高维数据中，相似的样本应该在低维空间中仍然相似，也就是说：相邻样本在低维空间中应该尽量靠近。
为了保持这种“局部结构”，图正则化应运而生——它通过构建样本之间的图结构（Graph），然后在特征提取过程中对这个结构进行保留和强化。

图正则化如何引导子空间？近样本的低维表示要彼此相近，才能共同重构彼此的原始表示。公式(10)。所以在这个正则项的引导下，低维表示空间会趋向于让同类样本聚集、异类远离。

Learning the Optimal Discriminant SVM With Feature Extraction

Junhong Zhang, Zhihui Lai, *Member, IEEE*, Heng Kong, and Jian Yang

***Abstract*—Subspace learning and Support Vector Machine (SVM) are two critical techniques in pattern recognition, playing pivotal roles in feature extraction and classification. However, how to learn the optimal subspace such that the SVM classifier can perform the best is still a challenging problem due to the difficulty in optimization, computation, and algorithm convergence. To address these problems, this paper develops a novel method named Optimal Discriminant Support Vector Machine (ODSVM), which integrates support vector classification with discriminative subspace learning in a seamless framework. As a result, the most discriminative subspace and the corresponding optimal SVM are obtained simultaneously to pursue the best classification performance. The efficient optimization framework is designed for binary and multi-class ODSVM. Moreover, a fast sequential minimization optimization (SMO) algorithm with pruning is proposed to accelerate the computation in multi-class ODSVM. Unlike other related methods, ODSVM has a strong theoretical guarantee of global convergence, highlighting its superiority and stability. Numerical experiments are conducted on thirteen datasets and the results demonstrate that ODSVM outperforms existing methods with statistical significance.**

***Index Terms*—Support vector machine, subspace learning, joint learning framework.**

I. Introduction

In THE field of pattern recognition, the data are usually with high-dimensional features, which causes the well-known “curse of dimensionality” [1]. To address this problem, feature extraction approaches are used to transform the data into low-dimensional features, and subspace learning is one of the most essential techniques for feature extraction.

The classical subspace learning methods were widely applied in feature extraction and classification tasks in previous years. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are two of the most representative subspace learning methods [2], [3]. PCA is an unsupervised method aiming to seek the subspace that maximizes the global variance of the projected data, which leads to reconstruction error minimization and maintains the global structures of data. In contrast, LDA is a supervised method that finds the discriminative subspace such that the projected data has a maximal between-class scatter and minimal within-class scatter [3]. However, these two methods have some limitations that may affect the classification performance. The common drawback of PCA and LDA is that they do not consider the manifold structure of data [4], which degrades the performance in feature extraction and classification tasks. Therefore, manifold learning has been paid more attention to feature extraction. Laplacian Eigenmap (LE) [5] and Locality Preserving Projection (LPP) [6] are the most representative graph-based manifold learning methods. LE and LPP aim to preserve the local geometric structure described by an undirected weighted graph, which is usually pre-defined with either the neighbor information or supervised information. In this way, the local manifold information can be preserved in the low-dimensional subspaces. Thus, these methods are more flexible and can achieve better classification performance than PCA and LDA. Recently, many novel discriminative feature extraction models have been developed based on the local geometric structure preservation strategy. Lai et al. and Wen et al. designed a new locality preservation method based on graph-regularized reconstruction error minimization, which demonstrate better performance than LPP [7], [8]. The latest works mostly focused on adaptive graph learning for feature extraction. For instances, Joint Sparse Locality-Aware Regression makes the graph optimizable to improve the locality-aware ability of the model [9], and [10] further proposed the anchor-based strategy and maximum entropy regularization to perform fast discriminant analysis.

Except for feature extraction, the classifier is another important component in pattern recognition systems. As one of the most popular classifiers, Support Vector Machine (SVM) has been developed for several decades [11]. SVM aims to seek a pair of parallel hyperplanes to separate different classes with maximal margin, leading to an outstanding classification performance and generalization ability [12]. Training SVM involves a large quadratic programming problem (QPP), which is generally solved by Sequential Minimization Optimization (SMO). In the

Received 29 September 2022; revised 2 January 2025; accepted 8 January 2025. Date of publication 14 January 2025; date of current version 6 March 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62476175 and Grant 62272319, in part by the Natural Science Foundation of Guangdong Province under Grant 2023A1515010677, Grant 2024A1515011637, and Grant 2023B1212060076, and in part by Science and Technology Planning Project of Shenzhen Municipality under Grant JCYJ20220818095803007 and Grant JCYJ20240813142206009. Recommended for acceptance by W. Zuo. *(Corresponding author: Zhihui Lai.)*

Junhong Zhang and Zhihui Lai are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: apple_zjh@163.com; lai_zhi_hui@163.com).

