Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

ℓ0-Regularized Quadratic Surface Support Vector Machines

Ahmad Mousavi

mousavi@american.edu

American University

Ramin Zandvakili

University of Minnesota

Zheming Gao

University of North Carolina at Charlotte

Method Article

Keywords: binary classification, quadratic support vector machines, cardinality constraint, penalty decomposition method

Posted Date: July 13th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9390397/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

ℓ_0-Regularized Quadratic Surface Support Vector Machines

Ahmad Mousavi[1*], Ramin Zandvakili[2] and Zheming Gao[3]

[1*]Department of Mathematics and Statistics, American University, Washington, DC, USA.
[2]Carlson School of Management, University of Minnesota, Minneapolis, MN, USA.
[3]Department of Industrial and Systems Engineering, University of North Carolina at Charlotte, Charlotte, NC, USA.

*Corresponding author(s). E-mail(s): mousavi@american.edu;
Contributing authors: zandv003@umn.edu; zgao7@charlotte.edu;

Abstract

Kernel-free quadratic surface support vector machines (QSVM) have recently gained traction due to their flexibility in modeling nonlinear decision boundaries without relying on kernel functions. However, the introduction of a full quadratic classifier significantly increases the number of model parameters, scaling quadratically with data dimensionality, which often leads to overfitting and makes interpretation difficult. To address these challenges, we propose sparse variants of the QSVM by enforcing a cardinality constraint on the model parameters. While enhancing generalization and promoting sparsity, leveraging the $\boldsymbol{\ell_0}$-norm inevitably incurs additional computational complexity. To tackle this, we develop a penalty decomposition algorithm that produces solutions that provably satisfy the first-order Lu-Zhang optimality conditions. We show that the subproblems arising in the algorithm either admit closed-form solutions or can be solved efficiently via dual formulations, thereby contributing to the method's overall effectiveness. Besides, we analyze the convergence behavior of the algorithm under both loss settings. In addition, the numerical experiments on public benchmark datasets indicate that the proposed model is competitive with commonly used SVM variants and produces sparse solutions as expected. Moreover, its strong performance on real-world credit datasets demonstrates its potential for credit scoring applications.

Keywords: binary classification, quadratic support vector machines, cardinality constraint, penalty decomposition method

1 Introduction

Soft-margin support vector machines (SVMs) remain a fundamental tool for binary classification, effectively balancing margin maximization with error tolerance [1–3]. In parallel, the development of sparse linear models has addressed the growing need for interpretability and overfitting control by deliberately restricting model complexity [4–8]. While valuable, linear sparsity is inherently limited by its inability to model nonlinear relationships. Kernel methods were introduced to SVMs to resolve this, enabling the discovery of complex patterns in a transformed feature space. This advance, however, comes at a price. The resulting models frequently lose the transparency of their linear counterparts, making them difficult to interpret. Furthermore, their effectiveness is heavily contingent on the careful, and often non-trivial, selection of the kernel and its associated hyperparameters, a process that can be computationally expensive and hinder practical applicability.

Kernel-free quadratic surface support vector machines (QSVMs) were developed to preserve model interpretability while avoiding the complexities of kernelization. These methods learn quadratic decision boundaries directly, optimizing an approximate geometric margin and maintaining robustness to noise and outliers [9]. Their explicit functional form offers additional flexibility, such as the ability to incorporate prior knowledge in the form of Universum data. Variants like twin QSVMs further enhance scalability by decomposing the primary problem into smaller subproblems [10]. Despite these advantages, a fundamental limitation persists: the number of parameters in a quadratic model grows quadratically with the input dimension, i.e., $\mathcal{O}(n^2)$. This quadratic scaling renders such models prone to overfitting and limits their practicality, even on moderately sized datasets, a challenge that subsequent research has sought to address.

The existing work further restricts the quadratic weight matrix to be diagonal [11], which reduces the number of trainable parameters to $\mathcal{O}(n)$ and alleviates the overfitting problem. However, this simplification rests on a strong assumption that feature interactions are negligible. In practice, this assumption is seldom valid, and discarding pairwise correlations can lead to the loss of informative structure, resulting in underfitting. To navigate this trade-off between model complexity and representational power, various regularization techniques have been explored. ℓ_1 norm regularization promotes sparsity and enhances interpretability, yet it may yield nonunique solutions [10, 12–15]. ℓ_2 norm regularization effectively curbs overfitting but does not inherently encourage sparsity [16]. More flexible approaches, such as ℓ_p norm regularization, allow finer control over the degree of sparsity, though this comes at the cost of careful tuning of the parameter p [17].

Sparse recovery theory provides a nuanced understanding of how different choices of p in ℓ_p regularization affect the structure of the solution. Values of $p > 1$ tend to yield dense solutions, offering little in the way of sparsity [18]. The case $p = 1$ preserves convexity and promotes sparsity, yet it remains less aggressive in this regard than norms with $0 < p < 1$. Within the nonconvex regime, values of p in the range $[1/2, 1)$ often produce the sparsest solutions overall, with diminishing returns as p decreases further [19]. Against this backdrop, ℓ_0 norm regularization occupies a unique position.

Unlike any ℓ_p surrogate with $p > 0$, it enables exact control over the number of nonzero parameters in the solution. This capability delivers not only precise sparsity but also stronger theoretical guarantees for recovery. Recent algorithmic advances have rendered direct ℓ_0 optimization increasingly practical. Methods such as penalty decomposition [20], greedy algorithms [21], and branch and bound techniques [22] demonstrate that convex relaxations are no longer necessary and are often suboptimal for ℓ_0 regularized problems.

Building on these insights, we develop novel kernel-free QSVM models that employ ℓ_0 regularization to achieve exact sparsity in both the quadratic and linear coefficients. Since the proposed ℓ_0 regularized models are intractable, an efficient algorithm is designed for their implementation. Numerical experiments are conducted to validate the sparsity and the classification capability of the proposed models. The contributions of this research are summarized as follows.

- Our method combines the representational capacity of quadratic decision boundaries with the simplicity of sparse models, offering both powerful nonlinear classification and clear interpretability. Different from the sparsity yielded by ℓ_1 regularizations [12, 23], the ℓ_0 constraint provides solutions with at most k nonzero elements, giving direct control over model complexity while automatically performing feature selection.
- To address the computational challenges brought up by the ℓ_0 regularization, we develop a penalty decomposition algorithm that recasts the original problem into a sequence of more tractable subproblems. By introducing an auxiliary variable that decouples the ℓ_0-norm constraint, the subproblem alternates between analytically tractable updates and efficiently solvable ones via duality theory. In addition, the convergence of the proposed penalty-decomposition algorithm is rigorously analyzed.
- Extensive numerical experiments are conducted to evaluate the effectiveness of the proposed models in classification tasks. The sparsity characteristics of the models and the influence of key parameters are systematically examined through computational analysis. In addition, the proposed model with quadratic loss is applied to real-world credit scoring datasets, demonstrating strong predictive performance and enhanced interpretability in practical credit risk assessment applications.

The remainder of the paper is structured as follows. Section 2 reviews relevant literature. Section 3 presents our ℓ_0-regularized QSVM models and describes the decomposition algorithm. Section 4 presents the numerical experiment results as well as the application of the proposed models to credit scoring. Section 5 concludes with a discussion of future directions.

2 Preliminaries and Related Work

This section briefly reviews several quadratic surface support-vector machine (QSVM) models. Consider a binary classification task with a training set

$$\mathcal{D} = \{(x_i, y_i)_{i=1,\dots,m} \mid x_i \in \mathbb{R}^n,\ y_i \in \{-1, 1\}\},$$

where m is the sample size, n is the number of features, and y_i is the label of x_i. $\mathcal{D}$ is called quadratically separable [9] if there exist $W \in \mathbb{R}^{n\times n}, b \in \mathbb{R}^n$, and $c \in \mathbb{R}$ if $y_i(x_i^\top W x_i + b^\top x_i + c) > 0$ for every $i \in [m]$. A linearly separable dataset is a special case of quadratic separability obtained by setting $W = 0$.

Standard SVMs seek a hyperplane that approximately separates a training set. In many real-world binary classification problems, however, data exhibit non-linear patterns, making linear separation difficult. Kernel methods overcome this limitation by mapping the data into a higher-dimensional feature space in which non-linear relationships appear linear. In that space, SVMs can identify optimal hyperplanes that more effectively capture complex structures. Consequently, kernel techniques markedly improve SVM performance on non-linear classification tasks and are widely applied in machine learning, pattern recognition, and data mining [24].

Despite their advantages, kernel methods also have notable limitations. They are computationally expensive because pairwise similarity calculations scale poorly with large datasets. Moreover, as the dimensionality of the feature space grows, kernel models become prone to overfitting. Choosing a kernel function demands domain expertise or extensive experimentation, and the resulting classifiers are less interpretable than their linear counterparts. Sensitivity to hyper-parameter tuning and limited scalability further underscore the trade-offs associated with kernel methods in complex classification tasks [25].

