Multi-group Support Vector Machines with measurement costs: a biobjective approach *

Emilio Carrizosa.
Facultad de Matemáticas. Universidad de Sevilla (Spain).
ecarrizosa@us.es
Belen Martin-Barragan.
Facultad de Matemáticas. Universidad de Sevilla (Spain).
belmart@us.es
Dolores Romero Morales.
Saïd Business School. University of Oxford (United Kingdom).
dolores.romero-morales@sbs.ox.ac.uk

July 28, 2005

Abstract

Support Vector Machine has shown to have good performance in many practical classification settings. In this paper we propose, for multi-group classification, a biobjective optimization model in which we consider not only the generalization ability (modelled through the margin maximization), but also costs associated with the features. This cost is not limited to an economical payment, but can also refer to risk, computational effort, space requirements, etc. We introduce a biobjective mixed integer problem, for which Pareto optimal solutions are obtained. Those Pareto optimal solutions correspond to different classification rules, among which the user would choose the one yielding the most appropriate compromise between the cost and the expected misclassification rate.

Keywords: Multi-group Classification, Pareto Optimality, Biobjective Mixed Integer Programming, Feature Cost, Support Vector Machines.

*This research was partially supported by projects BFM2002-11282-E and BFM2002-04525-C02-02 of Ministerio de Ciencia y Tecnología (Spain) and FQM-329 of Plan Andaluz de Investigación (Andalucía, Spain).

1 Introduction

In the last years operations researchers have made significant contributions to problems related with Data Mining (e.g. [2, 5, 8, 9, 23, 28]), such as Supervised Classification. Roughly speaking, supervised classification consists of building a rule to predict the class-membership of new objects from the same population than those in a given database. Support Vector Machine (SVM), e.g. [11, 13, 20], has shown to be a powerful tool for Supervised Classification. When only two groups exist, this method attempts to build a hyperplane with maximal margin that separates the two groups. Margin can be seen as a value that is zero when there are misclassified objects and otherwise it measures the confidence in the prediction, [1]. It has been shown (e.g. [11, 32, 33]) that this method enjoys good generalization properties, in the sense that one can expect the good behavior obtained in the available data to be generalized to the population which data come from, since the probability of misclassifying a forthcoming individual can be bounded by a function which is decreasing in the margin.

Generalization ability, addressed via margin maximization, will be our first goal. However, in real-word classification problems it is very convenient to obtain classification rules that, not only achieve good classification behavior, but are also cheap or quick. A typical example is medical diagnosis, where some tests are much more expensive or take much longer than others. If the classification rule does not use variables based on the most expensive tests, classifying new patients will be much cheaper or quicker, perhaps without deteriorating significantly the quality of classification.

Together with misclassification costs, which are related with the generalization ability of the rule, other costs, linked to the variables or attributes, can be defined. In the simplest model we associate equal costs to each feature; keeping the total cost below a given level amounts to stating an upper bound on the number of features to be used. Turney [31] proposed other types of nontrivial cost, for instance the test cost, also called measurement cost, where each test (attribute, measurement, feature) has an associated cost, such as economical payment, computational effort or some kind of complexity.

The aim of minimizing such costs has been mentioned before in the literature as a desirable consequence of feature selection, see e.g. [18], but hardly directly addressed.

In this paper, we address classification problems in which both misclassification rate and measurement costs are relevant. To do this, we formulate a biobjective program of simultaneous minimization of misclassification rate, via the maximization of the margin (the natural measure in SVM), and measurement costs. Pareto-optimal solutions, i.e. classifiers that cannot be improved at the same time in both objectives, are sought. The set of Pareto-optimal solutions of the biobjective program gives us a finite set of classification rules, in such a way that any rule which is not Pareto-optimal should be discarded, since it is beaten in terms of margin and cost by another rule. Choosing one out of the set of Pareto-optimal rules is done by choosing an appropriate compromise between the two criteria involved.

We have structured the paper as follows. In Section 2 the problem is formally in-

troduced. In Section 3 we model the first goal: the measurement cost. Maximizing the margin, as a surrogate of minimizing the misclassification rate, will be our second goal. Formal definitions of margin are given in Section 4, by generalizing the concept of margin for two groups. A Biobjective Mixed Integer Program formulation is given in Section 5, where a method to find the Pareto-optimal classifiers, the Two-Phase Method [34], is proposed. In Section 6, such biobjective formulations are modified to allow some points in the training sample to be misclassified. Doing this we avoid the problem called overfitting. Finally, some numerical results are presented in Section 7.

2 The problem

We have a finite set of classes $\mathcal{C} = \{1, 2, \ldots, C\}$, and a set of objects Ω, each object u having two components (x^u, c^u). The first component x^u is called the *predictor vector* and takes values in a set X. The set X is usually assumed to be a subset of $\mathbb{R}^p$, and then, the components x_l, $l = 1, 2, \ldots, p$, of the predictor vector x are called *predictor variables*. The other component c^u, with values in the set of classes $\mathcal{C}$, is called the *class-membership* of object u. Object u is said to belong to class c^u.

In general, class-membership of objects in Ω is known only for a subset I, called the *training sample*: both predictor vector and class-membership are known for $u \in I$, whereas only x^u is known for $u \in \Omega \setminus I$.

For any $c \in \mathcal{C}$, denote by I_c the set of objects in I belonging to class c : $I_c = \{u \in I : c^u = c\}$. We assume that each class is represented in the training sample, i.e., $I_c \neq \emptyset \, \forall c \in \mathcal{C}$.

We use a classification model in which a *score function*, $f = (f_c)_{c \in \mathcal{C}}$ with $f_c : X \longrightarrow \mathbb{R}$, enables us to classify (allocate) any $z \in X$ as member of one of the classes as follows

$$z \text{ is allocated to the class } c \text{ if } f_c(z) > f_j(z), \, \forall j \neq c, \tag{1}$$

i.e. z is allocated to the class c^* whose score function is highest:

$$c^* = \arg\max_{c \in \mathcal{C}} f_c(z). \tag{2}$$

Notice that in case of ties, the object will be unclassified by this rule, and can be later allocated randomly or by a prefixed order to some class in $\arg\max_{c \in \mathcal{C}} f_c(z)$. Following a worst-case approach, we will consider those objects as misclassified throughout the paper. Score functions f_c are assumed to have the form

$$f_c(x) = \sum_{k=1}^{N} \alpha_k^c \phi_k(x) + \beta^c, \tag{3}$$

where $\alpha^c \in \mathbb{R}^N$, $\beta^c \in \mathbb{R}$, and $\mathcal{G} = \{\phi_1, \phi_2, \ldots, \phi_N\}$ is a finite set of real-valued functions on X. Hence, each f_c belongs to a vector space $\mathcal{F}$, generated by $\mathcal{G}$. For instance, linear

classifiers correspond to scores generated by

$$\mathcal{G} = \{x_1, x_2, \ldots, x_p\}, \tag{4}$$

whereas quadratic classifiers, [15, 16], are obtained by setting

$$\mathcal{G} = \{x_1, x_2, \ldots, x_p\} \bigcup \{x_i x_j : 1 \leq i \leq j \leq p\} \tag{5}$$

i.e., the set of monomials of degree up to 2.

This framework also includes voting classifiers, such as boosting, e.g. [14, 17], in which $\mathcal{C} = \{1, 2\}$ and a set of primitive classifiers $\phi_k : X \to \{0, 1\}$

$$\phi_k(x) = 1 \text{ iff } x \text{ is allocated to class 1 via the } k\text{-th classifier}, \tag{6}$$

are combined linearly into a single score function of the form (3). For a very promising strategy for generating such primitive classifiers see e.g. [7].

Denote the coefficients of the score function by $A = (\alpha^1, \ldots, \alpha^C)$ and $b = (\beta^1, \ldots, \beta^C)$. The problem of choosing f is reduced to the choice of its coefficients (A, b).

Definition 1 *$f = (f_c)_{c \in \mathcal{C}}$ with $f_c : X \longrightarrow \mathbb{R}$, is said to* separate *$\{I_c : c \in \mathcal{C}\}$ if*

$$f_{c^u}(x^u) > f_j(x^u) \ \forall j \neq c^u, \quad \forall u \in I. \tag{7}$$

Moreover, $\{I_c : c \in \mathcal{C}\}$ is said to be separable by the space $\mathcal{F}$ if there exists $f = (f_c)_{c \in \mathcal{C}}$, with $f_c \in \mathcal{F}$, separating $\{I_c : c \in \mathcal{C}\}$.