Heng Kong is with the Department of Breast and Thyroid Surgery, BaoAn Central Hospital of Shenzhen, Shenzhen 518100, China (e-mail: generaldoc@126.com).

Jian Yang is with the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: csjyang@njust.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2025.3529711, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2025.3529711

last decades, many variants of SVM have been proposed for performance improvement. Least square SVM uses the least square loss instead of hinge loss and reduces the computation burden [13]. Xue et al. proposed structural regularized SVM which further exploits the within-class cluster structure of data via structural granularity. Recently, L_p-regularized SVM was studied to extend the classical SVM model [14], and [15] discussed the optimization of SVM with $L_{0,1}$ soft-margin loss with nonconvex optimization approaches. On the other hand, the nonparallel SVM classifiers were developed, which classify data with nonparallel hyperplanes. Twin SVM finds a pair of hyperplanes, each of which is close to the associated class and far away from the other class [16]. Inspired by Twin SVM, references [17], [18] further extend its objective function to improve discriminative performance. Yan et al. introduce $L_{2,p}$-norm metric into proximal SVM methods and enhance the robustness [19]. SVM and its improvements are widely applied in different pattern recognition applications [20], [21], [22], [23]. However, the methods mentioned above only considered the design of classifiers, but did not discuss how to improve the SVM classifiers' performance via feature extraction methods. In most applications utilizing SVM as the classifier, the feature extraction step and classification step are independent [24], which means they cannot take the feature extraction and classification into account as a whole to pursue the best performance. Thus, how to make the feature extraction consistent with classifiers (e.g., SVM) to improve the classification performance is still an unsolved problem.

To cope with this problem, one strategy is to find the subspace that maximizes the margin between the classes and then use SVM to classify the data in the corresponding subspace [25], [26], [27], [28]. However, in these methods, the learned subspaces are still independent of the optimization of SVM, which causes the suboptimality of both classifier and subspaces. Regarding this, the key is that the classifier and feature extractor should be optimized under the same criterion [29], leading to the subsequent joint learning approaches. Max-Margin Projection Pursuit (MMPP) [30] and Max-Margin Discriminant Feature Learning (MMLDF) [31] both tried to learn SVM and discriminative subspace via a unified optimization criterion. However, they have crucial drawbacks. As implied in [29], the optimal solution of MMPP is equivalent to SVM trained in the original high-dimensional subspace. Therefore, MMPP cannot further improve the classification performance of SVM. In addition, the convergence of the MMLDF algorithm cannot be theoretically guaranteed, and its classification performance is usually unstable. In short, *how to build a seamless framework that optimizes the discriminative subspace and the corresponding SVM jointly* is still a challenging problem. Furthermore, *how to design fast algorithms in such framework with a strict convergence guarantee* also needs to be explored.

To tackle these problems, we proposed Optimal Discriminant Support Vector Machine (ODSVM) in this paper, which integrates the discriminative subspace learning and support vector classification seamlessly. We then present efficient optimization algorithms with the strong theoretical guarantee of global convergence. Numerical experiments were conducted to evaluate the proposed method. The contributions of this paper are summarized as follows:

- We first present a graph-regularized reconstruction criterion for discriminative feature extraction, and devise a unified optimization framework for joint representation learning and SVM classification, i.e., ODSVM.
- The efficient algorithms for both binary-class and multi-class are presented. A novel block coordinate descent method and an SMO algorithm with pruning are developed to solve the induced large QPP. The global convergence to the local minimum is rigorously guaranteed in theory, i.e., the proposed ODSVM algorithms converge to the local minimum regardless of initialization. To the best of our knowledge, ODSVM for the first time integrates SVM with discriminative feature extraction that exhibits the theoretical guarantee of convergence.
- The experiments are conducted on 13 real-world datasets to evaluate the performance of the proposed ODSVM. The results demonstrate that ODSVM outperforms the related feature extraction methods. The efficiency of the proposed algorithms is also validated, which is consistent with the results of the theoretical analysis.

