# Strongly Agree or Strongly Disagree?: Rating Features in Support Vector Machines

Emilio Carrizosa
Universidad de Sevilla (Spain)
ecarrizosa@us.es
Amaya Nogales-Gómez
Universidad de Sevilla (Spain)
amayanogales@us.es
Dolores Romero Morales
University of Oxford (United Kingdom)
dolores.romero-morales@sbs.ox.ac.uk

October 15, 2013

**Abstract**

In linear classifiers, such as the Support Vector Machines (SVM), a score is associated with each feature and objects are assigned to classes based on the linear combination of the scores and the values of the features. Inspired by discrete psychometric scales, which measure the extent to which a factor is in agreement with a statement, we propose the Discrete Level Support Vector Machines (DILSVM) where the feature scores can only take on a discrete number of values, defined by the so-called feature rating levels. The DILSVM classifier benefits from interpretability as it can be seen as a collection of Likert scales, one for each feature, where we rate the level of agreement with the positive class. To build the DILSVM classifier, we propose a Mixed Integer Linear Programming approach, as well as a collection of strategies to reduce the building times. Our computational experience shows that the 3-point and the 5-point DILSVM classifiers have comparable accuracy to the SVM with a substantial gain in interpretability and sparsity, thanks to the appropriate choice of the feature rating levels.

*Keywords:* Support Vector Machines, Mixed Integer Linear Programming, Likert scale, interpretability, feature rating level.

## 1 Introduction

*Supervised Classification*, (Apte 2003, Hand et al. 2001, Wu et al. 2007), based on large datasets, plays an important role in many industries, with notable examples being credit scoring (Baesens et al. 2003), fraud detection (Cecchini et al. 2010), customer targeting (Archak et al. 2011, Cui et al. 2006), and surveillance (Fang et al. 2013). In Supervised Classification, we are given a set of objects $\Omega$ partitioned into classes and the aim is to build a procedure for classifying new objects. In its simplest form, each object $i \in \Omega$ has associated a pair $(x_i, y_i)$, where the feature vector $x_i$ takes values on a set $X \subseteq \mathbb{R}^d$ and $y_i \in \{-1, +1\}$ is the class membership of object $i$.

There are several desirable managerial properties in Supervised Classification methods. Incorporating domain knowledge at the time of building the classifier is a very appealing property (Cecchini et al. 2010). One may also like to keep the costs of learning the classifier low, where these costs arise from retrieving the features (Carrizosa et al. 2008, Turney 1995), obtaining feature information from the classifier (Saar-Tsechansky et al. 2009), or refreshing the classifier to include new data (Fang et al. 2013). Another desirable property is the comprehensibility/interpretability of the classifier. Since conciseness of the classifier is closely related with its interpretability, much effort has been devoted to increasing its sparsity, i.e., to reducing the number of active features in the classifier (Guyon et al. 2002), discretizing the features to detect active ranges of the features (Liu et al. 2002, Romero Morales and Wang 2009) or relevant thresholds for the features (Carrizosa et al. 2010). When trading off between accuracy and interpretability, another popular approach has been to extract easy–to–understand structures from powerful yet black–box type classifiers, such as if–then rules, decision trees and decision tables (Baesens et al. 2003, Bertsimas et al. 2011, Martens et al. 2007, Martens and Provost 2013, Orsenigo and Vercellis 2003, 2004). Yet another way of achieving interpretability is by building discrete linear classifiers (Chevaleyre et al. 2013, Golea and Marchand 1993). Classifier interpretability is a major challenge in datasets that contain a huge number of features, where most of these are irrelevant. To address this issue, models have been proposed aimed at increasing the sparsity of the classifier, as is done with the Lasso, see Li et al. (2006) and Roth (2004).

In Supervised Classification, linear classifiers are based on score functions. For instance, the CHADS$_2$ score (Gage et al. 2001, Letham et al. 2012) is widely used in medicine to predict the risk of stroke in patients with atrial fibrillation based on five different symptoms of the patient, whereas the extended CHA$_2$DS$_2$–VAS$_c$ score (Lip et al. 2010) includes three additional risk factors. A score is associated with each feature, and objects are assigned to classes based on the linear combination of the scores and the values of the features. The role each feature plays in the classifier is related to the magnitude of the corresponding score, while the sign gives information on how the feature points towards a given class.

A state-of-the-art method in Supervised Classification using a score function is the Support Vector Machines (SVM), Cristianini and Shawe-Taylor (2000), Vapnik (1995, 1998). The SVM aims at separating both classes by means of a hyperplane, $\omega^\top x + b = 0$, found by solving the following Quadratic Programming problem with linear constraints:

$$\min_{\omega,\, b,\, \xi} \ \frac{1}{2}\sum_{j=1}^{d} \omega_j^2 + \frac{C}{n}\sum_{i=1}^{n} \xi_i$$

s.t. (SVM)

$$\begin{aligned} y_i(\omega^\top x_i + b) &\geq 1 - \xi_i \qquad \forall i = 1, \ldots, n \\ \omega &\in \mathbb{R}^d \\ b &\in \mathbb{R} \\ \xi_i &\geq 0 \qquad \forall i = 1, \ldots, n, \end{aligned}$$

where $n$ is the size of the sample used to build the classifier, $C$ is a nonnegative tradeoff parameter, $(\xi_i)$ is the vector of deviation variables, the input feature vector $x_i$ takes values on $\mathbb{R}^d$ and $\sum_{i=1}^{n} \xi_i$ is the so-called *hinge loss* function. See Carrizosa and Romero Morales (2013) for a recent review on Mathematical Optimization and the SVM, and Bertsimas et al. (2008), Brooks (2011), Cecchini et al. (2010), Chaovalitwongse et al. (2008), Guyon et al. (2002), Martens et al. (2007), Romero Morales and Wang (2010) for successful applications of the SVM.

The classifier obtained by solving the SVM, like any linear classifier, provides valuable information on the role of each feature in the classifier. Indeed, the set of features can be partitioned into three clusters, namely those with positive $\omega_j$, those with negative $\omega_j$, and the ones with $\omega_j$ equal to zero. We can say that features in the first group have a positive contribution in the classifier, and thus such features point towards class $+1$, the positive class; similarly, features in the second group have a negative contribution, and thus such features point towards class $-1$, the negative class; and those features with $\omega_j = 0$ have no contribution in the classifier and are therefore irrelevant. However, the construction of the SVM classifier is aimed at margin maximization (a surrogate of classification accuracy), and interpretability issues are fully disregarded in the formulation.

Inspired by discrete psychometric scales, which measure the extent to which a factor is in agreement with a statement, (Keeney and Gregory 2005), and by linear classifiers in which the weights are allowed to take values on a discrete set, (Chevaleyre et al. 2013, Golea and Marchand 1993), we propose the Discrete Level Support Vector Machines (DILSVM) where the feature scores can only take on a discrete number of levels, defined by the so-called feature rating levels. The DILSVM classifier benefits from interpretability, as it can be seen as a collection of Likert scales, one for each feature, where we rate the level of agreement with the positive class.

We formulate the DILSVM as a Mixed Integer Linear Programming (MILP) problem. The parameters of our model, namely, the tradeoff parameter $C$ and the rating levels, may heavily influence the performance, i.e., the accuracy and the sparsity, of the DILSVM and, therefore, have to be carefully chosen. Thus, building the DILSVM classifier involves solving a series of MILP problems, which, if solved to optimality, may make the overall computational cost high for large datasets. For this reason we propose three strategies to alleviate the computational burden, with different tradeoffs between performance and reduction in computational cost. Such strategies use the guidance of related but simpler optimization models, and reduce the computational cost associated with each parameter vector and/or the number of parameter vectors to be inspected.

In our computational experience, we compare the DILSVM against the SVM classifier using ten real-life datasets. We show that the 3-point and 5-point DILSVM classifiers, built using the MILP approach, have comparable accuracy to the SVM, with a substantial gain in sparsity. Moreover, by the very nature of our procedure (only a few levels, to be interpreted as intensities, are allowed), interpretability is definitely improved allowing us to visualize the classification process via Likert scales. The tests illustrating the performance of the reduction strategies reveal a clear competitiveness in terms of sparsity, while the accuracy depends on the magnitude of the reduction, being close to the SVM accuracies. In our computational experience, we also compare the DILSVM against the 3-point DILSVM classifier with fixed parameters using ten real-life datasets. The 3-point DILSVM with fixed parameters is inspired by the model in (Chevaleyre et al. 2013). The results illustrate the relevance of tuning parameters to ensure accuracy and sparsity are not compromised.

Likert scale representations allow non-experts to easily understand the classifier, and there are fewer relevant features because of the improvement on sparsity. This contributes to the literature on SVM, but more generally, on Data Mining, because it significantly improves interpretability of learning methods and their visualization, without comprising accuracy. These advantages are obtained by extending existing SVM models, adding parameters right in the place they are needed to improve sparsity and thus interpretability.

The remainder of this paper is organized as follows. In Section 2 we introduce the DILSVM classifier. In Section 3 we discuss procedures for building the DILSVM classifier. In Section 4 we report our computational results using real-life datasets. We end the paper in Section 5 with some conclusions and directions for future research.

