



Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/eor

Feature selection in linear support vector machines via a hard cardinality constraint: A scalable conic decomposition approach

Immanuel Bomze ^a,*, Federico D'Onofrio ^b, Laura Palagi ^b, Bo Peng ^c

^a Faculty of Mathematics and Research Network Data Science, University of Vienna, Oskar-Morgenstern-Platz 1, 1090 Wien, Austria

^b Department of Computer, Control and Management Engineering, Sapienza University of Rome, Via Ariosto 25, 00185 Rome, Italy

^c VGSCO, University of Vienna, Oskar-Morgenstern-Platz 1, 1090, Wien, Austria

ARTICLE INFO

Keywords:

Machine learning
Biclassification
Interpretable AI
Fairness
Zero-norm

ABSTRACT

In this paper, we study the embedded feature selection problem in linear Support Vector Machines (SVMs), in which a cardinality constraint is employed, leading to an interpretable classification model. The problem is NP-hard due to the presence of the cardinality constraint, even though the original linear SVM amounts to a problem solvable in polynomial time. To handle the hard problem, we first introduce two mixed-integer formulations for which novel semidefinite relaxations are proposed. Exploiting the sparsity pattern of the relaxations, we decompose the problems and obtain equivalent relaxations in a much smaller cone, making the conic approaches scalable. To make the best usage of the decomposed relaxations, we propose heuristics using the information of its optimal solution. Moreover, an exact procedure is proposed by solving a sequence of mixed-integer decomposed semidefinite optimization problems. Numerical results on classical benchmarking datasets are reported, showing the efficiency and effectiveness of our approach.

1. Introduction

In supervised classification, Machine Learning (ML) models are trained on labeled datasets to learn how to predict the correct output for new, unseen data. Classification algorithms are prone to overfitting when the dataset is small and contains irrelevant features, or noise (Hastie et al., 2009). This problem can also occur with correlated features or features containing errors. Further, interpretability in ML is gaining more importance, and remains a cutting-edge research topic. Indeed, designing models which provide explanatory insights and help end users to understand the rationale behind decisions, is crucial from both predictive and debugging perspective (Kamath & Liu, 2021).

It is therefore beneficial to use a smaller set of features that can give a good approximation of the target function, i.e., the classifier. Feature Selection (FS) aims to eliminate less significant and/or redundant features to identify the possibly optimal subset for the classifier, which becomes crucial when dealing with real data. Among various feature selection techniques, embedded methods integrate feature selection within the classifier training process. In this paper, we study the embedded feature selection problem for SVMs.

From now on, we use lowercase letters either in the sans-serif Latin font, e.g., x , or in the boldface font for Greek letters, e.g., ξ , to indicate vectors. Transposition is denoted by T and the zero vector by \mathbf{o} ; given

two integers a, b with $a \leq b$, we abbreviate by $[a : b] = \{k \in \mathbb{Z} : a \leq k \leq b\}$.

Given a dataset with m samples, and n features, $\{(x^i, y^i) \in \mathbb{R}^n \times \{-1, 1\} : i \in [1 : m]\}$, the linear SVM classification problem defines a classifier as the sign function $f : \mathbb{R}^n \rightarrow \{-1, 1\}$,

$$f(x) = \text{sgn}(\mathbf{w}^{*T} x + b^*),$$

using the structural risk minimization principle (Vapnik, 1999).

In traditional SVMs (Cortes & Vapnik, 1995), coefficients $(\mathbf{w}^*, b^*) \in \mathbb{R}^n \times \mathbb{R}$ identify a separating hyperplane (if it exists) that maximizes its margin, i.e. the distance between the hyperplane and the closest data points of each class. Accounting for classes not linearly separable, the tuple (\mathbf{w}^*, b^*) is found by solving the following convex quadratic problem:

$$\begin{aligned} \text{(SVM)} \quad & \min_{\mathbf{w}, b, \xi} \quad \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & y^i (\mathbf{w}^T x^i + b) \geq 1 - \xi_i, \quad i \in [1 : m], \\ & \xi_i \geq 0. \end{aligned} \quad (1)$$

The non-negative variables ξ_i allow points to be on the wrong side of their “soft margin”, as well as being “too close” to the decision boundary.

* Corresponding author.

E-mail addresses: immanuel.bomze@univie.ac.at (I. Bomze), federico.donofrio@uniroma1.it (F. D'Onofrio), laura.palagi@uniroma1.it (L. Palagi), bo.peng@univie.ac.at (B. Peng).

<https://doi.org/10.1016/j.ejor.2025.03.017>

Received 17 April 2024; Accepted 17 March 2025

Available online 28 March 2025

0377-2217/© 2025 Published by Elsevier B.V.

The sum $\sum_{i=1}^m \xi_i$ in the objective function represents an upper bound on the total misclassification error, which is weighted by the hyperparameter $C > 0$ so to balance the maximization of the margin, which is inversely proportional to $\|w\|_2$. Problem (1) is categorized as ℓ_2 -regularized ℓ_1 -loss SVM (see Fan et al. (2008), Hastie et al. (2004), for instance). When needed, we can also refer to it as ℓ_2 -SVM to highlight the use of the ℓ_2 regularization contrasting with the use of ℓ_1 regularization, as in Hastie et al. (2004), Nguyen and De la Torre (2010), for instance.

The optimal solution w^* of (1) is usually dense, in the sense that usually all components of w^* are nonzero and thus every feature contributes to the definition of the classifier. Indeed, through duality theory, w^* can be expressed as the linear combination of a subset of samples, called support vectors. More precisely, $w^* = \sum_{i=1}^m \alpha_i^* y^i x^i$, with $\alpha_i^* \geq 0$ being the dual optimal solution of problem (1) with nonzero value only if sample i represents a support vector. Thus, it is unlikely to obtain $w_j^* = 0$ for some feature j , and sparsity in the vector w^* must be forced.

In this paper, we aim to strictly control sparsity by imposing hard cardinality constraints on the number of nonzero components in w^* . We approach the feature selection problem from a combinatorial perspective, using binary variables to clearly indicate which features are selected. More specifically, the SVM problem with feature selection we are interested in is:

$$\begin{aligned} \text{(FS-SVM)} \quad \min_{w, b, \xi} \quad & \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & y^i (w^\top x^i + b) \geq 1 - \xi_i, \quad i \in [1 : m], \\ & \|w\|_0 \leq B \\ & \xi \geq 0, \end{aligned} \quad (2)$$

where the ℓ_0 -pseudonorm is defined as $\|w\|_0 := \sum_{j=1}^n |\text{sgn}(w_j)|$. The user-defined parameter $B \in \mathbb{Z}_+$ constitutes a budget on the number of features used, indicating that the number of features involved in the classifier is at most B . Even though the objective function and all other constraints are convex quadratic or linear, the cardinality constraint $\|w\|_0 \leq B$ is of combinatorial type, which renders (2) NP-hard. The class of optimization problems involving cardinality constraint has gained significant attention over the past decades, for a comprehensive overview of such problems, we refer the interested readers to Bertsimas and Shioda (2009), Gao and Li (2013), Kanzow et al. (2021), Tillmann et al. (2024) and references therein.

By strictly controlling the budget value B , we can achieve a sparse, more transparent model which is usually easier to explain. Sparsity enhances interpretability by limiting the number of features, which simplifies the model structure and clarifies its decision-making, allowing clearer insights into predictions (Rudin et al., 2022). A model that selects at most B features offer precise control over sparsity, ensuring alignment with user-defined complexity requirements. Unlike penalty-based methods, such as Lasso regularization, which incorporates a ℓ_1 -norm penalty in the objective function, this approach provides a known feature count *a priori*, enhancing computational efficiency and reducing the need for extensive tuning (Bertsimas & Van Parys, 2020). Additionally, models with cardinality constraints are often generalized to budget-constrained models, where features are selected based on both relevance and cost, creating solutions that are sparse and cost-effective (Labbé et al., 2019; Lee et al., 2020).

Linear SVMs produce a linear classifier, that may cause performance to fall short in certain classification tasks. However, linearity in the original features is particularly valuable when developing explainable and interpretable machine learning models, as the original variables have a direct meaning more easily understood by practitioners, particularly if only a few of them (regulated by B) are used in the classifier. Although other approaches exist for obtaining a dimension reduction, such as Principal Component analysis, see e.g. Greenacre et al. (2022), or Linear Discriminant Analysis (Tharwat et al., 2017), these usually construct new artificial features as linear combinations of the original

ones, which obscure the direct contribution of the original features to classification. Similar considerations apply when comparing linear SVMs to more complex (kernel) models which map original features into high-dimensional feature spaces in a non-linear way which may also impede transparency of the decision model.

The main contributions of our work are:

- We introduce two novel Mixed Integer Quadratic optimization Problem (MIQP) formulations for the FS-SVM problem (2), building upon on known optimization techniques for tackling the ℓ_0 -pseudonorm constraint, either by means of a big-M reformulation or by modeling it with complementarity constraints.
- We present novel relaxations that decompose with respect to small blocks of variables, thus being easily solvable when dealing with datasets with a large number of features.
- We propose both exact and heuristic algorithms that are easily implementable and exploit the proposed relaxations.
- We conduct a battery of extensive numerical experiments that demonstrate the effectiveness of our algorithms in solving the proposed problem compared to off-the-shelf solvers, and validate the model from a machine learning perspective.

1.1. Related work

Embedding feature selection in linear SVM has been proposed in several papers. One of the most frequently used approaches consists in using the ℓ_1 -regularized SVM, the ℓ_1 -SVM, instead of the ℓ_2 -SVM. Indeed, the ℓ_1 -regularization, also known as LASSO regularization, leads to sparser solution and to an easier problem, being formulated as a Linear Programming problem (see Fung and Mangasarian (2004), Hastie et al. (2004), Nguyen and De la Torre (2010), Yuan et al. (2010), for instance).

While replacing the ℓ_2 -regularization by the ℓ_1 -regularization has a significant impact on reducing the computational requirement for solving the optimization problem, the ℓ_1 -SVM leads to relatively reduced classification accuracy due to the inability to maximize the margin of separation between the two classes of data. To overcome the disadvantages, in Wang et al. (2006), Zou and Hastie (2005) the elastic-net regularization technique has been proposed which uses both the ℓ_2 - and the ℓ_1 -regularization in the objective function.

Another approach penalizes the ℓ_0 -pseudonorm of w in the objective function, offering a better approximation than the ℓ_1 -norm. In this framework, (Bradley & Mangasarian, 1998) studied the model in which the ℓ_2 -regularization is substituted with the ℓ_0 -pseudonorm of w . They solve the resulting problem by approximating $\|w\|_0 \approx 1 - e^{-\lambda \|w\|_1}$, where $\lambda \geq 0$. The resulting problem is the minimization of a concave function over a polyhedral set and the authors proposed a successive linearization algorithm to find a local solution. Weston et al. (2003) tackled the same problem as Bradley and Mangasarian (1998) and proposed to solve it by iteratively rescaling the input features values, multiplying them by the absolute value of the weights w_j , obtained from the resolution of the SVM problem, until convergence is reached. Variables can be ranked by removing those features whose weights become zero during the iterative algorithm and computing the order of removal.

