ISSN 2281-4299

DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

Convex Mixed-Integer Quadratic Reformulations for Feature Selection in Polynomial Kernel SVMs

Ilaria Ciocci
Federico D'Onofrio
Sourour Elloumi
Laura Palagi

Technical Report n. 01, 2026

Convex Mixed-Integer Quadratic Reformulations for Feature Selection in Polynomial Kernel SVMs

Ilaria Ciocci[1], Federico D'Onofrio[1], Sourour Elloumi[2,3,4], and Laura Palagi[1]

[1]DIAG, Sapienza University of Rome
[2]Unité des Mathématiques Appliquées, ENSTA, Institut Polytechnique de Paris, 91120 Palaiseau, France
[3]CEDRIC lab, CNAM, Paris, France
[4]ensIIE, 1 place de la Résistance, Évry-Courcouronnes, France

Abstract

We address the problem of training nonlinear Support Vector Machines (SVMs) with polynomial kernels, incorporating a hard cardinality constraint which fixes the number of selected features to a user-defined value B. This formulation improves interpretability by enforcing exact sparsity in the input space, but leads to a Mixed-Integer Nonlinear Program (MINLP) whose continuous relaxation is nonconvex, making it intractable for general-purpose solvers. We derive two equivalent Mixed-Integer Convex Quadratic Programming (MIQP) reformulations of the original MINLP, obtained by linearizing, through auxiliary variables, the multilinear interactions between the binary feature-selection variables and the continuous dual variables of the kernel expansion. We prove that the continuous relaxations of both formulations are convex and establish formal equivalence with the original problem. The first one exploits the symmetry of the kernel expansion to eliminate redundant variables, yielding a symmetry-free model. The second one is a more compact formulation that eliminates the dependence of the auxiliary variables on the number of training samples, substantially reducing problem size. Both reformulations extend naturally to polynomial kernels of arbitrary degree. To the best of our knowledge, these are the first convex reformulations with certified global-optimality guarantees for embedded feature selection in kernel-based nonlinear SVMs. Computational experiments on benchmark datasets confirm their practical effectiveness, solving to certified global optimality instances where the original nonconvex formulation is computationally prohibitive.

Keywords: Nonlinear SVM, Feature selection, Mixed-Integer Nonlinear Programming

1 Introduction

The increasing use of machine learning (ML) models in high-stakes application domains such as medicine, finance, and public policy has made interpretability a fundamental requirement. In these contexts, decision makers often require models whose predictions are transparent and can be understood, validated, and trusted by domain experts. As a consequence, highly accurate but opaque black-box models are frequently not adopted in practice, despite their predictive performance [35, 19]. This has motivated the growing interest in interpretable machine learning, whose objective is to design predictive models that combine good prediction accuracy with transparency and explainability [36, 23].

One of the most widely used approaches to improve interpretability is *feature selection* (FS), which consists of identifying the most informative input features while discarding irrelevant or redundant ones. By restricting the prediction model to depend only on a small subset of features, FS simplifies the resulting decision rule and facilitates its interpretation. In addition, feature selection may improve generalization by reducing overfitting and mitigating noise [5, 9, 4, 30]. For a comprehensive survey of feature selection methods and their stability properties, we refer to [11]. In this paper, we focus on the feature selection problem in *Support Vector Machines* (SVMs) [13] with polynomial kernels. SVMs are among the most powerful tools for supervised classification, learning maximum-margin separating hyperplanes with strong generalization properties and robustness to high-dimensional data. When data are not linearly separable, nonlinear SVMs exploit kernel functions to implicitly map the input data into a high-dimensional feature space, enabling the learning of complex decision boundaries while retaining the convexity of the training problem. Despite their effectiveness, nonlinear SVMs typically rely on all available input features, resulting in models that are difficult to interpret. Embedding feature selection directly into the SVM training process is therefore a natural objective. Existing works imposing hard cardinality constraints mainly focus on the primal formulation of linear SVMs, where feature selection can be modeled through mixed-integer optimization in the original feature space. However, for nonlinear kernel SVMs, the problem becomes substantially more challenging as introducing binary feature selection variables directly into the kernel evaluation leads to mixed-integer nonlinear formulations whose continuous relaxations are nonconvex and computationally difficult to solve globally. As a consequence, most existing approaches for nonlinear SVMs rely on soft regularization techniques or approximation schemes that do not enforce an exact feature count (see Section 1.1). Instead, we develop an exact optimization framework for nonlinear SVMs with polynomial kernels and embedded hard feature selection, introducing binary variables directly into the dual kernel formulation to impose explicit cardinality constraints on the selected features. We then derive equivalent reformulations of the resulting nonconvex MINLP as Convex Mixed-Integer Quadratic Programs by exploiting the multilinear structure induced by the polynomial kernel expansion. The proposed reformulations admit convex continuous relaxations and can therefore be efficiently handled within a branch-and-bound framework using standard MIQP solvers.

1.1 Related work

We review the literature on embedded feature selection in SVMs. Most of the existing literature addresses the problem of feature selection in the context of linear SVMs, where the classifier operates directly in the input feature space. The most used approach is regularization, in which a sparsity-inducing penalty term is added to the SVM objective, encouraging the weight vector w to have few nonzero components ([8, 16, 41, 21, 32, 42, 39, 43]). While computationally tractable, regularization-based methods do not provide direct control over the number of selected features. Additionally, tuning of the regularization parameter to achieve a specific number or range of features together with good predictive performance can be particularly challenging [7].

When the user wishes to enforce exactly B features, a hard cardinality constraint is required and mixed-integer programming formulations are needed. Among these, [10] replaced the explicit cardinality constraint with a nonlinear condition involving ℓ_1 and ℓ_2-norms, admitting convex relaxations but potentially violating the upper bound on the feature budget. [17] addressed this weakness via an alternative relaxation based on bisection to adjust the parameters iteratively. A bilevel formulation solved via a genetic algorithm was proposed in [1]. Mixed-

Integer Linear Programming models with explicit cardinality constraints were introduced in [27] and extended in [24], the latter also incorporating margin maximization and proposing both heuristic and exact solution methods. [3] introduced binary mask variables in the ℓ_2-regularized SVM primal, solved via Generalized Benders Decomposition, though convergence difficulties were reported on larger instances. Most recently, [7] proposed a mixed-integer formulation with complementarity constraints and developed both exact and heuristic solution approaches based on semidefinite programming relaxations. Most of the above works operate on the primal linear SVM formulation, where binary variables act on the weight vector $w \in \mathbb{R}^n$, while feature□selection techniques for nonlinear SVM have not been much explored due to their substantially greater computational complexity. This approach does not extend to nonlinear SVMs, where w lives in a high-dimensional or infinite-dimensional space that is never computed explicitly. Working with nonlinearly separable data requires the use of kernel functions, which evaluate inner products in the transformed feature space without explicitly computing the mapping. We are interested in selecting features in the *original input space* and not in the projected kernel space, as considered in [41], so that the selected features retain their interpretability.

Most embedded methods for nonlinear SVMs are based on regularization approaches, where a penalization term is added to the objective function to seek a trade-off between the standard classification objective and the complexity of the classifier, without enforcing a hard feature constraint. Among regularization-based approaches, [31] proposed various continuous formulations solved via difference-of-convex programming; [26] added an ℓ_0-norm approximation to the dual objective, requiring the tuning of six hyperparameters; [2] introduced ℓ_1-penalization solved via alternating optimization; and [22] reformulated the problem as a min-max program and proposed a local optimization strategy. Approaches based on binary variables for nonlinear SVMs have received comparatively limited attention in the literature, mainly due to the significantly higher computational complexity induced by the resulting nonconvex mixed-integer optimization models. [40] minimized a radius-margin bound on the leave-one-out error of the hard-margin SVM, solving a penalized continuous problem via gradient methods. [29] proposed a mixed-integer model for nonlinear kernel classifiers with reduced features, but based on the generalized SVM paradigm of [28] rather than the standard dual formulation, and developed an algorithm to find a stationary point. Neither work enforces an explicit feature budget constraint nor aims at globally optimal solutions.

Gap addressed by this paper. Despite extensive literature on feature selection in linear SVMs, the integration of hard cardinality constraints within nonlinear kernel-based SVM formulations remains largely unexplored, especially in settings that have global optimality guarantees. Existing approaches either rely on relaxations or heuristic methods, which do not provide certificates of global optimality. This paper addresses this gap by introducing globally solvable mixed-integer convex reformulations of polynomial kernel SVMs with embedded feature selection.

1.2 Contribution

Feature selection in nonlinear SVMs poses a fundamental computational challenge, as embedding binary feature selection variables directly into the kernel evaluation leads to a Mixed-Integer Nonlinear Program (MINLP) whose continuous relaxation is nonconvex, making it intractable for standard solvers. While several approaches have been proposed for linear SVMs, the nonlinear case with hard feature selection constraints and global optimality guarantees remains largely unaddressed in the literature.

Our main contribution is twofold. First, we introduce a Convex MIQP reformulation of the original MINLP, obtained by linearizing, through auxiliary variables, the multilinear interactions between the binary feature-selection variables and the continuous dual variables arising from the polynomial kernel expansion. We prove that the resulting continuous relaxation is convex and establish the formal equivalence between the reformulation and the original MINLP. By exploiting the inherent symmetry of the kernel expansion, the auxiliary variables can be indexed without redundancy, yielding a symmetry-free model that avoids the need for explicit

symmetry-breaking constraints. Second, we derive a more compact reformulation that aggregates the auxiliary variables over the training samples, eliminating their dependence on the number of samples m, while preserving both convexity and the equivalence guarantee. Both reformulations extend naturally to polynomial kernels of arbitrary degree d. Computational experiments demonstrate that the resulting Convex MIQPs can be effectively solved to certified global optimality on benchmark datasets, particularly for the quadratic kernel, whereas the original nonconvex formulation is computationally intractable already on small instances.

1.3 Problem statement

For ease of notation, we denote $[m] := \{1, \dots, m\}$ and $[n] := \{1, \dots, n\}$, the sets of sample and feature indices, respectively. We consider a binary classification problem defined on a dataset $\{(x^i, y^i) \in \mathbb{R}^n \times \{-1, 1\},\ i \in [m]\}$, where x^i is the *feature vector* of sample i and $y^i \in \{-1, +1\}$ its *label*. Support Vector Machines [38] are supervised learning models designed to construct a separating hyperplane maximizing the margin between the two classes. To capture nonlinear decision boundaries, data are mapped into a higher dimensional feature space $\mathcal{H}$ via a nonlinear mapping $\phi : \mathbb{R}^n \to \mathcal{H}$. The classifier is defined as the function $f : \mathbb{R}^n \to \{-1, 1\}$ given by

$$f(x) = \text{sgn}({w^*}^\top \phi(x) + b^*), \tag{1}$$

where the tuple (w^*, b^*) is obtained as the optimal solution of the convex quadratic optimization problem

$$\begin{array}{rll} g^* = \min\limits_{w,b,\xi} & \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^m \xi_i & \\ \text{s.t.} & y^i(w^\top \phi(x^i) + b) \geq 1 - \xi_i & i \in [m], \\ & \xi_i \geq 0 & i \in [m]. \end{array} \tag{2}$$

A linear classifier is then learned in the transformed space $\mathcal{H}$, while computations are performed through a *kernel function* $k(x^i, x^h) := \langle \phi(x^i), \phi(x^h) \rangle$, which evaluates inner products in $\mathcal{H}$ directly from the input space, without requiring an explicit computation of ϕ. Indeed, a kernel can correspond to different maps ϕ in different spaces. The linear SVM formulation is recovered by taking ϕ to be the identity map. In the nonlinear SVM setting, it is standard to work with the dual formulation of the SVM model [25]:

$$\begin{array}{rll} f^* = \max\limits_{\alpha} & \mathbf{1}^\top \alpha - \frac{1}{2}\,\alpha^\top Q\,\alpha & \\ \text{s.t.} & \sum\limits_{i=1}^m y^i \alpha_i = 0 & \\ & 0 \leq \alpha_i \leq C, & i \in [m], \end{array} \tag{3}$$

where $Q \in \mathbb{R}^{m \times m}$ is the kernel matrix with entries $q_{ih} = y^i y^h k(x^i, x^h)$, and $C > 0$ is a regularization parameter. Since k is a kernel function, it is by definition positive semidefinite. Therefore, the matrix Q is symmetric positive semidefinite, and problem (3) is a convex quadratic program. Thus, strong duality holds between problems (2) and (3), so that $g^* = f^*$. Common choices of kernel include the polynomial kernel $k(x, z) = (\gamma x^\top z + c)^d$ and the Gaussian kernel $k(x, z) = \exp(-\gamma\|x - z\|^2)$, with $\gamma > 0$, $c \geq 0$, $d \geq 1$.

Given an optimal solution α^*, the SVM classifier is defined as

$$\text{sgn}\left(\sum_{i=1}^m \alpha_i^*\, y^i\, k(x^i, x) + b^*\right), \tag{4}$$

where b^* is recovered via standard complementary slackness conditions [13].