## 2 The DILSVM classifier

The Discrete Level Support Vector Machines (DILSVM) is a variant of the SVM classifier where for each feature $j$, the score $\omega_j$ can only take on a discrete number of values. Let $\mathcal{A} \subset \mathbb{R}$ be a finite set that includes the value 0, which models the marginal impact of the feature on the classifier. We can formulate the DILSVM as the following Discrete Quadratic Programming problem:

$$\min_{\omega,\, b,\, \xi} \frac{1}{2}\sum_{j=1}^{d} \omega_j^2 + \frac{C}{n}\sum_{i=1}^{n} \xi_i \tag{1}$$

s.t.

$$\begin{aligned} y_i(\omega^\top x_i + b) &\geq 1 - \xi_i & \forall i = 1, \ldots, n && (2)\\ \omega_j &\in \mathcal{A} & \forall j = 1, \ldots, d && (3)\\ b &\in \mathbb{R} & && (4)\\ \xi_i &\geq 0 & \forall i = 1, \ldots, n. && (5) \end{aligned}$$

See Burer and Letchford (2012) for a recent review of algorithmic tools for Mixed Integer Nonlinear Programming.

For adequate choices of $\mathcal{A}$, this model gains in interpretability and visualization. For instance, let us consider $\mathcal{A} = \{-a, 0, a\}$, $a > 0$. In the DILSVM classifier, some of the $\omega_j$ will be equal to $a$ or $-a$, while the rest will be equal to zero. Features for which

$\omega_j = a$ will have a positive impact on the classifier and therefore will point towards the positive class, those for which $\omega_j = -a$ will point towards the negative class, while the rest have no impact. We can view this in an alternative way: for a given rating level $a$, the DILSVM detects those features which *strongly agree* with the positive class, those which *strongly disagree* (and therefore strongly agree with the negative class), and those which are irrelevant to the classifier. This DILSVM classifier can be represented as a collection of 3-point Likert scales, one for each feature, measuring the extent to which the feature is in agreement with the positive class. When looking for more granularity of the scale, we can increase the size of $\mathcal{A}$. For $\mathcal{A} = \{-a_1, -a_2, 0, a_2, a_1\}$, $a_1 > a_2 > 0$, the DILSVM classifier can be seen as a collection of 5-point Likert scales where features $j$ with $\omega_j = a_1$ are seen to *strongly agree* with the positive class, those with $\omega_j = a_2$ *agree* (but not so strongly), while $\omega_j = -a_1$ $(-a_2)$ *strongly disagree* (*disagree*).

As an illustration, let us consider the well-known German credit dataset, `german`, which is one of the datasets used in our computational tests in Section 4, with 63 features in total. This is a credit scoring dataset, with *good* customers defining the positive class, and has been used in the context of Supervised Classification and interpretability, such as in Baesens et al. (2003) where rule extraction techniques are studied. Table 5 displays the DILSVM classifier with $\mathcal{A} = \{-1, 0, 1\}$ and $C/n = 10^6$ as a collection of 3-point Likert scales. Let us focus on three of the features, namely 'having no debtor', 'having a co-applicant' and 'having a guarantor'. The only relevant feature is 'having a guarantor', strongly contributing to being a good customer, while for the other two features the score is equal to zero. This is a pattern that can be extended to the overall classifier, where more than half of the features are irrelevant (41 out 63). The remaining features are roughly equally split between the left and right side of the scale. Thus, in addition to the gain in interpretability, this DILSVM classifier gains in sparsity too. The 5-point Likert scale representation of the DILSVM classifier with $\mathcal{A} = \{-1, -1/2, 0, 1/2, 1\}$ and $C/n = 10^6$ is given in Table 6. The feature 'having a guarantor' still strongly contributes to being a good customer, but 'having a co-applicant' now contributes negatively (but not strongly), while 'having no debtor' is still irrelevant. Similarly to the 3-point Likert scale classifier, the split between the left and right side of the scale is balanced, but the number of irrelevant features is about a third (27 out of 63).

Apart from the gain in interpretability and sparsity, once the DILSVM classifier has been obtained, its evaluation (i.e., classifying new objects) is as inexpensive as for the SVM. However, these advantages come with an increased computational burden, related to the choice of the set $\mathcal{A}$, to build the classifier. We analyze this issue in the next section.

## 3 Constructing the DILSVM classifier

Inspired by Likert scales, we assume that the set $\mathcal{A}$ is symmetric and defined as $\mathcal{A} = \{-a_1, \ldots, -a_K, 0, a_K, \ldots, a_1\}$, where $a_1 > \ldots > a_K > 0$ are the so-called rating levels telling us about the extent to which each feature is in agreement with the positive class. We denote this model by DILSVM$^{(K)}$.

Our model involves $K+1$ parameters, namely the $K$ rating levels as well as the tradeoff parameter $C$. In the following, we formulate the DILSVM$^{(K)}$, when the $K+1$ parameters are fixed, as an MILP problem. As we will illustrate in Section 4, tuning efficiently the parameters is a necessary and challenging problem, Carrizosa et al. (2014). In order to alleviate this computational burden, a collection of strategies is proposed.

## 3.1 An MILP formulation

In this section we formulate (1)–(5) with $\mathcal{A} = \{-a_1, \ldots, -a_K, 0, a_K, \ldots, a_1\}$ as an MILP problem. For each feature $j$ and each rating level $a_k$, let $\alpha_{jk}$ be equal to either $-1, 1$ or $0$, indicating whether $\omega_j$ is equal to $-a_k$, $a_k$ or none of these two. For each feature $j$, at most one $\alpha_{jk}$ variable can be different from zero. We can now rewrite

$$\begin{aligned} \omega_j &= \sum_{k=1}^{K} a_k \alpha_{jk} \\ \sum_{j=1}^{d} \omega_j^2 &= \sum_{j=1}^{d}\sum_{k=1}^{K} a_k^2 \alpha_{jk}^2 = \sum_{j=1}^{d}\sum_{k=1}^{K} a_k^2 |\alpha_{jk}|, \end{aligned}$$

where the latter follows from the fact that $\alpha_{jk}\alpha_{jk'} = 0$ for $k \neq k'$. It is straightforward to see that by making these substitutions in (1)–(5) and adding the constraints relating to $\alpha_{jk}$, the DILSVM$^{(K)}$ can be formulated as:

$$\min_{\alpha,b,\xi} \frac{1}{2}\sum_{j=1}^{d}\sum_{k=1}^{K} a_k^2 |\alpha_{jk}| + \frac{C}{n}\sum_{i=1}^{n} \xi_i$$

s.t.

$$\begin{aligned} y_i\left(\sum_{j=1}^{d}\sum_{k=1}^{K} a_k \alpha_{jk} x_{ij} + b\right) &\geq 1 - \xi_i & \forall i = 1, \ldots, n \\ \sum_{k=1}^{K} |\alpha_{jk}| &\leq 1 & \forall j = 1, \ldots, d \\ \alpha_{jk} &\in \{-1,\ 0,\ 1\} & \forall j = 1, \ldots, d,\ \forall k = 1, \ldots, K \\ b &\in \mathbb{R} & \\ \xi_i &\geq 0 & \forall i = 1, \ldots, n. \end{aligned}$$

Using the usual trick to transform an absolute value into linear constraints, i.e., $\alpha_{jk} = \alpha_{jk}^{+} - \alpha_{jk}^{-}$, and $|\alpha_{jk}| = \alpha_{jk}^{+} + \alpha_{jk}^{-}$, with $\alpha_{jk}^{+}, \alpha_{jk}^{-} \in \{0, 1\}$, we can reformulate the DILSVM$^{(K)}$ as an MILP problem:

$$\min_{\alpha^+,\alpha^-,b,\xi} \frac{1}{2}\sum_{j=1}^{d}\sum_{k=1}^{K} a_k^2 (\alpha_{jk}^{+} + \alpha_{jk}^{-}) + \frac{C}{n}\sum_{i=1}^{n} \xi_i \tag{6}$$

s.t. $\qquad (\mathrm{DILSVM}^{(K)})$

$$y_i\left(\sum_{j=1}^{d}\sum_{k=1}^{K} a_k(\alpha_{jk}^{+}-\alpha_{jk}^{-})x_{ij}+b\right)\geq 1-\xi_i \qquad \forall i=1,\ldots,n \tag{7}$$

$$\sum_{k=1}^{K}(\alpha_{jk}^{+}+\alpha_{jk}^{-})\leq 1 \qquad \forall j=1,\ldots,d \tag{8}$$

$$\alpha_{jk}^{+},\alpha_{jk}^{-}\in\{0,\ 1\} \qquad \forall j=1,\ldots,d,\ \forall k=1,\ldots,K \tag{9}$$

$$b\in\mathbb{R} \tag{10}$$

$$\xi_i\geq 0 \qquad \forall i=1,\ldots,n. \tag{11}$$

This MILP formulation has $n+d$ constraints, $2dK$ binary decision variables, $n$ nonnegative and 1 free, while the SVM formulation has a quadratic objective function but all the decision variables are continuous.