Therefore, a recent method, introduced by Dagher [26] and further refined by Luo et al. [9], seeks to explicitly separate data using a quadratic classifier directly in the original input space, avoiding any kernel-induced feature mapping:

$$f_{W,b,c}(x) = \frac{1}{2}x^\top W x + b^\top x + c = 0.$$

The soft-margin variant of this model, which penalizes misclassifications to accommodate noise and outliers, is formulated as follows:

$$\begin{aligned} \min_{W,b,c,\xi} \quad & \sum_{i=1}^{m} \|Wx_i + b\|_2^2 + \mathcal{C}\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y_i(\frac{1}{2}x_i^\top W x_i + b^\top x_i + c) \geq 1 - \xi_i, \quad i = 1, \ldots, m, \\ & W \in \mathbb{S}_n,\ b \in \mathbb{R}^n,\ c \in \mathbb{R},\ \xi \in \mathbb{R}^m_+. \end{aligned} \tag{QSVM}$$

Notice that adopting a kernel-free quadratic surface framework dramatically enlarges the model's degrees of freedom: the number of parameters grows on the order of $\mathcal{O}(n^2)$, compared with the $\mathcal{O}(n)$ parameters in a standard linear SVM. While this added flexibility can improve the fit to the training data, it also raises the risk of overfitting and may diminish generalization to unseen samples. The danger is especially acute for linearly separable datasets, where one would prefer the QSVM to revert to a linear separator—yet QSVM offers no such guarantee. A widely used remedy is sparsity-promoting ℓ_1-norm regularization; for example, Mousavi et al. [12] embed an

ℓ_1 penalty directly into the QSVM objective:

$$\begin{aligned} \min_{W,b,c,\xi} \quad & \sum_{i=1}^{m} \|Wx_i + b\|_2^2 + \mathcal{C}_1 \sum_{1\le i\le j\le n} |W_{ij}| + \mathcal{C}_2 \sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad & y_i(\frac{1}{2} x_i^\top W x_i + b^\top x_i + c) \ge 1 - \xi_i; \quad \forall i \in [m], \\ & W \in \mathbb{S}_n,\, b \in \mathbb{R}^n,\, c \in \mathbb{R},\, \xi \in \mathbb{R}^m_+, \end{aligned} \tag{ℓ_1-QSVM}$$

where $\mathcal{C} > 0$ is a positive penalty for incorporating noise and outliers. The authors demonstrate that the above model reduces to the standard SVM when λ is sufficiently large. Other ℓ_0-norm surrogates, such as ℓ_p norms, for $p \in (0, 1)$, have also been explored in the literature on linear SVMs, feature selection, and K-support vector classification regression. However, the non-convexity of these models poses significant challenges for algorithm design [27–29].

Although ℓ_p norms with $0 < p \le 1$ promote sparsity, only the ℓ_0 norm enforces it directly by minimizing the number of non-zero parameters. In compressive sensing, for example, ℓ_0 achieves higher compression rates than either the convex ($p = 1$) or non-convex ($0 < p < 1$) ℓ_p-norm relaxations, making it a superior tool for sparse representation. Motivated by this property, we next study sparse ℓ_0-norm, kernel-free quadratic surface SVMs and present a penalty decomposition algorithm that solves these models efficiently.

Kernel-free quadratic surface models differ from conventional quadratic program formulations in quadratic optimization. Hence, several notations and definitions must be introduced to rectify this and align them accordingly. We start with a square matrix $A = [a_{ij}]_{i=1,\dots,n;j=1,\dots,n} \in \mathbb{R}^{n\times n}$. Its vectorization is straightforwardly represented as $\text{vec}(A) := [a_{11}, \dots, a_{n1}, a_{12}, \dots, a_{n2}, \dots, a_{1n}, \dots, a_{nn}]^\top \in \mathbb{R}^{n^2}$. However, if A is symmetric, $\text{vec}(A)$ carries redundant data, hence we opt for its half-vectorization:

$$\text{hvec}(A) := [a_{11}, \dots, a_{n1}, a_{22}, \dots, a_{2n}, \dots, a_{nn}]^\top \in \mathbb{R}^{\frac{n(n+1)}{2}}.$$

For any $n \in \mathbb{N}$, there exists a unique elimination matrix $L_n \in \mathbb{R}^{\frac{n(n+1)}{2}\times n^2}$ such that $L_n\text{vec}(A) = \text{hvec}(A)$ for any $A \in S_n$, where S_n denotes the set of symmetric matrices. Additionally, this elimination matrix L_n has a full row rank. Conversely, for any $n \in \mathbb{N}$, there exists a unique duplication matrix $D_n \in \mathbb{R}^{n^2\times\frac{n(n+1)}{2}}$ such that $D_n\text{hvec}(A) = \text{vec}(A)$ for any $A \in S_n$, and $L_nD_n = I_{\frac{n(n+1)}{2}}$.

Definition 1 For $i \in [m]$, let

$$s_i := \tfrac{1}{2}\text{hvec}(x_ix_i^\top), \quad r_i := [s_i; x_i], \quad w := \text{hvec}(W), \quad z := [w; b], \quad V := \left[I_{\frac{n(n+1)}{2}} \;\; 0_{\frac{n(n+1)}{2}\times n}\right],$$
$$X_i := I_n \otimes x_i^\top, \quad M_i := X_iD_n, \quad H_i := \left[M_i \;\; I_n\right], \quad G := 2\textstyle\sum_{i=1}^m H_i^\top H_i, \quad X := [x_1, x_2, \dots, x_m]^\top.$$

Hence, simply $G \succeq 0$. And, for fixed $i \in [m]$, we have the following equations:

$$\begin{cases} Wx_i = X_i \mathrm{vec}(W) = X_i D_n \mathrm{hvec}(W) = M_i \mathrm{hvec}(W) = M_i w, \\ Wx_i + b = M_i w + I_n b = H_i z, \\ \frac{1}{2} x_i^\top W x_i + b^\top x_i + c = z^\top r_i + c. \end{cases}$$

Thus,

$$\sum_{i=1}^m \|Wx_i + b\|_2^2 = \sum_{i=1}^m (H_i z)^\top (H_i z) = z^\top \big(\sum_{i=1}^m H_i^\top H_i \big) z = \frac{1}{2} z^\top G z.$$

3 ℓ_0-Regularized Quadratic Surface SVMs and A Penalty Decompotion Algorithm

Empirical evidence suggests that many real-world datasets, being governed by underlying physical laws, are effectively controlled by only a few dominant factors. This makes ℓ_0-regularized models particularly appealing for capturing their essential structure. By isolating these critical components, ℓ_0 regularization delivers greater interpretability and practical utility. Conversely, ℓ_p surrogates with $0 < p \leq 1$ merely approximate sparsity: they shrink coefficients toward—but rarely all the way to—zero, and the degree of sparsity is controlled only indirectly through a tuning parameter. Because this parameter cannot set the exact number of non-zero entries, ℓ_p regularization often fails to capture the true zero pattern. These shortcomings highlight the benefits of employing ℓ_0 regularization when precise sparsity is crucial. However, the inclusion of a sparsity constraint renders the model intractable, necessitating the development of an efficient solution method. To address this, we propose a penalty decomposition algorithm and demonstrate how its subproblems can be solved effectively. We will also study the convergence of this algorithm.

To mitigate overfitting arising from the over-parameterization of the symmetric matrix W and vector b in the quadratic classifier f, we impose sparsity constraints on these parameters. This restriction lowers model complexity and improves generalization. Accordingly, we examine two sparse ℓ_0-norm kernel-free quadratic surface SVM formulations. The first employs the hinge loss and is expressed as follows:

$$\begin{aligned} \min_{W,b,c,\xi} \quad & \sum_{i=1}^m \|Wx_i + b\|_2^2 + \mathcal{C} \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & y_i(\frac{1}{2} x_i^\top W x_i + b^\top x_i + c) \geq 1 - \xi_i; \quad \forall i \in [m], \\ & \|[\mathrm{hvec}(W); b]\|_0 \leq k, \\ & W \in \mathbb{S}_n,\ b \in \mathbb{R}^n,\ c \in \mathbb{R},\ \xi \in \mathbb{R}_+^m. \end{aligned} \tag{ℓ_0-QSVM}$$

Using the quadratic loss function, we derive the least-squares version of the model as follows:

$$\begin{aligned}\min_{W,b,c,\xi} \quad & \sum_{i=1}^{m} \|Wx_i + b\|_2^2 + \mathcal{C} \sum_{i=1}^{m} \xi_i^2 \\ \text{s.t.} \quad & y_i(\frac{1}{2} x_i^\top W x_i + b^\top x_i + c) = 1 - \xi_i; \quad \forall i \in [m], \\ & \|[\text{hvec}(W); b]\|_0 \leq k, \\ & W \in \mathbb{S}_n,\ b \in \mathbb{R}^n,\ c \in \mathbb{R},\ \xi \in \mathbb{R}^m,\end{aligned} \tag{LS-ℓ_0-QSVM}$$

To present a penalty decomposition algorithm that works for both proposed models, with different corresponding subproblems, we introduced the following unified model:

$$\begin{aligned}\min \quad & \sum_{i=1}^{m} \|Wx_i + b\|_2^2 + \mathcal{C} \sum_{i=1}^{m} \text{H}(1 - y_i f_{W,b,c}(x_i)) \\ \text{s.t.} \quad & \|[\text{hvec}(W); b]\|_0 \leq k, \\ & W \in \mathbb{S}_n,\ b \in \mathbb{R}^n,\ c \in \mathbb{R}.\end{aligned}$$

which reduces to ℓ_0-QSVM for the well-received hinge loss $\text{H}(t) = \max(t, 0)$ and reduces to LS-ℓ_0-QSVM' for the quadratic loss $\text{H}(t) = t^2$. Based on the notation introduced in the previous section, the above model is equivalent to:

$$\begin{aligned}\min \quad & \frac{1}{2} z^\top G z + \mathcal{C} \sum_{i=1}^{m} \text{H}(1 - y_i(z^\top r_i + c)) \\ \text{s.t.} \quad & \|z\|_0 \leq k, \\ & z \in \mathbb{R}^{\frac{n(n+1)}{2}+n}, c \in \mathbb{R}.\end{aligned} \tag{1}$$