Feature selection in SVMs is often modeled via mixed-integer formulations. Classical approaches for linear SVMs are developed in the primal space and associate the binary variables directly with the primal weights, so that a weight is forced to zero whenever the corresponding feature is not selected [24, 3, 7]. This does not

extend to nonlinear SVMs, where w lives in the transformed space $\mathcal{H}$ and its components no longer correspond to input features. Following [29, 3], we instead introduce binary variables $\beta \in \{0,1\}^n$ that mask the *input* vectors, replacing x^i by $\beta \odot x^i$ inside the feature map. Specifically, $\beta_j = 1$ indicates that feature j is selected, while $\beta_j = 0$ excludes it. When feature selection is embedded, we define the feature-dependent kernel as

$$k_\beta(x^i, x^h) := k(\beta \odot x^i, \beta \odot x^h) = \langle \phi(\beta \odot x^i), \phi(\beta \odot x^h) \rangle,$$

where $\odot$ denotes the elementwise (Hadamard) product. For a given selection β, the classifier is obtained by replacing k with k_β in the expression (4), with α^* and b^* computed for the same β. The corresponding feature-selection SVM primal problem is

$$g^*_{FS} = \min_{w,b,\xi,\beta} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^m \xi_i \tag{5a}$$

$$\text{s.t.} \quad y^i\big(w^\top \phi(\beta \odot x^i) + b\big) \geq 1 - \xi_i, \quad i \in [m], \tag{5b}$$

$$\textstyle\sum_{j=1}^n \beta_j = B, \tag{5c}$$

$$\xi_i \geq 0, \quad i \in [m], \tag{5d}$$

$$\beta \in \{0,1\}^n. \tag{5e}$$

The cardinality constraint (5c) enforces that exactly B features are selected, where B is a user-defined parameter. Note that for $B = 1$, the model reduces to selecting a single feature, thus the optimal solution can be obtained by training n standard SVMs, one for each individual feature, and selecting the best one. Similarly, $B = n - 1$ reduces to selecting the single feature to discard, requiring the solution of n SVMs with one feature removed. Therefore, in the remainder of the paper we consider the non-trivial range

$$2 \leq B \leq n - 2.$$

In the experiments we also report, for reference, some budgets outside this range (e.g., $B = n - 1$ for the smallest datasets and $B = n$, which reduces to the standard SVM).

The feature map $\phi(\cdot)$ is not uniquely defined, and the nonlinear SVM model is usually defined only through kernel evaluations. Consequently, Problem (5) cannot be solved directly, since ϕ is typically high or infinite dimensional and is not available explicitly. Nevertheless, we can still work in the primal by using the SVM optimality conditions to eliminate w, so that the problem depends on the data only through kernel evaluations k_β, as detailed below.

Eliminating w via the optimality conditions. For any fixed β, problem (5) is a convex quadratic program with linear constraints, so the KKT conditions are necessary and sufficient for optimality. Let $\alpha_i \geq 0$ be the Lagrange multipliers associated with the margin constraints. Stationarity with respect to w yields

$$w = \sum_{i=1}^m \alpha_i\, y^i\, \phi(\beta \odot x^i). \tag{6}$$

Stationarity with respect to b yields

$$\sum_{i=1}^m y^i \alpha_i = 0, \tag{7}$$

while stationarity with respect to ξ together with dual feasibility gives

$$0 \leq \alpha_i \leq C, \quad i \in [m]. \tag{8}$$

The last two are the standard SVM dual feasibility conditions.

Substituting (6) into (5) and using $k_\beta(x^i, x^h) = \langle \phi(\beta \odot x^i), \phi(\beta \odot x^h) \rangle$ and $(Q_\beta)_{ih} = y^i y^h k_\beta(x^i, x^h)$, the objective becomes

$$\tfrac{1}{2} \Big\| \sum_{i=1}^{m} \alpha_i y^i \phi(\beta \odot x^i) \Big\|^2 = \tfrac{1}{2} \sum_{i=1}^{m} \sum_{h=1}^{m} y^i y^h \alpha_i \alpha_h \, k_\beta(x^i, x^h) = \tfrac{1}{2} \, \alpha^\top Q_\beta \, \alpha.$$

For the margin constraints (5b), expanding the product and using $(Q_\beta)_{ih} = y^i y^h k_\beta(x^i, x^h)$ gives

$$y^i \Big(\sum_{h=1}^{m} \alpha_h y^h k_\beta(x^i, x^h) + b \Big) = \sum_{h=1}^{m} \alpha_h \, (Q_\beta)_{ih} + y^i b = (Q_\beta \alpha)_i + y^i b,$$

so the i-th margin constraint becomes

$$(Q_\beta \alpha)_i + y^i b \;\geq\; 1 - \xi_i, \qquad i \in [m].$$

We thus obtain the single-level problem

$$\widetilde{g}^*_{FS} = \min_{\alpha, b, \xi, \beta} \quad \frac{1}{2} \, \alpha^\top Q_\beta \, \alpha + C \sum_{i=1}^{m} \xi_i \tag{9a}$$

$$\text{s.t.} \quad (Q_\beta \alpha)_i + y^i b \;\geq\; 1 - \xi_i, \quad i \in [m], \tag{9b}$$

$$\xi_i \geq 0, \quad i \in [m], \tag{9c}$$

$$\sum_{i=1}^{m} y^i \alpha_i = 0, \tag{9d}$$

$$0 \leq \alpha_i \leq C, \quad i \in [m], \tag{9e}$$

$$\textstyle\sum_{j=1}^{n} \beta_j = B, \quad \beta \in \{0,1\}^n. \tag{9f}$$

Although not strictly required for the equivalence with Problem (5), we retain the conditions $0 \leq \alpha_i \leq C$ and $\sum_i y^i \alpha_i = 0$ in Problem (9), since they provide valid bounds that help the solver: in our tests, the box constraint on α accounts for most of the strengthening, while the equality constraint has a smaller, complementary effect. Problem (9) has a nonconvex quadratic objective function and nonlinear multilinear constraints. In the following we show how to make the objective function convex and how to linearize the constraints.

Polynomial kernel. In this work, we consider polynomial kernels of the form

$$k_\beta(x^i, x^h) = \left(\gamma \sum_{j=1}^{n} \beta_j x^i_j \beta_j x^h_j + c \right)^d = \left(\gamma \sum_{j=1}^{n} \beta_j x^i_j x^h_j + c \right)^d,$$

where $c \geq 0$, $\gamma > 0$ and $d \geq 2$.

Reduction to homogeneous kernels. For simplicity and without loss of generality, we restrict to the case of homogeneous polynomials, i.e., $c = 0$. Indeed, as already observed in [15], any polynomial kernel with $c \neq 0$ can be rewritten as a homogeneous polynomial by augmenting the input vectors with an additional fictitious feature. Specifically, defining $\tilde{x}^i = (\sqrt{c/\gamma}, \, x^i)^\top \in \mathbb{R}^{n+1}$ yields

$$\Big(\gamma \, {x^i}^\top x^h + c \Big)^d = \Big(\gamma \, (\tilde{x}^i)^\top \tilde{x}^h \Big)^d.$$

In other words, the kernel in $\mathbb{R}^n$ is equivalent to a homogeneous kernel in $\mathbb{R}^{n+1}$. In the augmented space, the additional feature is constant across all samples and is always selected (i.e., $\beta_{n+1} = 1$), so the cardinality budget effectively applies only to the original n features.

Organization. The remainder of the paper is organized as follows. Section 2 reviews the linearization techniques used throughout the paper. Section 3 presents the proposed formulations for the quadratic kernel: a baseline nonconvex model (Sec. 3.1), the convex reformulation (Sec. 3.2) and its compact variant (Sec. 3.3). Section 4 extends both reformulations to polynomial kernels of arbitrary degree d and compares their sizes. Section 5 reports the computational experiments, including the warm-start heuristic, the comparison between formulations, and the predictive accuracy as a function of the budget. Section 6 concludes the paper.

2 Review on linearization techniques

Our reformulations rely on the exact linearization of products involving a bounded continuous variable and one or more binary variables. We briefly recall two strategies to linearize these terms. These strategies are inspired by the literature on linearization or quadrification of polynomials. We refer the reader for example to [14]. Given a number $k \geq 1$ and a monomial $\mathcal{M}$:

$$\mathcal{M} = a \prod_{j=1}^{k} b_j, \text{ with } a \in [a^L, a^U] \text{ and } b_j \in \{0,1\}, \tag{10}$$

The lower and upper bounds on variable a should satisfy $a^L < a^U$ and w.l.o.g we shall suppose $a^L \leq 0$ and $a^U \geq 0$. Otherwise, a^L can be replaced with $\min(0, a^L)$ and a^U can be replaced with $\max(0, a^U)$, which are also valid lower and upper bounds.

Standard linearization. This linearization requires the addition of a single variable z and the following set of $2k+2$ constraints

$$\begin{aligned}
&z \leq a^U b_j && \forall j \in [k], \\
&z \geq a^L b_j && \forall j \in [k], \\
&z \leq a - a^L \left(k - \sum_{j \in [k]} b_j \right), \\
&z \geq a - a^U \left(k - \sum_{j \in [k]} b_j \right).
\end{aligned} \tag{11}$$

Together with the initial constraints $a \in [a^L, a^U]$ and $b_j \in \{0,1\}$, Constraints (11) enforce z to be equal to $\mathcal{M}$.

Recursive linearization. Following a standard quadrification strategy, define auxiliary variables recursively as:

$$z^1 = ab_1, \qquad z^j = z^{j-1} b_j \quad j = 2, \ldots, k, \tag{12}$$

Each quadratic identity $z^j = z^{j-1} b_j$ can be enforced like in the above standard linearization (11) for the product of one bounded and one binary variable and adding 4 constraints. Using the bounds $a \in [a^L, a^U]$ and the fact that the variables b_j can be only 0 or 1, then $z^j \in [a^L, a^U]$ propagates for each j. Here, together with the initial constraints $a \in [a^L, a^U]$ and $b_j \in \{0,1\}$, Constraints (12) enforce z^k to be equal to $\mathcal{M}$. The precise

linearization is then:

$$
\begin{aligned}
& z^1 \leq a^U b_1, \\
& z^1 \geq a^L b_1, \\
& z^1 \leq a - a^L(1 - b_1), \\
& z^1 \geq a - a^U(1 - b_1), \\
& z^j \leq a^U b_j && \forall j \in [2:k], \\
& z^j \geq a^L b_j && \forall j \in [2:k], \\
& z^j \leq z^{j-1} - a^L(1 - b_j) && \forall j \in [2:k], \\
& z^j \geq z^{j-1} - a^U(1 - b_j) && \forall j \in [2:k].
\end{aligned}
\tag{13}
$$

The recursive linearization introduces k auxiliary continuous variables z^j and $4k$ linear constraints. It is known to yield a tighter continuous relaxation than the standard linearization. It is particularly interesting when one considers a polynomial that contains monomial $\mathcal{M}$ together with other monomials that are included in it. In this case the auxiliary variables z^j can be factorized with other monomials.

3 Mathematical Formulations

We aim to solve problem (9) to proven global optimality. This formulation is a Mixed-Integer Nonlinear Programming (MINLP) problem, whose continuous relaxation is nonconvex. The nonconvexity, combined with the presence of integer variables, makes the problem practically intractable for general-purpose global optimization solvers. Our main contribution is to derive two equivalent Convex Mixed-Integer Quadratic reformulations for polynomial kernels of arbitrary degree d, that enable the problem to be solved to certified global optimality by standard MIQP solvers.

Before introducing our reformulations, we analyze a natural baseline approach based on standard McCormick linearization. This preliminary step highlights the intrinsic difficulty of the problem, showing that such a straightforward approach fails to produce a tractable formulation.

In the following, for simplicity and clarity of exposition, we consider the case of polynomial kernels of degree $d = 2$. The extension to arbitrary degree d is reported in Section 4. For $d = 2$ and $c = 0$, the objective function of problem (9) is

$$
\frac{1}{2} \sum_{i=1}^{m} \sum_{h=1}^{m} y^i y^h \alpha_i \alpha_h \left(\gamma \sum_{j=1}^{n} \beta_j x_j^i x_j^h \right)^2 + C \sum_{i=1}^{m} \xi_i. \tag{14}
$$

It involves products of two continuous variables and two binary variables, which leads to a nonconvex formulation.

3.1 Baseline McCormick nonconvex quadratic formulation

We introduce a baseline reformulation of problem (9) obtained by applying McCormick linearization to the products between bounded continuous dual variables and binary feature-selection variables. Although this strategy preserves exactness, we show that it is insufficient to recover convexity of the continuous relaxation, motivating the need for a different approach.

Expanding the squared term in (14), the objective function can be written as

$$
\begin{aligned}
& \frac{\gamma^2}{2} \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} \sum_{i=1}^{m} \sum_{h=1}^{m} \left(y^i x_{j_1}^i x_{j_2}^i \right) \left(y^h x_{j_1}^h x_{j_2}^h \right) (\alpha_i \beta_{j_1})(\alpha_h \beta_{j_2}) + C \sum_{i=1}^{m} \xi_i = \\
& \frac{\gamma^2}{2} \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} \sum_{i=1}^{m} \sum_{h=1}^{m} \Delta_i^{j_1,j_2} \Delta_h^{j_1,j_2} (\alpha_i \beta_{j_1})(\alpha_h \beta_{j_2}) + C \sum_{i=1}^{m} \xi_i,
\end{aligned}
\tag{15}
$$

where we denote

$$\Delta_i^{j_1,j_2} := y^i x_{j_1}^i x_{j_2}^i. \tag{16}$$

We define the following auxiliary variables to model the products appearing in the objective function (15):

$$v_{i,j} \triangleq \alpha_i \beta_j \quad i \in [m],\ j \in [n].$$

Since $\alpha_i \in [0, C]$ and $\beta_j \in \{0, 1\}$, these products can be linearized exactly via the standard linearization (11).