In terms of the choice of the number of rating levels, the DILSVM$^{(K)}$ may lead to a higher classification accuracy than the DILSVM$^{(K')}$, with $K > K'$, but it is computationally more expensive as the number of zero–one decision variables increases and the gains in interpretability and sparsity are less dramatic. Similar observations can be made when comparing the SVM and the DILSVM$^{(K)}$, since at the end of the spectrum is the SVM, which can be seen as DILSVM$^{(\infty)}$. In the tradeoff between accuracy and interpretability, we recommend the $K = 1$ and $K = 2$ versions of the model, namely the DILSVM$^{(1)}$ and the DILSVM$^{(2)}$.

## 3.2 Reduction Strategies

We have formulated the DILSVM$^{(K)}$ as an MILP problem. Solving this MILP formulation to optimality is an $\mathcal{NP}$-Hard task even if we only consider one single rating level, namely $K = 1$ and $a_1 = 1$, and $C = \infty$, as shown in Chevaleyre et al. (2013). In addition, the performance of the DILSVM$^{(K)}$ classifier may be strongly influenced by the choice of the tradeoff parameter $C$ as well as the $K$ rating levels. Thus, building the DILSVM classifier involves solving a series of MILPs, which, if solved to optimality, may make the overall computational cost high for large datasets. In this section we present three different strategies to alleviate the computational burden of building the DILSVM$^{(K)}$ classifier.

The proposed strategies use the guidance of related but simpler optimization models, and reduce the computational cost associated with each parameter vector and/or the number of parameter vectors to be inspected. The first strategy is based on rounding the SVM classifier using each vector of rating levels. The second strategy proposes, for each parameter vector, the randomized rounding, Raghavan and Tompson (1987), of the Linear Programming (LP) relaxation of the DILSVM$^{(K)}$. The third strategy aims at speeding up the process for the DILSVM$^{(K)}$ based on information readily available for the DILSVM$^{(K')}$, $K' < K$. Clearly, these strategies are of diverse nature and, as the computational experience will illustrate, offer different tradeoffs between performance and reduction in computational cost.

The first and second strategies will be presented for general $K$, while, and for the sake of clarity, the third strategy will be described for $K = 2$, though the process gracefully

```
Step 1. For each $C$,
        (i) Solve the $SVM$ and obtain the (partial) optimal solution
            $(\omega, b)$.
        (ii) For each $(a_1, \dots, a_K)$,
                 For $j = 1, \dots, d$
                     For $k = 1, \dots, K$, set $\beta^+_{jk} = \beta^-_{jk} = 0$
                     For $k = 1, \dots, K$
                         If $a_k \leq \omega_j < a_{k-1}$ then $\beta^+_{jk} = 1$
                         ElseIf $-a_{k-1} < \omega_j \leq -a_k$ then $\beta^-_{jk} = 1$
                     end
                 end
                 Return the classifier $\sum_{k=1}^{K} a_k(\beta^+_k - \beta^-_k)^\top x + b$.
Step 2. Choose the optimal parameter vector using $\sum_{k=1}^{K} a_k(\beta^+_k - \beta^-_k)^\top x + b$.
```

**Figure 1:** Pseudocode for Strategy 1, the *SVM rounding* strategy.

extends to arbitrary $K$. Apart from a grid of parameter vectors, we assume that we are given a selection criterion, Carrizosa and Romero Morales (2013), to choose the optimal classifier among those generated for each vector of the grid. This is usually the accuracy on an independent sample different from the one used to build the classifier, see Section 4.1 for more details.

The first strategy, the *SVM rounding* (*RSVM*), is based on rounding the feature scores of the SVM classifier using the rating levels. For each value of $C$, the SVM classifier is obtained, and the rounding procedure is performed using each vector of rating levels. The pseudocode of this reduction strategy is given in Figure 1, where $a_0 = \infty$. While this strategy examines all the parameter vectors in the grid, it is appealing since it is cheaper to train an SVM than a DILSVM. In addition, for a given value of $C$, once the SVM classifier has been obtained, the rounding procedure takes $\mathcal{O}(dK)$ time for each vector of rating levels.

The second strategy is based on the randomized rounding of the LP relaxation of the DILSVM$^{(K)}$. We call this the *randomized rounding* (*RR*) strategy. For each parameter vector $(C, a_1, \dots, a_K)$, the *RR* strategy solves the LP relaxation of the DILSVM$^{(K)}$, where constraints (9) are relaxed to $\alpha^+_{jk}, \alpha^-_{jk} \in [0, 1]$. Let $(\alpha^+, \alpha^-)$ be the (partial) optimal solution obtained. Without loss of optimality, $\alpha^+_{jk} \cdot \alpha^-_{jk} = 0$. Thus, for a given feature $j$, $\alpha^+_{jk}$ ($\alpha^-_{jk}$) can be seen as the desirability of setting the score of feature $j$ to value $a_k$ ($-a_k$). Noting that the assignment constraints (8) are satisfied, a randomized rounding procedure can be applied to derive the DILSVM classifier. In order to ensure feasibility with respect

```
Step 1. For each $(C, a_1, \dots, a_K)$,

    (i) Solve the LP relaxation of DILSVM$^{(K)}$ and obtain the (partial)
        optimal solution $(\alpha^+, \alpha^-)$.
    (ii) For $j = 1, \dots, d$
             For $k = 1, \dots, K$, set $\beta^+_{jk} = \beta^-_{jk} = 0$
             Set $\mathcal{K} = \{1, \dots, K\}$
             while $(\mathcal{K} \neq \emptyset)$
                 Let $\bar{k}$ such that $\max\{\alpha^+_{j\bar{k}}, \alpha^-_{j\bar{k}}\} \geq \max\{\alpha^+_{jk}, \alpha^-_{jk}\}, \forall k \in \mathcal{K}$
                 Set
                         $\beta^+_{j\bar{k}} = \texttt{rand}(\alpha^+_{j\bar{k}})$
                         $\beta^-_{j\bar{k}} = \texttt{rand}(\alpha^-_{j\bar{k}})$
                 If $\beta^+_{j\bar{k}} = \beta^-_{j\bar{k}} = 0$, set $\mathcal{K} = \mathcal{K} \setminus \{\bar{k}\}$
                 Else $\mathcal{K} = \emptyset$
             end
         end
    (iii) Return the classifier $\sum_{k=1}^{K} a_k (\beta^+_k - \beta^-_k)^\top x + b$.

Step 2. Choose the optimal parameter vector using $\sum_{k=1}^{K} a_k (\beta^+_k - \beta^-_k)^\top x + b$.
```

**Figure 2:** Pseudocode for Strategy 2, the *randomized rounding* strategy.

to the assignment constraints, the rounding needs to take place in a predetermined order of the rating levels, such that once a rating level and a sign has been assigned to a feature, we move to the next feature. In the current version of the *RR* strategy, the rating levels are arranged in decreasing order of $\max\{\alpha^+_{jk}, \alpha^-_{jk}\}$. The pseudocode of this reduction strategy can be found in Figure 2, where $\texttt{rand}(p)$ is a subroutine of random numbers generation, returning the value 1 with probability p and 0 otherwise.

In the third strategy, the *fixing* strategy, we use the output of the DILSVM$^{(1)}$ classifier to alleviate the burden of building the DILSVM$^{(2)}$ classifier. In this case, the reduction is twofold. First, the size of the parameter space is narrowed down. Second, some of the decision variables in the MILP formulation (6)–(11) are fixed in advanced, and therefore eliminated. The pseudocode for this reduction strategy can be found in Figure 3. Using the grid corresponding to parameters $C$ and $a_1$, we first build the DILSVM$^{(1)}$ classifier yielding optimal parameters values $C^{(1)}$ and $a_1^{(1)}$. We then build the DILSVM$^{(2)}$ classifier using the MILP formulation (6)–(11) for $K = 2$, with $C = C^{(1)}$, $a_1 = a_1^{(1)}$ and the values of $a_2$ in the grid. In addition, we reduce the number of zero–one decision variables in the MILP formulation. In the current version of the *fixing* strategy, feature $j$ will have a strong positive rating in the DILSVM$^{(2)}$ classifier if that was the case in the DILSVM$^{(1)}$ classifier, and similarly for a strong negative rating. For the rest of features, $\beta_{jk}^+$ and $\beta_{jk}^-$, $k = 1, 2$, must be optimized.