This unified model remains intractable; therefore, we next apply an effective penalty decomposition method to solve it. After introducing a new variable u, this problem becomes:

$$\begin{aligned}\min \quad & \frac{1}{2} z^\top G z + \mathcal{C} \sum_{i=1}^{m} \text{H}(1 - y_i(z^\top r_i + c)) \\ \text{s.t.} \quad & \|z\|_0 \leq k, \\ & z - u = 0, \\ & z, u \in \mathbb{R}^{\frac{n(n+1)}{2}+n}, c \in \mathbb{R}.\end{aligned}$$

Thus, by penalizing the last constraint above, the problem 1 can be addressed by solving a sequence of penalty subproblems as follows:

$$\begin{aligned} \min \quad & q_\rho(z,c,u) := \frac{1}{2}z^\top Gz + \mathcal{C}\sum_{i=1}^{m} \mathrm{H}(1-y_i(z^\top r_i + c)) + \frac{1}{2}\rho\|z-u\|_2^2 \\ \text{s.t.} \quad & \|u\|_0 \le k, \\ & z,u \in \mathbb{R}^{\frac{n(n+1)}{2}+n}, c \in \mathbb{R}. \end{aligned} \tag{P_ρ}$$

Suppose we have a feasible point denoted as $(z,c)^{\text{feas}}$ of 1 in hand. Then, let:

$$\Upsilon \ge \max\{f((z,c)^{\text{feas}}), \min_{z,c} q_{\rho^{(0)}}(z,c,u_0^{(0)})\} > 0.$$

We solve the above penalty subproblem for a fixed positive ρ via a block coordinate descent method. The stopping criterion for the inner loop is the following:

$$\max\left\{\frac{\|z_l - z_{l-1}\|_\infty}{\max(\|z_l\|_\infty, 1)}, \frac{\|c_l - c_{l-1}\|_\infty}{\max(\|c_l\|_\infty, 1)}, \frac{\|u_l - u_{l-1}\|_\infty}{\max(\|u_l\|_\infty, 1)}, \right\} \le \epsilon_I. \tag{2}$$

And for the outer loop is:

$$\|z^{(j)} - u^{(j)}\|_\infty \le \epsilon_O. \tag{3}$$

Therefore, we present the following penalty decomposition algorithm for solving (1).

Algorithm 1 ℓ_0-Regularized QSVM Penalty Decomposition

1: Inputs: $\rho^{(0)} > 0, \beta > 1, k$, and $u_0^{(0)}$ such that $\|u_0^{(0)}\|_0 \le k$.
2: $j \leftarrow 0$.
3: **repeat**
4: $\quad l \leftarrow 0$.
5: $\quad$ **repeat**
6: $\quad\quad (z_{l+1}^{(j)}, c_{l+1}^{(j)}) \leftarrow \text{Argmin}_{(z,c)\in\mathbb{R}^{\frac{n(n+1)}{2}+n}\times\mathbb{R}} \; q_{\rho^{(j)}}(z,c,u_l^{(j)})$.
7: $\quad\quad u_{l+1}^{(j)} \leftarrow \text{Argmin}_{\{u\mid\|u\|_0\le k\}} \; q_{\rho^{(j)}}(z_{l+1}^{(j)}, c_{l+1}^{(j)}, u)$.
8: $\quad\quad l \leftarrow l+1$.
9: $\quad$ **until** stopping criterion (2) is met.
10: $\quad (z^{(j)}, c^{(j)}, u^{(j)}) \leftarrow (z_l^{(j)}, c_l^{(j)}, u_l^{(j)})$.
11: $\quad \rho^{(j+1)} \leftarrow \beta \cdot \rho^{(j)}$.
12: $\quad$ If $\min_{(z,c)\in\mathbb{R}^{\frac{n(n+1)}{2}+n}\times\mathbb{R}} \; q_{\rho^{(j+1)}}(z,c,u^{(j)}) > \Upsilon$, then $u_0^{(j+1)} \leftarrow z^{\text{feas}}$. Otherwise, $u_0^{(j+1)} \leftarrow u_l^{(j)}$.
13: $\quad j \leftarrow j+1$.
14: **until** stopping criterion (3) is met.

We now turn our attention to solving the restricted subproblems within the algorithm as efficiently as possible. Observe that the associated subproblem for u is:

$$\min_{u\in\mathbb{R}^{\frac{n(n+1)}{2}+n}} \|z-u\|_2^2 \qquad \text{s.t.} \qquad \|u\|_0 \leq k; \tag{4}$$

which obtains a solution

$$u = [z_{\mathcal{J}}; 0], \tag{5}$$

where $\mathcal{J}$ contains indices of the k largest components of z in absolute value, that is, Step 7 of Algorithm 0 has a closed-form solution. Next, we discuss how to tackle the associated subproblem for z for each loss function.

3.1 ℓ_0-Regularized Quadratic Surface SVM Model

For the hinge loss function $\mathrm{H}(t) = \max(0, t)$, the sparse quadratic model 1 reduces to ℓ_0-QSVM, which after using the notation in Section 2 is equivalent to:

$$\begin{array}{ll} \min\limits_{z,c,\xi} & \dfrac{1}{2}z^\top Gz + \mathcal{C}\sum\limits_{i=1}^{m}\xi_i \\ \text{s.t.} & y_i(z^\top r_i + c) \geq 1-\xi_i; \quad \forall i\in[m], \\ & \|z\|_0 \leq k. \\ & z\in\mathbb{R}^{\frac{n(n+1)}{2}+n}, c\in\mathbb{R}, \xi\in\mathbb{R}^m_+. \end{array} \tag{ℓ_0-QSVM'}$$

Recalling P_ρ, the subproblem associated to z for the hinge loss function becomes:

$$\min_{z\in\mathbb{R}^{\frac{n(n+1)}{2}+n}, c\in\mathbb{R}} \quad \frac{1}{2}z^\top Gz + \mathcal{C}\sum_{i=1}^{m}\max(1-y_i(z^\top r_i + c), 0) + \frac{1}{2}\rho\|z-u\|_2^2,$$

which is equivalent to the following feasible and bounded below convex quadratic optimization problem:

$$\begin{array}{ll} \min\limits_{z,c,\xi} & \dfrac{1}{2}z^\top(G+\rho I)z - \rho u^\top z + \mathcal{C}\sum\limits_{i=1}^{m}\xi_i \\ \text{s.t.} & y_i(z^\top r_i + c) \geq 1-\xi_i; \quad \forall i\in[m] \\ & z\in\mathbb{R}^{\frac{n(n+1)}{2}+n}, c\in\mathbb{R}, \xi\in\mathbb{R}^m_+, . \end{array} \tag{$P^h_{\rho,(z,c,\xi)}$}$$

Even though a standard solver can be used for this problem, standard practice in the SVM literature encourages applying the strong duality theorem in convex programming. Thus, let us bring the Lagrangian function of this model:

$$L(z,c,\xi,\alpha) = \frac{1}{2}z^\top(G+\rho I)z - \rho u^\top z + \mathcal{C}\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i(y_i(z^\top r_i + c) - 1 + \xi_i)$$

$$= \frac{1}{2}z^\top(G+\rho I)z - z^\top\Big(\sum_{i=1}^m \alpha_i y_i r_i + \rho u\Big) + \sum_{i=1}^m (\mathcal{C}-\alpha_i)\xi_i - c\sum_{i=1}^m \alpha_i y_i.$$

Therefore, the dual problem of $P^h_{\rho,(z,c,\xi)}$ becomes:

$$\begin{array}{ll} \min\limits_{\alpha\in\mathbb{R}^m} & -\sum\limits_{i=1}^m \alpha_i + \frac{1}{2}\Big(\sum\limits_{i=1}^m y_i\alpha_i r_i + \rho u\Big)^\top (G+\rho I)^{-1}\Big(\sum\limits_{j=1}^m y_j\alpha_j r_j + \rho u\Big) \\ \text{s.t.} & \sum\limits_{i=1}^m y_i\alpha_i = 0 \qquad \text{and} \qquad 0 \le \alpha \le \mathbf{1}\mathcal{C}. \end{array} \tag{D_α}$$

Therefore, the Karush-Kuhn-Tucker (KKT) conditions are:

$$\begin{cases} z = (G+\rho I)^{-1}\big(\sum_{i=1}^m \alpha_i y_i r_i + \rho u\big) \\ \alpha_i(1-\xi_i - y_i(z^\top r_i + c)) = 0; \quad \forall i \in [m] \\ (\mathcal{C}-\alpha_i)\xi_i = 0; \quad \forall i \in [m] \\ y_i(z^\top r_i + c) \ge 1-\xi_i; \quad \forall i \in [m] \\ \sum_{i=1}^m y_i\alpha_i = 0; \quad \forall i \in [m] \\ \xi \in \mathbb{R}^m_+ \qquad \text{and} \qquad 0 \le \alpha \le \mathbf{1}\mathcal{C}, \end{cases}$$

which implies that, once a solution to the dual D_α is provided, a solution to $P^h_{\rho,(z,c,\xi)}$ can be obtained from:

$$\begin{array}{l} z = (G+\rho I)^{-1}\Big(\sum\limits_{\alpha_i\neq 0} \alpha_i y_i r_i + \rho u\Big), \qquad c = \max\limits_{y_i=1,\alpha_i>0} -z^\top r_i, \\ \xi_i = \begin{cases} 1 - y_i(z^\top r_i + c) & \text{if} \quad \alpha_i > 0, \\ 0 \quad \text{if} \quad \alpha_i = 0. \end{cases} \end{array} \tag{6}$$

Corollary 3.1 For the hinge loss function, step 6 of Algorithm 0 can be obtained by solving D_α and then applying (6).