The variables $v_{i,j}$ are however not sufficient to linearize the margin constraints (9b), which also depend on β through $(Q_\beta \alpha)_i$.

Expanding the quadratic kernel $k_\beta(x^i, x^h) = \left(\gamma \sum_{j=1}^n \beta_j x_j^i x_j^h\right)^2 = \gamma^2 \sum_{j_1=1}^n \sum_{j_2=1}^n \beta_{j_1}\beta_{j_2}\, x_{j_1}^i x_{j_2}^i\, x_{j_1}^h x_{j_2}^h$ in

$$(Q_\beta \alpha)_i = \sum_{h=1}^m y^i y^h k_\beta(x^i, x^h) \alpha_h,$$

exchanging the order of summation and distributing y^i, y^h as in (16), we obtain

$$(Q_\beta \alpha)_i = \gamma^2 \sum_{j_1=1}^n \sum_{j_2=1}^n \Delta_i^{j_1,j_2} \sum_{h=1}^m \Delta_h^{j_1,j_2}\, \alpha_h \beta_{j_1} \beta_{j_2},$$

so that constraint (9b) becomes

$$\gamma^2 \sum_{j_1=1}^n \sum_{j_2=1}^n \Delta_i^{j_1,j_2} \sum_{h=1}^m \Delta_h^{j_1,j_2}\, \alpha_h \beta_{j_1} \beta_{j_2} + y^i b \ \geq\ 1 - \xi_i, \qquad i \in [m].$$

This constraint involves the triple product $\alpha_h \beta_{j_1} \beta_{j_2}$, which is not captured by the auxiliary variable $v_{i,j} = \alpha_i \beta_j$ alone, already used to linearize the objective. We therefore introduce a second auxiliary variable

$$z_i^{j_1,j_2} = v_{i,j_1}\, \beta_{j_2} = \alpha_i \beta_{j_1} \beta_{j_2}, \qquad i \in [m],\ j_1, j_2 \in [n],$$

obtained by applying McCormick linearization once more, this time to the product of the continuous variable $v_{i,j_1} \in [0, C]$ and the binary variable β_{j_2}. Substituting $\alpha_h \beta_{j_1} \beta_{j_2}$ with $z_h^{j_1,j_2}$ in the constraint above yields the linear margin constraint (17b), together with the exact linearization constraints (17h)–(17k) for z.

The resulting baseline formulation of problem (9) is obtained by substituting $\alpha_i \beta_j$ with $v_{i,j}$ in the kernel expansion and $\alpha_i \beta_{j_1} \beta_{j_2}$ with $z_i^{j_1,j_2}$ in the margin constraints, and by adding the corresponding linearization constraints (17d)–(17g) and (17h)–(17k):

$$
\begin{aligned}
(\texttt{MIQP}) \quad \min_{\alpha,b,\xi,\beta,v,z} \quad & \frac{\gamma^2}{2} \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} \sum_{i=1}^{m} \sum_{h=1}^{m} \Delta_i^{j_1,j_2} \Delta_h^{j_1,j_2} \, v_{i,j_1} v_{h,j_2} + C \sum_{i=1}^{m} \xi_i && && \text{(17a)} \\
\text{s.t.} \quad & \\
& \gamma^2 \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} \Delta_i^{j_1,j_2} \sum_{h=1}^{m} \Delta_h^{j_1,j_2} \, z_h^{j_1,j_2} + y^i b \;\geq\; 1 - \xi_i, && i \in [m] && \text{(17b)} \\
& \xi_i \geq 0, && i \in [m] && \text{(17c)} \\
& v_{i,j} \leq C\beta_j, && i \in [m],\, j \in [n] && \text{(17d)} \\
& v_{i,j} \leq \alpha_i, && i \in [m],\, j \in [n] && \text{(17e)} \\
& v_{i,j} \geq \alpha_i - C(1 - \beta_j), && i \in [m],\, j \in [n] && \text{(17f)} \\
& v_{i,j} \geq 0, && i \in [m],\, j \in [n] && \text{(17g)} \\
& z_i^{j_1,j_2} \leq C\beta_{j_2}, && i \in [m],\, j_1, j_2 \in [n] && \text{(17h)} \\
& z_i^{j_1,j_2} \leq v_{i,j_1}, && i \in [m],\, j_1, j_2 \in [n] && \text{(17i)} \\
& z_i^{j_1,j_2} \geq v_{i,j_1} - C(1 - \beta_{j_2}), && i \in [m],\, j_1, j_2 \in [n] && \text{(17j)} \\
& z_i^{j_1,j_2} \geq 0, && i \in [m],\, j_1, j_2 \in [n] && \text{(17k)} \\
& \sum_{i=1}^{m} y^i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C, && i \in [m] && \text{(17l)} \\
& \sum_{j=1}^{n} \beta_j = B, \quad \beta_j \in \{0, 1\}, && j \in [n] && \text{(17m)}
\end{aligned}
$$

This reformulation replaces the original objective function of degree four (15) with a quadratic one (17a). Note that, unlike the objective, the margin constraints can be linearized exactly through $z_i^{j_1,j_2}$, since $(Q_\beta \alpha)_i$ is linear in α for fixed β: each term involves the product of a *single* continuous variable α_h with the binary variables $\beta_{j_1}\beta_{j_2}$, which McCormick linearization handles exactly. However, although all products involving binary variables have been linearized exactly, the objective (17a) still contains quadratic terms in v that, in general, do not define a convex quadratic form. An example showing nonconvexity is reported in Appendix A.

This limitation is structural, as the variables $v_{i,j}$ capture only pairwise interactions between a dual variable α_i and a single feature j, but do not encode the joint feature activations (j_1, j_2) required by the polynomial kernel expansion. In other words, the quadratic form in (17a) couples different samples and features simultaneously.

Thus a straightforward application of standard linearization techniques is insufficient to obtain a tractable reformulation. This observation motivates the approach developed in the following subsection, where we exploit the structure of the kernel expansion to introduce auxiliary variables that explicitly encode joint feature activations. This allows the objective function to be rewritten as a sum of squared linear terms, leading to a convex MIQP formulation.

3.2 A new convex quadratic formulation

The analysis of the baseline reformulation shows that linearizing pairwise products of variables is insufficient to recover convexity, as this fails to capture the joint feature interactions induced by the polynomial kernel. We now present a reformulation that directly models these interactions, leading to a convex quadratic objective.

Since $\beta_j \in \{0,1\}$ implies $\beta_j^2 = \beta_j$, we can write the quadratic term of (15) as

$$\frac{\gamma^2}{2} \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} \Big(\sum_{i=1}^{m} \Delta_i^{j_1,j_2} \alpha_i \beta_{j_1} \beta_{j_2} \Big) \Big(\sum_{h=1}^{m} \Delta_h^{j_1,j_2} \alpha_h \beta_{j_1} \beta_{j_2} \Big),$$

which is a sum of squared terms:

$$\frac{\gamma^2}{2} \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} \left(\sum_{i=1}^{m} \Delta_i^{j_1,j_2} \alpha_i \beta_{j_1} \beta_{j_2} \right)^2 + C \sum_{i=1}^{m} \xi_i. \tag{18}$$

We further exploit the symmetry $\Delta_i^{j_2,j_1} = \Delta_i^{j_1,j_2}$. Thus, we decompose the double sum over the feature indices into diagonal and off-diagonal contributions, restricting the latter to $j_1 < j_2$. More precisely, separating the cases $j_1 = j_2$ (where we also use again the identity $\beta_j^2 = \beta_j$) and $j_1 \neq j_2$, we obtain the following writing of the objective function, which will be used throughout the remainder of the paper:

$$\frac{\gamma^2}{2} \left[\sum_{j=1}^{n} \left(\sum_{i=1}^{m} \Delta_i^{j,j} \alpha_i \beta_j \right)^2 + 2 \sum_{j_1=1}^{n} \sum_{j_2>j_1} \left(\sum_{i=1}^{m} \Delta_i^{j_1,j_2} \alpha_i \beta_{j_1} \beta_{j_2} \right)^2 \right] + C \sum_{i=1}^{m} \xi_i. \tag{19}$$

The key observation is that the multilinear terms $\alpha_i \beta_j$ and $\alpha_i \beta_{j_1} \beta_{j_2}$ in (19) are the only remaining source of nonconvexity. Our idea is to replace them with auxiliary variables in order to obtain convex quadratic functions. We do this following the recursive linearization (13). To this end, for each sample $i \in [m]$ and feature $j \in [n]$, we first define

$$z_i^j \triangleq \alpha_i \beta_j, \tag{20}$$

and then, for each pair $j_1 < j_2$, we define

$$z_i^{j_1,j_2} \triangleq z_i^{j_1} \beta_{j_2}. \tag{21}$$

We linearize identity (20) by Constraints (22d)–(22g), and identity (21) by Constraints (22h)–(22k).

We now express the margin constraints (9b) in terms of the same auxiliary variables z_i^j and $z_i^{j_1,j_2}$. Recall from the baseline reformulation that

$$(Q_\beta \alpha)_i = \gamma^2 \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} \Delta_i^{j_1,j_2} \sum_{h=1}^{m} \Delta_h^{j_1,j_2} \, \alpha_h \beta_{j_1} \beta_{j_2},$$

and apply to it the same diagonal/off-diagonal decomposition used for the objective. For $j_1 = j_2$, using $\beta_j^2 = \beta_j$, the diagonal terms reduce to $\gamma^2 \sum_j \Delta_i^{j,j} \sum_h \Delta_h^{j,j} \alpha_h \beta_j$. For $j_1 \neq j_2$, the symmetry $\Delta_i^{j_1,j_2} = \Delta_i^{j_2,j_1}$ implies that the contributions of (j_1, j_2) and (j_2, j_1) coincide, so summing over all ordered pairs with $j_1 \neq j_2$ is equivalent to twice the sum over $j_1 < j_2$. Combining both contributions yields

$$(Q_\beta \alpha)_i = \gamma^2 \left[\sum_{j=1}^{n} \Delta_i^{j,j} \sum_{h=1}^{m} \Delta_h^{j,j} \, \alpha_h \beta_j + 2 \sum_{j_1=1}^{n} \sum_{j_2>j_1}^{n} \Delta_i^{j_1,j_2} \sum_{h=1}^{m} \Delta_h^{j_1,j_2} \, \alpha_h \beta_{j_1} \beta_{j_2} \right].$$

Unlike the objective, this expression is already linear in α for fixed β, so substituting $\alpha_h \beta_j$ with z_h^j and $\alpha_h \beta_{j_1} \beta_{j_2}$ with $z_h^{j_1,j_2}$ directly yields the linear margin constraint (22b), using the same auxiliary variables already introduced for the objective.

We obtain the following convex MIQP formulation for polynomial kernels of degree $d = 2$:

$$(\texttt{ConvexMIQP}) \quad \min_{\alpha,b,\xi,\beta,z} \quad \frac{\gamma^2}{2}\left[\sum_{j=1}^{n}\left(\sum_{i=1}^{m}\Delta_i^{j,j}z_i^j\right)^2 + 2\sum_{j_1=1}^{n}\sum_{j_2>j_1}^{n}\left(\sum_{i=1}^{m}\Delta_i^{j_1,j_2}z_i^{j_1,j_2}\right)^2\right] + C\sum_{i=1}^{m}\xi_i \quad (22a)$$

s.t.

$$\gamma^2\sum_{h=1}^{m}\left[\sum_{j=1}^{n}\Delta_i^{j,j}\Delta_h^{j,j}z_h^j + 2\sum_{j_1=1}^{n}\sum_{j_2>j_1}^{n}\Delta_i^{j_1,j_2}\Delta_h^{j_1,j_2}z_h^{j_1,j_2}\right] + y^i b \;\geq\; 1-\xi_i, \quad i\in[m] \quad (22b)$$

$$\xi_i \geq 0, \qquad i\in[m] \quad (22c)$$

(9d), (9e), (9f)

$$z_i^j \leq \alpha_i, \qquad \forall i\in[m], j\in[n] \quad (22d)$$

$$z_i^j \leq C\beta_j, \qquad \forall i\in[m], j\in[n] \quad (22e)$$

$$z_i^j \geq \alpha_i - C(1-\beta_j), \qquad \forall i\in[m], j\in[n] \quad (22f)$$

$$z_i^j \geq 0 \qquad \forall i\in[m], j\in[n] \quad (22g)$$

$$z_i^{j_1,j_2} \leq z_i^{j_1}, \qquad i\in[m],\ j_1<j_2 \quad (22h)$$

$$z_i^{j_1,j_2} \leq C\beta_{j_2}, \qquad i\in[m],\ j_1<j_2 \quad (22i)$$

$$z_i^{j_1,j_2} \geq z_i^{j_1} - C(1-\beta_{j_2}), \qquad i\in[m],\ j_1<j_2 \quad (22j)$$

$$z_i^{j_1,j_2} \geq 0, \qquad i\in[m],\ j_1<j_2 \quad (22k)$$

Proposition 1. *Problem* (22) *is a convex MIQP equivalent to* (9)*.*

Proof. The proof is a particular case ($d = 2$) of the proof of Proposition 3 for general degree (Section 4). In short, convexity holds because the objective is a nonnegative combination of squared linear functions of z plus a linear term in ξ, and all constraints are linear. Equivalence follows from the exactness of the McCormick linearizations (11): for any fixed binary β, Constraints (22d)–(22k) force $z_i^j = \alpha_i\beta_j$ and $z_i^{j_1,j_2} = \alpha_i\beta_{j_1}\beta_{j_2}$, so that the objective and the margin constraints coincide with those of (9). □

Computational tricks. To improve numerical stability and solver efficiency, in the computational experiments of Section 5 we introduce auxiliary variables

$$t^{j,j} = \sum_{i=1}^{m}\Delta_i^{j,j}z_i^j \qquad \forall j\in[n],$$

$$t^{j_1,j_2} = \sum_{i=1}^{m}\Delta_i^{j_1,j_2}z_i^{j_1,j_2} \qquad \forall\, 1\leq j_1<j_2\leq n.$$

With this notation, the objective function can be rewritten as

$$\frac{\gamma^2}{2}\left[\sum_{j=1}^{n}\left(t^{j,j}\right)^2 + 2\sum_{j_1=1}^{n}\sum_{j_2>j_1}^{n}\left(t^{j_1,j_2}\right)^2\right] + C\sum_{i=1}^{m}\xi_i.$$

This eliminates repeated quadratic expressions involving the z variables and provides a more numerically stable representation of the objective. From a computational perspective, it reduces the density of the quadratic structure seen by the solver and improves scaling, which typically leads to faster convergence and more stable branch-and-bound behavior. The same trick extends to $d > 2$.