None of the approaches described above have a direct and hard control on the number $\|w\|_0$ of nonzero components in the vector w^* like the property $\|w\|_0 \leq B$.

Facing the problem (FS-SVM), Chan et al. (2007) proposed to replace the cardinality constraint by the non-linear constraint $\|w\|_1^2 \leq B \|w\|_2^2$. This non-convex constraint can be relaxed, leading to two convex relaxations of the sparse SVM. However, the resulting optimal solutions seem to be not as strong (Ghaddar & Naoum-Sawaya, 2018), and they violate the imposed desired features limit B . To overcome these drawbacks, Ghaddar and Naoum-Sawaya (2018) proposed an alternative relaxation, adjusting a parameter value iteratively to obtain

improving bounds and employing a bisection algorithm to ensure that $\|w\|_0$ does not exceed the maximum limit B .

Another approach based on bilevel optimization is proposed by Agor and Özaltın (2019), where binary variables in the upper-level problem are used to explicitly control the number of features selected. They proposed a genetic algorithm for the solution of the resulting problem.

Maldonado et al. (2014) proposed a Mixed Integer Linear Optimization Problem (MILP) model, introducing feature selection through a budget constraint to limit the number of features used in the classification process, but focusing just on minimizing the sum of the deviations ξ_i rather than considering also the margin. Extending this work, Labbé et al. (2019) proposed a MILP formulation which considers also the margin maximization, to select suitable features for constructing separating hyperplanes, incorporating the same budget constraint. The authors proposed both heuristic and exact approaches for solving their formulation when applied to large datasets.

As regards the ℓ_2 -SVM, Aytug (2015) proposed two feature selection (hard and soft constrained) models where binary variables act as a multiplicative “mask” on the w , and enter both the objective function and the constraints in a nonlinear way. An exact method based on Generalized Benders Decomposition is proposed to solve the two formulations. However, the experimental results indicate that the algorithm fails to converge in practice for datasets with a large number of features, reaching the preset limit on the decomposition iterations. Another approach employing ℓ_2 -SVMs and cardinality constraints is used in D’Onofrio et al. (2024), where the authors proposed a new MIQP model for building optimal classification trees with SVM sparse splits.

1.2. Paper organization and notation

This paper is structured as follows: in Section 2, two mixed-integer reformulations of the targeted problem (2) are proposed. Several tractable relaxations of these two mixed-integer problems are introduced in Section 3 with a theoretical comparison among them. In the same section, we further propose novel decomposed relaxations, exploiting the sparsity pattern of the problems. Section 4 focuses on algorithmic methods: a heuristic and an exact algorithm are proposed for solving the problem (2), based on the optimal solutions of the relaxations. Applying these algorithms to some benchmark datasets, we report the numerical results in Section 5, from the optimization point of view and from the ML perspective. The concluding Section 6 also discusses possible future research directions.

Throughout the paper, we denote by lower case letters, e.g. x , and by upper case letters, e.g. X , a vector and a matrix, respectively. Especially, \mathbf{e} and \mathbf{o} represent all-ones and all-zeros vectors of proper dimension. The dimension should be clear from the context unless otherwise mentioned. We employ subscripts to indicate a specific element of vectors or matrices. For instance, by x_i we denote the i th element of the vector x , and by X_{ij} we denote the ij -th entry of the matrix X . Furthermore, by $\text{diag } X = [X_{ii}]_{i=1}^n$ we denote the vector of diagonal of X . Finally, by S^n and S_+^n we denote the set of real symmetric $n \times n$ matrices, and the set of symmetric positive-semidefinite matrices of order n , respectively.

Additionally, let us introduce the set

$$\Xi = \{(w, b, \xi) \in \mathbb{R}^{n+1+m} : y^i(w^\top x^i + b) \geq 1 - \xi_i, i \in [1 : m], \xi \geq \mathbf{o}\}, \quad (3)$$

so that problem (2) reads as:

$$\begin{aligned} \min_{w, b, \xi} \quad & \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & (w, b, \xi) \in \Xi \\ & \|w\|_0 \leq B. \end{aligned}$$

In the next sections, we will also make use of big-M reformulations, so it is convenient to introduce the two sets

$$\Omega_M = \{(w, v) \in \mathbb{R}^{2n} : -Mv \leq w \leq Mv, e^\top v = B\} \quad (4)$$

and

$$B_M = \{(w, u) \in \mathbb{R}^{2n} : -M(1-u) \leq w \leq M(1-u), e^\top u = n-B\}, \quad (5)$$

where the big-M parameter $M > 0$ is large enough (see Remark 3.2 below) and the variables $u = 1 - v$ are the negated v , sometimes more convenient in notation.

2. Exact mixed-integer approaches

In this section, we propose two exact mixed-integer formulations of the original problem (2), using two different mixed-integer systems to present the cardinality constraint with auxiliary binary variables introduced as indicators, namely a big-M strategy and a complementarity constraint strategy.

Differently from Labbé et al. (2019), we study the following MIQP model incorporating the standard ℓ_2 -norm on the hyperplane weights as the regularization term:

$$\begin{aligned} (\text{BigMP}) \quad \min_{w, b, \xi, v} \quad & \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & (w, b, \xi) \in \Xi \\ & (w, v) \in \Omega_M \\ & v \in \{0, 1\}^n, \end{aligned} \quad (6)$$

where Ξ and Ω_M are defined in (3) and (4).

To show the equivalence between (2) and (6), we consider a B -sparse w , e.g., $\|w\|_0 \leq B$. If $\|w\|_0 = B$, there exists a $v \in \{0, 1\}^n$ such that $(w, v) \in \Omega_M$, where v is given by the rule that $v_j = 0$ if $w_j = 0$ and $v_j = 1$ if $w_j \neq 0$; If $\|w\|_0 < B$, there also exists a $v \in \{0, 1\}^n$ such that $(w, v) \in \Omega_M$, where v is given by the rule that $v_j = 1$ for $j \in I$ and $v_j = 0$ otherwise, where I is a index set covers all of features with $w_j \neq 0$, satisfying $|I| = B$.

Remark 2.1. It is worth noticing that the optimal value of model (6) remains the same if we substitute the equality constraint $e^\top v = B$ by the budget constraint $e^\top v \leq B$, which is derived directly from the targeted problem (2).

Alternatively, instead of expressing the cardinality constraint by the big-M formulation, we employ a set of binary variables $u \in \{0, 1\}^n$ and a complementarity constraint to count the nonzero elements of w and end up with another mixed-integer reformulation of (2):

$$\begin{aligned} (\text{CoP}) \quad \min_{w, b, \xi, u} \quad & \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & (w, b, \xi) \in \Xi \\ & u_j w_j = 0, \quad j \in [1 : n] \\ & e^\top u = n - B \\ & u \in \{0, 1\}^n. \end{aligned} \quad (7)$$

Unlike (6), the binary variables u_j count the number of zero elements of w . For a (7)-feasible point (w, b, ξ, u) , we let $v = 1 - u$ and the point (w, b, ξ, v) is (6)-feasible. To show this, we only need to show $(w, v) \in \Omega_M$, namely, that (w, v) satisfies the big-M constraint in (4). It is clear that the constraint holds for the coordinates of w with $w_j = 0$. On the other hand, the complementarity constraint $(1 - v_j)w_j = 0$ implies $v_j = 1$ if $w_j \neq 0$, and therefore the point $(w, v) \in \Omega_M$. Hence, the problem (7) is equivalent to the problem (6).

Remark 2.2. In the above problem (7), the constraint $e^\top u = n - B$ can be equivalently replaced by the constraint $e^\top u \geq n - B$ for the same reason as Remark 2.1, and the binary constraint $u_j \in \{0, 1\}$ can be equivalently reduced to the box constraint $0 \leq u_j \leq 1$ due to the existence of the complementarity constraint. To show this, we consider the model with the box constraint and assume that w is ρ -sparse with $\rho > B$. Due to the constraint $u_j w_j = 0$, $u_j = 0$ if w_j is nonzero and $0 \leq u_j \leq 1$ if w_j is zero. Hence, $e^\top u \leq n - \rho < n - B$, which contradicts the budget constraint. Therefore all w vectors with cardinality strictly larger than B are rendered infeasible to the model (7) and its relaxation (8) below.

In the first formulation, we need a big-M parameter M for the reformulation. It is known that the choice of M has a profound influence on the computational performance if we solve the problem with off-the-shell global solvers. In a branch-and-bound framework, we solve continuous relaxation at each node, and in many cases, the continuous relaxation can be as weak as the SVM problem without feature selection, if too large M is chosen, as shown later in [Theorem 3.1](#). In contrast, in the second formulation, the difficulty of the problem mainly goes to the complementarity constraint, and no big-M needed. The computational comparison between these two formulations will be presented in [Section 5](#).

3. Tractable relaxations

In this section, we propose several tractable relaxations, e.g., LP relaxations, and SDP relaxations, of the FS-SVM model, and a comparison is made among some of them. Moreover, a decomposition strategy is proposed by exploring the sparse patterns involved in both the objective and the feasible region of the original problem, leading to a much smaller and scalable SDP-based relaxation.

3.1. Box relaxation

In this subsection, we study the box relaxation (also known as a type of LP relaxation) of the two exact mixed-integer formulations presented in the last chapter and highlight the crucial role that M plays in the first formulation.

After relaxing the binarity constraints to the box constraints in the problem (6), we arrive at:

$$\begin{aligned} \text{(BoxMP)} \quad \min_{w,b,\xi,v} \quad & \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & (w, b, \xi) \in \Xi \\ & (w, v) \in \Omega_M \\ & 0 \leq v \leq e. \end{aligned} \quad (8)$$

The relaxation is a convex quadratic optimization problem solvable in polynomial time, therefore it has been widely used in many previous works, see references in [Labbé et al. \(2019\)](#). However, the tightness of the relaxation is highly dependent on the choice of M and there is a possibility that it can be as weak as the original SVM model without feature selection (1). To be more precise, if M is larger than a data-based threshold, the relaxation (8) is equivalent to the problem (1), as presented in the following theorem.

Theorem 3.1. *If the parameter M in the problem (8) satisfies*

$$M \geq \frac{\|w^*\|_1}{B},$$

where (w^, b^*, ξ^*) is an (1)-optimal solution, then (8) is equivalent to (1).*

Proof. It is clear to see that (1) is a relaxation of (8), as the former includes the constraint $(w, b, \xi) \in \Xi$. Hence the optimal value of (1) is not larger than that of (8). To show the converse, we assume that (w^*, b^*, ξ^*) is optimal to the problem (1). Let $v_j = \frac{|w_j^*|}{M}$ for all $j \in [1:n]$. Then the point (w^*, b^*, ξ^*, v) is (8)-feasible. Indeed, due to the construction of v_j , the constraint $-Mv_j \leq w_j \leq Mv_j$ is satisfied automatically for all $j \in [1:n]$. Since $e^T v = \frac{\|w^*\|_1}{M}$, the constraint $e^T v \leq B$ is implied by the assumption we made. Since the objective functions in both problems coincide, the claim follows. \square

On the other hand, M cannot be chosen smaller than:

$$M \geq \|\bar{w}^*\|_\infty,$$

where \bar{w}^* is part of an optimal solution to (2). Otherwise, the optimal solution will be cut off in the sense that the problem (6) is not a

reformulation for an insufficient choice of M . Hence, the proper range of the parameter M has to be:

$$M \in \left[\|\bar{w}^*\|_\infty, \frac{\|w^*\|_1}{B} \right). \quad (9)$$

However, this set can be empty depending on the problem data itself, which means no matter what M we chose, the box relaxation (8) ends up with one of the following situations: (i) no improvement compared to the original SVM problem (1); (ii) the optimal solution of the targeted problem (2) is cut off, because M is too small to render (8) a reformulation of (2). This is quite problematic in practice and it encourages us to explore the other formulation (7).