3.2 Least-squares ℓ_0-Regularized Quadratic Surface SVM Model

For the quadratic loss function $\mathrm{H}(t) = t^2$, the sparse quadratic model 1 reduces to the following least-squares ℓ_0-regularized kernel-free quadratic surface SVM model

LS-ℓ_0-QSVM, which through the notations presented in Section 2 becomes:

$$\begin{array}{ll}\min\limits_{z,c,\xi} & \frac{1}{2}z^\top Gz + \mathcal{C}\sum\limits_{i=1}^{m}\xi_i^2 \\ \text{s.t.} & y_i(z^\top r_i + c) = 1-\xi_i; \quad \forall i\in[m], \\ & \|z\|_0 \le k, \\ & z\in\mathbb{R}^{\frac{n(n+1)}{2}+n}, c\in\mathbb{R}, \xi\in\mathbb{R}^m,. \end{array} \tag{LS-ℓ_0-QSVM'}$$

The associated subproblem to z in this case is equivalent to:

$$\min_{z\in\mathbb{R}^{\frac{n(n+1)}{2}+n}, c\in\mathbb{R}} \frac{1}{2}z^\top Gz + \mathcal{C}\sum_{i=1}^{m}(1-y_i(z^\top r_i+c))^2 + \frac{1}{2}\rho\|z-u\|_2^2. \tag{$P^q_{(z,c)}$}$$

By letting $A=[r_1,r_2,\ldots,r_m]^\top$ and $D=\mathbf{diag}(y_1,y_2,\ldots,y_m)$, the objective function of the latter is equal to $T=\frac{1}{2}z^\top(G+\rho I)z+\mathcal{C}\|\mathbf{1}-D(Az+c\mathbf{1})\|_2^2-\rho u^\top z+\rho\|u\|_2^2$. Hence,

$$\begin{aligned}\frac{\partial T}{\partial z} &= (G+\rho I)z - 2\mathcal{C}A^\top D(\mathbf{1}-D(Az+c\mathbf{1})) - \rho u \\ \frac{\partial T}{\partial c} &= -2\mathcal{C}\mathbf{1}^\top D(\mathbf{1}-D(Az+c\mathbf{1}))\end{aligned}$$

which can be integrated into the following:

$$\begin{bmatrix} G+\rho I+2\mathcal{C}A^\top D^2A & 2\mathcal{C}A^\top D\mathbf{1} \\ 2\mathcal{C}\mathbf{1}^\top D^2A & 2\mathcal{C}\mathbf{1}^\top D^2\mathbf{1}\end{bmatrix}\begin{bmatrix}z\\c\end{bmatrix} = \begin{bmatrix}2\mathcal{C}A^\top D\mathbf{1}+\rho u \\ 2\mathcal{C}\mathbf{1}^\top D\mathbf{1},\end{bmatrix}$$

which has a positive definite coefficient matrix such that:

$$\begin{bmatrix}z\\c\end{bmatrix} = \begin{bmatrix} G+\rho I+2\mathcal{C}A^\top D^2A & 2\mathcal{C}A^\top D\mathbf{1} \\ 2\mathcal{C}\mathbf{1}^\top D^2A & 2\mathcal{C}\mathbf{1}^\top D^2\mathbf{1}\end{bmatrix}^{-1}\begin{bmatrix}2\mathcal{C}A^\top D\mathbf{1}+\rho u \\ 2\mathcal{C}\mathbf{1}^\top D\mathbf{1},\end{bmatrix} \tag{7}$$

Corollary 3.2 For the quadratic loss function, step 6 of Algorithm 0 is obtained from (7).

To conclude, we present two sparse ℓ_0-norm, kernel-free quadratic surface SVMs that employ hinge and quadratic loss functions, respectively. Because the presence of a sparsity constraint renders these formulations NP-hard, an efficient solution method is essential. We place both models within the unified framework (1) and develop a penalty decomposition algorithm that introduces an auxiliary variable u. The associated subproblem (4) admits a simple closed-form solution (5). For the hinge-loss-based model (ℓ_0-QSVM), the z-subproblem is solved most effectively via convex duality, with the primal variables recovered through (6). In the quadratic-loss-based model (LS-ℓ_0-QSVM), the z-subproblem itself has the closed-form solution given

in (7). Consequently, each iteration of our algorithm remains computationally efficient while enforcing exact sparsity.

3.3 Convergence of ℓ_0-Regularized QSVM Penalty Decomposition Algorithm

While convergence guarantees are well established for convex models, the presence of a nonconvex and nonsmooth ℓ_0-norm constraint introduces significant theoretical and computational challenges. The proposed models belong to the class of combinatorial optimization problems, for which finding global minimizers is generally intractable. To address this, we adopt a penalty-decomposition framework whose subproblems are either convex or admit closed-form solutions, depending on the chosen loss function. For the hinge loss, the main subproblem reduces to a linearly constrained quadratic program, which can be efficiently solved via duality theory. In the case of quadratic loss, the main subproblem simplifies to solving a system of linear equations, for which a closed-form solution is obtained. Specifically, Algorithm 0 is a direct application of the general penalty decomposition method proposed in [9], applied to the context of ℓ_0-regularized kernel-free quadratic SVMs. This framework circumvents the combinatorial difficulty of direct ℓ_0-minimization by solving a sequence of tractable subproblems, each enforcing sparsity in a relaxed yet principled manner.

In the subsequent analysis, we demonstrate that Algorithm 0 converges to a solution satisfying the Lu-Zhang stationarity conditions–a generalized notion of first-order optimality tailored to nonconvex problems with cardinality constraints. This concept extends the classical KKT framework and is particularly suitable for settings where conventional convex stationarity conditions fail to capture meaningful structure. To formally analyze convergence, we consider a general class of sparse optimization problems with structural constraints, as described in problem (P). These problems are motivated by applications in signal recovery, image reconstruction, and data compression [30] and can be formulated as:

$$\min_{v\in\mathbb{R}^{\hat{\jmath}}} \varphi(v) \qquad \text{subject to} \qquad \psi(v)\le 0, \quad \chi(v)=0, \quad \|v\|_0\le k, \tag{P}$$

where φ, ψ, and χ are continuously differentiable functions, though not necessarily convex, and $k<\hat{\jmath}$ denotes the desired sparsity level.

A point v satisfying $\psi(v)\le 0$ and $\chi(v)=0$ is called a *Lu-Zhang stationary point* for problem (P) if there exists an index set $\mathcal{L}\subseteq[\hat{\jmath}]$ with $|\mathcal{L}|=k$ such that $v_j=0$ for all $j\in\mathcal{L}^c:=[\hat{\jmath}]\setminus\mathcal{L}$, and there exist multipliers $\lambda\in\mathbb{R}^{\tilde{q}}$ and $\mu\in\mathbb{R}^{\check{r}}$ satisfying:

$$\begin{cases} \nabla\varphi(v)-\lambda^\top\nabla\psi(v)-\mu^\top\nabla\chi(v)-\omega=0, \\ \lambda_i\ge 0, \quad \lambda_i\cdot\psi_i(v)=0, \quad \forall i, \quad \text{and} \quad \omega_j=0, \quad \forall j\in\mathcal{L}^c. \end{cases} \tag{8}$$

While alternative stationarity concepts exist (e.g., basic feasibility [31]), Lu-Zhang stationarity provides a suitable theoretical foundation for convergence analysis in nonconvex sparse settings. It is known that basic feasible points are also Lu-Zhang stationary points, though the converse does not always hold [32].

To underscore the importance of Lu-Zhang stationarity, recall that Theorem 2.1 in [9] establishes that any local minimizer of problem (P) satisfies the Lu-Zhang stationarity condition under Robinson's constraint qualification. Furthermore, when φ and ψ are convex and χ is affine, any Lu-Zhang stationary point with support of size exactly k is a local minimizer. In our case, both proposed models—ℓ_0-QSVM' and LS-ℓ_0-QSVM'—satisfy these assumptions: the objective functions are convex (since $G \succeq 0$), and the structural constraints are linear. Under such linearity, Robinson's condition is automatically satisfied. Therefore, it suffices to show that Algorithm 0 converges to a Lu-Zhang stationary point, which implies that the obtained solution satisfies the necessary conditions for local optimality. Moreover, when such a solution has full support (i.e., $\|x\|_0 = k$), it is in fact a local minimizer of the corresponding model.