3.3 A compact convex quadratic formulation

The `ConvexMIQP` formulation derived in the previous section involves $O(mn^2)$ auxiliary variables z_i^j and $z_i^{j_1,j_2}$, one for each feature or feature pair retained by the recursive construction. Coming back to the expression of the objective function (19), we now show, following ideas from the compact linearizations in [18], that the dependence of the auxiliary variables on the sample index i can be eliminated by aggregating over i, reducing the number of auxiliary variables from $O(mn^2)$ to $O(n^2)$.

For each feature index j and for each pair of feature indices $j_1 < j_2$, we define the aggregated quantities

$$E_j(\alpha) \triangleq \sum_{i=1}^{m} \Delta_i^{j,j} \alpha_i, \qquad E_{j_1,j_2}(\alpha) \triangleq \sum_{i=1}^{m} \Delta_i^{j_1,j_2} \alpha_i.$$

These quantities are linear functions of α and they aggregate the contribution of all training samples for a given feature interaction. Since the binary variables do not depend on the sample index, they can be taken out of the sums over i in (19), which becomes

$$\frac{\gamma^2}{2}\left[\sum_{j=1}^{n} (E_j \beta_j)^2 + 2\sum_{j_1=1}^{n}\sum_{j_2>j_1}^{n} (E_{j_1,j_2}\beta_{j_1}\beta_{j_2})^2\right] + C\sum_{i=1}^{m}\xi_i. \tag{23}$$

The objective function (23) contains products of the form $E_j\beta_j$ and $E_{j_1,j_2}\beta_{j_1}\beta_{j_2}$, that are respectively bilinear and trilinear monomials given by the product of a continuous variable and binary variables. Suppose bounds L_j, U_j on E_j and bounds L_{j_1,j_2}, U_{j_1,j_2} on E_{j_1,j_2} are known; then these monomials can be linearized exactly via the standard linearization (11). We use the auxiliary variables

$$u^j \triangleq E_j\beta_j, \quad \forall j \in [n], \qquad u^{j_1,j_2} \triangleq E_{j_1,j_2}\beta_{j_1}\beta_{j_2}, \quad \forall j_1, j_2 \in [n],\ j_1 < j_2. \tag{24}$$

As shown before, the i-th margin constraint (9b) involves $(Q_\beta\alpha)_i$, which, using the same diagonal/off-diagonal decomposition, can be written as

$$(Q_\beta\alpha)_i = \gamma^2\left[\sum_{j=1}^{n}\Delta_i^{j,j}\Big(\sum_{h=1}^{m}\Delta_h^{j,j}\alpha_h\beta_j\Big) + 2\sum_{j_1=1}^{n}\sum_{j_2>j_1}^{n}\Delta_i^{j_1,j_2}\Big(\sum_{h=1}^{m}\Delta_h^{j_1,j_2}\alpha_h\beta_{j_1}\beta_{j_2}\Big)\right].$$

Remarkably, the terms in parentheses coincide exactly with the aggregated products defined in (24), since $\sum_{h=1}^{m}\Delta_h^{j,j}\alpha_h\beta_j = E_j\beta_j = u^j$ and $\sum_{h=1}^{m}\Delta_h^{j_1,j_2}\alpha_h\beta_{j_1}\beta_{j_2} = E_{j_1,j_2}\beta_{j_1}\beta_{j_2} = u^{j_1,j_2}$. No additional auxiliary variables are needed: the margin constraint is linear in the same variables u^j, u^{j_1,j_2} used in the objective, and reads

$$\gamma^2\left[\sum_{j=1}^{n}\Delta_i^{j,j}\,u^j + 2\sum_{j_1=1}^{n}\sum_{j_2>j_1}^{n}\Delta_i^{j_1,j_2}\,u^{j_1,j_2}\right] + y^i b \;\geq\; 1-\xi_i, \qquad i \in [m].$$

The resulting compact convex MIQP (`CompactMIQP`), equivalent to problem (9), is reported below, where Constraints (25e)–(25f) and (25h)–(25k) enforce the equalities (24).

$$
\begin{aligned}
(\texttt{CompactMIQP}) \quad \min_{\alpha,b,\xi,\beta,E,u} \quad & \frac{\gamma^2}{2}\left[\sum_{j=1}^{n}(u^j)^2 + 2\sum_{j_1=1}^{n}\sum_{j_2>j_1}^{n}(u^{j_1,j_2})^2\right] + C\sum_{i=1}^{m}\xi_i && \text{(25a)}\\
\text{s.t.} \quad & \\
& \gamma^2\left[\sum_{j=1}^{n}\Delta_i^{j,j}\,u^j + 2\sum_{j_1=1}^{n}\sum_{j_2>j_1}^{n}\Delta_i^{j_1,j_2}\,u^{j_1,j_2}\right] + y^i b \;\geq\; 1-\xi_i, \qquad i\in[m] && \text{(25b)}\\
& \xi_i \geq 0, && i\in[m] \quad \text{(25c)}\\
& \text{(9d), (9e), (9f)} && \\
& E_j = \sum_{i=1}^{m}\Delta_i^{j,j}\alpha_i, && \forall j\in[n], \quad \text{(25d)}\\
& L_j\beta_j \leq u^j \leq U_j\beta_j, && \forall j\in[n], \quad \text{(25e)}\\
& E_j - U_j(1-\beta_j) \leq u^j \leq E_j - L_j(1-\beta_j), && \forall j\in[n], \quad \text{(25f)}\\
& E_{j_1,j_2} = \sum_{i=1}^{m}\Delta_i^{j_1,j_2}\alpha_i, && \forall\, j_1<j_2, \quad \text{(25g)}\\
& L_{j_1,j_2}\beta_{j_1} \leq u^{j_1,j_2} \leq U_{j_1,j_2}\beta_{j_1}, && \forall\, j_1<j_2, \quad \text{(25h)}\\
& L_{j_1,j_2}\beta_{j_2} \leq u^{j_1,j_2} \leq U_{j_1,j_2}\beta_{j_2}, && \forall\, j_1<j_2, \quad \text{(25i)}\\
& E_{j_1,j_2} - U_{j_1,j_2}(2-\beta_{j_1}-\beta_{j_2}) \leq u^{j_1,j_2}, && \forall\, j_1<j_2, \quad \text{(25j)}\\
& u^{j_1,j_2} \leq E_{j_1,j_2} - L_{j_1,j_2}(2-\beta_{j_1}-\beta_{j_2}), && \forall\, j_1<j_2. \quad \text{(25k)}
\end{aligned}
$$

The objective function of problem (25) is convex, as it consists of a nonnegative sum of squared linear functions in the variables u, plus a linear term. Additionally, all constraints are linear, therefore the continuous relaxation of problem (25) is a convex quadratic program. The formulation is valid for any lower and upper bounds L and U under the condition that they are correct for any feasible solution in α and β. However, the strength of the continuous relaxation critically depends on the tightness of L and U.

Proposition 2. *Problem* (25) *is a convex MIQP equivalent to* (9)*, for any valid bounds* (L, U).

Proof. The proof is a particular case ($d = 2$) of the proof of Proposition 4 for general degree (Section 4.2). □

Computing $L_j, U_j, L_{j_1,j_2}, U_{j_1,j_2}$. A key component of the compact formulation is the availability of tight bounds on the aggregated quantities E_j and E_{j_1,j_2}. These bounds can be computed easily and efficiently, since

they are obtained by solving simple linear programs over the feasible region of the dual variables α. For each pair $j_1 \leq j_2$, the lower and upper bounds on E_{j_1,j_2} are given by the optimal values of the following linear programs:

$$L_{j_1,j_2} = \min_{(7),(8)} \quad \sum_{i=1}^{m} \Delta_i^{j_1,j_2} \alpha_i \tag{26a}$$

$$U_{j_1,j_2} = \max_{(7),(8)} \quad \sum_{i=1}^{m} \Delta_i^{j_1,j_2} \alpha_i \tag{26b}$$

The bounds L_j and U_j are obtained analogously by setting $j_1 = j_2$. Since the feasible region is common to all bound computations, each problem is a small linear program whose size depends only on the number of training samples. Consequently, all bounds can be precomputed before solving the mixed-integer formulation. By (7) and (8), $\alpha = 0$ is always feasible, thus the optimal solutions of problems (26) satisfy $L_{j_1,j_2} \leq 0$ and $U_{j_1,j_2} \geq 0$.

4 Extension to Higher-Degree Polynomial Kernels

The convex reformulations developed in Sections 3.2–3.3 for the quadratic kernel ($d = 2$) extend naturally to polynomial kernels of arbitrary degree d. In this section we introduce the general notation and rewrite the objective function in a form that exposes the same hidden sum-of-squares structure exploited in the quadratic case. This representation is then used to derive the two reformulations, which we denote again as Convex MIQP and Compact MIQP.

Kernel expansion and monomial structure. Consider the homogeneous polynomial kernel of degree d with feature selection,

$$k_\beta(x^i, x^h) = \left(\gamma \sum_{j=1}^{n} \beta_j x_j^i x_j^h \right)^d .$$

By the multinomial theorem [6], raising the inner sum to the d-th power produces one monomial for each possible selection of d terms from the sum (with repetition). Each monomial is uniquely characterized by the multiplicity t_j of each feature j, i.e., the number of times the j-th term appears in the product. Hence, the associated exponent vector $t = (t_1, \ldots, t_n) \in \mathbb{N}^n$ satisfies $\sum_{j=1}^{n} t_j = d$. We collect all such exponent vectors in the index set

$$T \triangleq \left\{ t \in \mathbb{N}^n : \sum_{j=1}^{n} t_j = d \right\}, \qquad |T| = \binom{n+d-1}{d},$$

whose cardinality equals the number of multisets of size d drawn from n features. Each $t \in T$ corresponds to a distinct monomial in the expansion, weighted by the multinomial coefficient

$$c_t = \frac{d!}{t_1! \cdots t_n!},$$

which counts the number of ordered selections of the d terms from the sum that correspond to the same monomial (equivalently, to the same exponent vector t). Grouping the expansion by exponent vector, the kernel can be written compactly as

$$k_\beta(x^i, x^h) = \gamma^d \sum_{t \in T} c_t \prod_{j=1}^{n} \left(\beta_j x_j^i x_j^h \right)^{t_j} .$$

Support sets and binary simplification. For each $t \in T$, we define its support as

$$J(t) \triangleq \{j \in [n] : t_j > 0\}.$$

The support is a set of feature indices and therefore contains each feature at most once, independently of its multiplicity in the exponent vector t. Since $\beta_j \in \{0,1\}$, for every $j \in J(t)$ we have $\beta_j^{t_j} = \beta_j$, thus

$$\prod_{j=1}^{n} \beta_j^{t_j} = \prod_{j \in J(t)} \beta_j^{t_j} = \prod_{j \in J(t)} \beta_j.$$

As a consequence, the binary component of a monomial depends only on its support and not on the vector t. Furthermore, the cardinality constraint (5c) allows exactly B features to be selected. Let

$$\widehat{d} = \min\{B, d\}.$$

Then, any monomial associated with a support of size larger than $\widehat{d}$ is identically zero at every feasible solution. Such monomials and all related quantities can therefore be removed without loss of exactness. We denote by

$$\widehat{T} \triangleq \{t \in T : \ |J(t)| \leq \widehat{d}\}$$

the set of retained monomials.