As already mentioned, the *fixing* strategy can be extended to an arbitrary $K$, where we use the output of the DILSVM$^{(K')}$ classifier with $K' < K$.

```
Step 1. For each $(C, a_1)$,
        (i)  Solve the DILSVM$^{(1)}$ and obtain the (partial) optimal solution
             $(\alpha^+, \alpha^-, b)$.
        (ii) Choose the optimal parameter vector $(C^{(1)}, a_1^{(1)})$ using $a_1(\alpha^+ - \alpha^-)^\top x + b$, and obtain $(\bar{\alpha}^+, \bar{\alpha}^-)$.
Step 2. For $C^{(1)}, a_1^{(1)}$, and for each $a_2$,
        (i)  For $j = 1, \ldots, d$
                     If $\bar{\alpha}_{j1}^+ = 1$ then $\beta_{j1}^+ = 1$ and $\beta_{j1}^- = \beta_{j2}^+ = \beta_{j2}^- = 0$
                     ElseIf $\bar{\alpha}_{j1}^- = 1$ then $\beta_{j1}^- = 1$ and $\beta_{j1}^+ = \beta_{j2}^+ = \beta_{j2}^- = 0$
             end
        (ii) Solve the DILSVM$^{(2)}$ and return the classifier $\sum_{k=1}^{2} a_k(\beta_k^+ - \beta_k^-)^\top x + b$.
Step 3. Choose the optimal parameter vector using $\sum_{k=1}^{2} a_k(\beta_k^+ - \beta_k^-)^\top x + b$.
```

**Figure 3:** Pseudocode for Strategy 3, the *fixing* strategy.

## 4 Computational Results

In this section we illustrate the performance of the DILSVM classifier in terms of accuracy and sparsity, where the latter is measured as the percentage of features with zero score. As benchmark procedure we use the SVM, whose performance in terms of both criteria is reported in Table 2. To build the DILSVM classifier, we use four approaches, namely the MILP approach and the three reduction strategies. We will show that, considered in its full generality and with small values of $K$ ($K = 1, 2$) to ensure interpretability, the DILSVM$^{(K)}$ is competitive against the SVM in terms of accuracy, being substantially sparser than the latter. Inspired by the model in Chevaleyre et al. (2013), we show in Table 3 results for the DILSVM$^{(1)}$ with fixed parameters $C = \infty$ and $a_1 = 1$. Comparing the DILSVM$^{(1)}$ with it shows the relevance of tuning parameters. The results for the DILSVM$^{(1)}$ are reported in Table 3, where for each dataset and each criterion, we underline the best results across the three approaches. (Note that the *fixing* strategy only applies to $K \geq 2$, and therefore it is not present in Table 3.) Similarly, Table 4 reports the results for the four approaches to build the DILSVM$^{(2)}$ classifier.

Our experiments have been conducted on a PC with an Intel® Core™ i7 processor, 16 Gb of RAM. We use the optimization engine CPLEX v12.4, (CPLEX 2012), for solving all optimization problems. We have set the time limit to 300 seconds, which is enough for most of the optimization problems we have solved. For the remaining ones, the classifiers derived in this way are of heuristic nature. The rest of this section is structured as follows. The tuning procedure used to choose the tradeoff parameter $C$ and the rating levels $a_k,\ k = 1, \ldots, K$, is given in Section 4.1. The datasets used to compare the models are described in Section 4.2. The computational results are presented in Sections 4.3 and 4.4.

## 4.1 Parameters Setting

As customary in Supervised Classification, the construction of the DILSVM classifier calls for tuning some parameters, namely the tradeoff parameter $C$ as well as the rating levels. The tuning procedure works as follows, (Carrizosa et al. 2014, Carrizosa and Romero Morales 2013). The dataset is split into three sets, the so-called training, testing and validation sets. For each vector of parameters, the DILSVM is run on the training set, which has size $n$. The different classifiers built in this way are compared according to their accuracy on the testing set. The vector of parameters with the highest accuracy on the testing set is chosen, and its accuracy and sparsity on the validation set are reported, see Figure 4.

Following the usual approach, for the DILSVM$^{(1)}$, parameters $C$ and $a_1$ are tuned by inspecting a grid of the form $\frac{C}{n} \in \{10^{-6}, \ldots, 10^{6}\}$ and of the form $a_1 \in \{2^0, \ldots, 2^{10}\}$. In order to show the relevance of tuning parameters, results for the DILSVM$^{(1)}$ with $C = \infty$ and $a_1 = 1$ are also resported in Section 4.3. For the DILSVM$^{(2)}$, $C$ and $a_1$ are tuned with the same grid, and $a_2 \in \{\frac{a_1}{2}, \frac{a_1}{2^2}\}$.

## 4.2 Datasets

The performance, in terms of accuracy and sparsity, of our model is illustrated using 10 real-life large datasets: `calhous` from the StatLib repository (Vlachos and Meyer 1989), `ijcnn1` and `cod-rna` from the LIBSVM repository (Chang and Lin 2011), and the remaining ones from the UCI repository (Blake and Merz 1998). Datasets containing categorical features have been transformed splitting the categories into binary features (`mushroom`, `german`). Regression datasets are transformed into 2-class datasets using the median (`abalone`, `calhous`), and multi-class datasets are transformed into 2-class, treating the largest class as class $+1$ and the remaining classes as class $-1$ (`careval`, `shuttle`). Please note the features are normalized and therefore we can safely assume that there is one single rating scale for all the features, and thus for the scores.

A description of these datasets can be found in Table 1, whose first four columns report the dataset name, full name given in the repository, number of features ($d$) and total size of the dataset ($|\Omega|$). The size of the training set ($n$) is set as the closest $5 \cdot 10^{s-1}$

```
Step 1. For each value of the parameter vector $(C, a_1, \ldots, a_K)$ in the grid,
    (i) Obtain the corresponding classifier $\sum_{k=1}^{K} a_k(\alpha_k^+ - \alpha_k^-)^\top x + b$ using
        the training set.
    (ii) Let $\pi^{\text{test}}(C, a_1, \ldots, a_K)$ be the accuracy of $\sum_{k=1}^{K} a_k(\alpha_k^+ - \alpha_k^-)^\top x + b$ in
        the testing set.
Step 2. Let $(\bar{C}, \bar{a}_1, \ldots, \bar{a}_K) \in \arg\max_{(C,a_1,\ldots,a_K)} \pi^{\text{test}}(C, a_1, \ldots, a_K)$ and $\sum_{k=1}^{K} \bar{a}_k(\bar{\alpha}_k^+ -$
    $\bar{\alpha}_k^-)x^\top + b$ be the corresponding classifier using the training set.
    (i) Report the performance of $\sum_{k=1}^{K} \bar{a}_k(\bar{\alpha}_k^+ - \bar{\alpha}_k^-)^\top x + b$ in the validation
        set.
```

**Figure 4:** Pseudocode of the tuning procedure for the DILSVM$^{(K)}$.

multiple to $|\Omega|/2$ where $|\Omega|$ is of $10^s$ order, see fifth column of Table 1, and the remaining records in the dataset are equally split between the testing and validation sets. The last column reports the class split in the training set.

To obtain sharp estimates for the accuracy and the sparsity, repeated random subsampling is used, where ten instances are run for each dataset. The ten instances differ in the seed used to reshuffle the dataset and then obtain different training, testing and validation sets.

## 4.3 Results for the MILP approach

In this section we compare the performance of the DILSVM$^{(1)}$ and the DILSVM$^{(2)}$ against that of the SVM, where the DILSVM results are generated using the MILP formulation in Section 3.1. Performance will be measured in terms of two criteria, accuracy, defined as the percentage of objects correctly classified, and sparsity, defined as the percentage of features having a zero score. The *ideal* model would be that one achieving the highest values in both criteria. When, for a given criterion, the difference in performance between two approaches is 1 percentage point (p.p.) or below, we will say that both approaches are comparable under that criterion.

Recall that Table 2 reports the performance of the SVM, while the MILP approach results for the DILSVM$^{(1)}$ can be found in the second set of columns of Table 3, and the ones for the DILSVM$^{(2)}$ in the first set of columns of Table 4. For each dataset, we report the mean validation accuracy across the ten instances, as well as the standard deviation and the median. The same statistics are presented for the sparsity. Below we discuss

Table 1: Datasets.

| Name | Name in Repository | $d$ | $\|\Omega\|$ | $n$ | Class split (in %) |
|---|---|---|---|---|---|
| `adult` | Adult | 123 | 30956 | 15000 | 24/76 |
| `mushroom` | Mushroom | 119 | 8124 | 4000 | 48/52 |
| `german` | Statlog (German Credit Data) | 63 | 1000 | 500 | 30/70 |
| `ijcnn1` | ijcnn1 | 22 | 35000 | 20000 | 9/91 |
| `careval` | Car Evaluation | 21 | 1728 | 1000 | 30/70 |
| `gamma` | MAGIC Gamma Telescope | 10 | 19020 | 10000 | 32/68 |
| `abalone` | Abalone | 10 | 4177 | 2000 | 50/50 |
| `shuttle` | Statlog (Shuttle) | 9 | 58000 | 30000 | 20/80 |
| `cod-rna` | cod-rna | 8 | 59535 | 30000 | 33/67 |
| `calhous` | California Housing | 8 | 20460 | 10000 | 50/50 |

mean values, but similar conclusions are derived if the median is used.