Remark 3.2. Since estimating the range of \bar{w}^* is as difficult as solving the problem itself, we usually choose M under a safe threshold, which is much larger than the tightest upper bound. In most practical cases, the condition $M \geq \frac{\|w^*\|_1}{B}$ holds then as well, which means the LP relaxation (8) does not provide any improvement compared to the plain problem (1).

Due to the presence of complementarity constraints, the box relaxation of (7) is equivalent to (7) itself, as mentioned in [Remark 2.2](#). However, the combinatorial nature of a complementarity constraint is the main difficulty in both the unrelaxed and the relaxed version, so this continuous reformulation does not solve the problem immediately without any further processing.

3.2. The Shor relaxation

In this subsection, we study the Shor relaxation, a type of SDP relaxation, of the two proposed mixed-integer reformulations.

Letting $W = ww^T$, $V = vv^T$ and dropping the rank one constraint, the Shor relaxation of (6) reads as:

$$\begin{aligned} \text{(SMP)} \quad \min_{w,W,b,\xi,v,V} \quad & \frac{1}{2} \text{trace}(W) + C \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & (w, b, \xi) \in \Xi \\ & (w, v) \in \Omega_M \\ & \text{diag } V = v \\ & \begin{bmatrix} 1 & w^T \\ w & W \end{bmatrix} \in S_+^{n+1}, \quad \begin{bmatrix} 1 & v^T \\ v & V \end{bmatrix} \in S_+^{n+1}. \end{aligned} \quad (10)$$

Theorem 3.3. *The Shor relaxation (10) is equivalent to the box relaxation (8).*

Proof. For a (10)-feasible point (w, W, b, ξ, v, V) , the projected point (w, b, ξ, v) satisfies all of the constraints of (8). Moreover, we have $\text{trace}(W) \geq \|w\|_2^2$, hence the relaxation (8) is weaker than the relaxation (10). To show the other direction, we suppose that the point (w, b, ξ, v) is feasible to the problem (8), and let $W = ww^T$ and $V = vv^T + Z$, where $Z \in S_+^n$ is a diagonal matrix with $Z_{jj} = v_j - v_j^2 \geq 0$ for all $j \in [1:n]$. The constructed point (w, W, b, ξ, v, V) then satisfies all constraints of (10). Indeed, the last positive-semidefiniteness constraint follows by the construction of V due to the Schur complement, which therefore completes the proof. \square

Letting $W = ww^T$, $R = wu^T$, $U = uu^T$ and dropping the rank-one constraint, we arrive at the Shor relaxation of (7):

$$\begin{aligned} \text{(SCoP)} \quad \min_{w,W,b,\xi,R,U} \quad & \frac{1}{2} \text{trace}(W) + C \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & (w, b, \xi) \in \Xi \\ & e^T u = n - B \\ & \text{diag } R = 0, \quad \text{diag } U = u \\ & \begin{bmatrix} 1 & w^T & u^T \\ w & W & R \\ u & R^T & U \end{bmatrix} \in S_+^{2n+1} \end{aligned} \quad (11)$$

Theorem 3.4. *If there exists part w of an optimal solution to (1) with $\|w\|_0 > B$, then the optimal value of (11) is strictly larger than the optimal value of (1).*

Proof. For a given (1)-feasible point (w, b, ξ) with $\|w\|_0 > B$, we restrict the corresponding variables of (11) to the same value and estimate the optimal value of the restricted version of (11). The matrix

$$\begin{bmatrix} 1 & w^T & u^T \\ w & W & R \\ u & R^T & U \end{bmatrix} \in S_+^{2n+1}, \quad \text{with} \quad \text{diag } R = o \text{ and } \text{diag } U = u,$$

has a principal submatrix

$$\begin{bmatrix} 1 & w_j & u_j \\ w_j & W_{jj} & 0 \\ u_j & 0 & u_j \end{bmatrix} \in S_+^3, \quad \text{for all } j \in [1 : n],$$

which implies that

$$\begin{cases} 0 \leq u_j \leq 1, & W_{jj} \geq w_j^2, & j \in [1 : n] \\ (W_{jj} - w_j^2)(u_j - u_j^2) - w_j^2 u_j^2 \geq 0, & j \in [1 : n]. \end{cases}$$

It follows that

$$\begin{cases} 0 \leq u_j \leq 1, & \text{if } j \in S_0 := \{j : w_j = 0\}, \\ 0 \leq u_j \leq \frac{W_{jj} - w_j^2}{W_{jj}} < 1 & \text{if } j \in S_+ := [1 : n] \setminus S_0 \end{cases}$$

(observe that $w_j > 0$ implies $W_{jj} > 0$). Recalling the assumption $\|w\|_0 > B$, it follows that

$$\sum_{j \in S_0} u_j < n - B.$$

Together with the constraint

$$e^T u = n - B,$$

we have $u_j > 0$ for some $j \in S_+$, which implies that $W_j > w_j^2$ for some $j \in S_+$. Hence, the optimal value of (11) is strictly larger than the optimal value of (1). \square

There is no direct comparison between the relaxation (10) and the relaxation (11). But given the case we have a valid estimate of M at hand satisfying (9), the relaxation (11) can be tightened with a small computational cost by adding the big-M constraint via B_M , which reads as:

$$\begin{aligned} (\text{SCoMP}) \quad & \min_{\substack{w, b, \xi, \\ R, u, U}} \frac{1}{2} \text{trace}(W) + C \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & (w, b, \xi) \in \Xi \\ & (w, u) \in B_M \\ & \text{diag } R = o, \quad \text{diag } U = u, \\ & \begin{bmatrix} 1 & w^T & u^T \\ w & W & R \\ u & R^T & U \end{bmatrix} \in S_+^{2n+1}. \end{aligned} \quad (12)$$

Theorem 3.5. *The relaxation (12) is at least as tight as the relaxations (10) and (11).*

Proof. It is clear that the relaxation (12) is tighter than the relaxation (11) due to the additional big-M constraint. To show the other claim, we construct a (10)-feasible point, given a (12)-feasible point. Suppose that the point (w, W, b, ξ, R, u, U) is (12)-feasible, let $v = e - u$ and $V = vv^T + Z$, where $Z \in S_+^n$ is a diagonal matrix with $Z_{jj} = v_j - v_j^2 \geq 0$ for all $j \in [1 : n]$. Then (w, W, b, ξ, v, V) is (10)-feasible with the same objective value. \square

3.3. Decomposed SDP-based relaxations

In this subsection, we propose a procedure for obtaining equivalent and much smaller SDP-based relaxations by exploiting the sparsity patterns of the Shor relaxations.

We notice that, for example: in problem (11), some sparsity patterns appear both in the objective and the constraints. e.g., only diagonal elements of matrices W , R , and U are involved. Based on this observation, we write down the following decomposed version of (11):

$$\begin{aligned} (\text{DSCoP}) \quad & \min_{w, \text{diag } W, b, \xi, u} \frac{1}{2} \text{trace}(W) + C \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & (w, b, \xi) \in \Xi \\ & e^T u = n - B \\ & \begin{bmatrix} 1 & w_j & u_j \\ w_j & W_{jj} & 0 \\ u_j & 0 & u_j \end{bmatrix} \in S_+^3, \quad j \in [1 : n] \end{aligned} \quad (13)$$

Theorem 3.6. *The relaxation (11) is equivalent to the relaxation (13).*

Proof. Suppose that (w, W, b, ξ, R, u, U) is (11)-feasible. It is clear that $(w, \text{diag } W, b, \xi, u)$ is (13)-feasible. In the opposite direction, we assume that the point $(w, \text{diag } W, b, \xi, u)$ is (13)-feasible. It is clear that all linear constraints in (11) are satisfied if we let $\text{diag } R = o$ and $\text{diag } U = u$. The only thing that needs to be checked is if the following partial positive-semidefinite matrix is positive-semidefinite completable:

$$\begin{bmatrix} 1 & w_1 & \dots & w_n & u_1 & \dots & u_n \\ w_1 & W_{11} & * & * & 0 & * & * \\ \vdots & * & \ddots & * & * & \ddots & * \\ w_n & * & * & W_{nn} & * & * & 0 \\ u_1 & 0 & * & * & u_1 & * & * \\ \vdots & * & \ddots & * & * & \ddots & * \\ u_n & * & * & 0 & * & * & u_n \end{bmatrix}.$$

By \mathcal{G} we denote the specification graph of the above partial positive-semidefinite matrix. Since the vertex w_i is not connected with w_j and u_j for all $i \neq j$, and the vertex u_i is not connected with u_j and w_j for all $i \neq j$, the only cycle we can find in the graph is the triangle $1 \rightarrow w_j \rightarrow u_j \rightarrow 1$ for all $[1 : n]$. Therefore \mathcal{G} is chordal. Due to Berman and Shaked-Monderer (2003, Theorem 1.39), if the specification graph of a partial positive-semidefinite matrix is chordal, it is always completable to a positive-semidefinite matrix. \square

Following a similar fashion, we propose the decomposed version of the relaxation (12):

$$\begin{aligned} (\text{DSCoMP}) \quad & \min_{w, \text{diag } W, b, \xi, u} \frac{1}{2} \text{trace}(W) + C \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & (w, b, \xi) \in \Xi \\ & (w, u) \in B_M \\ & \begin{bmatrix} 1 & w_j & u_j \\ w_j & W_{jj} & 0 \\ u_j & 0 & u_j \end{bmatrix} \in S_+^3, \quad j \in [1 : n]. \end{aligned} \quad (14)$$

Corollary 3.7. *The relaxation (14) is equivalent to the relaxation (12).*

Proof. See the proof of Theorem 3.6. \square

For the formulation (6), the decomposed version of its Shor relaxation (10) reads as:

$$\begin{aligned} (\text{DSMP}) \quad & \min_{w, \text{diag } W, b, \xi, v} \frac{1}{2} \text{trace}(W) + C \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & (w, b, \xi) \in \Xi \\ & (w, v) \in \Omega_M \\ & \begin{bmatrix} 1 & w_j \\ w_j & W_{jj} \end{bmatrix} \in S_+^2, \quad \begin{bmatrix} 1 & v_j \\ v_j & v_j \end{bmatrix} \in S_+^2, \quad j \in [1 : n]. \end{aligned} \quad (15)$$

Corollary 3.8. *The relaxation (15) is equivalent to the relaxation (10).*

Proof. See the proof of [Theorem 3.6](#). \square

4. The algorithms

In this section, we introduce both a heuristic and an exact algorithm for solving the feature selection problem we are interested in. In particular, we first present two heuristic strategies to find good upper bound values for our problem and a heuristic strategy to estimate a tight value for the big-M parameter M for the problem. All strategies are based on the resolution of the decomposed SDP-based relaxations presented in the previous section. These strategies will then be embedded in a heuristic algorithm which can ultimately improve the heuristic upper bound. Finally, we present an exact algorithm that implements one of the two upper bound procedures as a subroutine and consists of the resolution of a sequence of Mixed-Integer Second-Order Cone Optimization problems (MISOCPs).