Sum-of-squares structure of the objective. We now apply the procedure introduced for degree $d = 2$ to polynomial kernels of arbitrary degree d. We consider the quadratic term of the objective,

$$\frac{1}{2} \sum_{i=1}^{m} \sum_{h=1}^{m} y^i y^h \alpha_i \alpha_h \left(\gamma \sum_{j=1}^{n} \beta_j x_j^i x_j^h \right)^d = \frac{\gamma^d}{2} \sum_{i=1}^{m} \sum_{h=1}^{m} y^i y^h \alpha_i \alpha_h \left(\sum_{t \in T} c_t \prod_{j=1}^{n} \left(\beta_j x_j^i x_j^h \right)^{t_j} \right). \tag{27}$$

Exchanging the order of summation and using $\prod_{j \in J(t)} \beta_j = \left(\prod_{j \in J(t)} \beta_j \right)^2$, the right-hand side becomes

$$\frac{\gamma^d}{2} \sum_{t \in T} c_t \left(\sum_{i=1}^{m} y^i \alpha_i \prod_{j \in J(t)} \left(x_j^i\right)^{t_j} \right) \left(\sum_{h=1}^{m} y^h \alpha_h \prod_{j \in J(t)} \left(x_j^h\right)^{t_j} \right) \left(\prod_{j \in J(t)} \beta_j \right)^2.$$

For each $i \in [m]$ and $t \in T$, define

$$\Delta_{i,t} = y^i \prod_{j \in J(t)} (x_j^i)^{t_j}.$$

Note that $\Delta_{i,t}$ depends on the full exponent structure, and not only on the support $J(t)$: monomials sharing the same support but different exponent patterns generally yield different values of $\Delta_{i,t}$. Thus, we have

$$\frac{\gamma^d}{2} \sum_{t \in \widehat{T}} c_t \left(\sum_{i=1}^{m} \Delta_{i,t} \, \alpha_i \prod_{j \in J(t)} \beta_j \right)^2, \tag{28}$$

where the sum has been restricted to the retained monomials $t \in \widehat{T}$. This representation preserves the sum-of-squares structure of the objective, while the remaining nonconvexity is entirely due to the multilinear terms involving the binary variables β. By introducing suitable auxiliary variables, we obtain two equivalent convex MIQP reformulations, extending the `ConvexMIQP` and `CompactMIQP` formulations developed for the quadratic case ($d = 2$) to arbitrary polynomial degrees.

4.1 Convex MIQP Reformulation of the Degree-d Polynomial Kernel

We now extend the `ConvexMIQP` formulation introduced in Section 3.2 for the quadratic kernel to polynomial kernels of arbitrary degree d.

Nested support subsets. In (28), we aim to linearize the multilinear term

$$\alpha_i \prod_{j \in J(t)} \beta_j, \qquad t \in \widehat{T}.$$

Since this product depends only on the support $J(t)$ and not on the exponent vector t, monomials with the same support share the same binary-continuous multilinear product in the objective function. We collect all such distinct supports generated by the retained monomials in

$$\mathcal{P} \triangleq \{J(t) : t \in \widehat{T}\} = \{P \subseteq [n] : 1 \leq |P| \leq \widehat{d}\},$$

that is, all combinations of at most $\widehat{d}$ feature indices out of the n available ones. For each support $P = \{j_1, \ldots, j_\ell\} \in \mathcal{P}$, with $j_1 < \cdots < j_\ell$, we define its nested chain $\{P_0, P_1, \ldots, P_\ell\}$ as

$$P_0 = \emptyset, \qquad P_h = \{j_1, \ldots, j_h\}, \quad h = 1, \ldots, \ell, \qquad \text{with} \qquad P_0 \subset P_1 \subset \cdots \subset P_\ell = P.$$

Since the ordering of the feature indices is fixed, each support induces a unique sequence of nested subsets. This nested-chain construction follows the same principle as recursive McCormick (or recursive arithmetic interval) schemes for multilinear terms [37, 34]. Unlike these general schemes, which allow arbitrary choices in the recursive decomposition, our construction recursively extends the product according to the fixed increasing order of the feature indices.

Auxiliary variables. For each sample $i \in [m]$, we set

$$z_{i,P_0} = z_{i,\emptyset} = \alpha_i.$$

For each support $P = \{j_1, \ldots, j_\ell\} \in \mathcal{P}$, with $j_1 < \cdots < j_\ell$, the remaining auxiliary variables are defined recursively as

$$z_{i,P_{h+1}} = z_{i,P_h}\, \beta_{j_{h+1}}, \qquad h = 0, \ldots, \ell - 1. \tag{29}$$

Therefore, for every $t \in \widehat{T}$, observing that the set $J(t)$ with $|J(t)| = \ell$ is exactly $P_\ell = \{j_1, \ldots, j_\ell\}$, the terminal variable of the chain satisfies

$$z_{i,J(t)} = \alpha_i \prod_{j \in J(t)} \beta_j,$$

which is exactly the multilinear product appearing in the objective. Since $z_{i,P}$ depends only on the subset P and not on the monomial that generated it, monomials sharing a common nested support also share the corresponding auxiliary variable. Hence, the number of auxiliary variables scales with $|\mathcal{P}|$, the number of subsets involved in the recursive construction, which is at most the number of retained monomials $|\widehat{T}|$ (with equality for $d = 2$). The recursive construction generalizes (20)–(21), where the recursive chains have length at most two.

Example. Consider a polynomial kernel of degree $d = 3$ with $n = 4$ features. A monomial $\beta_1^{t_1}\beta_2^{t_2}\beta_3^{t_3}\beta_4^{t_4}$ is represented by its exponent vector $t \in T \subseteq \mathbb{N}^4$ with $\sum_j t_j = d$. For example, the exponent vectors

$$t^{(1)} = (1, 0, 2, 0), \qquad t^{(2)} = (2, 0, 1, 0)$$

correspond to the monomials $\beta_1\beta_3^2$ and $\beta_1^2\beta_3$, respectively. Although the two monomials are different, they have the same support $J(t^{(1)}) = J(t^{(2)}) = \{1, 3\}$. Therefore, both terms in the objective are represented by the same multilinear product $\alpha_i\beta_1\beta_3$ and share the same auxiliary variable $z_{i,\{1,3\}}$.

Now consider the monomial associated with $t^{(3)} = (1, 0, 1, 1)$, whose support is $J(t^{(3)}) = \{1, 3, 4\}$. Following the fixed ordering of the feature indices, this support generates the nested chain

$$\emptyset \subset \{1\} \subset \{1, 3\} \subset \{1, 3, 4\},$$

and the corresponding multilinear product in the objective function is constructed recursively as

$$\alpha_i \beta_1 \beta_3 \beta_4 = \underbrace{\underbrace{\underbrace{\underbrace{\alpha_i}_{z_{i,\emptyset}} \beta_1}_{z_{i,\{1\}}} \beta_3}_{z_{i,\{1,3\}}} \beta_4}_{z_{i,\{1,3,4\}}}.$$

In particular, the intermediate variable $z_{i,\{1,3\}}$ is the same variable already used for the monomials $t^{(1)}$ and $t^{(2)}$, and is therefore reused rather than duplicated.

Linearization. Each recursion step in (29) involves the product of a bounded continuous variable and a binary variable, where $z_{i,P} \in [0, C]$ for every $P \in \mathcal{P}$, with the convention $z_{i,P_0} = \alpha_i \in [0, C]$. Each such product can therefore be linearized exactly using the standard McCormick envelope for a binary-continuous product

$$\begin{aligned} &z_{i,P_{h+1}} \le z_{i,P_h}, \\ &z_{i,P_{h+1}} \le C\,\beta_{j_{h+1}}, \\ &z_{i,P_{h+1}} \ge z_{i,P_h} - C(1 - \beta_{j_{h+1}}), \\ &z_{i,P_{h+1}} \ge 0, \end{aligned} \qquad \forall i \in [m],\ P \in \mathcal{P},\ h = 0, \dots, |P| - 1. \tag{30}$$

Since a nested subset may appear in multiple chains, its auxiliary variable and the corresponding constraints in (30) are introduced only once.

Formulation. Similarly to the objective, the i-th margin constraint (9b) of Problem (9) involves $(Q_\beta \alpha)_i$. Following the same steps used above for the objective (substituting the kernel expansion and exchanging the order of summation), we obtain

$$(Q_\beta \alpha)_i = \gamma^d \sum_{t \in \widehat{T}} c_t\, \Delta_{i,t} \sum_{h=1}^{m} \Delta_{h,t}\, z_{h,J(t)}.$$

Unlike the objective, this expression is already linear in the auxiliary variables z (no squaring is involved, since the margin constraint is linear in α for fixed β), so substituting $z_{h,J(t)}$, already defined and linearized above, directly yields the linear margin constraint (31b) below, with no new variables. Substituting $z_{i,J(t)}$ for the multilinear products in (28) and enforcing (30) for every $P \in \mathcal{P}$, we obtain the following convex MIQP:

$$(\texttt{ConvexMIQP}^d) \qquad \min_{\alpha, b, \xi, \beta, z} \quad \frac{\gamma^d}{2} \sum_{t \in \widehat{T}} c_t \left(\sum_{i=1}^{m} \Delta_{i,t}\, z_{i,J(t)} \right)^2 + C \sum_{i=1}^{m} \xi_i \tag{31a}$$

s.t.

$$\gamma^d \sum_{t \in \widehat{T}} c_t\, \Delta_{i,t} \sum_{h=1}^{m} \Delta_{h,t}\, z_{h,J(t)} + y^i b \ \ge\ 1 - \xi_i, \qquad i \in [m] \tag{31b}$$

$$\xi_i \ge 0, \qquad i \in [m] \tag{31c}$$

(9d), (9e), (9f)

$$(30) \qquad i \in [m],\ P \in \mathcal{P} \tag{31d}$$

Proposition 3. *Problems* (9) *and* (31) *are equivalent, and the objective of* (31) *is convex.*

Proof. Convexity follows from the fact that the objective is a nonnegative linear combination of squared linear functions of z, plus a linear term in ξ. For equivalence, fix any feasible $\beta \in \{0,1\}^n$. We prove by induction on ℓ that, for every $P_\ell = \{j_1, \dots, j_\ell\} \in \mathcal{P}$,

$$z_{i,P_\ell} = \alpha_i \prod_{h=1}^{\ell} \beta_{j_h}.$$

For $\ell = 1$, this follows from the McCormick linearization of $z_{i,P_1} = z_{i,P_0}\beta_{j_1}$, with $z_{i,P_0} = \alpha_i \in [0, C]$. Assuming the claim holds for $P_{\ell-1}$, the same argument applied to

$$z_{i,P_\ell} = z_{i,P_{\ell-1}}\beta_{j_\ell}$$

gives the result for P_ℓ. Hence, for every retained monomial $t \in \widehat{T}$,

$$z_{i,J(t)} = \alpha_i \prod_{j \in J(t)} \beta_j,$$

and substituting these identities into the objective and into the margin constraints recovers the original kernel expansion, proving equivalence. □

4.2 Compact Convex MIQP Reformulation (`CompactMIQP`^{d})

Aggregated variables. Following the idea introduced in Section 3.3, we aggregate the contribution of all training samples associated with each exponent vector $t \in \widehat{T}$ and define the aggregated quantity

$$E_t = \sum_{i=1}^{m} \Delta_{i,t}\, \alpha_i, \qquad t \in \widehat{T}, \tag{32}$$

which collects the contribution of all training samples for the monomial t, weighted by the corresponding dual variables. Each E_t is a linear function of α only, decoupling the data-dependent part of the objective from the binary feature-selection variables β. Unlike the previous formulation, these variables are indexed by monomials rather than supports, since E_t depends on the full exponent vector t and not only on its support $J(t)$. Using this notation, the quadratic term of the objective function (28) can be rewritten entirely in terms of E_t and β:

$$\frac{\gamma^d}{2} \sum_{t \in \widehat{T}} c_t \left(E_t \prod_{j \in J(t)} \beta_j \right)^2.$$

We therefore introduce one auxiliary variable per monomial:

$$u_t = E_t \prod_{j \in J(t)} \beta_j, \qquad t \in \widehat{T},$$

so that the complete objective function becomes

$$\frac{\gamma^d}{2} \sum_{t \in \widehat{T}} c_t\, u_t^2 + C \sum_{i=1}^{m} \xi_i,$$

which is a convex quadratic function in u.

Bounding the aggregated terms. Valid bounds $L_t \le E_t \le U_t$ (with $L_t \le 0, U_t \ge 0$) are required to linearize the products $u_t = E_t \prod_{j \in J(t)} \beta_j$. As in Section 3.3, these are obtained by solving, for each $t \in \widehat{T}$, the pair of optimization problems

$$(L_t,\, U_t) = \left(\min_{\alpha} \sum_{i=1}^{m} \Delta_{i,t}\, \alpha_i,\ \max_{\alpha} \sum_{i=1}^{m} \Delta_{i,t}\, \alpha_i \right) \quad \text{s.t.} \quad \sum_{i=1}^{m} y^i \alpha_i = 0,\ 0 \le \alpha_i \le C, \tag{33}$$

which are linear programs whose size depends only on m and can be solved efficiently in the preprocessing phase prior to solving the MIQP. Since $\alpha = 0$ is feasible with zero value, we always get $L_t \le 0, U_t \ge 0$.