Now we compare the definition of separability given in Definition 1 with those existing in the literature, [1, 19, 20, 32].

For the two-group case, $\mathcal{C} = \{1, 2\}$, our definition is equivalent to the classical definition of separability stating that the convex hulls of $\{\phi(x^u) : u \in I_1\}$ and $\{\phi(x^u) : u \in I_2\}$ are contained in open halfspaces with a common hyperplane as boundary.

Property 2 *Let $\mathcal{C} = \{1, 2\}$. $\{I_c : c \in \{1, 2\}\}$ is separable iff there exists $(\omega, \gamma) \in (\mathbb{R}^N \setminus \{0\}) \times \mathbb{R}$ such that*

$$\begin{array}{l} \omega^\top \phi(x^u) + \gamma > 0 \quad \forall u \in I_1 \\ \omega^\top \phi(x^u) + \gamma < 0 \quad \forall u \in I_2. \end{array} \tag{8}$$

Proof. Take $\omega = \alpha^1 - \alpha^2$, $\gamma = \beta^1 - \beta^2$ and conversely, given (ω, γ), satisfying (8), setting $\alpha^1 = \omega$, $\beta^1 = \gamma$, $\alpha^2 = 0$ and $\beta^2 = 0$, we have a score function that correctly classifies $\{I_c : c \in \{1, 2\}\}$. □

For the multi-group case, $|\mathcal{C}| > 2$, we have that, together with the concept of separability given in Definition 1, a natural alternative exists: we will say that $\{I_c : c \in \mathcal{C}\}$ is one-against-rest separable (OAR-separable) iff for all $c_1 \in \mathcal{C}$, $\{I_{c_1}, \bigcup_{c \in \mathcal{C} \setminus \{c_1\}} I_c\}$ is separable.

Property 3 *One has*

$$\textit{OAR-separability} \Rightarrow \textit{separability}$$

Proof. Let $\{I_c : c \in \mathcal{C}\}$ be OAR-separable. It means that, for each class c_1, we have two score functions: f_{c_1} associated with I_{c_1}, and $f_{\overline{c_1}}$, associated with the objects in the remaining classes $\bigcup_{c\in\mathcal{C}\setminus\{c_1\}} I_c$. Since $(f_{c_1}, f_{\overline{c_1}})$ separates $\{I_{c_1}, \bigcup_{c\in\mathcal{C}\setminus\{c_1\}} I_c\}$, then

$$\begin{array}{lll} f_{c_1}(x^u) & > f_{\overline{c_1}}(x^u) & \forall u \in I_{c_1} \\ f'_{c_1}(x^u) & > f_{\overline{c_1}}(x^u) & \forall u \in \bigcup_{c\in\mathcal{C}\setminus\{c_1\}} I_c \end{array} \tag{9}$$

Set $g_c = f_c - f_{\overline{c}}$, for each $c \in \mathcal{C}$. Then $g_c(x^u) > 0$ iff $u \in I_c$. The function $g = (g_1, g_2, \ldots, g_C)$ trivially separates $\{I_c : c \in \mathcal{C}\}$. Hence, OAR-separability implies separability. □

Notice that the converse implication does not hold: for instance, in Figure 1 we have three classes $1, 2, 3$ with elements denoted respectively by crosses (points $(4, -3)$, $(1, 0)$ and $(4, 3)$), stars (points $(-1, -1)$ and $(3, -4)$) and circles (points $(-1, 1)$ and $(3, 4)$), which, as one can see in Figure 1, are not OAR-separable, but they are separable by the following score function,

$$\begin{aligned} f_1(x_1, x_2) &= x_1 \\ f_2(x_1, x_2) &= -x_2 \\ f_3(x_1, x_2) &= x_2. \end{aligned}$$

The definition of separability, as given in Definition 1, depends on the generator $\mathcal{G}$. Under weak assumptions, there exists a generator, $\mathcal{G}$, rich enough to enable separability of $\{I_c : c \in \mathcal{C}\}$.

Property 4 *If X is a subset of $\mathbb{R}^p$ and $x^u \neq x^v$, $\forall u, v \in I$ with $c^u \neq c^v$, then there exists a finite generator $\mathcal{G}$ such that $\{I_c : c \in \mathcal{C}\}$ is separable in the space $\mathcal{F}$ generated by $\mathcal{G}$.*

Proof. For each $c \in \mathcal{C}$, consider the function

$$f_c(x) = -\prod_{u\in I_c} d(x, x^u)^2$$

where $d(\cdot, \cdot)$ stands for the Euclidean distance. This function is zero for all x^u with $u \in I_c$ and strictly negative otherwise. Then, for $u \in I_c$, and $c' \neq c$,

$$f_c(x^u) - f_{c'}(x^u) = -f_{c'}(x^u) > 0,$$

thus, such set of functions separates $\{I_c : c \in \mathcal{C}\}$.

Moreover, each f_c is a polynomial in the variables $x_1, x_2, \ldots, x_p$, then it can be written as

$$f_c(x) = \sum_{\mathbf{h}=(h_1,\ldots,h_p)\in\{0,1,\ldots,2|I_c|\}^p} \alpha^c_{\mathbf{h}} \prod_{k=1}^{p} (x_k)^{h_k}, \tag{10}$$

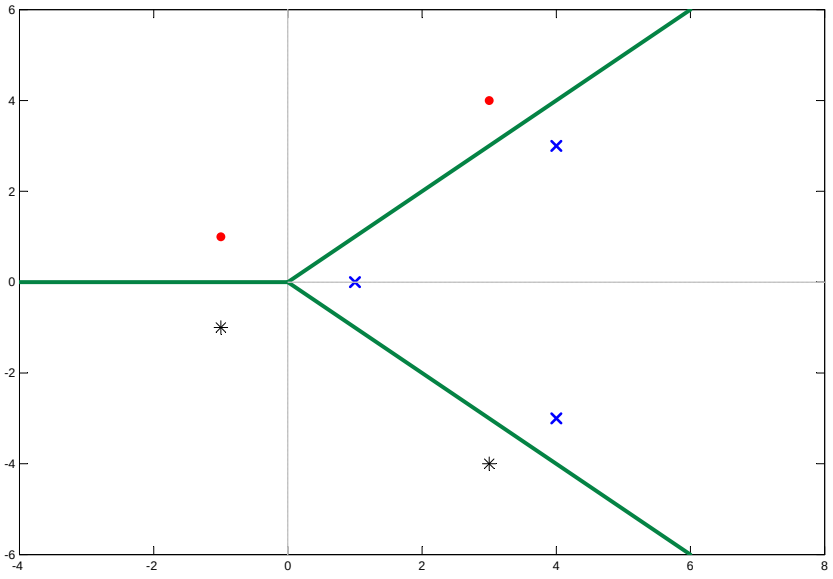

Figure 1: separable, but not OAR-separable

belonging to the space $\mathcal{F}$ generated by $\mathcal{G}$ the set of monomials of degree up to $2|I|$. $\square$

Suppose that $\mathcal{F}$ is rich enough to enable separability, which ensures the existence of separating functions f. However, uniqueness never holds. Indeed, it is easy to see that given $(\hat{\alpha}, \hat{\beta}) \in I\!R^{N+1}$ the classification rules obtained by (A, b) and $(\tilde{A}, \tilde{b})$ with $\tilde{\alpha}^c = \lambda\alpha^c + \hat{\alpha}$ and $\tilde{\beta}^c = \lambda\beta^c + \hat{\beta}$ for all $c \in \mathcal{C}$, are equivalent for all $\lambda > 0$, in the sense that both allocate objects to the same classes.

Moreover, there are also more than one score function that separates $\{I_c : c \in \mathcal{C}\}$ and they are not equivalent. For instance, given a score function separating $\{I_c : c \in \mathcal{C}\}$, let ε be any number satisfying:

$$0 < \varepsilon < \min_{u \in I} \min_{j \neq c^u} \{f_{c^u}(x^u) - f_j(x^u)\}.$$

The function $f^\varepsilon = (f_1 + \varepsilon, f_2, \ldots, f_C)$ also separates $\{I_c : c \in \mathcal{C}\}$. We need a criterion for choosing one of them. Following Vapnik's publications in generalization ability, e.g. [32], we will use the margin maximization criterion, as will be explained in Section 4.

3 Measurement costs

Finding classifiers separating conveniently the groups is a plausible criterion when obtaining the predictor vector x^u is costless. When this is not the case, we should also take into account the cost associated with the evaluation of the classification rule.