4.1. Two upper bounding strategies

In this subsection, we propose two strategies for obtaining an upper bound, namely a feasible objective value of a feasible solution $(\hat{w}, \hat{b}, \hat{\xi})$ to (2), based on the optimal solution of relaxations.

4.1.1. Local search

The first strategy, denoted as Local Search, consists of three main steps and employs a user-specified excess parameter k which will temporarily allow to work on just $k + B$ features instead of all. We first solve the relaxation by off-the-shelf software such as Mosek, an interior point solver. With the optimal solution u^* in hand, we want to search for a feasible point nearby. We then sort features by increasing values of u^* . Namely, if u_j^* is close to 1, feature j is unlikely to be selected because of the complementarity constraint $w_j u_j = 0$, while if u_j^* is closer to 0, then the feature j is more likely to be selected. Based on it, we select the first $k + B$ features with small corresponding u , denoted by a set $K \subseteq [1 : n]$ with $|K| = k + B$. In the end, we solve the restricted mixed-integer problem:

$$\begin{aligned}
 (\text{CoP}(K)) \quad & \min_{w, b, \xi, u} \quad \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^m \xi_i \\
 \text{s.t.} \quad & (w, b, \xi) \in \mathcal{E} \\
 & \sum_{j \in K} u_j = |K| - B \\
 & u_j w_j = 0, \quad j \in K, \\
 & u_j \in \{0, 1\}, \quad j \in K.
 \end{aligned} \tag{16}$$

which is much smaller than the original problem (7), as it only involves $|K|$ binary variables u_j . The problem is solved by the global solver Gurobi. This first upper bounding strategy is described in Algorithm 1.

Algorithm 1 Local Search

- 1: **Input:** $(x^i, y^i) \in \mathbb{R}^n \times \{1, -1\}$, B , C , parameter $k \in [0 : n - B]$
- 2: Solve the relaxation (13) and return the optimal solution u^*
- 3: Generate a subset $K \subseteq [1 : n]$ with $|K| = k + B$ such that $u_i^* \leq u_j^*$ for all $i \in K$ and for all $j \in [1 : n] \setminus K$
- 4: Solve the mixed-integer problem (16)
- 5: **Output:** The set of selected features and an upper bound UB.

4.1.2. Kernel search

Slightly enlarging the search set at each iteration, we propose the following search strategy for a feasible solution of (2). Our Kernel Search strategy consists in solving a sequence of small MIQPs considering an initial ranking on the features. This strategy takes inspiration from a heuristic proposed by Angelelli et al. (2010) for the general multi-dimensional knapsack problem which has been applied to different kinds of problems such as location problems (Guastaroba & Speranza, 2012) and portfolio optimization (Angelelli et al., 2012).

Recently, Labbé et al. (2019) also applied it to solve their embedded feature selection for ℓ_1 -SVMs. In this strategy, based on some ranking, a restricted set of promising features is kept and updated throughout each iteration. Usually, for ranking variables in general MILPs, LP theory is exploited using both relaxed solution values and reduced costs, a strategy followed by Labbé et al. (2019) for ranking features.

By contrast, in our case we decided to use as a ranking criterion the coordinates of an optimal solution u^* to the decomposed SDP relaxation (13). Features are sorted with respect to this vector u^* : the smaller the value of u_j^* , the higher the relevance of feature j . Since in practice (13) leads to much tighter lower bound solutions in contrast to the plain relaxation (8) similar to the model used by Labbé et al. (2019), there is a justified hope that the ranking provided by u^* is closer to the relevance of features.

Let K be the ordered set of features, ρ a user-defined parameter and $N := \lceil \frac{n-\rho}{\rho} \rceil$. Then K is divided into N subsets denoted as K_i for $i \in [1 : N]$. In particular, each subset K_i , $i \in [1 : N - 1]$, will be composed of ρ features, and K_N will contain the remaining features.

An initial value of the UB is set to ∞ . Also define a feature subset $\bar{K} \subseteq [1 : n]$ as the *kernel set*, containing features that are kept at the next iteration. To initialize, set $\bar{K} = \emptyset$. At each iteration k , the heuristic considers set $\mathcal{K}_k = \bar{K} \cup K_k$ i.e. the union of the features in the kernel set and the features in the set K_k . To update the UB, at each iteration we solve problem $\text{CoP}(\mathcal{K}_k)$ plus the following two constraints:

$$\frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^m \xi_i \leq \text{UB}, \tag{17}$$

$$\sum_{j \in K_k} u_j \leq |K_k| - 1. \tag{18}$$

Constraint (17) restricts the objective function to take a value smaller than or equal to the current upper bound, while constraint (18) makes sure that at least one feature from the new set K_k is selected. Note that, due to the addition of these constraints, the optimization problems may potentially be infeasible. If the optimization problems are feasible, the new features selected are then added to the kernel set \bar{K} used in the next iteration since adding these features obtains an identical or better upper bound. Conversely, the set of features of \bar{K} that have not been chosen in the optimal solution in the previous iterations is removed from the kernel set \bar{K} . The removal of some of the features from the kernel set is decisive in that it does not excessively increase the number of binary variables considered in each iteration. We decided to remove the features that were not selected in the previous two iterations. The set of added features is denoted as K^+ and the set of removed features as K^- . The resulting kernel set for the next iteration is $\bar{K} \cup K^+ \setminus K^-$. Conversely, if the problem is infeasible, the kernel set is not modified and the procedure skips to the next iteration. The Kernel Search strategy is described in Algorithm 2.

4.2. A strategy to tighten the big-M parameter

Given an upper bound UB of problem (2), one can estimate the upper bound of w_j by solving the following problem:

$$\begin{aligned}
 \bar{M}_j := \max_{w, \text{diag}} \quad & w_j \\
 \text{s.t.} \quad & (w, b, \xi) \in \mathcal{E} \\
 & \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^m \xi_i \leq \text{UB} \\
 & e^\top u = n - B \\
 & \begin{bmatrix} 1 & w_j & u_j \\ w_j & W_{jj} & 0 \\ u_j & 0 & u_j \end{bmatrix} \in S_+^3, \quad j \in [1 : n].
 \end{aligned} \tag{19}$$

Similarly, the lower bound of w_j can be obtained by solving the same optimization problem with the max replaced by min and we denote the optimal value by \underline{M}_j . The proper choice of M is $\max_{j \in [1 : n]} \{\max\{|\bar{M}_j|, |\underline{M}_j|\}\}$.

Algorithm 2 Kernel Search

```

1: Input:  $(x^i, y^i) \in \mathbb{R}^n \times \{1, -1\}$ ,  $B, C$ , parameter  $\rho \in [1 : n]$ 
2: Solve relaxation (13) and return the optimal solution  $u^*$ 
3: Sort features according to the numerical size of corresponding  $u^*$ 
   increasingly, denoted by a vector  $K \in \mathbb{Z}_+^n$ 
4: Separate  $K$  into  $N := \lfloor \frac{n}{\rho} \rfloor$  groups evenly such that  $K = [K_1, \dots, K_N]$ 
   and set  $\bar{K} = \emptyset$ ,  $k = 1$ ,  $UB = \infty$ 
5: while  $k \leq N$  do
6:    $\mathcal{K}_k \leftarrow K_k \cup \bar{K}$ 
7:   Solve problem  $\text{CoP}(\mathcal{K}_k) + (17) + (18)$ 
8:   if problem is feasible then
9:     Return solution  $u^*$  and optimal value  $UB^*$ 
10:     $UB \leftarrow UB^*$ 
11:    Build  $K^+ := \{j \in K_k : j \text{ selected at current iteration}\}$ 
12:    Build  $K^- := \{j \in \bar{K} : j \text{ not selected in the last two iterations}\}$ 
13:    Update  $\bar{K} \leftarrow \bar{K} \cup K^+ \setminus K^-$ 
14:   end if
15:    $k \leftarrow k + 1$ 
16: end while
17: Output: The set of selected features and an upper bound  $UB$ .

```

Proposition 4.1. Let $[\hat{w}, \hat{b}, \hat{\xi}]$ be a feasible solution of (2) and $UB := \frac{1}{2} \|\hat{w}\|_2^2 + C \sum_{i=1}^m \hat{\xi}_i$ be the corresponding objective value. Then

$$M = \max_{j \in [1:n]} \{\max\{|\bar{M}_j|, |\underline{M}_j|\}\}$$

is a valid bound, where \bar{M}_j and \underline{M}_j are obtained by (19).

Proof. obvious. \square

Remark 4.2. The proposed bound M is tighter than most of the currently used big-Ms. Nevertheless, empirically, the tightening of any such bound is of little relevance unfortunately, because they still fall outside of the range specified in (9).

4.3. The heuristic algorithm

In this subsection, we propose a procedure for generating a good feasible solution of (2) taking one of the two upper bound strategies and the big-M estimation strategy into account. At first, a feasible solution is computed with the chosen upper bound strategy. Then, according to Proposition 4.1, we solve a series of $2n$ many SDP problems to find, for each feature j , values \bar{M}_j , \underline{M}_j and then obtain the big-M parameter M . If this value satisfies (9), we know that the SDP formulation (14) is better than (13), thus we compute a new relaxed solution u^* of (14). We thus use vector u^* as a ranking criterion for the chosen upper bound strategy, we run the strategy again and we update the heuristic solution if a better one is found.

4.4. The exact algorithm

In this subsection, we propose an exact procedure for (2) by solving a sequence of MISOCPs. With a slight abuse of notation, let us define, given a subset of features $K \subseteq [1 : n]$, the set

$$B_M(K) = \{(w, u) \in \mathbb{R}^n \times \mathbb{R}_+^n : -M(1 - u_j) \leq w_j \leq M(1 - u_j), j \in K, e^T u = n - B\}.$$

Algorithm 3 Heuristic procedure

```

1: Input:  $(x^i, y^i) \in \mathbb{R}^n \times \{1, -1\}$ ,  $B, C$ , the parameter  $k \in [0 : n - B]$  or the
   parameter  $\rho \in [1 : n]$ 
2: Run Algorithm 1 or Algorithm 2 to obtain a feasible solution  $(\hat{w}, \hat{b}, \hat{\xi})$ 
   to (2) with upper bound value  $\hat{UB}$ 
3: Set  $(w^*, b^*, \xi^*) \leftarrow (\hat{w}, \hat{b}, \hat{\xi})$ 
4: Obtain  $M$  according to Proposition 4.1
5: Solve problem (1) and obtain  $w^*$ 
6: if  $M < \frac{\|w^*\|_1}{B}$  then
7:   Solve problem (14) and obtain  $u^*$ 
8:   Run Algorithm 1 or Algorithm 2 starting with  $u^*$ , get  $(\bar{w}, \bar{b}, \bar{\xi})$ 
   feasible to (2) and upper bound  $\bar{UB}$ 
9:   if  $\bar{UB} \leq \hat{UB}$  then
10:     $(w^*, b^*, \xi^*) \leftarrow (\bar{w}, \bar{b}, \bar{\xi})$ 
11:   end if
12: end if
13: Output  $(w^*, b^*, \xi^*)$  solution of (2)

```

The subproblem considered in the exact procedure is

$$\begin{aligned}
 (\text{SR-DSCoMP}(K)) \quad & \min_{w, b, \xi, u} \frac{1}{2} \text{trace}(W) + C \sum_{i=1}^m \xi_i \\
 \text{s.t.} \quad & (w, b, \xi) \in \Xi \\
 & (w, u) \in B_M(K) \\
 & \begin{bmatrix} 1 & w_j & u_j \\ w_j & W_{jj} & 0 \\ u_j & 0 & u_j \end{bmatrix} \in S_+^3, \quad j \in [1 : n] \setminus K, \\
 & u_j \in \{0, 1\}, \quad j \in K, \\
 & \begin{bmatrix} 1 & w_j \\ w_j & W_{jj} \end{bmatrix} \in S_+^2, \quad j \in K,
 \end{aligned} \tag{20}$$

where SR is the abbreviation for semi-relaxation. In the problem (20), we essentially relax the complementarity constraints associated with features in the set $[1 : n] \setminus K$ via decomposed SDP relaxation, and express the other by the big-M mixed-integer formulation. We note that the problem (20) is a mixed-integer positive semi-definite optimization problem (MISDP) and there is no off-the-shelf solver dealing with it. In the following part, we show a way to express the problem in the form of MISOCP, which can be handled by the off-the-shelf solvers like Copt.