Next, observe that Algorithm 0 is a direct application of the general penalty decomposition framework developed in [9], and thus inherits its convergence guarantees without further modification. In particular, Theorem 4.3 in [9] guarantees convergence to a Lu-Zhang stationary point, provided that Robinson's constraint qualification holds—an assumption automatically satisfied in our models due to the linearity of their constraint sets $\mathbb{K}$. Furthermore, since the objective functions in both formulations are convex, any Lu-Zhang stationary point with full support of size k is a local minimizer. Hence, Algorithm 0 produces feasible k-sparse solutions where the gradient of the Lagrangian vanishes on the support, certifying convergence to a Lu-Zhang stationary point of problem (1). More specifically, in solving the hinge loss model ℓ_0-QSVM', Algorithm 0 obtains a point $(z, c, \xi) \in \mathbb{R}^{\frac{n(n+1)}{2}+n} \times \mathbb{R} \times \mathbb{R}^m$ such that z is supported on a set $\mathcal{L} \subseteq \left[\frac{n(n+1)}{2} + n\right]$ with $|\mathcal{L}| = k$, and the following conditions hold:

$$\begin{cases} Gz - \sum_{i=1}^m \lambda_i y_i r_i - \omega = 0, & \text{where } \omega_j = 0 \text{ for } j \in \mathcal{L} \\ \sum_{i=1}^m \lambda_i y_i = 0, & \\ \mathcal{C} - \lambda_i - \bar{\lambda}_i = 0, & \forall i \in [m] \\ \lambda_i \geq 0, \quad \lambda_i \cdot \big(y_i(z^\top r_i + c) + \xi_i - 1\big) = 0, & \forall i \in [m] \\ \bar{\lambda}_i \geq 0, \quad \bar{\lambda}_i \cdot \xi_i = 0, & \forall i \in [m] \\ y_i(z^\top r_i + c) \geq 1 - \xi_i, \quad \xi_i \geq 0, & \forall i \in [m] \end{cases}$$

Furthermore, when solving the quadratic loss model LS-ℓ_0-QSVM', Algorithm 0 returns a point (z, c, ξ) satisfying $z_j = 0$ for all $j \in \mathcal{L}^c := \left[\frac{n(n+1)}{2} + n\right] \setminus \mathcal{L}$ with $|\mathcal{L}| = k$, and the following Lu-Zhang stationarity conditions hold:

$$\begin{cases} Gz - \sum_{i=1}^m \mu_i y_i r_i - \omega = 0, \text{ where } \omega_j = 0 \text{ for } j \in \mathcal{L} \\ \sum_{i=1}^m \mu_i y_i = 0, \\ 2\mathcal{C}\xi_i + \mu_i = 0, \qquad \forall i \in [m] \\ y_i(z^\top r_i + c) = 1 - \xi_i, \qquad \forall i \in [m]. \end{cases}$$

4 Numerical Experiments

In this section, we conduct numerical experiments to evaluate the effectiveness of the proposed ℓ_0-QSVM and LS-ℓ_0-QSVM models for classification. We first describe the experimental settings and the benchmark models used for comparison. In Section 4.1, we present results on several public benchmark datasets. Subsequently, in Section 4.2, we apply the relatively more efficient LS-ℓ_0-QSVM model to credit scoring using real-world credit datasets.

The implementation of the proposed models ℓ_0-QSVM and LS-ℓ_0-QSVM follows Algorithm 0, guided by the instructions in Corollaries 3.1 and 3.2 for Step 6. For comparison, we implement several widely used SVM-based models, including QSVM, linear SVM (LSVM), least squares SVM (LS-SVM), radial basis function kernel SVM (SVM-rbf), and quadratic kernel SVM (SVM-Q2). In addition, we include several sparse SVM models, including the linear ℓ_1-regularized SVM (ℓ_1-SVM), the quadratic ℓ_1-regularized SVM (ℓ_1-QSVM), and the least squares ℓ_0-regularized SVM (LS-ℓ_0-SVM) [33].

Model	Algorithm/Package/Solver	Parameters
LSVM	Scikit-learn	$\mathcal{C}$
LS-SVM	Numpy	$\mathcal{C}$
ℓ_1-SVM	COPT	$\mathcal{C}$
LS-ℓ_0-SVM	ADMM	$\mathcal{C}, \gamma, \lambda$
SVM-rbf	Scikit-learn	$\mathcal{C}, \gamma$
SVM-Q2	Scikit-learn	$\mathcal{C}, \gamma$
QSVM	COPT	$\mathcal{C}$
ℓ_1-QSVM	COPT	$\mathcal{C}_1, \mathcal{C}_2$
ℓ_0-QSVM	Algorithm 0, COPT	$\mathcal{C}, k$
LS-ℓ_0-QSVM	Algorithm 0, (7), COPT	$\mathcal{C}, k$

Table 1: Information of Implemented Models

All experiments are implemented in Python 3.9.12 and performed on a MacBook Pro equipped with an Apple M3 Pro Chip and 18GB of Memory. The Cardinal Optimizer (COPT) solver, version 8.0.2, is used to solve the optimization problems in the implementation. The Python package *scikit-learn* [34] is also used to implement linear and kernel-based SVM models.

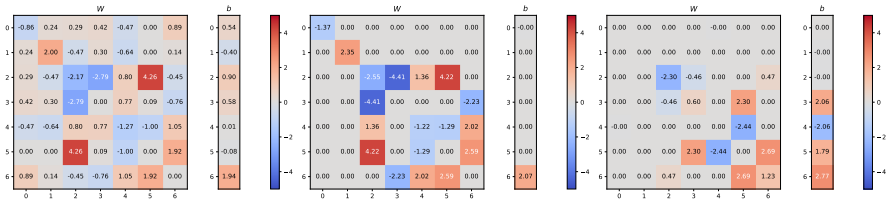
4.1 Benchmark Datasets

All models are implemented on the datasets listed in Table 2, which are widely used benchmarks for assessing newly developed methods in the literature. All datasets are binary except the *iris* dataset, which contains three classes. For the multi-class setting, we adopt a one-vs-rest scheme and assign the predicted label by majority voting. The grid search with a five-fold cross validation is performed to tune the parameters of all implemented models. The mean accuracy and standard deviation are recorded for each model on each dataset in Table 2.

Table 2: Results on Public Benchmark Datasets

Model		abalone (2854, 8)		CTG (2126, 22)		Ecoli (336, 7)		glass (214, 9)		haberman (306, 3)		Immunotherapy (90, 7)		iris (150, 4)	
		acc	f1	acc	f1	acc	f1	acc	f1	acc	f1	acc	f1	acc	f1
LSVM	mean	87.21	87.18	96.90	98.02	97.02	96.48	92.97	84.25	73.57	17.17	81.11	89.06	96.67	96.80
	std	1.31	0.69	0.56	0.37	1.83	2.31	2.96	4.86	9.12	18.32	7.45	4.79	3.33	3.10
LS-SVM	mean	86.51	86.48	95.16	96.98	97.03	96.55	92.04	78.17	73.57	17.17	80.00	88.47	83.33	83.59
	std	1.43	0.86	0.79	0.48	1.48	1.65	3.20	16.21	9.12	18.32	8.43	5.44	8.50	6.31
SVM-rbf	mean	**89.70**	**89.95**	**98.50**	**99.03**	97.02	96.48	**94.39**	**86.75**	71.28	27.37	78.89	87.92	96.67	97.02
	std	1.14	0.75	0.35	0.24	1.06	1.40	2.11	8.36	9.68	19.29	9.94	6.25	4.71	4.11
SVM-Q2	mean	89.28	89.43	98.31	98.91	97.03	96.60	93.92	85.56	72.58	18.06	80.00	88.19	96.67	97.19
	std	1.08	0.70	0.58	0.38	1.03	0.87	3.90	12.17	6.92	13.00	10.83	6.81	5.77	4.64
ℓ_1-SVM	mean	87.70	87.73	97.18	98.19	96.43	95.80	92.98	84.32	71.28	3.33	80.00	88.27	96.67	96.80
	std	1.10	0.53	0.55	0.35	1.32	1.58	2.38	3.71	8.17	7.45	11.52	7.23	3.33	3.10
LS-ℓ_0-SVM	mean	86.69	86.69	95.39	97.12	97.03	96.55	92.04	78.17	73.57	17.17	80.00	88.47	84.00	83.95
	std	1.60	1.22	0.56	0.34	1.48	1.65	3.20	16.21	9.12	18.32	8.43	5.44	5.48	4.09
QSVM	mean	88.89	89.11	98.21	98.85	97.03	96.60	92.98	83.68	72.58	18.06	81.11	88.79	97.33	97.43
	std	1.50	1.01	0.21	0.15	1.03	0.87	4.68	14.01	6.92	13.00	11.52	7.17	2.79	2.77
ℓ_1-QSVM	mean	88.79	88.99	98.21	98.85	97.03	96.60	94.39	86.49	73.56	19.37	81.11	88.83	97.33	97.32
	std	1.47	0.98	0.27	0.18	1.03	0.87	3.54	12.14	6.85	13.60	11.52	7.27	1.49	1.55
ℓ_0-QSVM	mean	89.21	89.32	98.40	98.97	97.02	96.48	93.44	86.48	73.24	19.11	81.11	88.85	**97.33**	**97.56**
	std	1.44	1.03	0.35	0.24	1.06	1.40	3.86	4.17	6.98	13.18	10.83	6.47	3.65	3.37
LS-ℓ_0-QSVM	mean	88.44	88.54	96.24	97.62	**97.33**	**96.98**	93.43	85.27	**74.22**	**32.10**	**83.33**	**89.56**	96.67	97.00
	std	1.65	1.18	0.92	0.58	1.62	1.70	3.55	8.54	8.10	11.55	14.16	8.94	5.77	4.68

From the results, we can observe that the proposed ℓ_0-QSVM and the LS-ℓ_0-QSVM provide the highest in both accuracy and f1-score on the *Ecoli*, *haberman*, *Immunotherapy*, and *Iris* datasets. Specifically, the LS-ℓ_0-QSVM model outperforms ℓ_0-QSVM on most of these cases. Overall, the proposed models are consistently competitive with commonly used linear and kernel-based SVM benchmark models.