In many practical applications, as medical diagnosis, the predictor variables of the data may be some diagnosis test (such as blood test, ...) that have associated a cost, either money, or risk/damage incurred to the patient. If the classifier built does not depend on some of these variables, we could avoid their measurement (and the corresponding cost) in the diagnosis of new patients. In this situation, we should seek a classifier that enjoys good generalization properties, and at the same time, has low cost.

Obtaining cheaper or quicker classification rules have been mentioned as one of the desirable consequences of feature selection, where the aim is to reduce the number of variables or features used by the classification rule. However costs associated with such variables or features have seldom been considered.

Several authors have addressed measurement cost issues related with classification. For instance, [24, 25, 30] consider classification trees whose branching rule takes such costs into account. See [31] for a comparison of such methods and [3, 31] and the references therein for other proposals. In most cases, the unique goal is to minimize some surrogate of the expected misclassification cost, and, since the algorithm takes somehow into account measurement costs, it is hoped that the measurement cost of individuals with the rule obtained this way is not too high.

In this paper, however, we explicitly consider the minimization of measurement costs as one criterion, whose trade-off with margin optimization is to be determined by the user.

Costs are modelled as follows: Denote by Π_k the cost associated with evaluating the feature $\phi_k \in \mathcal{G}$ at a given x. For instance, if we are following a linear approach, as given by (4), Π_l represents the cost of measuring the predictor variable l in a new object.

Given the parameter $A = (\alpha^1, \ldots, \alpha^C)$, define

$$S(A) = \{k \,|\, \exists c \in \mathcal{C} : \, \alpha_k^c \neq 0, \, 1 \leq k \leq N\}.$$

In other words, $S(A)$ represents the set of features we have to use in order to classify new objects. In principle, these are the features we have to pay for, so a score function with coefficients (A, b) will have associated a measurement cost equal to

$$\pi(A, b) = \sum_{k \in S(A)} \Pi_k. \tag{11}$$

Pure linearity, as assumed in (11), may be unrealistic in some practical situations. For instance, it may be the case that, once we have incurred a cost for obtaining some feature ϕ_k, some other features may be given for free or at reduced cost. This may happen, for example, in a medical context when the measurement of a variable requires a blood

extraction, and some other variables can be measured using the same blood test. Another context where one encounters this, is the case in which some features are functions of other features: In model (5), feature $\phi(x) = x_i x_j$ is obtained for free if both features $\phi(x) = x_i$ and $\phi(x) = x_j$ have been previously inspected.

In Table 1 one can see the costs of a simple example with two classes $C = 2$, and $\mathcal{G} = \{\phi_1, \ldots, \phi_5\}$ with different costs.

features	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
costs	2	5	3	0	2

Table 1: Example of feature cost.

The score function given by $f_1 = \phi_1 + 4\phi_5$ and $f_2 = 3\phi_1 + 2$ incurs a cost of $2 + 2 = 4$.

Suppose that precedence constraints, in the form of a partial order $\prec$ between the features, is given. This means that if $h \preceq k$, the use of the feature ϕ_k requires also the payment for feature ϕ_h. Moreover, in computing the total cost, the cost for every feature has to be summed at most once. In order to formalize this, define an auxiliary variable $z_k \in \{0, 1\}$ for each $k = 1, \ldots, N$, representing

$$z_k = \begin{cases} 1 & \text{if payment of } \Pi_k \text{ is needed} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

in other words:

$$z_k = \begin{cases} 1 & \text{if } h \in S(A) \text{ for some } h \text{ with } k \preceq h \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

Thus, cost associated with a score function with coefficients (A, b) will be

$$\pi(A, b) = \sum_{k=1}^{N} z_k \Pi_k. \tag{14}$$

Particular cases already suggested in the literature can be easily accommodated into our framework. For instance, in [26] variables are grouped in a way that, if one variable from a group is requested, then all the others in the same group are available for zero additional cost. To model this case in our setting, define the cost of one variable from each group to be equal to the cost of the group it belongs to, and set the remaining variables to have zero cost. Moreover, choose a partial order $\prec$ for which $h \prec j$ iff variables h and j are in the same group and h has nonzero cost.

Moreover, this modelling technique allows us to use, but it is not limited to, polynomial kernels. Indeed, suppose a kernel $k(x, y) = \Phi(x)^\top \Phi(y)$ for some $\Phi : X \to F$. If Φ holds

- F is a finite dimensional *feature space*, $F \subset I\!R^N$,

- for any component ϕ_k, $k = 1, 2, \ldots, N$ of $\Phi = (\phi_1, \phi_2, \ldots, \phi_N)$, the information about what original variables are needed to calculate ϕ_k is available,

then, the cost associated to a score function can be modelled using the methodology explained in this section.

We will show in Sections 5 and 6, that this modelling technique allows formulations as Biobjective Mixed Integer Programs. For these models there exist suitable techniques for finding their Pareto-optimal solutions. Biobjective problems for more general problems, such as e.g. measurement cost minimization using kernels which are not of polynomial type, [36], can also be formulated. However, they yield combinatorial problems which are much harder to solve in practice.

Minimizing (14) will be one of our goals. However, our main goal is finding classifiers with good generalization properties. This, the second objective in our model, will be discussed in detail in the following section.

4 Margin optimization

Throughout this section, unless explicitly stated, we assume that $\mathcal{F}$ is rich enough to enable separability:

Assumption 1 *$\{I_c : c \in \mathcal{C}\}$ is separable by $\mathcal{F}$.*

We may observe that we can always consider F as in Property 4, and therefore Assumption 1 will be hold. However we expect in practice to attain separability with smaller generators.

Since by Assumption 1 objects in I will be correctly classified, the substantial matter is the classification of objects $u \in \Omega \setminus I$. Hence, we are interested in obtaining classifiers with good generalization properties, via margin maximization,[11, 32, 33]). The concepts of functional and geometrical margin, introduced in Cristianini and Shawe-Taylor [13] for the case of two groups, are extended below to the multi-group case.

Definition 5 *The* functional margin of an object u *with respect to the score function f, with coefficients (A, b), is the quantity*

$$\hat{\theta}^u(A, b) = \min_{j \neq c^u} \{f_{c^u}(x^u) - f_j(x^u)\} \tag{15}$$

The functional margin of a score function f, *with coefficients (A, b) with respect to a training sample I is equal to*

$$\hat{\theta}^I(A, b) = \min_{u \in I} \hat{\theta}^u. \tag{16}$$

We immediately obtain

Property 6 *A score function f with coefficients (A, b) separates $\{I_c : c \in \mathcal{C}\}$ if and only if, the margin $\hat{\theta}^I(A, b)$ is strictly positive.*

The choices (A, b) and $(\lambda A, \lambda b)$ yield the same classification rule, but have different functional margins. Hence, as in the two-group case, we need to normalize this quantity in order to be able to compare score functions.

The normalization done here is made dependent on a norm $\|\cdot\|$, which can be different from the standard choice of the Euclidean norm, [13]. This will allow us, as shown in Section 5, to formulate the resulting optimization problems as mixed integer linear problems, solvable with existing commercial software.

Definition 7 *Let $\|\cdot\|$ be a norm in $\mathbb{R}^{C\times N}$. The* geometrical margin of an object u *with respect to the score function (A, b), with $A \neq 0$, is the quantity*

$$\theta^u(A, b) = \frac{\hat{\theta}^u}{\|A\|}. \tag{17}$$

The geometrical margin of a score function (A, b) *with respect to a training sample I is the minimum:*

$$\theta^I(A, b) = \min_{u\in I} \theta^u. \tag{18}$$

Now, we consider the problem of maximizing the geometrical margin

$$\max_{A\neq 0,\, b\in\mathbb{R}^C} \frac{\min_{u\in I} \hat{\theta}^u(A, b)}{\|A\|}. \tag{19}$$

We have an alternative formulation, in terms of the functional margin, as given by the following proposition.

Proposition 8 *Problem* (19) *is equivalent to:*

$$\begin{array}{ll} \max & \min_{u\in I} \hat{\theta}^u(A, b) \\ s.t: & \|A\| \leq 1, \end{array} \tag{20}$$

in the sense that any optimal solution of (20) is also optimal for (19), and for any optimal solution (A^*, b^*) of (19),

$$(\hat{A}, \hat{b}) = \frac{1}{\|A^*\|}(A^*, b^*) \tag{21}$$

is an optimal solution of (20).