Before specifying the MISOCP reformulation, we present a useful result: some special slices \mathcal{F} of the positive-semidefinite cone are second-order cone representable, due to Lemma 4.3 below which seems well known in the literature, see, e.g. Agler et al. (1988), Cheramin et al. (2022). We include its short, and elementary, proof for the sake of completeness.

Lemma 4.3.

$$\begin{aligned}
 \mathcal{F} &:= \left\{ (w, u, W) \in \mathbb{R}^3 : \begin{bmatrix} 1 & w & u \\ w & W & 0 \\ u & 0 & u \end{bmatrix} \in S_+^3 \right\} \\
 &= \left\{ (w, u, W) \in \mathbb{R} \times \mathbb{R}_+^2 : \sqrt{(W - s)^2 + 4w^2} \leq W + s, \right. \\
 &\quad \left. s := 1 - u \geq 0. \right\}
 \end{aligned}$$

Proof. Consider all principal minors of the matrix (with a chordal sparsity pattern, see Agler et al. (1988)):

$$\begin{bmatrix} 1 & w & u \\ w & W & 0 \\ u & 0 & u \end{bmatrix} \in S_+^3 \iff (u, W) \in \mathbb{R}_+^2 \text{ and } u \leq 1 \text{ and } (1 - u)W \geq w^2.$$

This way we complete the proof. \square

Table 1
First small datasets.

Dataset	m	n	Class (%)
Breast Cancer D.	569	30	63/37
Breast Cancer P.	198	33	65/35
Breast Cancer W.	683	9	65/35
Cleveland	297	13	54/46
Diabetes	768	8	65/35
German	1000	24	70/30
Ionosphere	351	33	36/64
Parkinsons	195	22	25/75
Wholesale	440	7	68/32

Applying Lemma 4.3, the problem (20) can be rewritten of the form

$$\begin{aligned}
 \text{(SR-DLMP}(K)) \quad & \min_{\mathbf{w}, \text{diag}} \frac{1}{2} \text{trace}(\mathbf{W}) + C \sum_{i=1}^m \xi_i \\
 \text{s.t.} \quad & (\mathbf{w}, \mathbf{b}, \xi) \in \Xi \\
 & (\mathbf{w}, \mathbf{u}) \in B_M(K) \\
 & \mathbf{d}_j := \begin{bmatrix} W_{jj} - s_j \\ 2w_j \end{bmatrix} \in \mathbb{R}^2, \quad j \in [1:n] \setminus K, \\
 & \|\mathbf{d}_j\|_2 \leq s_j + W_{jj}, \quad j \in [1:n] \setminus K, \\
 & s_j + u_j = 1, \quad j \in [1:n] \setminus K, \\
 & u_j \in \{0, 1\}, \quad j \in K, \\
 & w_j^2 \leq W_{jj}, \quad j \in K,
 \end{aligned} \tag{21}$$

where $n - |K|$ more variables $s_j, j \in [1:n] \setminus K$, are introduced.

Now we can introduce the exact algorithm reported as Algorithm 4. In this exact procedure, we solve a sequence of semi-relaxed problems SR-DLMP(K) associated with a subset $K \subseteq [1:n]$ of features. In such problems, only variables u_j with $j \in K$ will be considered as binary and the remaining ones are relaxed. To obtain initial bounds on the objective value, the exact procedure exploits the upper bound strategies detailed above. The main step of the exact procedure consists of solving the sequence of semi-relaxed problems (21) to improve the lower bound of the objective value.

To start with, we must select a subset of features $K \subseteq [1:n]$ whose associated u variables will be considered as binary variables in the first semi-relaxed problem. The upper-bound strategy run at the beginning provides a subset of features that allows us to obtain a good bound on the optimal objective value. Therefore, the exact procedure will consider the set provided by the heuristic as the initial K and it will obtain an initial LB solving SR-DLMP(K). Then the set K is updated by adding and removing some of the features, in order to improve the lower bound of the objective value. To this end, two sets (denoted by K^+ and K^-) are built in each iteration. Set K^+ consists of some of the features in $[1:n] \setminus K$ whose associated u variables will be considered as binary in the next iteration, i.e. features of K^+ will be added to K . Similarly, K^- consists of features in K that will not be considered as binary in the next iteration.

While we cannot rule out that a once selected set K is visited again during the iterations, the imposed time limit ensures finiteness of this procedure.

In addition and if possible, we will update the UB in the main step running the upper bound strategies and using vector $\tilde{\mathbf{u}}$ as the initial ranking. There could be many different ways of defining set K^+ and updating K . In the strategy we adopted, we set K^+ as the set of s features j such that the relaxed solution \tilde{u}_j of SR-DLMP(K) was “less binary”. This is why we first create a ranking on the features based on terms $\tilde{u}_j - \tilde{u}_j^2$ in descending order, and then select the first s , which is a user-defined parameter.

Algorithm 4 Exact procedure

```

1: Input:  $(x^i, y^i) \in \mathbb{R}^n \times \{1, -1\}$ ,  $B, C, s \in [1:n]$ ,  $\text{LB} = -\infty$ ,  $\text{UB} = \infty$ ,  $K = \emptyset$ 
2: Run Algorithm 1 or Algorithm 2, obtain set  $K$  as the set of the selected features and update the UB value
3: while  $\frac{\text{UB} - \text{LB}}{\text{UB}} \cdot 100 \geq 0.01$  do
4:   Solve problem SR-DLMP( $K$ )
5:   if a time limit of 1800 seconds is reached then
6:     STOP
7:   end if
8:   Obtain optimal value  $\widehat{\text{LB}}$  and optimal solution  $\tilde{\mathbf{u}}$ 
9:   Run Algorithm 1 or Algorithm 2 with the first step skipped
   vector  $\tilde{\mathbf{u}}$  used for initial ranking
10:  Obtain the solution  $(\hat{\mathbf{w}}, \hat{\mathbf{b}}, \hat{\xi}, \hat{\mathbf{u}})$  and upper bound value  $\widehat{\text{UB}}$ 
11:  Sort features using terms  $\tilde{u}_j - \tilde{u}_j^2$  in descending order
12:  Build  $K^+$  with the first  $s$  features
13:  Build  $K^- := \{j \in K : j \text{ not selected in the last two iterations}\}$ 
14:   $K \leftarrow K \cup K^+ \setminus K^-$ 
15:  if  $\widehat{\text{UB}} < \text{UB}$  then
16:     $\text{UB} \leftarrow \widehat{\text{UB}}$ 
17:     $(\mathbf{w}^*, \mathbf{b}^*, \xi^*, \mathbf{u}^*) \leftarrow (\hat{\mathbf{w}}, \hat{\mathbf{b}}, \hat{\xi}, \hat{\mathbf{u}})$ 
18:  end if
19:  if  $\widehat{\text{LB}} > \text{LB}$  then
20:     $\text{LB} \leftarrow \widehat{\text{LB}}$ 
21:  end if
22: end while
23: Output:  $(\mathbf{w}^*, \mathbf{b}^*, \xi^*)$  solution of (2)

```

Table 2
Second large datasets.

Dataset	m	n	Class (%)
Arrhythmia	452	274	54/46
Colorectal	62	2000	65/35
DLBCL	77	7129	25/75
Lymphoma	96	4026	35/65
Madelon	2000	500	50/50
Mfeat	2000	649	10/90

5. Numerical experiments

The numerical experiments were performed on a computer with an Apple M1 processor and 16 GB of RAM. All models and algorithms were implemented in Python. Results were performed using Gurobi 11.0 for the resolution of the MIQPs, Mosek 10.1 for the resolution of the SDP relaxations and Copt for the resolution of the MISOCP models solved in the Exact algorithm.

The experiments were carried out on fifteen different datasets. Nine of them can be found in the UCI repository (Dua & Graff, 2017). See Table 1, where m is the number of data points, n is the number of features and the last column shows the percentage of samples in each class. As can be observed, they contain a small number of features. The other six datasets used in the experiments have a larger number of features, as shown in Table 2. The Arrhythmia, Madelon, and MFeat datasets are available in the UCI repository as well. The remaining three datasets, Colorectal, DLBCL, and Lymphoma, are microarray datasets with thousands of features but smaller sample sizes. Further descriptions of these last datasets can be found in Alon et al. (1999), Guyon et al. (2002), Shipp et al. (2002).

