uci.edu/ mlearn/MLRepository.html`.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Chapman & Hall, NY, 1984.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

J. Cid-Sueiro and A. R. Figueiras-Vidal. On the structure of strict sense Bayesian cost functions and its applications. *IEEE Transactions on Neural Networks*, 12(3):445–455, May 2001.

C. Drummond and R. C. Holte. Explicitly representing expected cost: An alternative to ROC representation. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 198–207. ACM Press, 2000.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, 2001.

Y. C. Eldar and N. Merhav. Minimax approach to robust estimation of random parameters. *IEEE Trans. on Signal Processing*, 52(7):1931–1946, July 2004.

Y. C. Eldar, A. Ben-Tal, and A. Nemirovski. Linear minimax regret estimation of deterministic parameters with bounded data uncertainties. *IEEE Trans. on Signal Processing*, 52(8):2177–2188, August 2004.

M. Feder and N. Merhav. Universal composite hypothesis testing: A competitive minimax approach. *IEEE Trans. on Information Theory*, 48(6):1504–1517, June 2002.

A. Guerrero-Curieses, R. Alaiz-Rodriguez, and J. Cid-Sueiro. A fixed-point algorithm to minimax learning with neural networks. *IEEE Transactions on Systems, Man and Cybernetics Part C*, 34 (4):383–392, November 2004.

H. A. Güvenir, N. Emeksiz, N. Ikizler, and N. Örmeci. Diagnosis of gastric carcinoma by classification on feature projections. *Artificial Intelligence in Medicine*, 31(3), 2004.

N. Ikizler. Benefit maximizing classification using feature intervals. Technical Report BU-CE-0208, Bilkent University, Ankara, Turkey, 2002.

N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis Journal*, 6(5):429–450, November 2002.

M. G. Kelly, D. J. Hand, and N. M. Adams. The impact of changing populations on classifier performance. In *Proceedings of Fifth International Conference on SIG Knowledge Discovery and Data Mining (SIGKDD)*, pages 367–371, San Diego, CA, 1999.

H. J. Kim. On a constrained optimal rule for classification with unknown prior individual group membership. *Journal of Multivariate Analysis*, 59(2):166–186, November 1996.

M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In *Proceedings 14th International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann, 1997.

M. Kubat, R. Holte, and S. Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30(2/3):195–215, 1998.

S. Lawrence, I. Burns, A. D. Back, A. C. Tsoi, and C. L. Giles. Neural network classification and unequal prior class probabilities. In G. Orr, K.-R. Müller, and R. Caruana, editors, *Tricks of the Trade*, Lecture Notes in Computer Science State-of-the-Art Surveys, pages 299–314. Springer Verlag, 1998.

T. K. Moon and W. C. Stirling. *Mathematical Methods and Algorithms for Signal Processing*. Prentice Hall, 2000.

A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *ICML '05: Proceedings of the 22nd International Conference on Machine learning*, pages 625–632, New York, NY, USA, 2005. ACM Press. ISBN 1-59593-180-5.

E. Polak. *Optimization: Algorithms and Consistent Approximations*. Springer, 1997.

F. Provost. Learning with imbalanced data sets 101. In *Invited paper for the AAAI 2000 Workshop on Imbalanced Data Sets*. AAAI Press. Technical Report WS-00-05, 2000.

F. Provost and T. Fawcett. Robust classification systems for imprecise environments. *Machine Learning*, 42(3):203–231, March 2001.

M. Saerens, P. Latinne, and C. Decaestecker. Adjusting a classifier for new a priori probabilities: A simple procedure. *Neural Computation*, 14:21–41, January 2002.

E. Takimoto and M. Warmuth. The minimax strategy for Gaussian density estimation. In *Proceedings 13th Annual Conference on Computational Learning Theory*, pages 100–106. Morgan Kaufmann, San Francisco, 2000.

K. M. Ting. A study of the effect of class distribution using cost-sensitive learning. In *Proceedings of the Fifth International Conference on Discovery Science*, pages 98–112. Berlin: Springer-Verlag, 2002.

H. L. VanTrees. *Detection, Estimation and Modulation Theory*. John Wiley and Sons, 1968.

G. I. Webb and K. M. Ting. On the application of ROC analysis to predict classification performance under varying class distributions. *Machine Learning*, 58(1):25–32, 2005.

W. Wei, T. K. Leen, and E. Barnard. A fast histogram-based postprocessor that improves posterior probability estimates. *Neural Computation*, 11(5):1235 – 1248, July 1999.

B. Zadrozny and C. Elkan. Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 204–213. ACM Press, 2001.

B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Eighth International Conference on Knowledge Discovery and Data Mining*, 2002.

B. Zadrozny, J. Langford, and N. Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Proceedings of the third IEEE International Conference on Data Mining*, pages 435–442, 2003.

Z. H. Zhou and X. Y. LiuJ. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77, January 2006.