Property 9 *Problem (20) has finite optimal value.*

Proof. Let $(A, b) = (\alpha^1, \ldots, \alpha^C; \beta^1, \ldots, \beta^C)$ be a feasible solution of (20).

Let $u \in I$ and $j \neq c^u$, then

$$\begin{aligned} & |\alpha^{c^u}\phi(x^u) + \beta^{c^u} - \alpha^j\phi(x^u) - \beta^j| \\ = \; & |(\alpha^{c^u} - \alpha^j)\phi(x^u) + \beta^{c^u} - \beta^j| \\ \leq \; & |(\alpha^{c^u} - \alpha^j)\phi(x^u)| + |\beta^{c^u} - \beta^j| \end{aligned} \tag{22}$$

To bound the first term, observe that, since all norms are equivalent, there exists K such that $|\alpha_k^c| \leq K$ for all $k = 1, 2, \ldots, N$, $c \in \mathcal{C}$.

Hence,

$$\begin{aligned} & |(\alpha^{c^u} - \alpha^j)\phi(x^u)| \\ \leq \; & \sum_{k=1}^{N} |\alpha_k^{c^u} - \alpha_k^j||\phi_k(x^u)| \\ \leq \; & 2KN \max_{1\leq k\leq N,\, u\in I} |\phi_k(x^u)| = K' < \infty \end{aligned}$$

Now, we will bound the term $|\beta^{c^u} - \beta^j|$. Since each class is represented, $I_j \neq \emptyset$, let $v \in I_j$. Solution (A, b) feasible for (20) implies both u and v are correctly classified,

$$\begin{aligned} \alpha^{c^u}\phi(x^u) + \beta^{c^u} - (\alpha^j\phi(x^u) + \beta^j) > 0 \\ \alpha^{c^u}\phi(x^v) + \beta^{c^u} - (\alpha^j\phi(x^v) + \beta^j) < 0 \end{aligned}$$

yielding,

$$(\alpha^{c^u} - \alpha^j)\phi(x^v) < \beta^j - \beta^{c^u} < (\alpha^{c^u} - \alpha^j)\phi(x^u). \tag{23}$$

Thus

$$\begin{aligned} & |\beta^j - \beta^{c^u}| \\ \leq \; & \max\{|(\alpha^{c^u} - \alpha^j)\phi(x^u)|, |(\alpha^{c^u} - \alpha^j)\phi(x^v)|\} \\ \leq \; & \max_{v\in I}\{|(\alpha^{c^u} - \alpha^j)\phi(x^v)|\} \\ \leq \; & 2KN \max_{1\leq k\leq N,\, v\in I} |\phi_k(x^v)| = K'. \end{aligned}$$

Hence the objective function is bounded by

$$\min_{u\in I} \theta^u = \min_{u\in I} \min_{j\neq c^u} |\alpha^{c^u}\phi(x^u) + \beta^{c^u} - \alpha^j\phi(x^u) - \beta^j| \leq 2K'. \tag{24}$$

□

We have assumed that $\mathcal{F}$ is rich enough to enable separability of $\{I_c : c \in \mathcal{C}\}$. However, it may be useful to have a method to check such separability. In case we do not know if $\{I_c : c \in \mathcal{C}\}$ is separable in a space $\mathcal{F}$, solving Problem (20) allow us to check it. Indeed we have the property:

Property 10 *$\{I_c : c \in \mathcal{C}\}$ is separable if and only if Problem (20) has strictly positive optimal value.*

Another reduction of Problem (20) is even possible. For all $\lambda \in \mathbb{R}$ the score functions defined by (A, b) and $(A, \tilde{b})$, with $\tilde{b}^c = b^c + \lambda$ for all $c \in \mathcal{C}$, are equivalent in the sense that both classify objects to the same classes, and both have the same margins. Then, we can restrict the coefficients β^c to be nonnegative, yielding the problem:

$$\begin{array}{ll} \max & \min_{u \in I} \theta^u(A, b) \\ s.t.: & \|A\| \leq 1 \\ & (A, b) \in \mathbb{R}^{NC} \times \mathbb{R}^C_+. \end{array} \tag{25}$$

Property 11 *Problems (20) and (25) are equivalent in the sense that every optimal solution of (25) is also optimal for (20), and, for any optimal solution of (20), there exists a feasible solution of (25) that is also optimal in both problems.*

5 A biobjective approach

In the last sections we have described the two objectives of our problem, namely, maximizing the margin and minimizing the measurement cost. Hence we have the following biobjective problem:

$$\begin{array}{ll} \max & \theta(A, b) \\ \min & \pi(A, b) \\ s.t.: & \|A\| \leq 1 \\ & (A, b) \in \mathbb{R}^{NC} \times \mathbb{R}^C_+. \end{array} \tag{26}$$

Property 12 *The set of Pareto-optimal outcomes of the biobjective problem (26) is finite.*

Proof. The set of all outcomes of (26) can be calculated by solving the problem

$$\begin{array}{ll} \max & \theta(A, b) \\ s.t.: & \|A\| \leq 1 \\ & \pi(A, b) \leq \pi \\ & (A, b) \in \mathbb{R}^{NC} \times \mathbb{R}^C_+ \end{array} \tag{27}$$

for any π in the set of possible costs:

$$\{\pi(A, b) : (A, b) \in \mathbb{R}^{NC} \times \mathbb{R}^C_+\},$$

which is contained in the finite set $\left\{\sum_{k\in S}\Pi_k : S \subseteq \{1,2,\ldots,N\}\right\}$. □

Using the notation of section above, (26) can also be reformulated as

$$\begin{array}{lll}
\max & y & \\
\min & \sum_{k=1}^{N}\Pi_k z_k & \\
s.t.: & \sum_{k=1}^{N}\phi_k(x^u)\left(\alpha_k^i-\alpha_k^j\right)+\beta^i-\beta^j-y\geq 0, & \forall i\neq j;\ i,j\in\mathcal{C},\ u\in I_i \\
& \|A\|\leq 1 & \\
& -z_k\leq\sum_{k:h\preceq k}\sum_{c=1}^{C}\alpha_k^c\leq z_h & \forall h=1,2,\ldots,N \\
& \alpha_k^c \text{ unrestricted} & \forall k=1,2,\ldots,N;\ c\in\mathcal{C} \\
& y \text{ unrestricted} & \\
& \beta^c\geq 0 & \forall c\in\mathcal{C} \\
& z_k\in\{0,1\} & \forall k=1,2,\ldots,N
\end{array} \tag{28}$$

In this formulation if $\|\cdot\|$ is the L_∞, then the normalization constraint is redundant.

Due to the presence of a nonlinear constraint ($\|A\|\leq 1$), Problem (28) is a biobjective mixed integer nonlinear program.

Many classical SVM implementations have used the Euclidean norm [13], yielding a quadratic program. Mangasarian [22] proposes the use of other norms. In particular Linear Programming approaches have been implemented by different authors using polyhedral norms, [4, 29, 35]. Empirical results are shown in [27], where it is concluded that 'in terms of separation performance, L_1, L_∞ and Euclidean norm-based support vector machines tend to be quite similar'.

Instead of using the Euclidean norm, we suggest the use of a polyhedral norm, such as, for instance, a scaled L_1-norm, $\|A\|_1=\frac{1}{N}\sum_{k=1}^{N}\sum_{c=1}^{C}|\alpha_k^c|$. Then Problem (26), can be rewritten as a biobjective mixed integer *linear* problem, as stated below.

Property 13 *Let $\|\cdot\|$ be a scaled L_1-norm, $\|A\|_1=\frac{1}{N}\sum_{k=1}^{N}\sum_{c=1}^{C}|\alpha_k^c|$. Then, Problem (26) can be formulated as the following Biobjective Mixed Integer Problem,*

$$\begin{array}{lll}
\max & y & \\
\min & \sum_{k=1}^{N}\Pi_k z_k & \\
s.t.: & \sum_{k=1}^{N}\phi_k(x^u)\left(\alpha_{+k}^i-\alpha_{-k}^i-\alpha_{+k}^j+\alpha_{-k}^j\right)+\beta^i-\beta^j-y\geq 0, & \\
& & \forall i\neq j;\ i,j\in\mathcal{C},\ u\in I_i \\
& \sum_{c=1}^{C}\sum_{k=1}^{N}\left(\alpha_{+k}^c+\alpha_{-k}^c\right)\leq N & \\
& \sum_{k:h\preceq k}\sum_{c=1}^{C}\left(\alpha_{+k}^c+\alpha_{-k}^c\right)\leq N\,z_h & \forall h=1,2,\ldots,N \\
& y \textit{ unrestricted} & \\
& \alpha_{+k}^c\geq 0 & \forall k=1,2,\ldots,N;\ c\in\mathcal{C} \\
& \alpha_{-k}^c\geq 0 & \forall k=1,2,\ldots,N;\ c\in\mathcal{C} \\
& \beta^c\geq 0 & \forall c\in\mathcal{C} \\
& z_k\in\{0,1\} & \forall k=1,2,\ldots,N
\end{array} \tag{29}$$

We focus on the generation of Pareto-optimal solutions of Problem (26) for a scaled L_1-norm by using formulation (29) as dicussed below. The very same approach can be used if one chooses any other polyhedral norm, such as the L_∞ norm, instead of the L_1 norm, in the definition of geometrical margin.