5.1. Computational analysis on the MIQP models

To start with, we used Gurobi (Gurobi Optimization, LLC, 2024) to solve the two proposed MIQP models BigMP and CoP. The reason behind these first comparisons is that Gurobi is used in the two upper

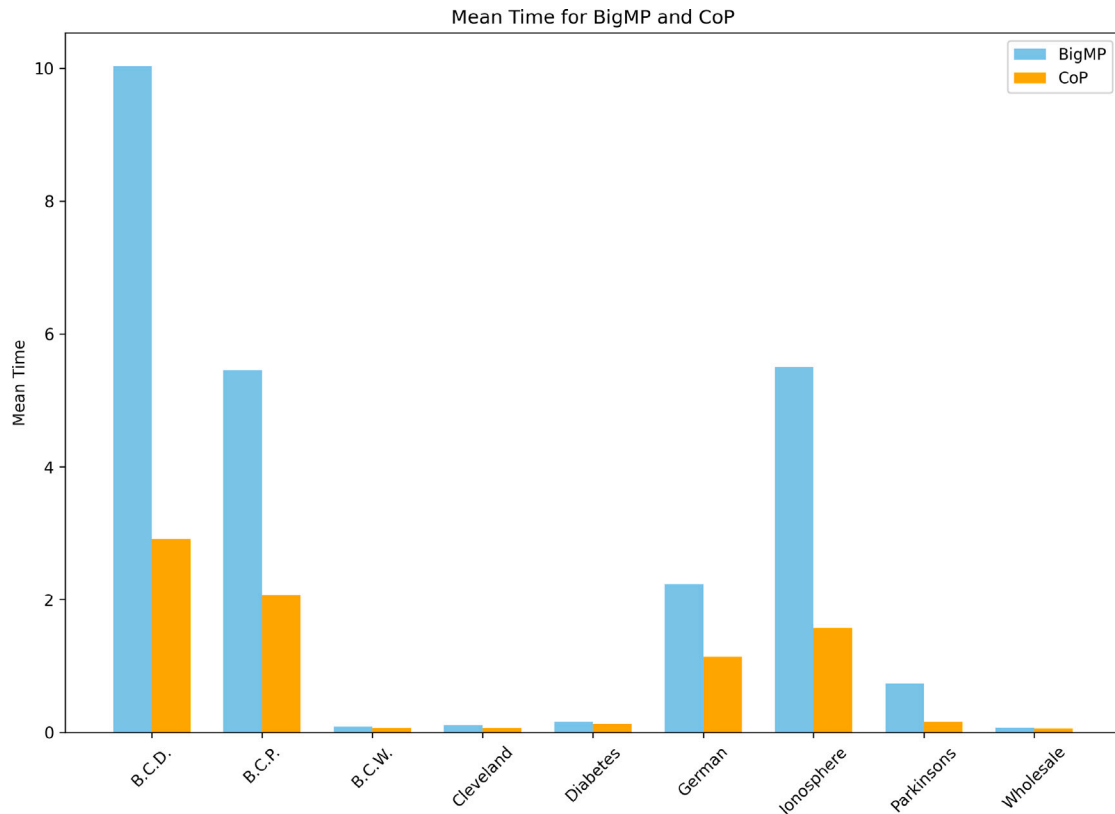


Fig. 1. Comparison of solving BigMP and CoP with Gurobi for the small datasets: mean computational times (in seconds).

bounding strategies we proposed. The two strategies, the Local Search and the Kernel Search, require the resolution of the FS-SVM model on instances created on a smaller subset of features, and this can be either done by using the BigMP or the CoP formulation.

From these initial tests, we understand that CoP is the model which can be solved faster with Gurobi when dealing with datasets with a small number of features. Thus we will use the resolution of model CoP as the subroutine used by the heuristic algorithms to find, and update the values of upper bounds.

We designed two sets of experiments: the first using small datasets, described in Table 1 and the second using larger ones, described in Table 2.

For the first set of results based on the small size datasets, we generated different instances ranging the integer B from 1 to n and C values in the set $\{10^r : r \in [-3 : 3]\}$. We present the results in Fig. 1, where we report for each dataset, the average computation times for solving the MIQP formulations, CoP and BigMP, on all the instances obtained varying B, C . Indeed, for this first set of problems, Gurobi was able to reach the global optimum in all the cases, although the formulation CoP (7) was solved faster than BigMP (6) by a factor of 1.4 on average across these datasets.

For the results on datasets in Table 2, we set the hyperparameter $C = 10$, and, similarly to Labbé et al. (2019), we ranged B values in $\{10, 20, 30\}$. In Table 3 we show the computational performance of Gurobi and report the returned value of the objective function of problem (2) (ObjFun) and the optimal gap value (MipGap) returned by Gurobi.

We can notice how the resulting optimization problems are much harder to solve in contrast to the problems related to the smaller datasets. Indeed for these results a time limit of 1 hour was set, and the final MipGap values for the majority of these datasets are notably high. This is certainly due to the fact that the lower bound values computed

Table 3

Comparison on large datasets of solving BigMP and CoP with Gurobi: optimal objective function (ObjFun) and gap values (MipGap).

Dataset	B	BigMP		CoP	
		ObjFun	MipGap	ObjFun	MipGap
Arrhythmia	10	2398.06	0.64	2151.56	0.82
	20	2047.67	0.67	1712.64	0.81
	30	1733.28	0.67	1441.63	0.78
Colorectal	10	2.99	0.64	2.12	0.98
	20	0.64	0.13	0.68	0.93
	30	0.41	0.04	0.43	0.89
DLBCL	10	0.63	1.00	0.56	1.00
	20	0.18	0.99	0.15	0.99
	30	0.16	0.99	0.09	0.98
Lymphoma	10	18.07	1.00	1.16	0.99
	20	0.71	0.99	0.29	0.98
	30	0.34	0.98	0.16	0.96
Madelon	10	16 680.48	0.32	16 800.07	0.33
	20	16 419.87	0.32	18 558.98	0.40
	30	16 075.70	0.30	18 425.44	0.39
Mfeat	10	0.92	0.59	0.68	0.93
	20	0.26	0.21	0.26	0.81
	30	0.18	0.15	0.18	0.71

in the Branch & Bound tree of Gurobi are not very effective for this type of formulations. This is particularly the case for model CoP whose final MipGap values tend to be worse than the ones of model BigMP. In contrast, Gurobi manages to find much better incumbent solutions when solving model CoP for all datasets, except for Colorectal and Madelon.

Table 4

Results on large datasets of the Heuristic algorithm implementing either the Local search (H-LS) or the Kernel Search (H-KS) as the upper bound strategies.

Dataset	B	H-LS		H-KS	
		ObjFun	Time	ObjFun	Time
Arrhythmia	10	2491.04	5.37	2142.67	41.24
	20	1982.08	9.55	1652.83	168.83
	30	1729.55	7.59	1380.19	311.62
Colorectal	10	2.13	15.38	2.04	15.88
	20	0.67	20.12	0.65	57.51
	30	0.411	37.26	0.406	69.46
DLBCL	10	0.25	51.01	0.23	26.56
	20	0.104	46.97	0.101	67.74
	30	0.065	134.95	0.063	117.29
Lymphoma	10	0.69	28.73	0.69	98.64
	20	0.24	40.87	0.23	54.42
	30	0.14	98.25	0.13	109.33
Madelon	10	16 660.85	25.07	16 615.43	607.32
	20	16 236.87	48.57	16 236.87	612.62
	30	15 917.52	176.92	15 917.99	605.93
Mfeat	10	0.70	21.27	0.66	341.87
	20	0.25	76.68	0.25	621.16
	30	0.17	276.90	0.17	606.96

5.2. Results on the heuristics

In this section we analyze the performance of the heuristic algorithm we proposed and we compare it with Gurobi. We have two versions of the heuristic algorithm depending on whether the Local Search or the Kernel Search upper bound strategies are used. We denote by H-LS the heuristic algorithm which implements the Local Search strategy, and by H-KS the heuristic implementing Kernel Search.

Table 4 shows the optimization performance of H-LS and H-KS. Both heuristics were given a time limit of 600 s. In particular for H-LS we set the parameter $k = 10$, which means that we solved the CoP model using the $B + 10$ most promising features based on the information of the optimal solution to the relaxation. For H-KS we also set parameter $s = 10$, which means that the overall set of features was divided in subsets of ten features each. A smaller time limit of 60 s was set for the resolution of each CoP subproblem solved in the Kernel Search strategy.

As expected, H-KS manages to find better upper bounds compared to H-LS, at the cost of generally longer computation times. In some instances, such as Lymphoma, Madelon and Mfeat, both heuristics find the same solutions. The time limit of 600 s was reached by H-KS for the instances related to datasets Madelon and Mfeat. For the rest of the datasets, the average computational times do not exceed 300 s for both heuristics. In general, compared to the objective function values of the solutions found by Gurobi in 1 hour (Table 3), the solutions found by the heuristics are much better.

Another set of results we carried out was the one of using H-KS as a warm start heuristic in order to evaluate if Gurobi was able to find better solution. In Table 5 we compare the optimal value of the solution of Gurobi after 1 hour for model CoP, the value of the H-KS heuristic, and the value of the solution found by Gurobi using as a warm start solution the one of H-KS (denoted by CoP*).

Regarding Table 5, we can see that Gurobi was unable to find a better solution than the heuristic after 1 hour for all datasets, apart from Arrhythmia, and this is probably due to the fact that the heuristic solutions are already optimal or nearly-optimal. In this table, column d_f indicate the percentage of improvement of the objective function found by Gurobi using H-KS starting solution, against the objective function of the solution found by Gurobi alone (see Fig. 2 and Table 6).

Table 5

Results on large datasets of Gurobi, the Heuristic Kernel Search algorithm, Gurobi with H-KS as a warm start: returned objective functions of CoP. d_f is the % improvement Gurobi with H-KS warm start against Gurobi alone.

Dataset	B	ObjFun			d_f
		CoP	H-KS	CoP*	
Arrhythmia	10	2151.56	2142.67	2098.55	2%
	20	1712.64	1652.83	1652.83	3%
	30	1441.63	1380.19	1366.94	5%
Colorectal	10	2.12	2.04	2.04	4%
	20	0.68	0.65	0.65	5%
	30	0.43	0.41	0.41	6%
DLBCL	10	0.56	0.23	0.23	60%
	20	0.15	0.10	0.10	35%
	30	0.09	0.06	0.06	29%
Lymphoma	10	1.16	0.69	0.69	41%
	20	0.29	0.22	0.22	21%
	30	0.16	0.13	0.13	16%
Madelon	10	16 800.07	16 615.43	16 615.43	1%
	20	18 558.98	16 236.86	16 236.86	13%
	30	18 425.44	15 917.51	15 917.51	14%
Mfeat	10	0.68	0.66	0.66	3%
	20	0.26	0.25	0.25	4%
	30	0.17	0.17	0.17	3%

Table 6

Comparison on large datasets among the SDP lower bound and the starting bound found by Gurobi for the BigMP and CoP models.

Dataset	B	Initial bounds		
		DSCoP	BigMP	CoP
Arrhythmia	10	754.46	451.10	274.79
	20	622.44	395.92	274.79
	30	538.86	303.66	274.79
Colorectal	10	1.07	0.05	0.05
	20	0.54	0.05	0.05
	30	0.38	0.05	0.05
DLBCL	10	0.16	0.00	0.00
	20	0.08	0.00	0.00
	30	0.06	0.00	0.00
Lymphoma	10	0.35	0.00	0.01
	20	0.18	0.00	0.01
	30	0.12	0.00	0.01
Madelon	10	11 243.14	0.00	0.00
	20	11 209.44	0.00	0.00
	30	11 197.73	0.00	0.00
Mfeat	10	0.36	0.05	0.05
	20	0.20	0.05	0.05
	30	0.15	0.05	0.05

5.3. Results on the exact approach

We now present the results on the Exact procedure. As shown in Table 7, the exact algorithm performs much better than Gurobi with a warm start heuristic, both on optimal value and computation time. In particular, the exact algorithm is able to globally solve four problems under a time limit of 1 hour. It is worth mentioning that the initial lower bounds obtained through the relaxation (13) for these problems are already quite strong. In the process of the exact procedure, after several updates of the set K , the semi-relaxation SR-DLMP(K) provides an almost tight bound, allowing the algorithm to terminate within the time limit. For the unsolved problems, the exact algorithm provides a better objective value and a smaller MipGap.