(a) ℓ_1-QSVM ($\mathcal{C}_1 = 256, \mathcal{C}_2 = 100$)

(b) ℓ_0-QSVM ($\mathcal{C} = 1024, k = 12$)

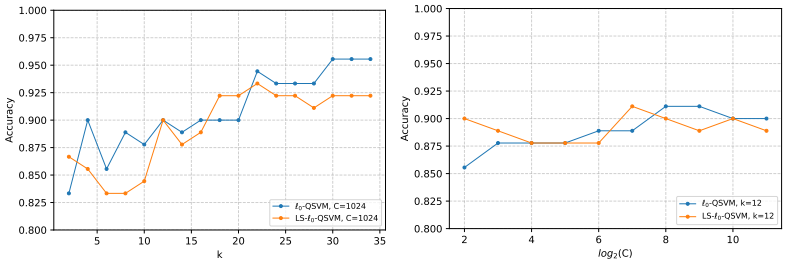
(c) LS-ℓ_0-QSVM ($\mathcal{C} = 1024, k = 12$)

Fig. 1: The sparsity of optimal coefficients W^* and b^* by three sparse quadratic models on the *Immunotherapy* dataset.

The proposed ℓ_0-QSVM and LS-ℓ_0-QSVM models produce sparse solutions, as expected from their formulation. To illustrate this property, we visualize the optimal coefficients (W, b) of the two proposed models and the ℓ_1-QSVM model [12] on the *Immunotherapy* dataset in Figure 1. It is evident that both ℓ_0-QSVM and LS-ℓ_0-QSVM yield well-controlled sparsity patterns. Although LS-ℓ_0-QSVM produces a different sparsity structure in (W^*, b^*) compared with ℓ_0-QSVM, this difference stems from their distinct classification mechanisms.

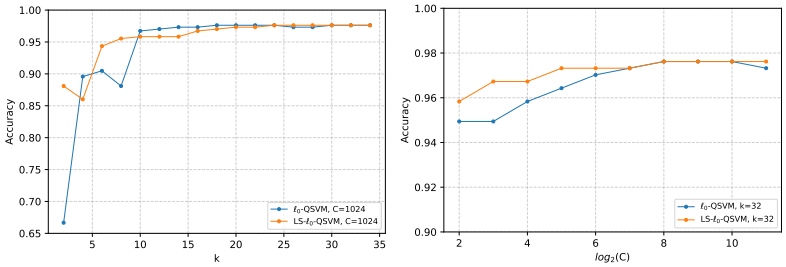
In contrast, while increasing the penalty parameter $\mathcal{C}_2$ in ℓ_1-QSVM can encourage sparsity of solutions, it does not allow precise control over the number of nonzero coefficients. As a result, it is difficult to obtain solutions with a predetermined sparsity level using ℓ_1-QSVM. By comparison, the proposed ℓ_0-QSVM and LS-ℓ_0-QSVM models enable direct control of sparsity, allowing solutions with an explicitly specified number of nonzero coefficients.

Next, we investigate the effects of the parameters $\mathcal{C}$ and k on classification performance. Figures 2c and 2a present the accuracy curves of ℓ_0-QSVM (blue) and LS-ℓ_0-QSVM (yellow) on the *Ecoli* and *Immunotherapy* datasets, respectively, by fixing $\mathcal{C}$ and varying k. We observe that when k increases from a small value, the classification accuracy improves steadily. However, after k exceeds a certain threshold, the performance gain becomes marginal, particularly for the LS-ℓ_0-QSVM model. This behavior indicates that once a sufficient number of informative coefficients are selected, adding additional nonzero terms contributes little to further improvement. On both datasets, especially *Ecoli*, accuracy increases rapidly for small k, then the curve becomes flat. This indicates that only a relatively small subset of interacted and linear terms is necessary to achieve an optimal performance. This strongly supports the effectiveness of sparsity control.



(a) *Immunotherapy*: Accuracy vs. k ($\mathcal{C} = 1024$)

(b) *Immunotherapy*: Accuracy vs. C ($k = 12$)



(c) *Ecoli*: Accuracy vs. k ($\mathcal{C} = 1024$)

(d) *Ecoli*: Accuracy vs. C ($k = 32$)

Fig. 2: The trend of accuracy against parameters C and k by the proposed ℓ_0 regularized QSVM models on the *Immunotherapy* and *Ecoli* datasets.

Similarly, Figures 2d and 2b illustrate the accuracy curves obtained by fixing k and varying $\mathcal{C}$. The results show that the performance of both models remains relatively stable across a wide range of $\mathcal{C}$ values, provided that k is appropriately chosen. This suggests that the proposed models are not highly sensitive to the penalty parameter $\mathcal{C}$, and that sparsity control through k plays a more dominant role in determining predictive performance.

4.2 Application: Credit Scoring

Credit scoring plays a crucial role in financial institutions by supporting risk assessment, underwriting decisions, and portfolio management. Accurate credit scoring helps lenders reduce default risk, improve pricing strategies, and enhance operational efficiency and financial stability. Traditional credit scoring approaches, such as expert-driven scorecards and statistical models [35, 36], are largely based on handcrafted features and domain expertise, which may limit their flexibility when dealing with complex borrower behavior.

Credit scoring can be naturally formulated as a classification problem, where the objective is to assign each applicant to a risk category based on observed attributes. In recent years, machine learning methods have been increasingly adopted for credit

scoring, including decision trees [37], logistic regression [38], and neural networks [39], due to their ability to capture nonlinear relationships and feature interactions. Among these methods, SVM have demonstrated competitive performance in credit scoring applications [40]. In this section, we evaluate the effectiveness of the proposed LS-ℓ_0-QSVM model for credit scoring using five publicly available credit scoring datasets. Five-fold cross validation is also applied.

We consider five credit datasets from both public repositories and industry sources, including personal and corporate credit datasets with varying sample sizes and feature sets. The basic information for the datasets, including their sources and dimensions, is listed in Table 3. Among them, the *German Credit Dataset* (*GCD*), *Japanese Credit* (*JAP*), and *Australian Credit* (*AUS*) datasets are obtained from the UCI Machine Learning Repository. The *GCD* contains 1000 loan applicants described by 20 attributes. These features include financial indicators, employment information, demographic characteristics, and other credit-related records. The response variable is binary, indicating whether an applicant is classified as a good credit risk or a bad credit risk. Similarly, the *AUS* dataset comprises 690 samples and 14 attributes. It is a famous binary benchmark dataset with Australian credit card applications. The *JAP* dataset contains 653 instances with 15 attributes. For both the *GCD* and *JAP* datasets, the original attribute names and categorical values were replaced with symbolic representations to protect data confidentiality.

In addition to the UCI datasets, we also include two smaller datasets. The *CCC* dataset is obtained from a private credit union and focuses on corporate loan evaluation. It contains 106 samples with 6 interpretable financial and compliance-related features, such as the number of collaborations, integrity violations, and blacklist penalties. The response label indicates the corporation's credit status. The *credit small* dataset is collected from Kaggle and consists of 164 personal credit assessments and 7 demographic and socioeconomic features, including age, income, education, marital status, and home ownership. The response variable is binary and reflects the credit risk level of individuals.

In this application The proposed LS-ℓ_0-QSVM model is implemented on the datasets in Table 3 along with benchmark models. The mean classification accuracies and the F1-scores are recorded in Table 4.

From Table 4, we observe that the proposed LS-ℓ_0-QSVM model achieves the highest mean accuracy and F1-scores on most of the tested credit datasets. Although it does not outperform SVM-Q2 on the *CCC* dataset, its performance remains highly competitive. These results provide strong empirical evidence that the proposed LS-ℓ_0-QSVM model is effective across diverse credit evaluation scenarios.

We further examine the impact of the sparsity parameter k in the proposed LS-ℓ_0-QSVM model on the *GCD*, *AUS*, and *CCC* datasets. Figure 3 illustrates the classification accuracy as a function of k while fixing $\mathcal{C}$.