Problem (29) is a biobjective mixed integer linear problem, which can be tackled for instance, by adapting the two-phase method of [34] designed for solving biobjective knapsack problems.

In the first phase, one obtains the so-called supported solutions, namely, those which are found as solution of the scalarized problem

$$\begin{array}{ll} \max & \lambda_1\theta(A,b) - \lambda_2\pi(A,b) \\ s.t.: & \|A\| \leq 1 \\ & (A,b) \in I\!R^{NC} \times I\!R^C_+ \end{array} \tag{30}$$

for some weights $\lambda_1, \lambda_2 \in [0,1]$, with $\lambda_1 + \lambda_2 = 1$. These points describe, in the outcome space, the frontier of the convex hull of the Pareto-optimal outcomes.

Since we face a bi-objective problem, the set of possible weights

$$\Lambda = \{(\lambda_1, \lambda_2) \in I\!R^2_+ : \lambda_1 + \lambda_2 = 1\}$$

that describe the supported efficient outcomes is unidimensional, and only a finite number of weights describe different outcomes. This fact can be exploited to find all supported outcomes in a sequential way.

A solution with minimal (zero) cost is the trivial solution $(A,b) = (0,0)$. Note that with this solution, points are classified arbitrarily by the tie-break rules, since all the score functions will be zero.

When we are optimizing only the first objective, namely maximizing the margin, the optimal value can be obtained by solving Problem (20), which can be easily reformulated as a linear program. Denote by θ^* its optimal value. Given an optimal solution (A^*, b^*) of (20), a feasible solution (A^*, b^*, z^*) of the biobjective problem (26) can be built by setting

$$z_i^* = \begin{cases} 1, & \text{if } \alpha_i^{*c} \neq 0 \text{ for some } c \in C, \\ 0, & \text{otherwise .} \end{cases}$$

If (A^*, b^*) is the unique optimal solution, then (A^*, b^*, z^*) will be a Pareto-optimal point. Otherwise, a Pareto-optimal point of (26) can be found by maximizing the margin, i.e., by solving,

$$\begin{array}{ll} \min & \pi(A,b) \\ s.t.: & \|A\| \leq 1 \\ & \theta(A,b) \geq \theta^* \\ & (A,b) \in I\!R^{NC} \times I\!R^C_+ \end{array}$$

Once we have both a Pareto-optimal solution with minimal cost, i.e. $(0,0)$, and a Pareto-optimal solution with maximal margin, namely (A_0, b_0), we construct an ordered

list (sorted by either margin or by cost) whose elements can be built from any two consecutive already known elements (A_1, b_1) and (A_2, b_2) by the scalarized Problem (30) for certain λ_1 and λ_2. Denote θ_1 and θ_2 the margin of solution (A_1, b_1) and (A_2, b_2) respectively and costs Π^1 and Π^2. The scalarization needed in the problem is

$$\lambda_1 = \frac{\theta^2 - \theta^1}{\theta^2 - \theta^1 + \Pi^2 - \Pi^1}$$

$$\lambda_2 = \frac{\Pi^2 - \Pi^1}{\theta^2 - \theta^1 + \Pi^2 - \Pi^1}.$$

All optimal solutions of such scalarized problem are Pareto-optimal points. If both (or any of) (A_1, b_1) and (A_2, b_2) are solutions of the scalarized problem, the set of its optimal solutions yield the only supported Pareto outcomes between those of (A_1, b_1) and (A_2, b_2), so we do not need to seek more supported Pareto points between them. Since the number of Pareto outcomes is finite, the process ends in finite time.

When all the supported Pareto outcomes are found, the non-supported ones may be obtained in the following way. Let (A_1, b_1) be any Pareto-optimal point with cost $\Pi^1 > 0$. Let $\hat{\pi}$ be the minimal feature cost that is positive,

$$\hat{\pi} = \min_{k=1,2,\ldots,N} \{\Pi_k : \Pi_k > 0\}.$$

Then a Pareto-optimal point, with cost strictly lower than Π^1, is obtained by solving the problem

$$\begin{array}{ll} \max & \theta(A, b) \\ s.t.: & \|A\| \leq 1 \\ & \pi(A, b) \leq \Pi^1 - \hat{\pi} \\ & (A, b) \in I\!R^{NC} \times I\!R^C_+. \end{array} \tag{31}$$

Then, the next Pareto-optimal point can be found in the same way. Thus, starting from any supported Pareto-optimal point with cost greater than zero, the non-supported Pareto-optimal outcomes between it and the next supported one can be found.

6 Soft-margin biobjective optimization

In classification problems, when the number of parameters to be fitted is large, model may incur a phenomenon called overfitting. It is said to happen when a classification rule achieves very good performance in the training sample I, but does not generalize well, thus yielding a bad performance in future objects.

Moreover, it may happen that I is not separable in the feature space. Then, the models proposed in the previous section do not apply, since they look for rules which correctly classify all the objects in I. As stated in Property 4, other feature space could be used, but usually they are more complicated and thus the model would incur overfitting.

In order to both avoid overfitting and deal with the non-separability of I, the typical SVM approach, called soft-margin maximization [13], is based on allowing some objects in I to be misclassified. This is done by adding to the model some slack variables $\xi \in I\!R^n_+$, where n is the cardinal of the training sample. Using this idea, the biobjective Problem (28) is replaced by the following problem:

$$
\begin{array}{lll}
\max & y & \\
\min & \sum_{k=1}^{N} \Pi_k z_k & \\
s.t.: & \sum_{k=1}^{N} \phi_k(x^u)\left(\alpha_k^i - \alpha_k^j\right) + \beta^i - \beta^j - y + \xi^u \geq 0, & \forall i \neq j;\ i,j \in \mathcal{C},\ u \in I_i \\
& \|A\| + \gamma \sum_{u \in I} \xi^u \leq N & \\
& -N\, z_k \leq \sum_{k:h \preceq k} \sum_{c=1}^{C} \alpha_k^c \leq N\, z_h & \forall h = 1,2,\ldots,N \\
& \alpha_k^c \text{ unrestricted} & \forall k = 1,2,\ldots,N;\ c \in \mathcal{C} \\
& y \text{ unrestricted} & \\
& \beta^c \geq 0 & \forall c \in \mathcal{C} \\
& z_k \in \{0,1\} & \forall k = 1,2,\ldots,N \\
& \xi^u \geq 0 & \forall u \in I,
\end{array}
\tag{32}
$$

for some user-defined value γ, which trades off the perturbations ξ^u and the margin.

In the same way as for the hard-margin approach, when $\|\cdot\|$ is a polyhedral norm, this problem can be formulated as a Biobjective Mixed Integer Problem. For instance, if $\|\cdot\|$ is a scaled L_1-norm, then Problem (32) can be formulated as follows:

$$
\begin{array}{lll}
\max & y & \\
\min & \sum_{k=1}^{N} \Pi_k z_k & \\
s.t.: & \sum_{k=1}^{N} \phi_k(x^u)\left(\alpha_{+k}^i - \alpha_{-k}^i - \alpha_{+k}^j + \alpha_{-k}^j\right) + \beta^i - \beta^j - y + \xi^u \geq 0, & \\
& & \forall i \neq j;\ i,j \in \mathcal{C},\ u \in I_i \\
& \sum_{c=1}^{C} \sum_{k=1}^{N} \left(\alpha_{+k}^c + \alpha_{-k}^c\right) + \gamma \sum_{u \in I} \xi^u \leq N & \\
& \sum_{k:h \preceq k} \sum_{c=1}^{C} \left(\alpha_{+k}^c + \alpha_{-k}^c\right) \leq N\, z_h & \forall h = 1,2,\ldots,N \\
& y \text{ unrestricted} & \\
& \alpha_{+k}^c \geq 0 & \forall k = 1,2,\ldots,N;\ c \in \mathcal{C} \\
& \alpha_{-k}^c \geq 0 & \forall k = 1,2,\ldots,N;\ c \in \mathcal{C} \\
& \beta^c \geq 0 & \forall c \in \mathcal{C} \\
& z_k \in \{0,1\} & \forall k = 1,2,\ldots,N \\
& \xi^u \geq 0 & \forall u \in I
\end{array}
\tag{33}
$$

The Two-Phase Method proposed in Section 5 to find the Pareto-optimal classifiers can also be used for solving (33). Note that in this case, the solution with minimal (zero) cost is not the trivial solution $(A, b) = (0, 0)$, but any optimal solution (33) with A set equal to the null matrix. The following steps of the method remain analogous to the hard-margin approach, and will not be repeated here.