5.4. Validation

Finally, we conducted a series of experiments to compare our feature selection model (2) which is based on the ℓ_2 -norm, against

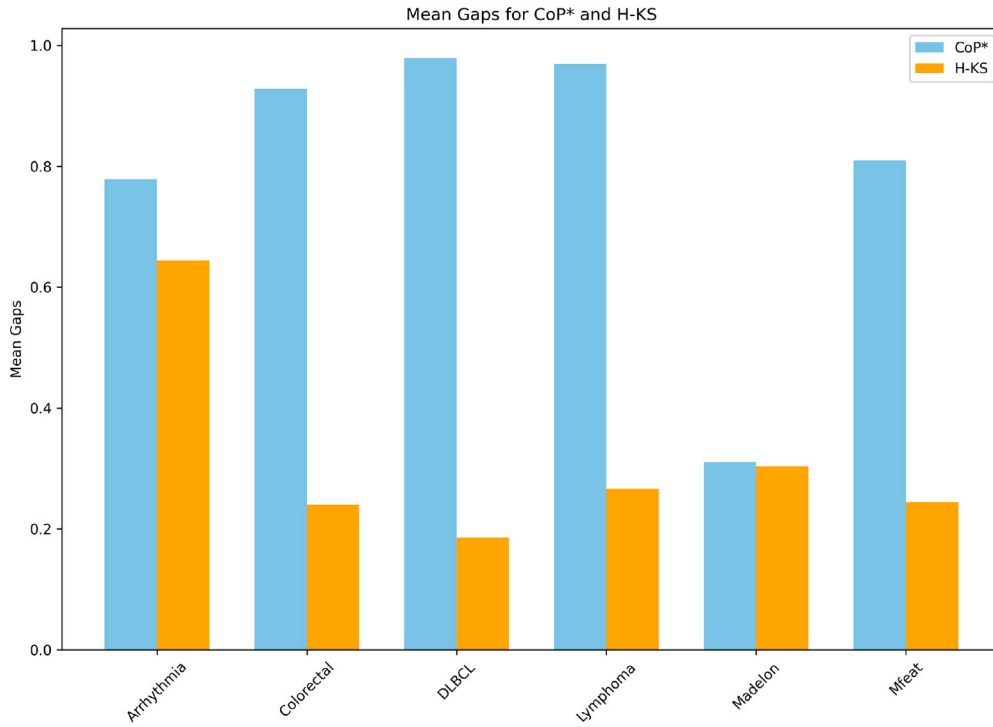


Fig. 2. Comparison on large datasets of the mean MipGap values of the solution found by Gurobi solving CoP after 1 hour, and the Relaxed Gap of the Heuristic Kernel Search solution using the solution of DSCoP as a lower bound.

Table 7

Comparison on large datasets of the solution to the CoP returned by Gurobi with H-KS as a warm start, and by the Exact algorithm using the Kernel Search procedure as the upper bound strategy.

Dataset	B	CoP*			Exact KS		
		ObjFun	Time	MipGap	ObjFun	Time	MipGap
Arrhythmia	10	2098.55	3600	0.82	2003.98	3600	0.22
	20	1652.83	3600	0.80	1632.32	3600	0.27
	30	1366.94	3600	0.77	1362.75	3600	0.32
Colorectal	10	2.04	3600	0.98	2.04	407.2	0
	20	0.65	3600	0.93	0.65	431.5	0
	30	0.41	3600	0.88	0.41	407.6	0
DLBCL	10	0.23	3600	0.99	0.23	451.3	0
	20	0.10	3600	0.98	0.10	542.7	0
	30	0.06	3600	0.97	0.06	746.2	0
Lymphoma	10	0.69	3600	0.99	0.69	314.6	0
	20	0.22	3600	0.97	0.22	532.3	0
	30	0.13	3600	0.95	0.13	823.4	0
Madelon	10	16 615.43	3600	0.33	16 499.38	3600	0.17
	20	16 236.86	3600	0.31	16 142.54	3600	0.18
	30	15 917.51	3600	0.30	15 765.83	3600	0.21
Mfeat	10	0.66	3600	0.93	0.64	397.5	0
	20	0.25	3600	0.80	0.24	573.5	0
	30	0.17	3600	0.71	0.17	1045.5	0

other machine learning models in the literature. Firstly, we compare it against the feature selection model studied in [Labbé et al. \(2019\)](#) which employs the ℓ_1 -norm instead; secondly we compare it with a model which adds an ℓ_1 penalty term in the objective function (similarly to the elastic-net models proposed in [Wang et al. \(2006\)](#), [Zou and Hastie \(2005\)](#)). This approach, aimed at achieving solution sparsity, requires tuning of the penalty parameter. The aim of the first set of experiments is to validate, from a machine learning perspective, our model compared to the one by [Labbé et al. \(2019\)](#) which was already proven to be more effective than other SVM-based feature selection models. The second set of experiments is instead aimed at emphasizing

the main difference between our model, which is based on a strict cardinality constraint, and an elastic-net-type model, which has no direct control over the number of features that will be selected

As for the comparison with the ℓ_1 model, we varied the parameter C among values in $\{10^r : r \in [-3 : 3]\}$. Regarding the parameter B , similarly to [Labbé et al. \(2019\)](#), we ranged it among all possible values in-between 1 and n for the small datasets, while, for the larger ones, B varied in the set $\{10, 20, 30, 40, 50\}$.

Under a ten-fold cross-validation framework, for each of the ten folds and for each C and B in the grid, we computed the accuracy (ACC) values on the portion of the dataset left for validation. We then

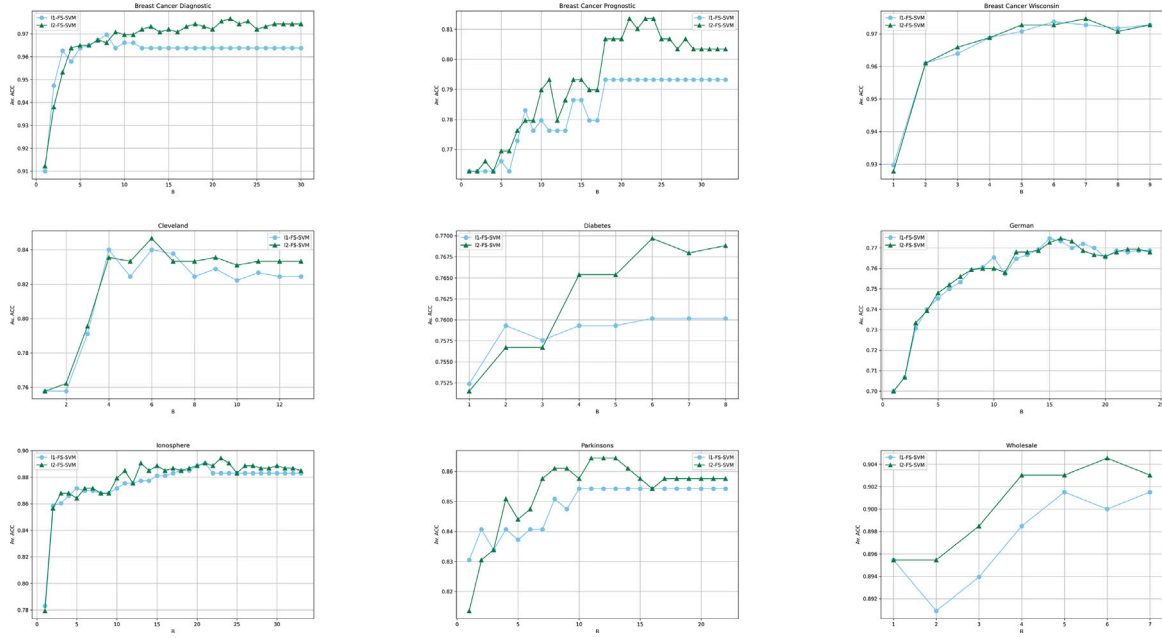


Fig. 3. The ℓ_2 -FS-SVM (dark green) versus ℓ_1 -FS-SVM (light blue): best average validation ACC values over a 10-fold cross validation as B changes - small datasets (problems solved to optimality). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

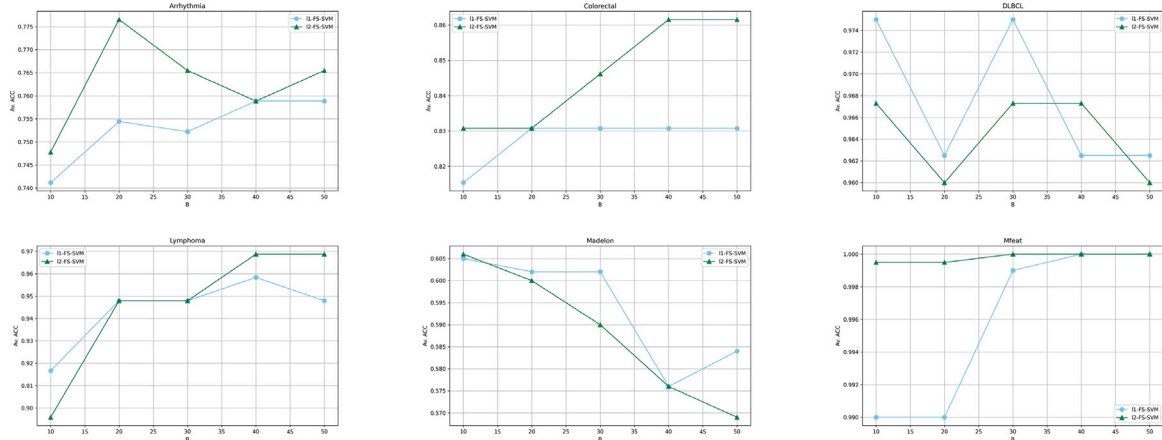


Fig. 4. The ℓ_2 -FS-SVM (dark green) versus ℓ_1 -FS-SVM (light blue): best average validation ACC values over a 10-fold cross validation as B changes - large datasets (solutions returned by the Heuristic Kernel Search with a time limit of 600 sec). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

computed the average ACC values among the ten folds. In Fig. 3, we plot the best mean ACC values for each B in the grid for the small datasets. We recall that in this case, all the optimization problems could be solved to optimality by Gurobi. As we can see, in most of the cases, employing the ℓ_2 -norm leads to better generalization performance, resulting in higher average accuracy values.

Similar results for the large datasets are reported in Fig. 4. In this case, each optimization problem was solved with the Kernel Search strategy alone. This is because, as shown in the computational experience, the strategy was able to find in most cases very good, if not optimal, solutions. The same was done in Labbé et al. (2019). To differentiate the two models, the FS-SVM model we studied in this paper is here named as the ℓ_2 -FS-SVM model, while the one presented in Labbé et al. (2019) is referred to as the ℓ_1 -FS-SVM. The Kernel Search heuristic we implemented for the ℓ_1 -FS-SVM is the same as described in the relative paper. In order to conduct a fair comparison, we set the same time limit to 600 s for each procedure.

As regards the comparison with the elastic-net-type SVM model (Wang et al., 2006; Zou & Hastie, 2005), we perform experiments to check whether the approaches can obtain the same performance. For this set of results we did not carry out a k -fold cross validation but we partitioned the datasets in a training set and a test set (80% and 20% of the datapoints respectively). We fixed the parameter C to 10, and the budget value was set to $B = 5$ for the set of the small datasets (Table 1), and to $B = 10$ for the larger datasets (Table 2). The results are reported in Tables 8 and 9.