Linearization. Applying the standard linearization strategy (11), the product $u_t = E_t \prod_{j \in J(t)} \beta_j$ can be enforced exactly via

$$\begin{aligned}
&u_t \le U_t\, \beta_j && \forall j \in J(t),\\
&u_t \ge L_t\, \beta_j && \forall j \in J(t),\\
&u_t \ge E_t - U_t \left(|J(t)| - \sum_{j \in J(t)} \beta_j \right), &&\\
&u_t \le E_t - L_t \left(|J(t)| - \sum_{j \in J(t)} \beta_j \right). &&
\end{aligned} \tag{34}$$

These constraints enforce $u_t = 0$ whenever at least one $\beta_j = 0$ for $j \in J(t)$, while guaranteeing $u_t = E_t$ when $\beta_j = 1$ for all $j \in J(t)$, which generalizes the compact quadratic reformulation to arbitrary monomial supports of cardinality up to $\widehat{d}$. The tightness of the relaxation depends on the sharpness of (L_t, U_t).

Formulation. Similarly to the objective, the i-th margin constraint (9b) of Problem (9) involves $(Q_\beta \alpha)_i = \sum_{h=1}^{m} y^i y^h k_\beta(x^i, x^h) \alpha_h$. Substituting the kernel expansion $k_\beta(x^i, x^h) = \gamma^d \sum_{t \in T} c_t \prod_{j=1}^{n} (\beta_j x_j^i x_j^h)^{t_j}$ and the binary simplification $\prod_{j=1}^{n} \beta_j^{t_j} = \prod_{j \in J(t)} \beta_j$ established above, and exchanging the order of summation exactly as done for the objective, we obtain

$$(Q_\beta \alpha)_i = \gamma^d \sum_{t \in \widehat{T}} c_t \Big(\prod_{j \in J(t)} \beta_j \Big) \Big(y^i \prod_{j \in J(t)} (x_j^i)^{t_j} \Big) \sum_{h=1}^{m} \Big(y^h \prod_{j \in J(t)} (x_j^h)^{t_j} \Big) \alpha_h.$$

Using $\Delta_{i,t} = y^i \prod_{j \in J(t)} (x_j^i)^{t_j}$ and recalling $E_t = \sum_{h=1}^{m} \Delta_{h,t} \alpha_h$ from (32) and $u_t = E_t \prod_{j \in J(t)} \beta_j$, the product of the binary term and E_t is exactly u_t, so

$$(Q_\beta \alpha)_i = \gamma^d \sum_{t \in \widehat{T}} c_t\, \Delta_{i,t}\, u_t.$$

Unlike the objective, this expression is already linear in the auxiliary variables u (no squaring is involved, since the margin constraint is linear in α for fixed β), so it does not require any further linearization beyond (34), already introduced for the objective. Substituting into constraint (9b) yields the linear margin constraint (35b)

below, expressed in the same variables u_t used in the objective. The full degree-d compact convex MIQP is:

$$(\texttt{CompactMIQP}^d) \quad \min_{\alpha,b,\xi,\beta,E,u} \quad \frac{\gamma^d}{2}\sum_{t\in\widehat{T}} c_t\, u_t^2 + C\sum_{i=1}^{m}\xi_i \tag{35a}$$

s.t.

$$\gamma^d \sum_{t\in\widehat{T}} c_t\,\Delta_{i,t}\,u_t + y^i b \;\geq\; 1-\xi_i \quad i\in[m] \tag{35b}$$

$$\xi_i \geq 0 \qquad i\in[m] \tag{35c}$$

(9d), (9e), (9f)

$$(32) \qquad \forall\, t\in\widehat{T} \tag{35d}$$

$$(34) \qquad \forall\, t\in\widehat{T} \tag{35e}$$

Proposition 4. *Problems* (9) *and* (35) *are equivalent for any valid bounds* (L_t, U_t)*, and the objective of* (35) *is convex.*

Proof. Convexity follows immediately since the objective is a nonnegative linear combination of squared linear terms in u, plus a linear term in ξ. For equivalence, fix any feasible (α, β). The defining equations for E_t uniquely determine the aggregated quantities. Consider a monomial $t \in \widehat{T}$. If at least one feature in its support is not selected, i.e., $\beta_j = 0$ for some $j \in J(t)$, the first two groups of constraints in (34) imply

$$u_t \leq U_t\beta_j = 0, \qquad u_t \geq L_t\beta_j = 0,$$

and hence

$$u_t = 0 = E_t \prod_{j\in J(t)} \beta_j.$$

Conversely, if all features in $J(t)$ are selected, then $\beta_j = 1$ for every $j \in J(t)$. The first two constraints in (34) become redundant, while the last two reduce to

$$u_t \geq E_t, \qquad u_t \leq E_t,$$

which gives $u_t = E_t$. Hence, also in this case,

$$u_t = E_t \prod_{j\in J(t)} \beta_j.$$

Thus, the auxiliary variables exactly reproduce the multilinear terms in (28), and substituting them into the objective and the margin constraints recovers the original kernel expansion. □

4.3 Variable count comparison

We now compare the sizes of the two convex reformulations proposed for polynomial kernels of arbitrary degree d, namely the $\texttt{ConvexMIQP}^d$ and the $\texttt{CompactMIQP}^d$, together with the baseline of Section 3.1. Table 1 reports the asymptotic number of auxiliary continuous variables and of the additional linearization constraints introduced beyond the original MINLP (9). Structural constraints shared by all formulations are excluded from the count. For the $\texttt{CompactMIQP}^d$, the defining equalities of E_t are also omitted, since they only define aggregated continuous quantities and do not affect the combinatorial size of the formulation.

The table highlights the trade-off between model size and convexity. For $d = 2$, the baseline McCormick reformulation has the same order of growth as the ConvexMIQP (it does not exploit the symmetry of the kernel

Table 1: Comparison of the proposed formulations for a polynomial kernel of degree d: order of growth of the auxiliary continuous variables (# Aux. vars) and of the non-trivial additional constraints (# Add. constr.). The baseline is reported for $d = 2$. $\mathcal{P}$ is the set of distinct nested support subsets generated by the recursive construction, and $\widehat{T}$ is the set of retained monomials, as defined in Section 4, with $\widehat{d} = \min\{B, d\}$. The last column indicates whether the continuous relaxation is convex.

Formulation	**# Aux. vars**	**# Add. constr.**	**Convex relaxation**
`MINLP` (9)	$O(1)$	$O(1)$	No
`MIQP` (17)	$O(mn^2)$	$O(mn^2)$	No
`ConvexMIQP`^{d} (22), (31)	$O(m\,\lvert\mathcal{P}\rvert)$	$O(m\,\lvert\mathcal{P}\rvert)$	Yes
`CompactMIQP`^{d} (25), (35)	$O(\lvert\widehat{T}\rvert)$	$O(\widehat{d}\,\lvert\widehat{T}\rvert)$	Yes

expansion, so it introduces roughly twice as many variables of type z), but its continuous relaxation is nonconvex and therefore does not provide global optimality guarantees.

The auxiliary variables $z_{i,P}$ of the `ConvexMIQP`^{d} are indexed by samples $i \in [m]$ and by the nested support subsets $P \in \mathcal{P}$ generated during the recursive construction. Therefore, the formulation introduces $O(m|\mathcal{P}|)$ auxiliary variables. Since each recursive linearization step introduces only a constant number of constraints for each auxiliary variable, the number of additional constraints has the same order, namely $O(m|\mathcal{P}|)$. The `CompactMIQP`^{d} eliminates the dependence on the number of training samples in the auxiliary variables by aggregating the sample-dependent terms before linearization. Since in most applications the number of training samples m is large, the `CompactMIQP`^{d} can substantially reduce the model size compared to the `ConvexMIQP`^{d}.

5 Computational Experiments

5.1 Experimental setup

Datasets. We evaluate our approach on 10 benchmark binary classification datasets drawn from the UCI Machine Learning Repository. All features are standardized to zero mean and unit variance prior to training. Each dataset is randomly split into a training and a test set (80/20). The optimization experiments are performed on the training set, whose size m and number of features n are reported in Table 2; the test set is used only in Section 5.5 to assess predictive accuracy.

Table 2: Number of training samples m and features n per dataset after preprocessing.

Dataset	**Samples (m)**	**Features (n)**
Breast Cancer Diagnostic (BCD)	455	30
Breast Cancer Prognostic (BCP)	155	33
Breast Cancer Wisconsin (BCW)	359	9
Cleveland	237	13
Climate Model	432	18
German	800	24
Ionosphere	280	33
Parkinsons	156	22
Sonar	166	60
Spectf	80	44

Hyperparameters. We set $C = 1$ and $c = 0$. The kernel scale parameter is fixed to $\gamma = 1/n$, in line with the default setting of `scikit-learn` [33] and LIBSVM [12] after data standardization, since $\gamma = 1/(n \cdot \mathrm{Var}(X)) \approx 1/n$ when features are standardized to unit variance. The number of selected features B is varied over $\{2, 5, 8\}$ to assess performance across different sparsity levels.

Implementation and hardware. All formulations are implemented in Python using the Gurobi [20] solver (version 13.0). Experiments are run on an Intel(R) Xeon(R) Gold 5218 CPU with a time limit of 7200 seconds per instance. The implementation of all formulations and the scripts used to reproduce the computational experiments are publicly available at `https://github.com/IlariaCiocci/FS-SVM`.

Optimality gap. We report two gap measures. The *gap at node 0* (GAP_0) measures the quality of the root relaxation before the branch-and-bound process starts. The corresponding lower bound LB_0 is obtained by explicitly solving the continuous relaxation of the model via `model.relax()`, outside the branch-and-bound framework, so that it is not affected by any algorithmic choices made by Gurobi during the search (in particular, it does not include the cuts and heuristics applied by Gurobi at the root node). We define

$$\mathrm{GAP}_0 = \frac{\mathrm{UB}_{\mathrm{WS}} - \mathrm{LB}_0}{|\mathrm{UB}_{\mathrm{WS}}|},$$

where $\mathrm{UB}_{\mathrm{WS}}$ is the objective value of the solution provided by the warm-start heuristic (Section 5.2). This represents the initial optimality gap available at the beginning of the branch-and-bound process, combining the externally computed root relaxation with the best feasible solution available before any node is explored. The *final gap* (GAP_f) is the optimality gap reported by Gurobi at termination, computed as the relative distance between the best incumbent solution found during branch-and-bound and the best lower bound at termination:

$$\mathrm{GAP}_f = \frac{\mathrm{UB}_f - \mathrm{LB}_f}{|\mathrm{UB}_f|},$$

where UB_f is the value of the best integer-feasible solution found and LB_f is the final lower bound reported by Gurobi.

5.2 Warm start

Despite the convexity of the proposed MIQP reformulations, the problem remains computationally demanding, particularly for higher degrees d, owing to the presence of binary variables and to the high dimensionality of the formulation. Supplying a high-quality feasible solution is therefore essential to accelerate the branch-and-bound process. To this end, we use the local search (LS) heuristic of [15]: for a fixed feature-selection vector β, the optimization over α reduces to a standard SVM problem restricted to the selected features, so every candidate β can be evaluated exactly and the best one found provides a valid upper bound. The resulting solution is given to the solver as the initial incumbent.

5.3 Results

The original MINLP (9) cannot be handled directly by Gurobi, since the objective contains products of the form $\alpha_i \alpha_h \beta_{j_1} \beta_{j_2}$ that are not in a form accepted by the solver, which requires the objective to be linear or quadratic. We therefore compare the nonconvex baseline McCormick reformulation `MIQP` (17) with the compact convex reformulation `CompactMIQP` (25) and its degree-d version $\texttt{CompactMIQP}^d$ (35), which is the formulation whose size does not depend on the number of training samples (Section 4).

Baseline reformulation. The baseline McCormick reformulation `MIQP` (17) linearizes the products $\alpha_i\beta_j$ and $\alpha_i\beta_{j_1}\beta_{j_2}$ via auxiliary variables $v_{i,j}$ and $z_i^{j_1,j_2}$, reducing the original degree-four objective to a quadratic one. However, as shown in Section 3.1 and Appendix A, the resulting quadratic form is generally not convex, and the continuous relaxation of the baseline formulation remains nonconvex. As a consequence, Gurobi cannot certify global optimality and the formulation is not suitable for branch-and-bound with standard MIQP solvers. Table 3 reports the performance of the baseline formulation on three representative instances, chosen to illustrate different failure modes of the solution process, under the easiest configuration considered ($d = 2, B = 2$). Even in this minimal setting, the results are conclusive: `Cleveland` and `Parkinsons` reach final gaps of over 4000% and 3400%, respectively, after one hour, and `German` gets stuck during presolve and never starts the branch-and-bound. These results confirm that the nonconvex baseline is computationally intractable even on small instances, and motivate the need for the convex reformulations proposed in this work.

Table 3: Performance of the baseline formulation `MIQP` on three datasets, illustrating different failure modes ($d = 2, C = 1, B = 2$).

Dataset	(m, n)	# Nodes	Stop Criterion	GAP_f
Cleveland	(237,13)	1	Time Limit (3600 s)	4341.88%
German	(800,24)	–	Stuck in presolve	–
Parkinsons	(156,22)	1	Time Limit (3600 s)	3472.84%

`CompactMIQP` and $\texttt{CompactMIQP}^d$ reformulations. The reported computational times include all preprocessing operations required to build the reformulation. In particular, they include the computation of the valid bounds (L, U) used in the McCormick linearizations, and the time required to generate and provide the warm-start solution. We present results for degree $d = 2$ and $d = 3$.