7 Numerical results

In order to explore both, costs and quality, of the Pareto score functions obtained, we have performed a series of numerical tests on four standard databases, publicly available from the UCI Machine Learning Repository [6], namely, the BUPA Liver-disorders Database, called here `bupa`; the Pima Indians Diabetes Database, called here `pima`; the New Diagnostic Database, contained in the Wisconsin Breast Cancer Databases, called here `wdbc`, and the Credit Screening Databases, called here `credit`.

For each database, the name of the file (as called in the database), the total number of objects $|\Omega|$, the number of groups C and the number of variables (all quantitative) p are given in Table 2.

Database	filename	$\|\Omega\|$	C	p
`bupa`	bupa.data	345	2	6
`pima`	pima-indians-diabetes.data	768	2	8
`wdbc`	wdbc.data	569	2	30
`credit`	crx.data*	768	2	8

Table 2: Parameters of the databases. *only the numerical variables were used.

For the sake of simplicity, the features are chosen as the original variables in the database $x_1, x_2, \ldots, x_p$ and their products, yielding monomials of degree up to g. However, other feature spaces, as those proposed by [7], might give better classification rates.

Two types of costs are considered for the original variables. For the four databases, costs are independently chosen, randomly in the interval $(0, 1)$. Moreover, for the databases `bupa` and `pima` there exists a file, donated by Turney [31] and publicly available in the UCI repository [6], which contains an example for possible costs for the measurement of the variables. The cost information comes from the Ontario Health Insurance Program's fee schedule. For these databases we have also considered such given costs. The remaining features have zero cost. The partial order is given as follows: feature $\phi = x_k$ precedes all features of the form $\phi(x) = x_k q(x)$ for some monomial $q(x)$ of degree up to $g - 1$.

Data were standardized by subtracting its mean and dividing by its standard deviation. Then, from each database, a random sample with two thirds of the objects is drawn and used as training sample I. The supported Pareto-optimal solutions of Problem (32) were computed by the first phase of the Two-Phase Method [34], described in Section 5. The non-supported Pareto-optimal solutions can also be computed using formulation (31). The trade-off parameter γ is chosen to be equal to the number of objects in I.

The results are plotted in Figures 2-9. In the right side of such figures, measurement costs of the Pareto-optimal rules (except for zero-cost solutions) are plotted against the margin. Since only Pareto-optimal solutions are considered, we see that, the higher the cost, the higher the margin.

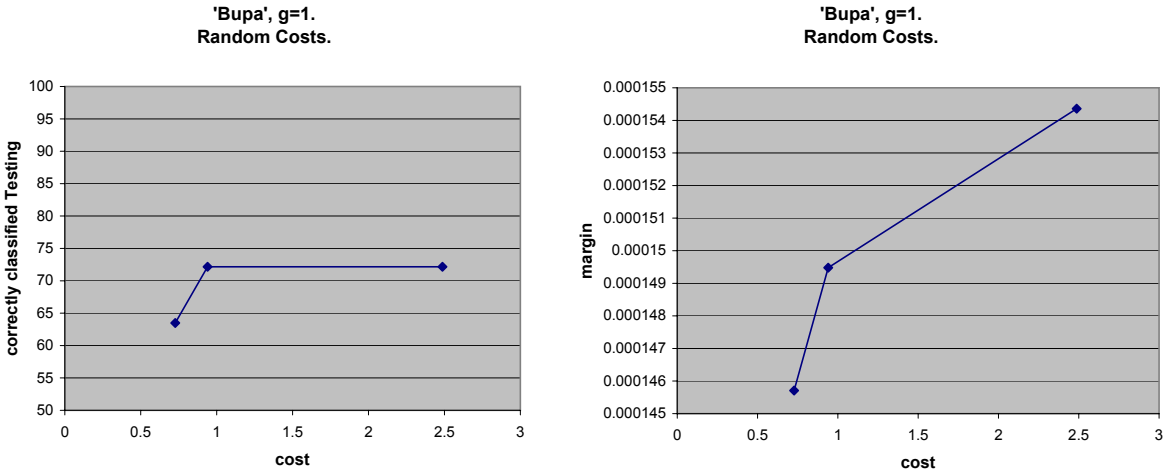

Figure 2: Database 'bupa', $g = 1$, random costs.

This is the plot the final user will obtain in real-world applications, and chose, with this information, one classification rule.

However, margin maximization is only a surrogate for the minimization of the misclassification rate, which will remain unknown. In the right side of Figures 2-9 we have plotted, for the Pareto-optimal classifiers obtained, costs against the percentage of correctly classified objects in the testing sample. Figures show clearly that high correct classification rates correspond to high costs. Moreover, the trade-off between measurement costs and margin translates into a similar trade-off between measurement costs and percentage of correctly classified objects.

method	'bupa'	'pima'	'wdbc'	'credit'
1-Nearest Neighbor	60.87	64.84	94.74	72.07
2-Nearest Neighbor	57.39	69.14	94.21	70.72
3-Nearest Neighbor	60.00	72.27	95.26	73.87
4-Nearest Neighbor	60.87	72.27	95.26	72.52
5-Nearest Neighbor	62.61	71.48	95.79	72.07
Classification Tree	67.83	70.31	90.53	72.97
SVM with linear kernel	72.17	74.22	95.79	77.48
SVM with polynomial kernel, grade =2	66.96	38.28	94.21	65.32
SVM with polynomial kernel, grade =3	59.13	66.41	93.68	69.37
SVM with polynomial kernel, grade =4	58.26	62.89	89.47	59.01
SVM with polynomial kernel, grade =5	57.39	67.19	91.58	75.23
SVM with radial basis function kernel	68.70	64.84	63.16	77.48

Table 3: Behavior of other methods.

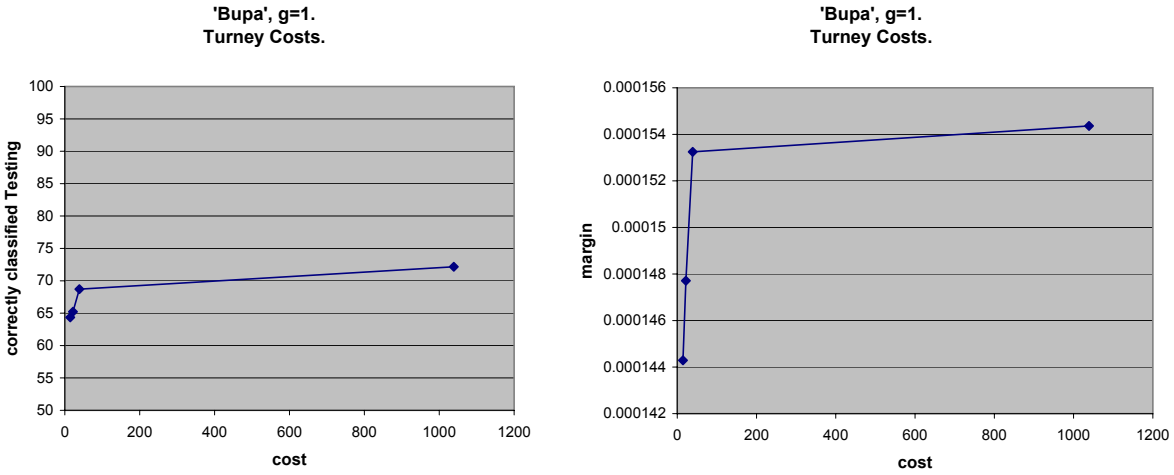

Figure 3: Database 'bupa', $g = 1$, Turney's costs.