Our model, denoted as FS-SVM, always selects the desired number of features, due to the strict cardinality constraint. The elastic-net-type SVM model, denoted as EN-SVM, does not use the cardinality constraint, instead an additional penalization term $\alpha\|w\|_1$ is added to the objective function. We ranged values of α in $\{2^i : i \in [0:10]\}$. For each dataset we reported the training and test accuracy values, as well as the number of features selected (underlined values indicate that the number of selected features exceeds the limit B). If there are

Table 8

Strict cardinality control (FS) vs. elastic net penalty control (EN) on the first (small) datasets.

Dataset	FS-SVM			EN-SVM			
	ACC train	ACC test	Feat. sel.	α	ACC train	ACC test	Feat. sel.
Breast Cancer D.	98.4	93.7	5	2 ⁹	96.2	93.0	5
Breast Cancer P.	84.8	77.6	5	2 ⁵	80.7	79.6	<u>16</u>
Breast Cancer W.	97.1	96.5	5	2 ¹⁰	95.5	95.3	<u>6</u>
Cleveland	85.1	78.7	5	2 ⁸	78.8	70.7	<u>8</u>
Diabetes	77.6	76.6	5	2 ⁹	76.9	76.6	4
German	75.3	73.2	5	2 ⁸	74.4	71.6	<u>12</u>
Ionosphere	89.0	89.8	5	2 ⁸	88.2	89.8	<u>7</u>
Parkinsons	89.0	85.7	5	2 ⁷	86.3	83.7	5
Wholesale	92.1	90.0	5	2 ⁷	91.5	89.1	5

Table 9

Strict cardinality control (FS) vs. elastic net penalty control (EN) on the second (larger) datasets.

Dataset	FS-SVM			EN-SVM			
	ACC train	ACC test	Feat. sel.	α	ACC train	ACC test	Feat. sel.
Arrhythmia	80.8	75.2	10	2 ⁹	72.6	74.3	10
Colorectal	100.0	75.0	10	2 ⁷	86.9	75.0	<u>11</u>
DLBCL	100.0	95.0	10	2 ⁷	98.2	100	<u>27</u>
Lymphoma	100.0	95.8	10	2 ⁸	93.6	83.3	<u>14</u>
Madelon	62.6	63.1	10	2 ⁹	65.4	61.2	<u>84</u>
Mfeat	100	100	10	2 ¹⁰	99.8	99.8	<u>17</u>

α values for the EN-SVM model which yielded a number of features not exceeding B , we chose the model with the best training accuracy, otherwise we chose the model with the lowest number of features selected. In the latter case, increasing α values for the selected α , yielded models in which no features were selected (i.e. for those α values, the optimal w had all zero components).

The results show that the FS-SVM models consistently achieve better ACC values, not just on the training set, but also on the test set. In most of the experiments, the EN-SVM model was not able to select a number of features that was under the predefined value B . Results indicate that, while in some cases such as for the Breast Cancer P., Ionosphere, Colorectal, and DLBCL datasets, selecting more features led EN-SVM to have better or equal ACC values on the test set than FS-SVM, in all other cases where EN-SVM selected more than B features, it did not lead to improved ACC values in the test set. Thus our model performed better even with fewer features being selected.

6. Conclusions

In this study, we addressed the NP-hard feature selection problem in linear SVMs under a cardinality constraint, which ensures that the model remains interpretable by selecting a limited number of features. Our approach involves formulating the problem as a MIQP problem and introducing novel SDP relaxations to handle its complexity and enhance scalability.

We developed two MIQP formulations: one employing a big-M method and another one using a complementarity constraint. These formulations accommodate the cardinality constraint by integrating binary variables to select features. To solve these challenging formulations, we proposed several tractable SDP relaxations and a decomposed SDP approach, exploiting the sparsity pattern inherent to the problems. This decomposition significantly reduces computational complexity, making the conic relaxations scalable even for datasets with a large number of features.

For practical implementation, we designed both heuristic and exact algorithms based on the SDP relaxations. The heuristic algorithms, informed by the solutions of the relaxations, efficiently search for good upper bounds and feasible solutions. Meanwhile, the exact algorithm iteratively refines these bounds by solving a sequence of MISOCPs, eventually converging to the global optimum.

Numerical experiments on various benchmark datasets demonstrate the effectiveness of our approach. The proposed methods not only

outperform Gurobi but also provide competitive classification accuracy. By allowing a flexible adjustment of the number of features, our approach enhances interpretability without significantly compromising predictive performance.

Future work may explore extensions of this framework to non-linear SVMs or Support Vector Regression problems. Moreover, the reformulations and algorithms proposed in this paper can be extended to other linear-SVM-based models where feature selection plays a crucial role in model interpretation and performance, e.g. optimal classification trees with margins (D'Onofrio et al., 2024).

CRedit authorship contribution statement

Immanuel Bomze: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis. **Federico D'Onofrio:** Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Laura Palagi:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Bo Peng:** Writing – review & editing, Writing – original draft, Visualization, Validation, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Acknowledgments

Laura Palagi acknowledges financial support from Progetto di Ricerca Medio Sapienza Uniroma1 (2022) - n. RM1221816BAE8A79. Research of Bo Peng supported by the doctoral programme Vienna Graduate School on Computational Optimization, FWF (Austrian Science Fund), Project W1260-N35. All authors are indebted to excellent reviewers' reports who provided significant remarks for improving presentation; based upon their suggestion, some new experiments have been added as well, corroborating our approach.

References

- Agler, J., Helton, W., McCullough, S., & Rodman, L. (1988). Positive semidefinite matrices with a given sparsity pattern. *Linear Algebra and its Applications*, 107, 101–149.

- Agor, J., & Özaltın, O. Y. (2019). Feature selection for classification models via bilevel optimization. *Computers & Operations Research*, 106, 156–168.
- Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., & Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12), 6745–6750.
- Angelesli, E., Mansini, R., & Speranza, M. G. (2010). Kernel search: A general heuristic for the multi-dimensional knapsack problem. *Computers & Operations Research*, 37(11), 2017–2026.
- Angelesli, E., Mansini, R., & Speranza, M. G. (2012). Kernel search: A new heuristic framework for portfolio selection. *Computational Optimization and Applications*, 51, 345–361.
- Aytug, H. (2015). Feature selection for support vector machines using generalized Benders decomposition. *European Journal of Operational Research*, 244(1), 210–218.
- Berman, A., & Shaked-Monderer, N. (2003). *Completely Positive Matrices*. World Scientific.
- Bertsimas, D., & Shioda, R. (2009). Algorithm for cardinality-constrained quadratic optimization. *Computational Optimization and Applications*, 43(1), 1–22.
- Bertsimas, D., & Van Parys, B. (2020). Sparse high-dimensional regression. *The Annals of Statistics*, 48(1), 300–323.
- Bradley, P. S., & Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines. In *International conference on machine learning*.
- Chan, A. B., Vasconcelos, N., & Lanckriet, G. R. (2007). Direct convex relaxations of sparse SVM. In *Proceedings of the 24th international conference on machine learning* (pp. 145–153).
- Cheramin, M., Cheng, J., Jiang, R., & Pan, K. (2022). Computationally efficient approximations for distributionally robust optimization under moment and Wasserstein ambiguity. *INFORMS Journal on Computing*, 34(3), 1768–1794.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.
- D'Onofrio, F., Grani, G., Monaci, M., & Palagi, L. (2024). Margin optimal classification trees. *Computers & Operations Research*, 161, Article 106441.
- Dua, D., & Graff, C. (2017). *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences, <http://archive.ics.uci.edu/ml>.
- Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9, 1871–1874.
- Fung, G., & Mangasarian, O. L. (2004). A feature selection Newton method for support vector machine classification. *Computational Optimization and Applications*, 28, 185–202.
- Gao, J., & Li, D. (2013). Optimal cardinality constrained portfolio selection. *Operations Research*, 61(3), 745–761.
- Ghaddar, B., & Naoum-Sawaya, J. (2018). High dimensional data classification and feature selection using support vector machines. *European Journal of Operational Research*, 265(3), 993–1004.
- Greenacre, M., Groenen, P. J., Hastie, T., d'Enza, A. I., Markos, A., & Tuzhilina, E. (2022). Principal component analysis. *Nature Reviews Methods Primers*, 2(1), 100.
- Guastaroba, G., & Speranza, M. G. (2012). Kernel search for the capacitated facility location problem. *Journal of Heuristics*, 18(6), 877–917.
- Gurobi Optimization, LLC (2024). Gurobi Optimizer Reference Manual. URL <https://www.gurobi.com>.
- Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46, 389–422.
- Hastie, T., Rosset, S., Tibshirani, R., & Zhu, J. (2004). The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5(Oct), 1391–1415.
- Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.
- Kamath, U., & Liu, J. (2021). *Explainable artificial intelligence: An introduction to interpretable machine learning*. Springer.
- Kanzow, C., Raharja, A. B., & Schwartz, A. (2021). An augmented Lagrangian method for cardinality-constrained optimization problems. *Journal of Optimization Theory and Applications*, 189(3), 793–813.
- Labbé, M., Martínez-Merino, L. I., & Rodríguez-Chía, A. M. (2019). Mixed integer linear programming for feature selection in support vector machine. *Discrete Applied Mathematics*, 261, 276–304.
- Lee, I. G., Zhang, Q., Yoon, S. W., & Won, D. (2020). A mixed integer linear programming support vector machine for cost-effective feature selection. *Knowledge-Based Systems*, 203, Article 106145.
- Maldonado, S., Pérez, J., Weber, R., & Labbé, M. (2014). Feature selection for support vector machines via mixed integer linear programming. *Information Sciences*, 279, 163–175.
- Nguyen, M. H., & De la Torre, F. (2010). Optimal feature selection for support vector machines. *Pattern Recognition*, 43(3), 584–591.
- Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., & Zhong, C. (2022). Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistic Surveys*, 16, 1–85.
- Shipp, M. A., Ross, K. N., Tamayo, P., Weng, A. P., Kutok, J. L., Aguiar, R. C., Gaasenbeek, M., Angelo, M., Reich, M., Pinkus, G. S., et al. (2002). Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, 8(1), 68–74.
- Tharwat, A., Gaber, T., Ibrahim, A., & Hassanien, A. E. (2017). Linear discriminant analysis: A detailed tutorial. *AI Communications*, 30(2), 169–190.
- Tillmann, A. M., Bienstock, D., Lodi, A., & Schwartz, A. (2024). Cardinality minimization, constraints, and regularization: a survey. *SIAM Review*, 66(3), 403–477.
- Vapnik, V. (1999). *The nature of statistical learning theory*. Springer Science & Business Media.
- Wang, L., Zhu, J., & Zou, H. (2006). The doubly regularized support vector machine. *Statistica Sinica*, 16(2), 589–615.
- Weston, J., Elisseeff, A., Schölkopf, B., & Tipping, M. (2003). Use of the zero norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3, 1439–1461.
- Yuan, G.-X., Chang, K.-W., Hsieh, C.-J., & Lin, C.-J. (2010). A comparison of optimization methods and software for large-scale l1-regularized linear classification. *Journal of Machine Learning Research*, 11, 3183–3234.
- Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B*, 67(2), 301–320.