Degree $d = 2$. Table 4 reports the results for the quadratic kernel. The `CompactMIQP` formulation is solved to certified optimality within the time limit on 27 out of the 30 instances. The three exceptions are `BCP` with $B = 8$ and `Sonar` with $B = 5$ and $B = 8$, which terminate with final gaps of 0.16%, 1.06% and 2.44%, respectively. Solving times increase with the budget B: on the larger datasets the time at $B = 8$ is between one and two orders of magnitude larger than at $B = 2$ (e.g., `BCD`: 217.7 s vs. 5051.2 s; `German`: 105.6 s vs. 3194.3 s; `Ionosphere`: 134.5 s vs. 4825.9 s; `Spectf`: 123.2 s vs. 3330.0 s), whereas the small dataset `BCW` is solved in about one second for all budgets. The root gap GAP_0 is often substantial (up to 43.8% on `Ionosphere`), so that certifying optimality relies on branch-and-bound rather than on the root relaxation alone.

Degree $d = 3$. Table 5 reports the results for $d = 3$, which are considerably harder. For $B = 2$, eight datasets out of ten (`BCD`, `BCP`, `BCW`, `Cleveland`, `Climate Model`, `German`, `Ionosphere` and `Parkinsons`) are solved to certified optimality, in at most 370.3 s, while `Sonar` and `Spectf` terminate with final gaps of 2.67% and 0.18%. For $B = 5$ only `BCW`, `Cleveland` and `Parkinsons` are solved to optimality, and for $B = 8$ only `BCW` and `Cleveland`; overall, 13 out of 30 instances are certified. For $B = 5$ and $B = 8$, the final gaps of the instances that are not solved range from 4.5% to 39.5%, and in some cases it is not much smaller than the root gap, meaning that branch-and-bound makes limited progress within the time limit. The difficulty of these instances is consistent with the growth of the model size with the degree (Section 4).

5.4 Computational performance across the feature budget

While Tables 4–5 report results for the three budgets $B \in \{2, 5, 8\}$, we now examine the trend across a wider range of budgets for the quadratic kernel ($d = 2$), expressed as a fraction B/n of the total number of features, ranging from 10% to 100% of the available features.

Table 4: `CompactMIQP` reformulation (degree $d = 2$): total time (s), root gap GAP_0 (%), and final gap GAP_f (%) for each feature budget B. Times close to 7200 s indicate that the time limit was reached.

	$B = 2$			$B = 5$			$B = 8$		
Dataset	Time	GAP_0	GAP_f	Time	GAP_0	GAP_f	Time	GAP_0	GAP_f
BCD	217.71	34.08	0.00	1113.72	27.27	0.00	5051.20	21.67	0.01
BCP	58.88	6.34	0.00	1408.33	8.64	0.00	7201.07	8.27	0.16
BCW	1.09	40.28	0.00	1.12	15.75	0.00	1.07	3.02	0.00
Cleveland	1.52	20.14	0.00	2.10	13.97	0.00	2.02	7.70	0.00
Climate Model	19.95	11.08	0.00	38.30	11.09	0.00	347.01	10.33	0.00
German	105.60	17.03	0.00	317.20	15.13	0.00	3194.34	13.76	0.00
Ionosphere	134.54	42.47	0.00	315.15	43.78	0.00	4825.85	38.55	0.00
Parkinsons	9.33	14.88	0.00	19.43	9.18	0.00	26.43	6.73	0.00
Sonar	1371.96	14.57	0.00	7201.22	29.85	1.06	7201.48	36.86	2.44
Spectf	123.24	16.11	0.01	895.08	21.73	0.01	3329.95	22.39	0.01

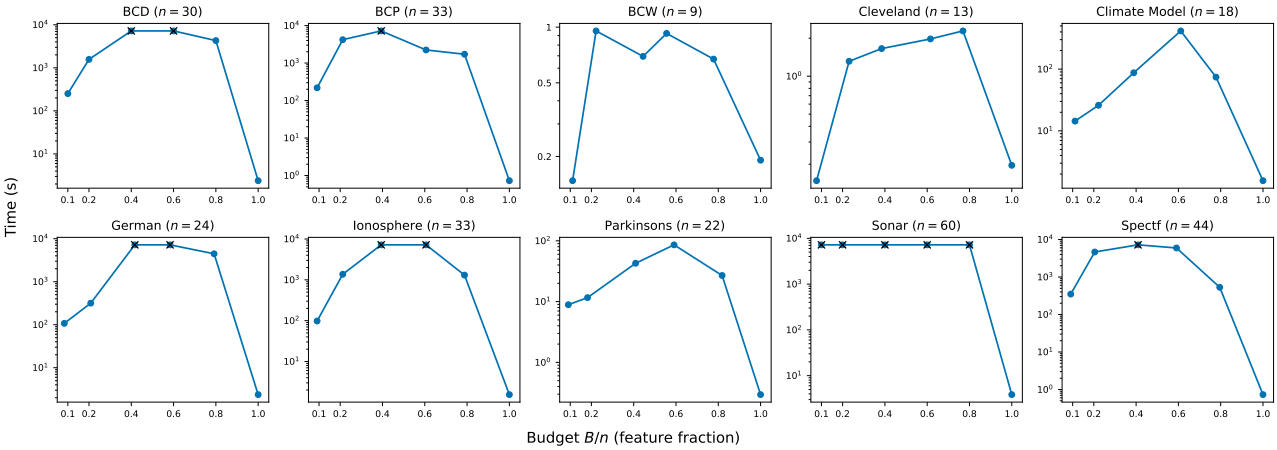

Figure 1: Solving time (s, log scale) of the `CompactMIQP` formulation as a function of the feature budget B/n, for degree $d = 2$. Black crosses mark instances that reached the 7200 s time limit.

Figure 1 shows that the solving time of the `CompactMIQP` formulation is not monotonic in the budget. It is lowest at the smallest budgets, grows over the intermediate range $B/n \in [0.4, 0.6]$, and decreases again for $B/n \geq 0.8$. In the intermediate range `BCD`, `BCP`, `German`, `Ionosphere` and `Spectf` reach the time limit (black crosses), and `Sonar` reaches it for all budgets up to $B/n = 0.8$, whereas the smaller datasets (`BCW`, `Cleveland`, `Climate Model` and `Parkinsons`) are solved in less than about 400 s at every budget. This is broadly consistent with the combinatorial nature of the cardinality constraint: the number of candidate feature subsets $\binom{n}{B}$ is largest for B close to $n/2$, and the constraint is least restrictive when B/n approaches its extremes. At $B/n = 1$ in particular, the constraint no longer restricts the choice of features, β is forced to the all-ones vector, and the problem reduces to solving a single convex QP, namely a standard SVM, which explains the sharp drop in solving time (a few seconds at most) at that extreme.

Table 5: `CompactMIQP`^{d} reformulation (degree $d = 3$): total time (s), root gap GAP_0 (%), and final gap GAP_f (%) for each feature budget B. Times close to 7200 s indicate that the time limit was reached. *Gurobi terminated with status 13 (suboptimal termination), which is neither a certified optimum nor the time limit.

	$B = 2$			$B = 5$			$B = 8$		
Dataset	Time	GAP_0	GAP_f	Time	GAP_0	GAP_f	Time	GAP_0	GAP_f
BCD	322.08	52.47	0.00	7221.01	54.04	38.19	7250.33	44.35	35.00
BCP	322.88	3.65	0.00	7204.44	17.04	16.04	7205.82	15.80	8.65
BCW	1.01	63.16	0.00	3.52	36.14	0.00	6.38	4.32	0.00
Cleveland	3.55	46.70	0.00	18.03	40.10	0.00	148.58	27.01	0.00
Climate Model	42.96	9.05	0.00	7201.60	37.80	4.46	7201.45	35.74	9.34
German	370.32	15.61	0.00	7214.11	30.59	13.25	7209.05	27.81	16.71
Ionosphere	144.81	18.87	0.00	7205.78	50.55	24.46	7214.28	47.06	28.80
Parkinsons	25.48	18.84	0.00	6723.87	18.33	0.00	7203.07	11.62	4.55
Sonar	7207.26	2.74	2.67	7218.90	34.27	33.78	7220.04	40.14	39.48
Spectf	3652.03*	11.54	0.18	7205.63	23.95	23.74	7205.57	22.46	22.04

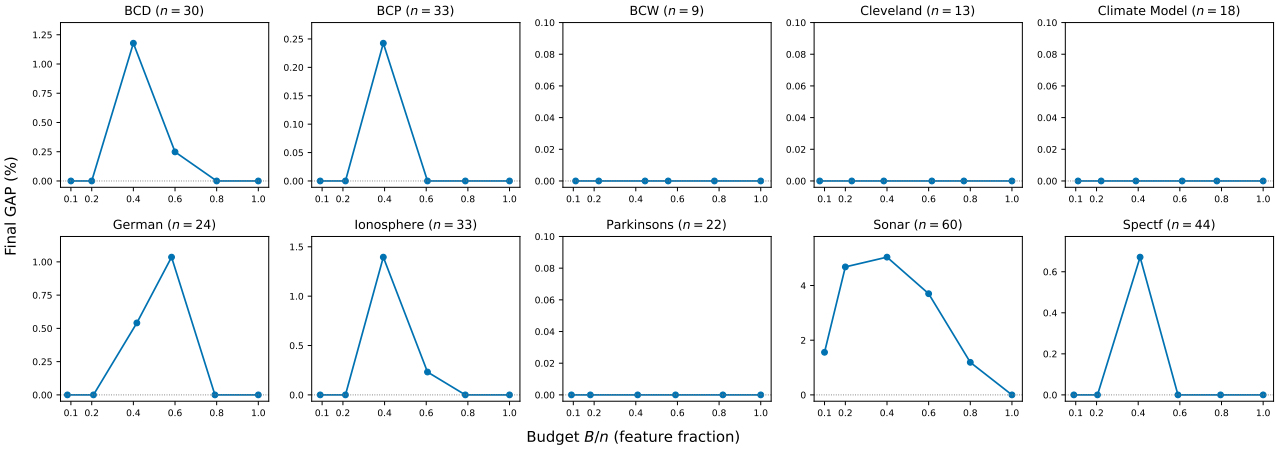

Figure 2: Final optimality gap GAP_f (%) of the `CompactMIQP` formulation as a function of the feature budget B/n, for degree $d = 2$. Note that the vertical scale differs across panels.

Figure 2 shows that, on the budgets of this grid, the final gap is different from zero only for intermediate budgets, i.e., in the same range where the time limit is reached (`BCD`, `BCP`, `German`, `Ionosphere`, `Spectf` and `Sonar`). It is below about 1.4% on all these datasets except `Sonar`, where it reaches about 5% at $B/n = 0.4$. The gap vanishes again for $B/n \geq 0.8$ on all datasets but `Sonar`, where it is still about 1.2% at $B/n = 0.8$, and the datasets `BCW`, `Cleveland`, `Climate Model` and `Parkinsons` are solved to optimality at every budget.

5.5 Accuracy varying the budget B

The previous experiments assess the optimization performance of the proposed reformulations for solving the MINLP model (9). In this section, we complement them by assessing the *machine-learning* performance of the resulting model. In particular, we report the training and test accuracy attained by the final predictive model

$$\texttt{Class}(x) = \text{sgn}\left(\sum_{i=1}^{m} \alpha_i^* y^i k_{\beta^*}(x^i, x) + b^*\right),$$

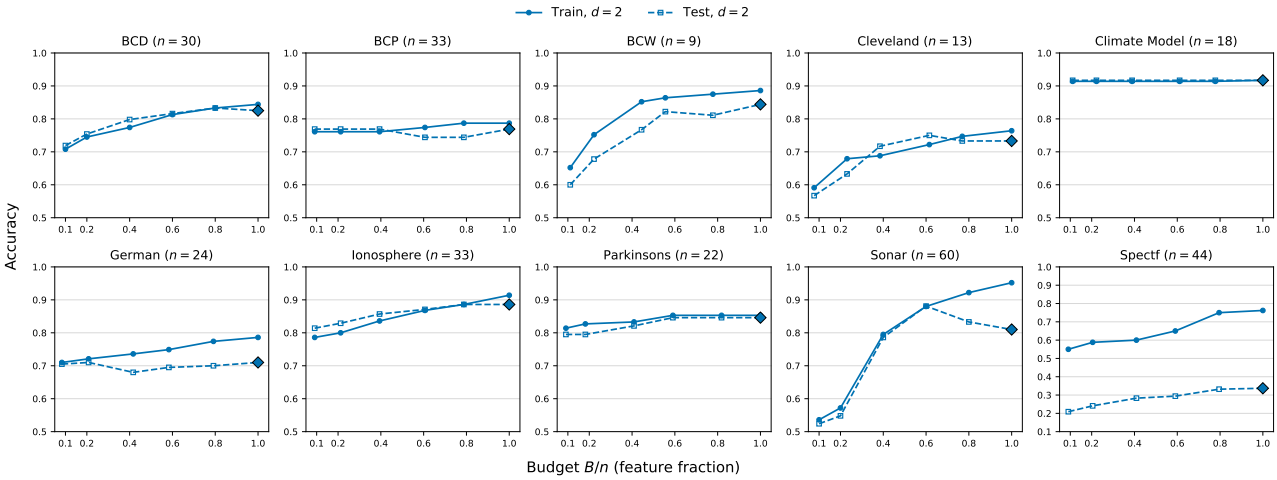

Figure 3: Train (solid) and test (dashed) accuracy of the `CompactMIQP` formulation as a function of the feature budget B/n, for degree $d = 2$, with $C = 1$ and $\gamma = 1/n$ fixed. The diamond marker ($\diamond$) at $B/n = 1$ denotes the case $B = n$, where the cardinality constraint no longer restricts the choice of features and the model reduces to a standard SVM.

where b^* satisfies

$$y^i \left(\sum_{j=1}^{m} \alpha_j^* y^j k_{\beta^*}(x^i, x^j) + b^* \right) = 1 \quad \text{for some } i \text{ such that } 0 < \alpha_i^* < C,$$

and α^*, β^* are the solutions of (9) obtained via the `CompactMIQP` formulation.