We start with the analysis of the mean accuracy, and show that the DILSVM is competitive against the SVM. We first compare with the mean accuracy of the SVM. For three datasets, `adult`, `german` and `careval`, the DILSVM$^{(1)}$ outperforms the SVM by 5.54, 2.36 and 1.16 p.p., respectively. For three datasets, `mushroom`, `gamma` and `shuttle`, the DILSVM$^{(1)}$ and the SVM are comparable. In three datasets, `ijcnn1`, `abalone` and `calhous`, the SVM outperforms the DILSVM$^{(1)}$ by 1.20, 1.24 and 1.61 p.p., respectively. In `cod-rna`, the DILSVM$^{(1)}$ is clearly inefficient compared to the SVM. For datasets such as `cod-rna`, the DILSVM$^{(1)}$ is too restrictive, and the accuracy may benefit from the additional flexibility built in by the DILSVM$^{(2)}$. An improvement of more than 1 p.p. can be observed from the DILSVM$^{(1)}$ to the DILSVM$^{(2)}$ in three datasets. The first one is `careval`, for which the DILSVM$^{(1)}$ is already better than the SVM, and the improvement on the SVM increases from 1.66 to 3.66 p.p. The second one is `cod-rna`, where now the difference in mean accuracy between the SVM and the DILSVM has been reduced from 15.69 to 3.22 p.p. The third one is `calhous`, where the SVM is better than the DILSVM$^{(1)}$, but now the DILSVM$^{(2)}$ has a comparable performance to the SVM. Across all datasets, the DILSVM$^{(2)}$ outperforms the SVM in three datasets, they are comparable in five datasets, while the SVM outperforms the DILSVM$^{(2)}$ in two datasets. Thus, the DILSVM$^{(2)}$ is competitive against the SVM in terms of mean accuracy. We now show the relevance of tuning parameters by comparing mean accuracy results between the DILSVM$^{(1)}$ and the DILSVM$^{(1)}$ with fixed parameters $C = \infty$ and $a_1 = 1$. Using this criterion, the fixed DILSVM$^{(1)}$ is not competitive and is outperformed by the DILSVM$^{(1)}$ in eight datasets, where the increase in mean accuracy achieved ranges between 2.80 and 18.41 p.p., while the mean accuracy of both methods is basically the same for `mushroom` and `ijcnn1`.

We now focus on the second criterion, and show that the DILSVM is the best in terms of mean sparsity. Indeed, for each dataset except `cod-rna`, the best model is the DILSVM$^{(1)}$, followed by the DILSVM$^{(2)}$, and the SVM being the worse. In `cod-rna`, DILSVM$^{(2)}$ has the highest mean sparsity, then the DILSVM$^{(1)}$ followed by the SVM. Note that the sparsity performance of the SVM is rather poor, being always fully dense (except for `german`), that is, all features have non-zero coefficients, and therefore play a role in the classifier. In terms of mean sparsity, the DILSVM$^{(1)}$ with fixed parameters is clearly outperformed by the DILSVM$^{(1)}$, except for `cod-rna`, in which the mean sparsity is exactly the same for both methods. We now take a closer look at the mean sparsity of our model and show how useful the MILP approach is to determine a high number of irrelevant features which may make the classifier harder to interpret and also may negatively affect accuracy due to overfitting. The mean sparsity for the DILSVM$^{(1)}$ is always 50% or above except for `calhous`, with 31.25%, while for the DILSVM$^{(2)}$ the mean sparsity is always above 35% except for `calhous`, with 17.50%. Thus, except for `calhous`, at least half of the features are irrelevant in the DILSVM$^{(1)}$, while this becomes at least one third in the DILSVM$^{(2)}$. The results are even more encouraging for five of these datasets, where the mean sparsity of the DILSVM$^{(1)}$ is at least 70%, while the mean sparsity of the DILSVM$^{(2)}$ is at least 50%. In addition, in the `mushroom` dataset, both the DILSVM$^{(1)}$ and the DILSVM$^{(2)}$ report a 100% accuracy with a 91.60% mean sparsity.

In summary, the model we propose, the DILSVM, is competitive against the SVM in terms of accuracy, being substantially sparser than the latter. When faced with the choice between one or two rating levels, we should observe that the DILSVM$^{(1)}$ is, in general, more appealing: it has comparable accuracies to those of the DILSVM$^{(2)}$ in seven datasets, while the DILSVM$^{(1)}$ is at least as sparser as the DILSVM$^{(2)}$. Thus, one rating level will, in general, suffice. In the remaining three datasets, including `cod-rna`, the DILSVM$^{(1)}$ underperforms in terms of accuracy, while the additional rating level available in the DILSVM$^{(2)}$ boosts the accuracy to a competitive level, at the cost of a lower sparsity.

The MILP approach to build the DILSVM yields attractive results in terms of both criteria against both benchmarks. That comes with a high computational cost though, where an MILP problem needs to be solved for each parameter vector. This can be illustrated by the median ratio across the ten datasets between the building time of the DILSVM and that of the SVM, which is equal to 210.91 for $K = 1$ and 429.60 for $K = 2$. When comparing with the fixed DILSVM$^{(1)}$, the median ratio is 361.18 for $K = 1$ and 1054.53 for $K = 2$. We should emphasize that this high computational effort only affects the phase of building the classifier (off-line). The evaluation phase (on-line), when new objects are to be classified, is even faster than with the SVM, since, due to the high sparsity of the classifier, fewer terms are computed. In addition, as shown in the next section, the three strategies proposed in Section 3.2 are able to reduce the time to build the DILSVM classifier. In terms of performance, while the sparsity is not compromised, the accuracy will depend on the magnitude of the time reduction.

### 4.4 Results for the Reduction Strategies

In this section we illustrate the performance of the three reduction strategies proposed in Section 3.2, as well as their building times. The accuracy and the sparsity results can be found in Tables 3 ($K = 1$) and 4 ($K = 2$), where we present the mean, the standard deviation and the median of both criteria. Clearly, there is a drop in accuracy with respect to the corresponding MILP approach. Below we show that these strategies behave well against the benchmark. As before, we start with the mean accuracy.

In Strategy 1, the *RSVM* strategy, we round the feature scores of the SVM classifier using the rating levels in the grid. This is a cheap option, in which the optimization problems involved have only continuous decision variables. We now analyze the mean accuracy, and show that the *RSVM* strategy is dominated by the SVM. For $K = 1$, the *RSVM* strategy is clearly inefficient against the SVM, where the loss in accuracy is at least 1 p.p. in all datasets, while this becomes 47.97 p.p. in `mushroom` and 22.81 p.p. in `cod-rna`. The improvement of $K = 2$ is dramatic for `mushroom`, where the *RSVM* strategy becomes comparable to the SVM, also in `ijcnn1`, while in the remaining eight datasets this strategy is still not competitive against the SVM.

In Strategy 2, the *RR* strategy, we reduce the computational cost of solving an MILP problem for each parameter vector. Instead, the *RR* strategy solves the LP relaxation of the DILSVM$^{(K)}$ and applies a randomized rounding to its optimal solution. Below, we show that the *RR* strategy yields mean accuracies close to those of the SVM. For $K = 1$, the loss in accuracy of the *RR* strategy compared to the SVM ranges from 0.51 to 17.73 p.p., and therefore, using this criterion, the *RR* strategy is dominated by the SVM. For $K = 2$, the *RR* strategy outperforms the SVM in `adult` by 4.15 p.p. and in `careval` by 3.71 p.p., for three datasets both methods are comparable, while in the remaining five the losses in accuracy (in p.p.) with respect to the SVM are 1.12 (`ijcnn1`), 1.87 (`shuttle`), 1.88 (`abalone`), 3.12 (`calhous`) and 13.76 (`cod-rna`). Except for `cod-rna`, we can conclude that the *RR* strategy with $K = 2$ and the SVM are comparable in terms of accuracy.

In Strategy 3, the *fixing* strategy, we first build the DILSVM$^{(1)}$, and use its classifier to reduce both the number of parameter vectors to be inspected as well as the number of zero–one decision variables in the MILP. For each dataset, this strategy reduces the number of MILP problems to be solved from 286 (full grid inspected) to 2 (only grid for $a_2$ inspected), after solving the 143 MILPs associated with the DILSVM$^{(1)}$. Roughly speaking, this halves the number of MILPs to be solved. We now analyze the mean accuracy, and show that the *fixing* strategy yields results close to those of the SVM. Compared to the SVM, the *fixing* strategy outperforms the SVM in `adult` by 4.19 p.p. and in `careval` by 1.79 p.p., for three datasets both methods are comparable, while in the remaining five the SVM performs better. If we ignore `cod-rna` with a 12.54 p.p. loss, one can see that the improvement of the SVM for the remaining four datasets is below 1.36 p.p. Except for `cod-rna`, we can conclude that the *fixing* strategy has a comparable behavior to the SVM.

In terms of mean sparsity, the strategies clearly dominate the SVM, as the latter is always fully dense (except for `ijcnn1`). We now take a closer look at the mean sparsity of each strategy. For $K = 1$, the mean sparsity of the *RSVM* strategy is at least 50% for all datasets with eight datasets above 70%, while for $K = 2$ the mean sparsity is above 25% for all datasets with seven above 50%. For $K = 1$, the mean sparsity of the *RR* strategy is at least 65% for all datasets with eight datasets above 70%, while for $K = 2$ the mean sparsity is above 15% for all datasets with eight datasets above 50%. Finally, the sparsity of the *fixing* strategy is at least 20% for all datasets with five datasets above 50%.