For the *CCC* and *AUS* datasets, the classification accuracy remains relatively stable across a wide range of k values once an appropriate $\mathcal{C}$ is selected, indicating that performance is not highly sensitive to the exact sparsity level. In contrast, on the *GCD* dataset, the accuracy initially improves as k increases and then gradually reaches a plateau. This behavior suggests that adopting more features in the model

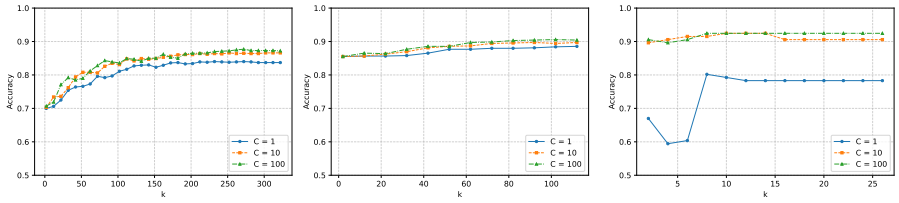
Table 3: Credit Data Information

Dataset	Source	Type	Dimension (m, n)	Features
CCC	private credit union	corporation loan	$(106, 6)$	1. Number of collaborations or total deliveries within the past three years 2. Number of integrity violations 3. Number of delivery-related violations 4. Number of quality-related violations 5. Credit China – number of positive records 6. Credit China – number of blacklist penalties
credit small	Kaggle	personal credit	$(164, 7)$	1. Age 2. Gender 3. Income 4. Education 5. Marital Status 6. Number of Children 7. Home Ownership
GCD	UCI	personal credit	$(1000, 20)$	1. Status of existing checking account 2. Duration in month 3. Credit history 4. Loan purpose 5. Credit amount 6. Status of savings account/bonds 7. Years of current employment 8. Installment rate in percentage of disposable income 9. Gender and marriage status 10. Other debtors / guarantors 11. Length of present residence 12. Property status 13. Age 14. Other installment plans 15. Housing 16. Number of existing credits at this bank 17. Job status 18. Number of people being liable to provide maintenance for 19. Phone indicator 20. Foreign worker indicator
JAP	UCI	personal credit	$(653, 15)$	Processed numeric variables
AUS	UCI	personal credit	$(690, 14)$	Processed numeric variables

Table 4: Results on Credit Data

Model	AUS		JAP		GCD		CCC	
	acc	f1	acc	f1	acc	f1	acc	f1
LSVM	87.10±1.88	87.83±2.05	86.83±1.60	86.56±1.42	76.90±1.85	55.79±2.47	89.70±9.57	85.23±13.71
LS-SVM	86.38±2.37	86.82±2.92	86.67±1.71	86.46±1.66	76.80±2.11	54.81±2.99	87.75±5.38	84.24±8.16
SVM-rbf	86.67±1.75	87.27±2.25	86.21±1.36	85.93±1.71	76.10±3.38	54.53±6.71	90.56±7.53	86.49±11.95
SVM-Q2	86.23±3.28	87.39±3.46	86.37±1.40	86.12±1.59	75.70±3.29	56.56±5.86	**92.47±7.94**	**89.43±11.85**
ℓ_1-SVM	85.51±1.85	85.93±2.07	86.67±1.42	86.45±1.77	77.20±1.25	55.85±2.95	91.56±8.41	88.27±12.06
LS-ℓ_0-SVM	86.23±2.51	86.69±2.83	86.67±1.71	86.46±1.66	77.20±1.72	55.75±2.67	87.75±5.38	84.24±8.16
QSVM	85.94±2.64	87.09±2.50	86.52±1.53	86.31±1.59	76.90±2.10	57.31±5.07	91.56±8.41	88.27±12.06
ℓ_1-QSVM	86.38±2.87	87.53±2.95	86.52±1.53	86.31±1.59	77.40±0.55	57.42±2.25	91.56±8.41	88.27±12.06
LS-ℓ_0-QSVM	**87.25±1.67**	**88.28±2.06**	**87.14±1.63**	**86.59±1.85**	**77.50±1.73**	**57.48±5.50**	91.56±8.41	88.27±12.06

Model	credit small	
	acc	f1
LSVM	98.18±2.71	98.13±2.82
LS-SVM	90.87±8.55	76.97±19.36
SVM-rbf	98.18±2.71	98.13±2.82
SVM-Q2	98.18±2.71	98.13±2.82
ℓ_1-SVM	98.18±2.71	98.13±2.82
LS-ℓ_0-SVM	90.87±8.55	76.97±19.36
QSVM	97.58±2.54	97.66±2.62
ℓ_1-QSVM	98.18±2.71	98.13±2.82
LS-ℓ_0-QSVM	**99.39±1.36**	**99.33±1.49**



(a) GCD (b) AUS (c) CCC

Fig. 3: The trend of classification accuracy from LS-ℓ_0-QSVM changes as k increases. Parameter $\mathcal{C}$ is fixed to be $1, 10$ or 100.

benefits performance up to a certain point, beyond which further increases in model complexity provide negligible improvement in classification accuracy.

We further analyze the sparsity of the proposed LS-ℓ_0-QSVM model on the *GCD* data. By the cross validation results in Table 4, the optimal parameters of the proposed LS-ℓ_0-QSVM model are $(\mathcal{C}, k) = (256, 47)$. The corresponding optimal solution (W^*, b^*) is plot in Figure 4 along with the result of the logistic regression (LR) model on the *GCD* data. In the plot, the non-zero entries of the solution from LS-ℓ_0-QSVM are highlighted in blue, and the coefficients that are statistically significant at the 95% confidence level in the LR model are highlighted in green.

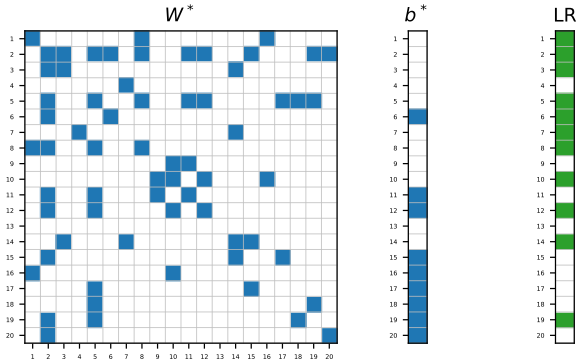


Fig. 4: W^* and b^* are sparse optimal solutions provided by LS-ℓ_0-QSVM with non-zero entries colored in blue. Significant coefficients by LR that are statistically significant at the 95% confidence level are highlighted in green.

We have the following observations from Figure 4.

1. The proposed LS-ℓ_0-QSVM model tells that the credit risk is not driven by single features alone, but by how the core financial variables combine with applicant profile and repayment burdens. Many of these affects are not shown in optimal b^*, such as the *Duration (2)*, the *Credit amount (5)* and the *Installment rate (8)*, they appear in the interactions in the optimal quadratic coefficient W^*. These features

indicates that the default risk is better characterized by interactions between core financial terms and applicant context.

2. In contrast, the linear vector b^* retains a smaller set of main effects, including *Savings account/bonds (6)*, *Length of present residence (11)*, *Property status (12)*, *Housing (15)*, *Number of existing credits (16)*, *Job status (17)*, *Dependents (18)*, *Phone indicator (19)*, and *Foreign worker (20)*, suggesting these factors contribute more directly without requiring interaction terms.
3. Notice that LR identifies a larger number of significant predictors among the financial variables (Features 1 to 8) than among the applicant context variables (Features 9 to 20). Compared with LR, the proposed LS-ℓ_0-QSVM model shifts much of the explanatory power of the financial features into the interaction matrix W^*, indicating that their effects are primarily conditional on other variables rather than purely linear. For the applicant context features, the proposed LS-ℓ_0-QSVM model captures their influence through both the linear coefficients b^* and the quadratic terms represented by the diagonal elements of W^*.

The empirical results demonstrate that the proposed LS-ℓ_0-QSVM model is well-suited for credit scoring applications, and has practical value for real-world credit risk assessment and decision-making.

5 Conclusion

This paper introduces ℓ_0 regularized kernel free quadratic surface support vector machine models for binary classification. By enforcing an explicit sparsity constraint, the proposed models produce sparse solutions that improve interpretability while mitigating the risk of overfitting. Although these models are inherently intractable due to the combinatorial nature of the ℓ_0 penalty, we develop an efficient penalty decomposition algorithm for their implementation. For the hinge loss formulation, we exploit duality theory to solve the resulting subproblem at low computational cost. For the quadratic loss, we show that the subproblem admits a closed form solution. We provide a rigorous convergence analysis of the algorithm, establishing a theoretical foundation that supports its practical use. Numerical experiments confirm the effectiveness of the proposed models on classification tasks. We systematically examine their sparsity properties and investigate the influence of key parameters. Finally, the proposed model LS-ℓ_0-QSVM is applied to real world credit scoring datasets, where it demonstrates strong predictive performance and interpretability in practical credit risk assessment applications.

The work presented in this paper opens several avenues for future investigation. First, integrating the proposed models with the twin SVM framework could yield more effective solutions for multiclass and imbalanced classification problems. Second, developing adaptive strategies for parameter selection would enhance computational efficiency, particularly when scaling to large datasets. Finally, the interpretability inherent in sparse quadratic decision surfaces positions these models as promising tools for high stakes domains such as healthcare and transportation, where datasets are often high dimensional and transparency is critical for informed decision making.

Declarations

Competing Interests

The authors declare that they have no financial or non-financial interests that are directly or indirectly related to the work submitted for publication.

Funding

The authors declare that they have not received any funding, financial support, or sponsorship from any organization or agency for the preparation of this work.

Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, the authors used generative AI technologies to improve language and readability with extreme caution. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

[1] Lyaqini, S., Hadri, A., Ellahyani, A., Nachaoui, M.: Primal dual algorithm for solving the nonsmooth twin svm. Engineering Applications of Artificial Intelligence **128**, 107567 (2024)

[2] Ke, T., Liao, Y., Wu, M., Ge, X., Huang, X., Zhang, C., Li, J.: Maximal margin hyper-sphere svm for binary pattern classification. Engineering Applications of Artificial Intelligence **117**, 105615 (2023)

[3] Chen, Y., Huang, W., Nguyen, L., Weng, T.-W.: On the equivalence between neural network and support vector machine. Advances in Neural Information Processing Systems **34**, 23478–23490 (2021)

[4] Chen, X., Lai, Z.: Jointly sparse least square support vector machine. Computers and Electrical Engineering **101**, 108078 (2022)

[5] Landeros, A., Lange, K.: Algorithms for sparse support vector machines. Journal of Computational and Graphical Statistics **32**(3), 1097–1108 (2023)

[6] Xie, Z., Xu, Y.: Sparse group lasso based uncertain feature selection. International Journal of Machine Learning and Cybernetics **5**, 201–210 (2014)

[7] Pappu, V., Panagopoulos, O.P., Xanthopoulos, P., Pardalos, P.M.: Sparse proximal support vector machines for feature selection in high dimensional datasets. Expert Systems with Applications **42**(23), 9183–9191 (2015)

[8] Moosaei, H., Hladík, M.: Sparse solution of least-squares twin multi-class support vector machine using ℓ_0 and ℓ_p-norm for classification and feature selection. Neural Networks **166**, 471–486 (2023)

[9] Lu, Z., Zhang, Y.: Sparse approximation via penalty decomposition methods. SIAM Journal on Optimization **23**(4), 2448–2478 (2013)

[10] Moosaei, H., Mousavi, A., Hladík, M., Gao, Z.: Sparse l1-norm quadratic surface support vector machine with universum data. Soft Computing **27**(9), 5567–5586 (2023)

[11] Gao, Z., Wang, Y., Huang, M., Luo, J., Tang, S.: A kernel-free fuzzy reduced quadratic surface ν-support vector machine with applications. Applied Soft Computing **127**, 109390 (2022)

[12] Mousavi, A., Gao, Z., Han, L., Lim, A.: Quadratic surface support vector machine with l1 norm regularization. Journal of Industrial & Management Optimization **18**(3) (2022)

[13] Xiao, J., Liu, G., Huang, M., Yin, Z., Gao, Z.: A kernel-free l1 norm regularized ν-support vector machine model with application. International Journal of Industrial Engineering Computations **14**(4), 691–706 (2023)

[14] Si, Q., Yang, Z., Zhao, J., Chen, R.: A kernel-free quadratic surface twin support vector machine with capped l1-norm distance metric for robust classification. Applied Soft Computing **180**, 113317 (2025)

[15] Zhou, J., Tian, Y., Luo, J., Zhai, Q.: A kernel-free laplacian quadratic surface optimal margin distribution machine with application to credit risk assessment. Applied Soft Computing **133**, 109931 (2023)

[16] Wang, J., Guo, F., Shen, J.: An l_2 regularization reduced quadratic surface support vector machine model. Journal of Combinatorial Optimization **49**(2), 29 (2025)

[17] Yang, X., Yang, Z., Ye, J., Chen, Y.: Kernel-free quadratic proximal support vector machine with lp-norm regularization. Engineering Applications of Artificial Intelligence **151**, 110658 (2025)

[18] Shen, J., Mousavi, S.: Least sparsity of p-norm based optimization problems with $p > 1$. SIAM Journal on Optimization **28**(3), 2721–2751 (2018)

[19] Zong-Ben, X., Hai-Liang, G., Yao, W., *et al.*: Representative of l1/2 regularization among lq ($0 < q \leq 1$) regularizations: an experimental study based on phase diagram. Acta Automatica Sinica **38**(7), 1225–1228 (2012)

[20] Lu, Z., Sun, Z., Zhou, Z.: Penalty and augmented lagrangian methods for constrained dc programming. Mathematics of Operations Research **47**(3), 2260–2285 (2022)

[21] Gupta, V.: Greedy algorithm for multiway matching with bounded regret. Operations Research **72**(3), 1139–1155 (2024)

[22] Hazimeh, H., Mazumder, R., Saab, A.: Sparse regression at scale: Branch-and-bound rooted in first-order optimization. Mathematical Programming **196**(1), 347–388 (2022)

[23] Huang, X., Liu, Y., Shi, L., Van Huffel, S., Suykens, J.A.: Two-level ℓ_1 minimization for compressed sensing. Signal Processing **108**, 459–475 (2015)

[24] Chandra, M.A., Bedi, S.: Survey on svm and their application in image classification. International Journal of Information Technology **13**(5), 1–11 (2021)

[25] Hofmann, T., Schölkopf, B., Smola, A.: Kernel methods in machine learning. The Annals of Statistics **36**(3), 1171–1220 (2008)

[26] Dagher, I.: Quadratic kernel-free non-linear support vector machine. Journal of Global Optimization **41**(1), 15–30 (2008)

[27] Yao, L., Zeng, F., Li, D.-H., Chen, Z.-G.: Sparse support vector machine with lp penalty for feature selection. Journal of Computer Science and Technology **32**(1), 68–77 (2017)

[28] Blanco, V., Puerto, J., Rodriguez-Chia, A.M.: On lp-support vector machines and multidimensional kernels. Journal of Machine Learning Research **21**(14), 1–29 (2020)

[29] Ma, J., Zhou, S., Chen, L., Wang, W., Zhang, Z.: A sparse robust model for large scale multi-class classification based on k-svcr. Pattern Recognition Letters **117**, 16–23 (2019)

[30] Mousavi, A., Rezaee, M., Ayanzadeh, R.: A survey on compressive sensing: classical results and recent advancements. Journal of Mathematical Modeling **8**(3), 309–344 (2020)

[31] Beck, A., Hallak, N.: On the minimization over sparse symmetric sets: projections, optimality conditions, and algorithms. Mathematics of Operations Research **41**(1), 196–223 (2016)

[32] *et al.*: Convergent inexact penalty decomposition methods for cardinality-constrained problems. Journal of Optimization Theory and Applications **188**(2), 473–496 (2021)

[33] Tang, Q., Li, G.: Sparse l0-norm least squares support vector machine with

feature selection. Information Sciences **670**, 120591 (2024)

[34] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., *et al.*: Scikit-learn: Machine learning in python. the Journal of Machine Learning Research **12**, 2825–2830 (2011)

[35] Mays, E.: Handbook of Credit Scoring. Global Professional Publishing, Chicago (1995)

[36] Splett, N.S., Barry, P.J., Dixon, B.L., Ellinger, P.N.: A joint experience and statistical approach to credit scoring. (1994)

[37] Dumitrescu, E., Hué, S., Hurlin, C., Tokpavi, S.: Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. European Journal of Operational Research **297**(3), 1178–1192 (2022)

[38] Bolton, C.: Logistic regression and its application in credit scoring. PhD thesis, University of Pretoria, South Africa (2009)

[39] West, D.: Neural network credit scoring models. Computers & Operations Research **27**(11-12), 1131–1152 (2000)

[40] Huang, C.-L., Chen, M.-C., Wang, C.-J.: Credit scoring with a data mining approach based on support vector machines. Expert Systems with Applications **33**(4), 847–856 (2007)

A Tuning Parameters

For each implemented model, the values of optimal parameter s are reported through the 5-fold cross validation with grid search in the corresponding ranges. The ranges of tuning parameters are provided as follows.

- For the proposed ℓ_0-QSVM and LS-ℓ_0-QSVM models, the penalty parameter is selected from $\mathcal{C} \in \{4^{-1}, \ldots, 4^{8}\}$. The sparsity parameter k is chosen from a uniformly spaced sequence starting at 2 and increasing in increments of δ, up to the maximum possible number of coefficients $\frac{n(n+1)}{2} + n$, where δ is typically selected from $\{2, 5\}$ depending on the dimensionality of the dataset.
- For LSVM, LS-SVM, ℓ_1-SVM, LS-ℓ_0-SVM, and QSVM, the penalty parameter is selected from $\mathcal{C} \in \{2^{-2}, \ldots, 2^{10}\}$.
- For the LS-ℓ_0-SVM model, the parameters are selected from $\mathcal{C} \in \{2^{-2}, \ldots, 2^{10}\}$, $\gamma \in \{10^{2}, \ldots, 10^{4}\}$, and $\lambda \in \{10^{-1}, \ldots, 10^{1}\}$.
- For the kernel-based models SVM-rbf and SVM-Q2, the penalty parameter is selected from $\mathcal{C} \in \{2^{-2}, \ldots, 2^{10}\}$, and the kernel parameter γ is set to the default value provided in the *scikit-learn* package.

- For ℓ_1-QSVM, the parameters are chosen from $\mathcal{C}_1 \in \{2^{-2}, \ldots, 2^{10}\}$ and $\mathcal{C}_2 \in \{10^{-1}, \ldots, 10^2\}$.

B Data source

The sources of the datasets utilized in Sections 4 are available in the GitHub repository.

https://github.com/tonygaobasketball/L0SQSSVM.