We consider the same feature budgets B, expressed as fractions of n, used in Figures 1 and 2, for degree $d = 2$. Unlike a typical machine-learning pipeline, we do not perform hyperparameter tuning via cross-validation here and, consistently with the computational experiments above, the kernel parameters are fixed to $C = 1$ and $\gamma = 1/n$, and a single train/test split is used for each dataset. The goal of this section is therefore not to report the best achievable accuracy for each configuration, but to isolate how the predictive performance of the model varies with the feature budget B, under the same fixed experimental setup used to evaluate its computational performance.

Figure 3 shows that, for most datasets, train and test accuracy increase with the feature budget, with the largest gains at low-to-intermediate budgets (e.g., `BCW`, from about 0.60 to 0.82; `Sonar`, from about 0.52 to 0.88), and are already close to the full-feature model (diamond, $B/n = 1$) at intermediate budgets, suggesting that a fraction of the features often suffices. `Climate Model` and `BCP` are essentially insensitive to the budget, staying respectively around 0.92 and between 0.74–0.79 throughout. On `Sonar` and `German`, the train–test gap widens at larger budgets, with `Sonar`'s test accuracy peaking at $B/n = 0.6$ (about 0.88) and declining to 0.81 for the full-feature model, suggesting that a moderate budget acts as a regularizer. `Spectf` shows a large train–test gap at all budgets, likely due to its small sample size ($m = 80$).

6 Conclusion

In this work, we addressed the problem of training nonlinear Support Vector Machines with polynomial kernels under a hard cardinality constraint that fixes the number of selected features to a user-defined value B. Masking the input features with binary variables inside the kernel and eliminating the primal weights through the optimality conditions yields a Mixed-Integer Nonlinear Program whose continuous relaxation is nonconvex, and which is therefore intractable for general-purpose global optimization solvers. We showed that a direct

application of standard McCormick linearization is insufficient to recover convexity, and we proposed instead two equivalent Mixed-Integer Convex Quadratic reformulations of the original problem. The key idea is to expose a hidden convex structure by rewriting the polynomial kernel expansion as a sum of squared linear forms, and to linearize exactly the resulting multilinear products between the continuous dual variables and the binary feature-selection variables. The first formulation exploits the symmetry of the kernel expansion to eliminate redundant variables, while the second further aggregates over the dual variables, eliminating the dependence of the auxiliary variables on the number of training samples m. Both reformulations admit a convex continuous relaxation, are provably equivalent to the original problem, and extend naturally to polynomial kernels of arbitrary degree d.

Computational experiments on ten benchmark datasets show that, for the quadratic kernel, the compact reformulation can be solved to certified global optimality within two hours on 27 out of 30 instances (budgets $B \in \{2, 5, 8\}$), whereas the nonconvex baseline is intractable already on small instances. Solving times grow with the budget B and with the degree d: at $d = 3$, 13 out of 30 instances are certified, with final gaps that can be large for larger budgets and higher-dimensional datasets. The root relaxation is often far from tight, so that branch-and-bound is essential to close the gap.

Several directions remain open for future research. A natural extension is to investigate tighter linearizations and valid inequalities that strengthen the continuous relaxation, especially for higher degrees and larger budgets, where the instances remain challenging. The aggregation idea underlying the compact formulation could also be combined with decomposition or column-generation schemes to scale to larger datasets and higher polynomial degrees. Finally, extending the framework beyond polynomial kernels, for instance to other kernels admitting a finite or approximate feature-space expansion, would broaden the applicability of exact embedded feature selection in nonlinear SVMs.

A Nonconvexity of the baseline reformulation

We provide a simple instance showing that the quadratic form induced by the baseline reformulation is not convex. The quadratic term of (17a) is

$$\frac{\gamma^2}{2} \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} \sum_{i=1}^{m} \sum_{h=1}^{m} \Delta_i^{j_1,j_2} \Delta_h^{j_1,j_2} \, v_{i,j_1} \, v_{h,j_2}.$$

By stacking the variables $v_{i,j}$ into a vector v indexed by (i,j), so that $v = (v_{11} \; v_{12} \; v_{21} \; v_{22})^\top$, the quadratic term can be written as $\frac{\gamma^2}{2} \, v^\top Q v$, where the matrix Q is defined as

$$Q_{(i,j_1),(h,j_2)} = \Delta_i^{j_1,j_2} \Delta_h^{j_1,j_2} = y^i x_{j_1}^i x_{j_2}^i \, y^h x_{j_1}^h x_{j_2}^h.$$

Consider $m = 2$ samples and $n = 2$ features, with labels and feature vectors given by

$$y^1 = 1, \quad y^2 = -1, \qquad x^1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \quad x^2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix},$$

and $\gamma = 1$. We obtain

$$Q = \begin{pmatrix} 16 & 4 & -4 & -4 \\ 4 & 1 & -4 & -4 \\ -4 & -4 & 1 & 4 \\ -4 & -4 & 4 & 16 \end{pmatrix}.$$

This matrix is indefinite, as it admits negative eigenvalues. Therefore, the quadratic form is not convex, and the continuous relaxation of the baseline formulation is nonconvex.

References

[1] Joseph Agor and Osman Y Özaltın. "Feature selection for classification models via bilevel optimization". In: *Computers & Operations Research* 106 (2019), pp. 156–168.

[2] Genevera I Allen. "Automatic feature selection via weighted kernels and regularization". In: *Journal of Computational and Graphical Statistics* 22.2 (2013), pp. 284–299.

[3] Haldun Aytug. "Feature selection for support vector machines using Generalized Benders Decomposition". In: *European Journal of Operational Research* 244.1 (2015), pp. 210–218.

[4] Rafael Blanquero et al. "Selection of time instants and intervals with support vector regression for multivariate functional data". In: *Computers & Operations Research* 123 (2020), p. 105050.

[5] Verónica Bolón-Canedo, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. *Feature selection for high-dimensional data*. Springer, 2015.

[6] DW Bolton. "The multinomial theorem". In: *The Mathematical Gazette* 52.382 (1968), pp. 336–342.

[7] Immanuel Bomze et al. "Feature selection in linear support vector machines via a hard cardinality constraint: A scalable conic decomposition approach". In: *European Journal of Operational Research* (2025).

[8] Paul S. Bradley and Olvi L. Mangasarian. "Feature Selection via Concave Minimization and Support Vector Machines". In: *International Conference on Machine Learning*. 1998.

[9] Boseon Byeon and Khaled Rasheed. "Simultaneously removing noise and selecting relevant features for high dimensional noisy data". In: *2008 Seventh International Conference on Machine Learning and Applications*. IEEE. 2008, pp. 147–152.

[10] Antoni B Chan, Nuno Vasconcelos, and Gert RG Lanckriet. "Direct convex relaxations of sparse SVM". In: *Proceedings of the 24th international conference on Machine learning*. 2007, pp. 145–153.

[11] Girish Chandrashekar and Ferat Sahin. "A survey on feature selection methods". In: *Computers & Electrical Engineering* 40.1 (2014), pp. 16–28.

[12] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM: A library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology* 2 (3 2011). Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`, 27:1–27:27.

[13] Corinna Cortes and Vladimir Vapnik. "Support-Vector Networks". In: *Machine Learning* 20.3 (Sept. 1995), pp. 273–297. ISSN: 0885-6125. DOI: `10.1023/A:1022627411411`. URL: `https://doi.org/10.1023/A:1022627411411`.

[14] Evrim Dalkiran and Laleh Ghalami. "On linear programming relaxations for solving polynomial programming problems". In: *Computers & Operations Research* 99 (2018), pp. 67–77.

[15] Yuri Faenza Federico D'Onofrio and Laura Palagi. *Combinatorial Approaches for Embedded Feature Selection in Nonlinear SVMs*. arXiv:2507.23711.

[16] Glenn Fung and O.L. Mangasarian. "A Feature Selection Newton Method for Support Vector Machine Classification". In: *Computational Optimization and Applications* 28 (July 2004), pp. 185–202. DOI: `10.1023/B:COAP.0000026884.66338.df`.

[17] Bissan Ghaddar and Joe Naoum-Sawaya. "High dimensional data classification and feature selection using support vector machines". In: *European Journal of Operational Research* 265.3 (2018), pp. 993–1004.

[18] Fred Glover. "Improved linear integer programming formulations of nonlinear integer problems". In: *Management Science* 22.4 (1975), pp. 455–460.

[19] Bryce Goodman and Seth Flaxman. "European Union regulations on algorithmic decision-making and a "right to explanation"". In: *AI magazine* 38.3 (2017), pp. 50–57.

[20] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2026. URL: https://www.gurobi.com.

[21] Trevor Hastie et al. "The entire regularization path for the support vector machine". In: *Journal of Machine Learning Research* 5.Oct (2004), pp. 1391–1415.

[22] Asunción Jiménez-Cordero, Juan Miguel Morales, and Salvador Pineda. "A novel embedded min-max approach for feature selection in nonlinear Support Vector Machine classification". In: *European Journal of Operational Research* 293.1 (2021), pp. 24–35. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2020.12.009. URL: https://www.sciencedirect.com/science/article/pii/S0377221720310195.

[23] Uday Kamath and John Liu. *Explainable artificial intelligence: An introduction to interpretable machine learning*. Springer, 2021.

[24] Martine Labbé, Luisa I Martínez-Merino, and Antonio M Rodríguez-Chía. "Mixed integer linear programming for feature selection in support vector machine". In: *Discrete Applied Mathematics* 261 (2019), pp. 276–304.

[25] Chih-Jen Lin. "Formulations of Support Vector Machines: A Note from an Optimization Point of View". In: *Neural Computation* 13.2 (2001), pp. 307–317. DOI: 10.1162/089976601300014547.

[26] Sebastián Maldonado, Richard Weber, and Jayanta Basak. "Simultaneous feature selection and classification using kernel-penalized support vector machines". In: *Information Sciences* 181.1 (2011), pp. 115–128. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2010.08.047. URL: https://www.sciencedirect.com/science/article/pii/S0020025510004287.

[27] Sebastián Maldonado et al. "Feature selection for support vector machines via mixed integer linear programming". In: *Information Sciences* 279 (2014), pp. 163–175.

[28] Olvi Mangasarian. *Generalized support vector machines*. Tech. rep. 1998.

[29] Olvi L. Mangasarian and Gang Kou. "Feature Selection for Nonlinear Kernel Support Vector Machines". In: *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*. 2007, pp. 231–236. DOI: 10.1109/ICDMW.2007.30.

[30] Sergio Muñoz-Romero et al. "Informative variable identifier: Expanding interpretability in feature selection". In: *Pattern Recognition* 98 (2020), p. 107077.

[31] Julia Neumann, Christoph Schnörr, and Gabriele Steidl. "Combined SVM-based feature selection and classification". In: *Machine learning* 61 (2005), pp. 129–150.

[32] Minh Hoai Nguyen and Fernando De la Torre. "Optimal feature selection for support vector machines". In: *Pattern Recognition* 43.3 (2010), pp. 584–591.

[33] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[34] Arvind U. Raghunathan et al. "Recursive McCormick Linearization of Multilinear Programs". In: *INFORMS Journal on Computing* (2023). arXiv:2207.08955.

[35] Cynthia Rudin. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". In: *Nature machine intelligence* 1.5 (2019), pp. 206–215.

[36] Cynthia Rudin et al. "Interpretable machine learning: Fundamental principles and 10 grand challenges". In: *Statistic Surveys* 16 (2022), pp. 1–85.

[37] Hong Seo Ryoo and Nikolaos V. Sahinidis. "Analysis of bounds for multilinear functions". In: *Journal of Global Optimization* 19.4 (2001), pp. 403–424.

[38] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.

[39] Li Wang, Ji Zhu, and Hui Zou. “The doubly regularized support vector machine”. In: *Statistica Sinica* 16.2 (2006), pp. 589–615.

[40] Jason Weston et al. “Feature selection for SVMs”. In: *Advances in neural information processing systems* 13 (2000).

[41] Jason Weston et al. “Use of the zero norm with linear models and kernel methods”. In: *The Journal of Machine Learning Research* 3 (2003), pp. 1439–1461.

[42] Guo-Xun Yuan et al. “A comparison of optimization methods and software for large-scale l1-regularized linear classification”. In: *Journal of Machine Learning Research* 11 (2010), pp. 3183–3234.

[43] Hui Zou and Trevor Hastie. “Regularization and variable selection via the elastic net”. In: *Journal of the Royal Statistical Society Series B* 67.2 (2005), pp. 301–320.