We now illustrate the reduction in building time achieved by the strategies. As before, we measure the ratio between the building time of a given strategy and that of the SVM, and report the median ratio across the ten datasets. For the *RSVM* strategy, the rounding time is negligible for small values of $K$. Thus, for $K = 1, 2$, the ratio between the building time of the $RSVM$ and the SVM is roughly equal to 1 for each dataset. For the $RR$ strategy, when comparing with the SVM, the median ratio is equal to 10.32 for $K = 1$ and 34.41 for $K = 2$. Finally, the *fixing* strategy is designed to reduce the building time when $K = 2$ using the DILSVM$^{(1)}$ classifier, and the median ratios are very similar to the ones reported for the MILP approach with $K = 1$, namely 211.60 against the SVM.

In conclusion, we have presented three strategies of very diverse nature that are able to reduce the building time of the DILSVM classifier. When compared to the SVM, the $RR$ and the *fixing* strategies are competitive in terms of mean accuracy, but less so is the $RSVM$, while the dominance of the three strategies in terms of sparsity is overwhelming.

## 5 Conclusions

In this paper we propose the DILSVM$^{(K)}$ classifier, an SVM-type classifier in which there are $K$ possible levels of agreement of each feature with the positive class. We recommend small values of $K$, such as $K = 1$ and $K = 2$. In this case, our classifier enjoys two properties. First, it can be visualized as a collection of Likert scales, and therefore, since $K$ is small, the DILSVM$^{(K)}$ classifier gains in interpretability. Second, once the classifier has been built, the evaluation of the DILSVM$^{(K)}$ (i.e., classifying new objects) is at least as inexpensive as for the SVM: classifying new objects amounts to evaluating a linear function. In addition, as shown by our computational experience, many coefficients are set to zero in both the DILSVM$^{(1)}$ and the DILSVM$^{(2)}$, while the SVM is fully dense.

The gain in visualization and sparsity achieved by the DILSVM is made without paying any price in accuracy. As our computational experience shows, the DILSVM is competitive against the SVM in terms of accuracy. Our classifier is much harder to obtain than the SVM, since an MILP problem is to be solved for each parameter vector, and we have illustrated that parameter tuning is crucial if we do not want to compromise accuracy. In terms of the number of parameters, there are now one ($K = 1$) or two ($K = 2$) more parameters than in the standard SVM to tune. In order to alleviate the computational burden, we propose three reduction strategies. Our computational tests show that we are able to preserve an accuracy comparable to the SVM and significantly better sparsities. This means that, at the expense of an increase in off-line computational cost, the DILSVM is able to extract easy-to-interpret information from datasets without sacrificing classification accuracy.

We conclude with three promising extensions of our approach. First, knowledge domain can also be incorporated into the model. Given a family of features, constraints of the type “at least one feature of this family should be selected” or “no more than one feature of this family should be chosen” simply lead to new linear constraints in the MILP formulation. Second, accuracy and interpretability are usually contradicting objectives. In this paper we fix the number of rating levels, and aim at optimizing accuracy. It is natural to consider the problem of simultaneous optimization of both accuracy and interpretability, see for instance Hooker and Williams (2012), Müssel et al. (2012). This biobjective model deserves further analysis and testing. Third, our approach can also be extended to other linear classifiers, such as the classical Linear Discriminant Analysis (Fisher 1936) and the Logistic Regression (Hastie et al. 2001), where building the new classifier yields Nonlinear Integer Programming problems (Burer and Letchford 2012). Solving these nonlinear problems efficiently remains an important future challenge.

## Acknowledgments

This work has been partially supported by projects MTM2012-36163, Ministerio de Economía y Competitividad, Spain, and FQM-329 of Junta de Andalucía, Spain, both with EU ERD Funds.

## References

Apte, C. 2003. The big (data) dig. *OR/MS Today* **30**(1) 24–29.

Archak, N., A. Ghose, P.G. Ipeirotis. 2011. Deriving the pricing power of product features by mining consumer reviews. *Management Science* **57**(8) 1485–1509.

Baesens, B., R. Setiono, C. Mues, J. Vanthienen. 2003. Using neural network rule extraction and decision tables for credit-risk evaluation. *Management Science* **49**(3) 312–329.

Bertsimas, D., M.V. Bjarnadóttir, M.A. Kane, J.Ch. Kryder, R. Pandey, S. Vempala, G. Wang. 2008. Algorithmic prediction of health-care costs. *Operations Research* **56**(6) 1382–1392.

Bertsimas, D., A. Chang, C. Rudin. 2011. ORC: Ordered rules for classification. A discrete optimization approach to associative classification. Tech. Rep. OR 386-11, Massachusetts Institute of Technology.

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

Brooks, J.P. 2011. Support vector machines with the ramp loss and the hard margin loss. *Operations Research* **59**(2) 467–479.

Burer, S., A.N. Letchford. 2012. Non-convex mixed-integer nonlinear programming: A survey. *Surveys in Operations Research and Management Science* **17**(2) 97–106.

Carrizosa, E., B. Martín-Barragán, D. Romero Morales. 2008. Multi-group support vector machines with measurement costs: A biobjective approach. *Discrete Applied Mathematics* **156** 950–966.

Carrizosa, E., B. Martín-Barragán, D. Romero Morales. 2010. Binarized support vector machines. *INFORMS Journal on Computing* **22**(1) 154–167.

Carrizosa, E., B. Martín-Barragán, D. Romero Morales. 2014. A nested heuristic for parameter tuning in support vector machines. *Computers and Operations Research* **43**(0) 328–334.

Carrizosa, E., D. Romero Morales. 2013. Supervised classification and mathematical optimization. *Computers and Operations Research* **40** 150–165.

Cecchini, M., H. Aytug, G.J. Koehler, P. Pathak. 2010. Detecting management fraud in public companies. *Management Science* **56**(7) 1146–1160.

Chang, C.C., C.J. Lin. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* **2** 1–27.

Chaovalitwongse, W. A., Y.-J. Fan, R. C. Sachdeo. 2008. Novel optimization models for abnormal brain activity classification. *Operations Research* **56**(6) 1450–1460.

Chevaleyre, Y., F. Koriche, J.-D. Zucker. 2013. Rounding methods for discrete linear classification. S. Dasgupta, D. McAllester, eds., *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, vol. 28. JMLR Workshop and Conference Proceedings, 651–659.

CPLEX, ILOG. 2012. www.ilog.com/products/cplex.

Cristianini, N., J. Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

Cui, G., M.L. Wong, H.-K. Lui. 2006. Machine learning for direct marketing response models: Bayesian networks with evolutionary programming. *Management Science* **52**(4) 597–612.

Fang, X., O.R. Liu Sheng, P. Goes. 2013. When is the right time to refresh knowledge discovered from data? *Operations Research* **61**(1) 32–44.

Fisher, R.A. 1936. The use of multiple measurements in taxonomy problems. *Annals of Eugenics* **7** 179–188.

Gage, B. F., A. D. Waterman, W. Shannon, M. Boechler, M.W. Rich, M.J. Radford. 2001. Validation of clinical classification schemes for predicting stroke: Results from the national registry of atrial fibrillation. *Journal of the American Medical Association* **285**(22) 2864–2870.

Golea, M., M. Marchand. 1993. On learning perceptrons with binary weights. *Neural Computation* **5**(5) 767–782.

Guyon, I., J. Weston, S. Barnhill, V. Vapnik. 2002. Gene selection for cancer classification using support vector machines. *Machine Learning* **46** 389–422.

Hand, H., H. Mannila, P. Smyth. 2001. *Principles of Data Mining*. MIT Press.

Hastie, T., R. Tibshirani, J. Friedman. 2001. *The Elements of Statistical Learning*. Springer, New York.

Hooker, J.N., H.P. Williams. 2012. Combining equity and utilitarianism in a mathematical programming model. *Management Science* **58**(9) 1682–1693.

Keeney, R.L., R.S. Gregory. 2005. Selecting attributes to measure the achievement of objectives. *Operations Research* **53**(1) 1–11.

Letham, B., C. Rudin, T.H. McCormick, D. Madigan. 2012. Building interpretable classifiers with rules using bayesian analysis. Tech. Rep. tr609, University of Washington.

Li, F., Y. Yang, E. Xing. 2006. From Lasso regression to feature vector machine. Y. Weiss, B. Schölkopf, J. Platt, eds., *Advances in Neural Information Processing Systems*, vol. 18. MIT Press, Cambridge, MA, 779–786.

Lip, G.Y.H., R. Nieuwlaat, R. Pisters, D.A. Lane, H.J.G.M. Crijns. 2010. Refining clinical risk stratification for predicting stroke and thromboembolism in atrial fibrillation using a novel risk factor-based approach: The Euro heart survey on atrial fibrillation. *CHEST Journal* **137**(2) 263–272.

Liu, H., F. Hussain, C. Tan, M. Dash. 2002. Discretization: An enabling technique. *Data Mining and Knowledge Discovery* **6**(4) 393–423.

Martens, D., B. Baesens, T.V. Gestel, J. Vanthienen. 2007. Comprehensible credit scoring models using rule extraction from support vector machines. *European Journal of Operational Research* **183**(3) 1466–1476.

Martens, D., F. Provost. 2013. Explaining data-driven document classifications. Technical report. Forthcoming in *MIS Quarterly*.