For comparative purposes, in Table 3, the percentage of correctly classified objects is shown for different classification methods, such as classification trees [10], $k-$nearest neighbor classifier [12] and the classical SVM approach as implemented in SVMlight [21]. It can be observed that the classification behavior of the Pareto-optimal classifiers are among the best ones, even for low classification costs.

The method proposed in this paper, can thus be seen as a procedure that generates a series of classification rules with different costs, and expected good classification behavior supported by the theoretical generalization properties of the margin maximizer (e.g. Vapnik [33]). Choosing one classification rule among them can be done by the user after plotting the measurement costs against margins, as illustrated in the examples.

References

[1] E.L. Allwein, R.E. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:369–409, 2000.

[2] C. Apte. The big (data) dig. *OR/MS Today*, February 2003.

[3] V. Bayer Zubek. *Learning Cost-Sensitive Diagnostic Policies from Data*. PhD thesis, Oregon State University, July 2003. http://eecs.oregonstate.edu/library/?call=2003-13.

[4] K. Bennet. Combining support vector and mathematical programming methods for induction. In *Advances in Kernel Methods - Support Vector Learning*, 1999.

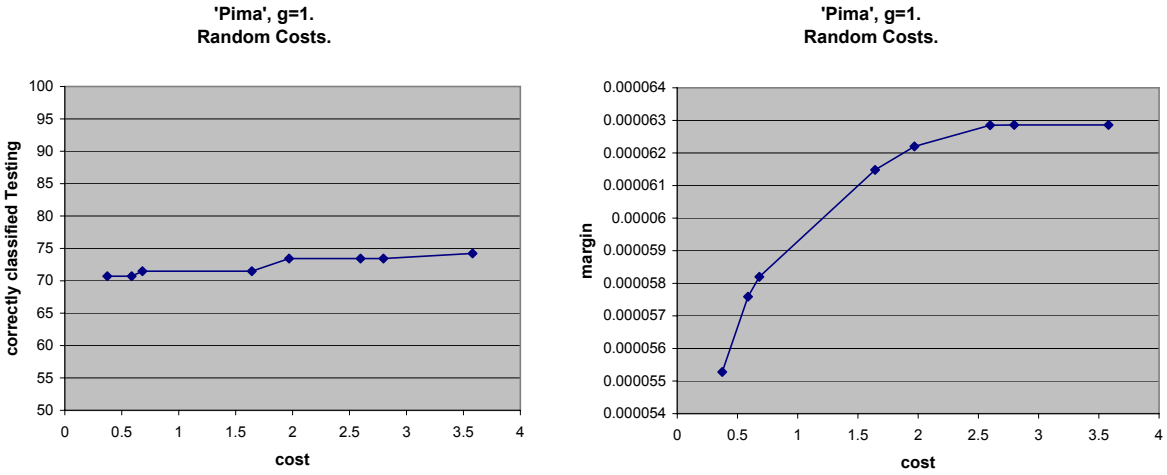

Figure 4: Database 'pima', $g = 1$, random costs.

[5] K.P. Bennet and O.L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–24, 1992.

[6] C.L. Blake and C.J. Merz. UCI Repository of Machine Learning Databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html, 1998. University of California, Irvine, Dept. of Information and Computer Sciences.

[7] E. Boros, P. L. Hammer, T. Ibaraki, and A. Kogan. A logical analysis of numerical data. *Mathematical Programming*, 79:163–190, 1997.

[8] P. Bradley, O. Mangasarian, and D. Musicant. Optimization methods in massive datasets. In J. Abello, P.M. Pardalos, and M.G.C. Resende, editors, *Handbook of Massive Datasets*, pages 439–472. Kluwer Academic Pub., 2002.

[9] P.S. Bradley, U.M. Fayyad, and O.L. Mangasarian. Mathematical programming for data mining: formulations and challenges. *INFORMS Journal on Computing*, 11(3):217–238, 1999.

[10] L. Breiman, J.H. Friedmann, R.A. Olshen, and C.J. Stone. *Classification and Regresion Trees*. Wadsworth, Belmont, CA, 1984.

[11] C. Cortes and V. Vapnik. Support-vector network. *Machine Learning*, 1:113–141, 1995.

[12] T.M. Cover and P.E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theroy*, 13:21–27, 1967.

[13] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines*. Cambridge University Press, 2000.

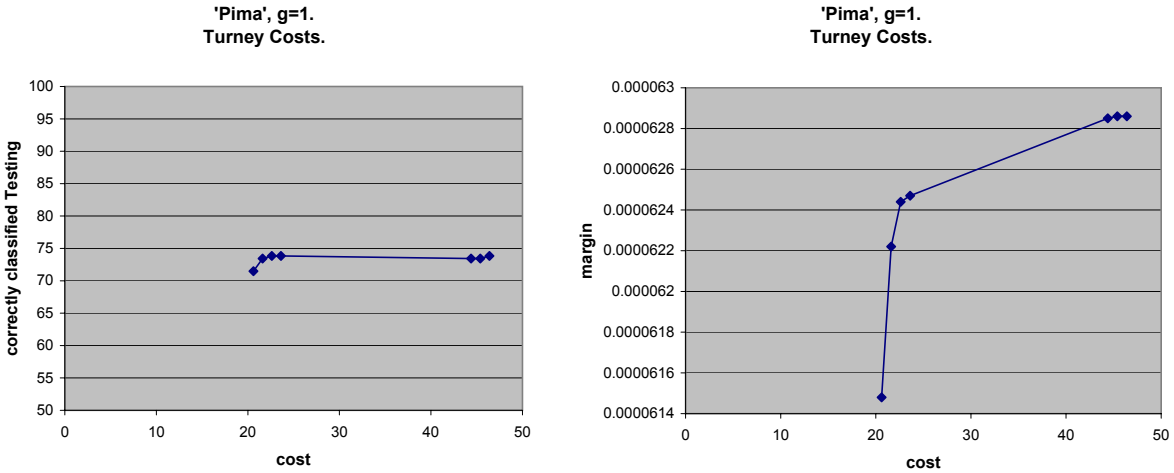

Figure 5: Database 'pima', $g = 1$, Turney's costs.

[14] A. Demiriz, K. P. Bennett, and J. Shawe-Taylor. Linear programming boosting via column generation. *Machine Learning*, 46(1):225–254, 2002.

[15] A.P. Duarte Silva and A. Stam. Second order mathematical programming formulations for discriminant analysis. *European Journal of Operational Research*, 72:4–22, 1994.

[16] J.E. Falk and V.E. Karlov. Robust separation of finite sets via quadratics. *Computers and Operations Research*, 28:537–561, 2001.

[17] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[18] I. Guyon and A. Elisseff. An introduction to variable and feature selecion. *Journal of Machine Learning Research*, 3(1157-1182), 2003.

[19] T. Hastie and R. Tibshirani. Classification by pairwise coupling. *The Annals of Statistics*, 26(2):451–471, 1998.

[20] R. Herbrich. *Learning Theory Classifiers. Theory and Algorithms.* MIT Press, 2002.

[21] T. Joachims. *Learning to Classify Text Using Support Vector Machines.* Kluwer, 2002.

[22] O.L. Mangasarian. Linear and nonlinear separation of patterns by linear programming. *Operations Research*, 13:444–452, 1965.

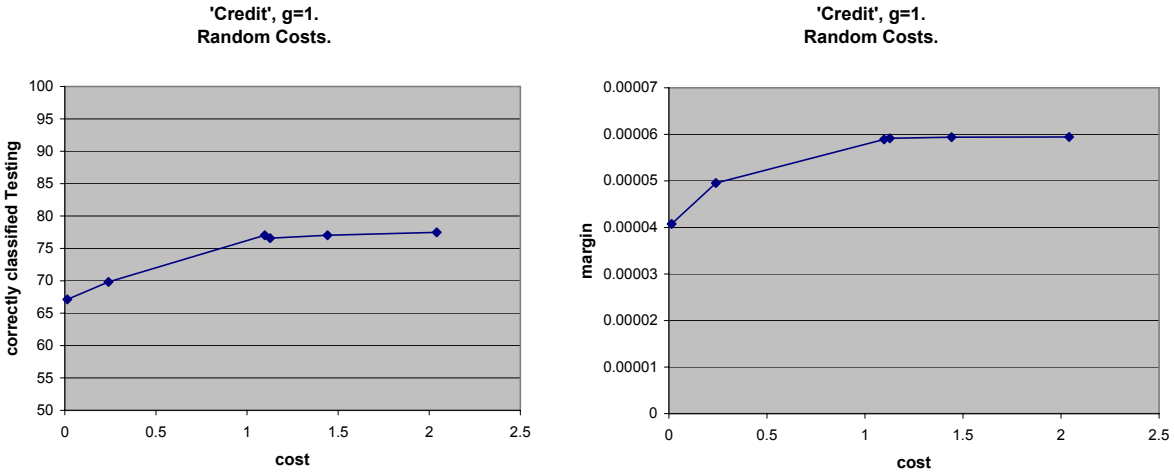

Figure 6: Database 'credit', $g = 1$, random costs.