Müssel, C., L. Lausser, M. Maucher, H.A. Kestler. 2012. Multi-objective parameter selection for classifiers. *Journal of Statistical Software* **46**(5).

Orsenigo, C., C. Vercellis. 2003. Multivariate classification trees based on minimum features discrete support vector machines. *IMA Journal of Management Mathematics* **14**(3) 221–234.

Orsenigo, C., C. Vercellis. 2004. Discrete support vector decision trees via tabu search. *Computational Statistics and Data Analysis* **47**(2) 311–322.

Raghavan, P., C. D. Tompson. 1987. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica* **7**(4) 365–374.

Romero Morales, D., J. Wang. 2009. A parallel discretization algorithm for cancellation rate forecasting in revenue management. Working Paper, Saïd Business School, University of Oxford, UK.

Romero Morales, D., J. Wang. 2010. Forecasting cancellation rates for services booking revenue management using data mining. *European Journal of Operational Research* **202**(2) 554–562.

Roth, V. 2004. The generalized lasso. *IEEE Transactions on Neural Networks* **15**(1) 16–28.

Saar-Tsechansky, M., P. Melville, F. Provost. 2009. Active feature-value acquisition. *Management Science* **55**(4) 664–684.

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

Vapnik, V. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag.

Vapnik, V. 1998. *Statistical Learning Theory*. Wiley.

Vlachos, P., M. Meyer. 1989. StatLib. http://lib.stat.cmu.edu.

Wu, X., V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, P.S. Yu, Z.-H. Zhou, M. Steinbach, D.J. Hand, D. Steinberg. 2007. Top 10 algorithms in data mining. *Knowledge and Information Systems* **14** 1–37.

Table 2: Validation performance in % with $C/n \in \{10^{-6}, \dots, 10^{6}\}$: the SVM.

| *Name* | *SVM* | | | | | |
|---|---|---|---|---|---|---|
| | Accuracy | | | Sparsity | | |
| | mean | std | med | mean | std | med |
| **adult** | 84.89 | 0.28 | 84.98 | 0.00 | 0.00 | 0.00 |
| **mushroom** | 99.98 | 0.04 | 100.00 | 0.00 | 0.00 | 0.00 |
| **german** | 72.96 | 2.37 | 72.60 | 4.12 | 1.05 | 3.97 |
| **ijcnn1** | 91.37 | 0.36 | 91.33 | 0.00 | 0.00 | 0.00 |
| **careval** | 95.05 | 0.86 | 94.91 | 0.00 | 0.00 | 0.00 |
| **gamma** | 79.32 | 0.57 | 79.47 | 0.00 | 0.00 | 0.00 |
| **abalone** | 79.52 | 1.14 | 79.73 | 0.00 | 0.00 | 0.00 |
| **shuttle** | 98.17 | 0.08 | 98.15 | 0.00 | 0.00 | 0.00 |
| **cod-rna** | 93.86 | 0.12 | 93.89 | 0.00 | 0.00 | 0.00 |
| **calhous** | 83.59 | 0.29 | 83.59 | 0.00 | 0.00 | 0.00 |

Table 3: Validation performance in % with $C/n \in \{10^{-6}, \dots, 10^{6}\}$: the MILP approach and the reduction strategies for the DILSVM$^{(1)}$.

| *Name* | *MILP* $C=\infty, a_1=1$ | | | | | | *MILP* $a_1 \in \{2^0,\dots,2^{10}\}$ | | | | | | *RSVM* $a_1 \in \{2^0,\dots,2^{10}\}$ | | | | | | *RR* $a_1 \in \{2^0,\dots,2^{10}\}$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Accuracy | | | Sparsity | | | Accuracy | | | Sparsity | | | Accuracy | | | Sparsity | | | Accuracy | | | Sparsity | | |
| | mean | std | med | mean | std | med | mean | std | med | mean | std | med | mean | std | med | mean | std | med | mean | std | med | mean | std | med |
| **adult** | 82.93 | 0.48 | 82.91 | 55.93 | 9.65 | 55.69 | 90.43 | 0.64 | 90.69 | 79.27 | 8.80 | 74.80 | 75.81 | 0.28 | 75.90 | 100.00 | 0.00 | 100.00 | 82.33 | 0.36 | 82.19 | 97.36 | 0.93 | 97.56 |
| **mushroom** | 99.98 | 0.04 | 100.00 | 49.08 | 3.32 | 50.42 | 100.00 | 0.00 | 100.00 | 91.60 | 0.00 | 91.60 | 52.01 | 1.05 | 51.87 | 100.00 | 0.00 | 100.00 | 99.47 | 0.23 | 99.47 | 97.56 | 2.27 | 98.32 |
| **german** | 72.52 | 2.14 | 72.40 | 51.43 | 7.31 | 50.00 | 75.32 | 5.70 | 73.20 | 70.95 | 12.17 | 69.84 | 69.24 | 3.00 | 69.20 | 98.89 | 2.84 | 100.00 | 69.28 | 2.72 | 68.80 | 93.02 | 6.49 | 94.05 |
| **ijcnn1** | 90.18 | 0.36 | 90.03 | 81.82 | 22.27 | 100.00 | 90.17 | 0.36 | 90.03 | 100.00 | 0.00 | 100.00 | 90.18 | 0.36 | 90.03 | 98.18 | 3.64 | 100.00 | 90.17 | 0.37 | 90.04 | 96.59 | 4.22 | 100.00 |
| **careval** | 86.92 | 2.47 | 86.81 | 30.00 | 15.21 | 33.30 | 96.21 | 1.98 | 96.84 | 59.52 | 5.73 | 61.90 | 82.03 | 5.32 | 84.48 | 62.86 | 15.18 | 61.90 | 87.61 | 1.62 | 87.78 | 81.19 | 7.18 | 80.95 |
| **gamma** | 70.37 | 0.60 | 70.47 | 31.00 | 5.39 | 30.00 | 79.09 | 0.64 | 79.22 | 50.00 | 10.95 | 45.00 | 77.95 | 0.71 | 77.79 | 80.00 | 0.00 | 80.00 | 78.31 | 0.49 | 78.26 | 73.00 | 4.00 | 75.00 |
| **abalone** | 74.28 | 0.81 | 73.95 | 15.00 | 5.00 | 15.00 | 78.28 | 0.99 | 78.17 | 70.00 | 8.94 | 70.00 | 77.19 | 1.85 | 77.85 | 72.00 | 6.00 | 70.00 | 76.91 | 1.32 | 77.30 | 75.50 | 5.22 | 75.00 |
| **shuttle** | 80.83 | 1.26 | 80.63 | 15.55 | 7.37 | 16.66 | 97.32 | 0.11 | 97.33 | 63.33 | 5.09 | 66.67 | 84.30 | 5.09 | 82.76 | 75.56 | 9.69 | 77.78 | 93.75 | 1.77 | 92.81 | 68.33 | 7.88 | 66.67 |
| **cod-rna** | 66.56 | 0.38 | 66.58 | 87.50 | 0.00 | 87.50 | 78.17 | 0.28 | 78.20 | 87.50 | 0.00 | 87.50 | 71.05 | 3.09 | 71.35 | 88.75 | 3.75 | 87.50 | 76.13 | 1.51 | 75.70 | 93.75 | 0.00 | 93.75 |
| **calhous** | 63.57 | 2.47 | 63.48 | 0.00 | 0.00 | 0.00 | 81.98 | 0.50 | 81.78 | 31.25 | 8.39 | 37.50 | 78.60 | 2.00 | 79.06 | 50.00 | 0.00 | 50.00 | 78.17 | 1.17 | 78.16 | 65.00 | 5.73 | 68.75 |

Table 4: Validation performance in % with $C/n \in \{10^{-6}, \dots, 10^{6}\}$: the MILP approach and the reduction strategies for the DILSVM$^{(2)}$.

| *Name* | *MILP* $a_1 \in \{2^0,\dots,2^{10}\}, a_2 \in \{a_1/2, a_1/2^2\}$ | | | | | | *RSVM* $a_1 \in \{2^0,\dots,2^{10}\}, a_2 \in \{a_1/2, a_1/2^2\}$ | | | | | | *RR* $a_1 \in \{2^0,\dots,2^{10}\}, a_2 \in \{a_1/2, a_1/2^2\}$ | | | | | | *fixing* $C^{(1)}, a_1^{(1)}, a_2 \in \{a_1^{(1)}/2, a_1^{(1)}/2^2\}$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Accuracy | | | Sparsity | | | Accuracy | | | Sparsity | | | Accuracy | | | Sparsity | | | Accuracy | | | Sparsity | | |
| | mean | std | med | mean | std | med | mean | std | med | mean | std | med | mean | std | med | mean | std | med | mean | std | med | mean | std | med |
| **adult** | 90.56 | 0.46 | 90.75 | 77.40 | 11.64 | 78.45 | 77.88 | 2.41 | 77.22 | 90.33 | 7.01 | 90.25 | 89.04 | 1.99 | 89.14 | 93.33 | 8.39 | 95.93 | 89.08 | 1.63 | 88.66 | 67.07 | 19.10 | 73.98 |
| **mushroom** | 100.00 | 0.00 | 100.00 | 91.60 | 0.00 | 91.60 | 99.41 | 0.14 | 99.42 | 95.80 | 0.00 | 95.80 | 100.00 | 0.00 | 100.00 | 100.00 | 0.00 | 100.00 | 100.00 | 0.00 | 100.00 | 91.60 | 0.00 | 91.60 |
| **german** | 75.96 | 5.32 | 74.00 | 55.87 | 22.06 | 55.56 | 70.92 | 3.38 | 71.00 | 57.78 | 20.21 | 47.62 | 73.00 | 4.51 | 72.20 | 54.76 | 20.59 | 46.03 | 71.60 | 2.26 | 71.60 | 35.56 | 10.56 | 36.51 |
| **ijcnn1** | 90.18 | 0.36 | 90.03 | 95.46 | 13.63 | 100.00 | 90.46 | 0.45 | 90.51 | 65.91 | 7.67 | 70.46 | 90.25 | 0.33 | 90.14 | 70.45 | 4.66 | 72.73 | 90.17 | 0.36 | 90.03 | 100.00 | 0.00 | 100.00 |
| **careval** | 98.71 | 0.44 | 98.77 | 35.72 | 2.39 | 35.72 | 89.81 | 3.22 | 89.97 | 43.34 | 6.88 | 42.86 | 98.76 | 0.43 | 98.90 | 53.81 | 30.57 | 38.10 | 96.84 | 0.81 | 97.25 | 51.90 | 12.68 | 61.90 |
| **gamma** | 79.31 | 0.65 | 79.11 | 50.00 | 16.73 | 40.00 | 78.28 | 0.39 | 78.18 | 52.00 | 12.49 | 50.00 | 79.19 | 0.84 | 79.34 | 57.00 | 14.87 | 60.00 | 79.36 | 0.64 | 79.49 | 32.00 | 7.48 | 30.00 |
| **abalone** | 79.05 | 1.07 | 78.77 | 55.00 | 9.22 | 50.00 | 78.28 | 0.76 | 78.03 | 57.00 | 14.87 | 65.00 | 77.64 | 1.36 | 77.62 | 65.00 | 19.62 | 65.00 | 78.43 | 1.05 | 78.44 | 61.00 | 13.75 | 50.00 |
| **shuttle** | 97.59 | 0.27 | 97.51 | 45.55 | 13.57 | 44.44 | 92.16 | 3.77 | 92.16 | 43.33 | 20.76 | 33.33 | 96.30 | 0.83 | 96.29 | 43.33 | 27.87 | 33.33 | 97.32 | 0.09 | 97.34 | 47.78 | 15.76 | 50.00 |
| **cod-rna** | 90.64 | 0.88 | 90.57 | 42.50 | 11.46 | 50.00 | 73.94 | 5.88 | 74.03 | 82.50 | 8.29 | 81.25 | 80.10 | 4.60 | 82.46 | 62.50 | 12.50 | 62.50 | 81.32 | 2.18 | 80.25 | 46.25 | 11.25 | 50.00 |
| **calhous** | 83.03 | 0.24 | 83.09 | 17.50 | 6.12 | 12.50 | 80.14 | 0.84 | 80.23 | 25.00 | 19.36 | 12.50 | 80.47 | 0.64 | 80.50 | 17.50 | 6.12 | 12.50 | 82.25 | 0.44 | 82.12 | 20.00 | 6.12 | 25.00 |

Table 5: 3-point Likert scale for `german` dataset with the DILSVM.

The following features contribute to being a good customer:

| | | Strongly Agree | Neutral | Strongly Disagree |
|---|---|---|---|---|
| Status of existing checking account | $\ldots < 0$ DM | | × | |
| | $0 \leq \ldots < 200$ DM | | × | |
| | $\ldots \geq 200$ DM | × | | |
| | No checking account | × | | |
| Credit duration | | | | × |
| Credit history | no credits taken/all credits paid back duly | | | × |
| | all credits at this bank paid back duly | | | × |
| | existing credits paid back duly till now | | × | |
| | delay in paying off in the past | | × | |
| | critical account/other credits existing (not at this bank) | × | | |
| Purpose | new car | | | × |
| | used car | | × | |
| | furniture/equipment | | × | |
| | radio/television | | × | |
| | domestic appliances | | | × |
| | repairs | | × | |
| | education | | | × |
| | vacation | | × | |
| | retraining | × | | |
| | business | | × | |
| | others | | × | |
| Credit amount | | | | × |
| Saving accounts | $\ldots < 100$ DM | | × | |
| | $100 \leq \ldots < 500$ | | × | |
| | $500 \leq \ldots < 1000$ | × | | |
| | $\ldots \geq 1000$ | × | | |
| | unknown/no saving accounts | × | | |
| Duration of present employment | unemployed | | × | |
| | $\ldots < 1$ year | | × | |
| | $1 \leq \ldots < 4$ years | | × | |
| | $4 \leq \ldots < 7$ years | | × | |
| | $\ldots \geq 7$ years | | × | |
| Installment rate in % of disposable income | | | | × |
| Personal status and sex | male (divorced/separated) | | | × |
| | female (divorced/separated/married) | | × | |
| | male (single) | | × | |
| | male (married/widowed) | | × | |
| | female (single) | | × | |
| Other debtors/guarantors | none | | × | |
| | co-applicant | | × | |
| | guarantor | × | | |
| Duration of present residence | | | × | |
| Property | real estate | | × | |
| | building society savings agreement/life insurance | | × | |
| | car/others | | × | |
| | unknown/no property | | | × |
| Age | | × | | |
| Other installment plans | bank | | × | |
| | stores | | × | |
| | none | | × | |
| Housing | rent | | × | |
| | own | | × | |
| | for free | × | | |
| Number of existing credits at this bank | | | | × |
| Job | unemployed/unskilled (non-resident) | | × | |
| | unskilled (resident) | | × | |
| | skilled employee/official | | × | |
| | management/self-employed/highly qualified employee officer | | × | |
| Number of people being liable to provide maintenance for | | | × | |
| Telephone | None | | × | |
| | Yes | | × | |
| Foreign worker | Yes | | × | |
| | No | × | | |

Table 6: 5-point Likert scale for `german` dataset with the DILSVM.

The following features contribute to being a good customer:

| | | Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree |
|---|---|---|---|---|---|---|
| Status of existing checking account | . . . < 0 DM | | | | × | |
| | 0 ≤ . . . < 200 DM | | | × | | |
| | . . . ≥ 200 DM | × | | | | |
| | No checking account | × | | | | |
| Credit duration | | | | | | × |
| Credit history | no credits taken/all credits paid back duly | | | | | × |
| | all credits at this bank paid back duly | | | | | × |
| | existing credits paid back duly till now | | | × | | |
| | delay in paying off in the past | | × | | | |
| | critical account/other credits existing (not at this bank) | × | | | | |
| Purpose | new car | | | | | × |
| | used car | | | × | | |
| | furniture/equipment | | | | × | |
| | radio/television | | | | × | |
| | domestic appliances | | | | | × |
| | repairs | | | | × | |
| | education | | | | | × |
| | vacation | | | × | | |
| | retraining | × | | | | |
| | business | | | | × | |
| | others | | | × | | |
| Credit amount | | | | | | × |
| Saving accounts | . . . < 100 DM | | × | | | |
| | 100 ≤ . . . < 500 | | × | | | |
| | 500 ≤ . . . < 1000 | × | | | | |
| | . . . ≥ 1000 | × | | | | |
| | unknown/no saving accounts | × | | | | |
| Duration of present employment | unemployed | | | × | | |
| | . . . < 1 year | | | × | | |
| | 1 ≤ . . . < 4 years | | | × | | |
| | 4 ≤ . . . < 7 years | | × | | | |
| | . . . ≥ 7 years | | | × | | |
| Installment rate in % of disposable income | | | | | | × |
| Personal status and sex | male (divorced/separated) | | | | | × |
| | female (divorced/separated/married) | | | | × | |
| | male (single) | | | × | | |
| | male (married/widowed) | | | × | | |
| | female (single) | | | × | | |
| Other debtors/guarantors | none | | | × | | |
| | co-applicant | | | | × | |
| | guarantor | × | | | | |
| Duration of present residence | | | | × | | |
| Property | real estate | | | × | | |
| | building society savings agreement/life insurance | | | × | | |
| | car/others | | | × | | |
| | unknown/no property | | | | | × |
| Age | | × | | | | |
| Other installment plans | bank | | | × | | |
| | stores | | | × | | |
| | none | | × | | | |
| Housing | rent | | × | | | |
| | own | | × | | | |
| | for free | × | | | | |
| Number of existing credits at this bank | | | | | | × |
| Job | unemployed/unskilled (non-resident) | | | × | | |
| | unskilled (resident) | | | × | | |
| | skilled employee/official | | | × | | |
| | management/self-employed/highly qualified employee officer | | | × | | |
| Number of people being liable to provide maintenance for | | | | × | | |
| Telephone | None | | | × | | |
| | Yes | | | × | | |
| Foreign worker | Yes | | | × | | |
| | No | × | | | | |