[23] O.L. Mangasarian. Mathematical programming in data mining. *Data Mining and Knowledge Discovery*, 42(1):183–201, 1997.

[24] S.W. Norton. Generating better decision trees. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, IJCAI-89*, pages 800–805, Detroit, Michigan, 1989.

[25] M. Núñez. The use of background knowledge in decision tree induction. *Machine Learning*, 6:231–250, 1991.

[26] P. Paclík, R.P.W. Duin, G.M.P. van Kempen, and R. Kohlus. On feature selection with measurement cost and grouped features. *Lecture Notes in Computer Science*, 2396(461-469), 2002.

[27] J.P. Pedroso and N. Murata. Support vector machines with different norms: motivation, formulations and results. *Pattern recognition letters*, 22:1263–1272, 2001.

[28] A.M. Rubinov, A.M. Bagirovand, N.V. Soukhoroukova, and J. Yearwood. Unsupervised and supervised data classification via nonsmooth and global optimization. *TOP*, 11(1):1–93, 2003.

[29] A. Smola, T.T. Friess, and B. Schölkopf. Semiparametric support vector and linear programming machines. In *Advances in Neural Information Processing Systems 10*, pages 585–591, 1999.

[30] M. Tan. Cost-sensitive learning of classification knowledge and its applications in robotics. *Machine Learning*, 13:7–33, 1993.

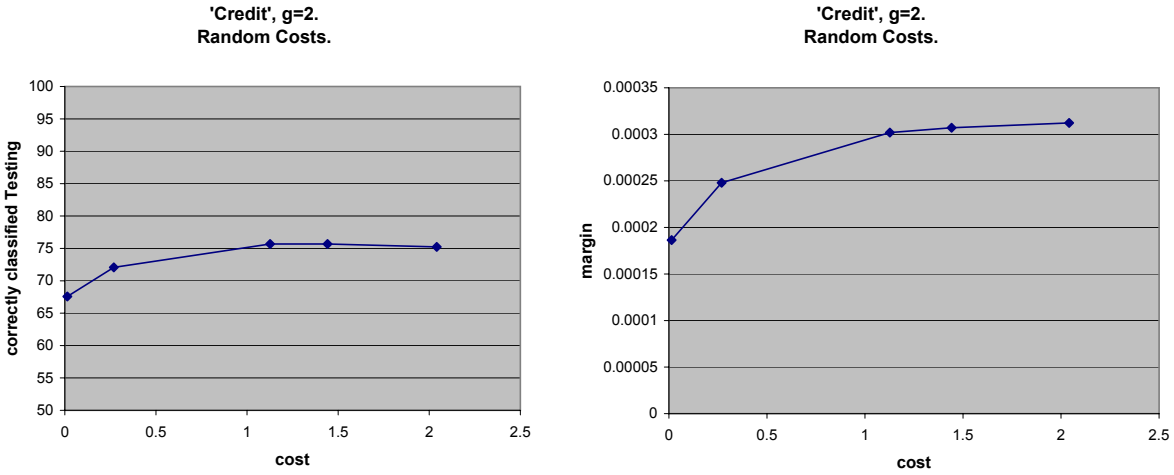

Figure 7: Database 'credit', $g = 2$, random costs.

[31] P.D. Turney. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research*, 2:369–409, 1995.

[32] V. Vapnik. *The nature of statistical learning theory*. Springer-Verlag, 1995.

[33] V. Vapnik. *Statistical learning theory*. Wiley, 1998.

[34] M. Visée, J. Teghem, M. Pirlot, and E.L. Ulungu. Two-phases method and branch and bound procedures to solve the bi-objective knapsack problem. *Journal of Global Optimization*, 12:139–155, 1998.

[35] J. Weston, A. Gammerman, M.O. Stitson, V. Vapnik, V. Vovk, and C. Watkins. Support vector density estimation. In *Advances in Kernel Methods - Support Vector Learning*, 1999.

[36] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs. In *Advances in Neural Information Processing Systems 13*, 2001.

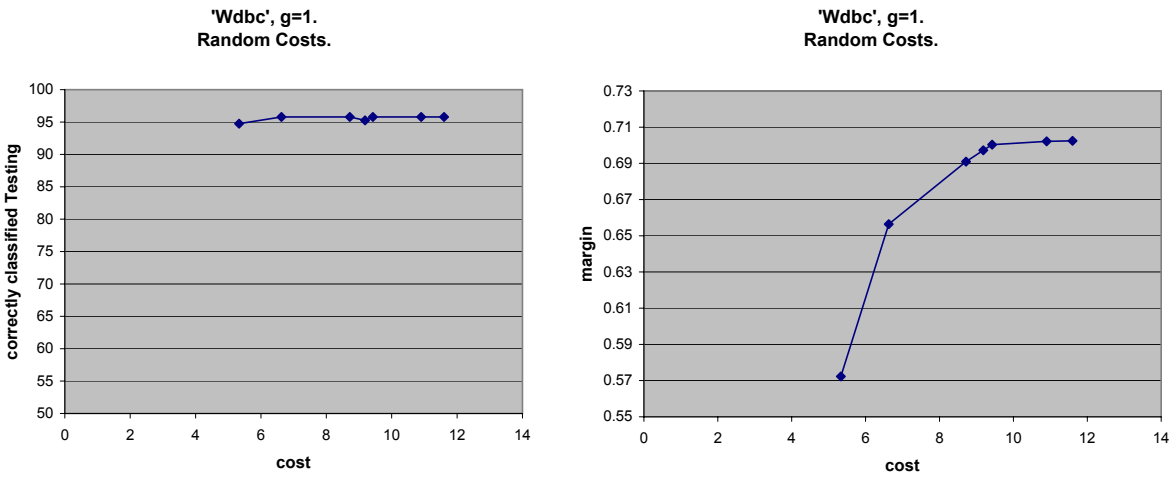

Figure 8: Database 'wdbc', $g = 1$, random costs.

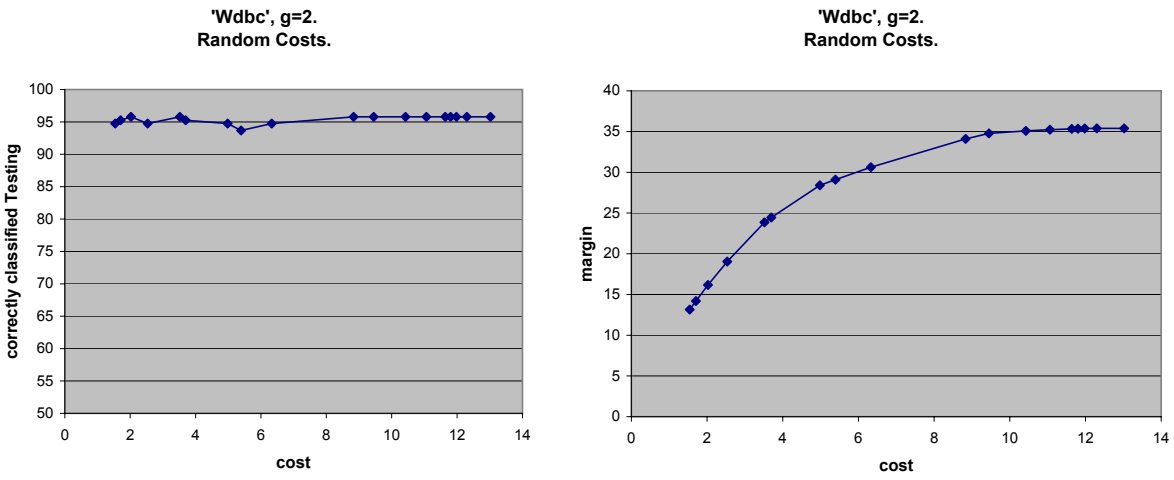

Figure 9: Database 'wdbc', $g = 2$, random costs.