The remainder of this paper is organized as follows. In Section II, as related works, binary-class and multi-class SVM are briefly recapped. In Section III, we analyze the existing issues in the previous related methods and present the learning framework of ODSVM. The optimization framework of both binary-class and multi-class ODSVM are introduced, and the iterative algorithms are proposed to learn the discriminative subspace and SVM classifier. In addition, the global convergence of the algorithm is analyzed in this section. In Section V, the results of the numerical experiments on thirteen real-world datasets are reported. Finally, we conclude our work and provide some future directions in Section VI.

II. Preliminaries

In this section, we briefly review SVM in both binary-class and multi-class cases.

A. Binary-Class SVM

Given a binary-class dataset with the training sample matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{m\times n}$ and label vector $\mathbf{y} = [y_1, y_2, \ldots, y_n]^\top \in \{-1, 1\}^n$, where $\mathbf{x}_i \in \mathbb{R}^m$ denotes the i-th data point and y_i is its label. SVM aims to maximize the margin between the support hyperplanes between two classes, which leads to the following optimization problem:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 + c\sum_{i=1}^{n}[1 - y_i\mathbf{w}^\top\mathbf{x}_i]_+, \tag{1}$$

where $[z]_+ = \max(0, z)$ and $c > 0$ is penalty parameter. According to duality, this problem can be converted to the following quadratic programming problem (QPP):

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha} - \mathbf{e}^\top\boldsymbol{\alpha}, \text{s.t. } 0 \le \alpha_i \le c, \tag{2}$$

where $\mathbf{e}$ is the all-ones vector in the appropriate dimension and matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$ is defined as $K_{ij} = y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$. There exist many efficient solvers for problem (2), such as SMO [32] and Successive Over-Relaxation algorithm (SOR) [33].

B. Multi-Class SVM by Crammer and Singer

There exist many strategies extending binary-class SVM for multi-class classification, such as commonly used one-versus-one and one-versus-rest strategies [12]. Differently, Crammer and Singer proposed a direct multi-class extension of SVM, also known as Crammer-Singer SVM (CS-SVM) [34]. For a multi-class dataset with κ classes, i.e., $y_i \in \{1, 2, \ldots, \kappa\}$, CS-SVM aims to find κ hyperplanes and maximize the margin between different hyperplanes, leading to the following optimization problem.

$$\min_{\mathbf{W}} \frac{1}{2}\|\mathbf{W}\|_F^2 + c\sum_{i=1}^{n}[1 + \max_{j\neq y_i} \mathbf{w}_j^\top \mathbf{x}_i - \mathbf{w}_{y_i}^\top \mathbf{x}_i]_+, \quad (3)$$

where $\mathbf{W} \in \mathbb{R}^{m\times\kappa}$ is the weight matrix, and $\mathbf{w}_j$ represents the j-th column of $\mathbf{W}$. Problem (3) can be also converted to the following large-scale QPP:

$$\min_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\ldots,\boldsymbol{\alpha}_n} \sum_{j=1}^{\kappa}\|\mathbf{w}_j\|_2^2 + \sum_{i=1}^{n}\sum_{y_i\neq j}\alpha_{ij},$$

$$\text{s.t.} \sum_{j=1}^{\kappa}\alpha_{ij} = 0, \quad \alpha_{ij} \leq c_{y_i}^j, \quad i = 1, 2, \ldots, n, \quad (4)$$

where

$$\mathbf{w}_j = \sum_{i=1}^{n}\alpha_{ij}\mathbf{x}_i, \quad c_{y_i}^j = \begin{cases} 0, & y_i \neq j, \\ c, & y_i = j. \end{cases} \quad (5)$$

The previous research proposed many efficient algorithms for problem (4), such as fixed point algorithm [34] and sequential dual method [35]. Differing from binary-class SVM, the decision function of CS-SVM is

$$h(\mathbf{x}) = \arg\max_{j\in\{1,2,\ldots,\kappa\}} \mathbf{w}_j^\top \mathbf{x}. \quad (6)$$

In practice, LIBLINEAR [36] provides an implementation for both binary-class SVM and CS-SVM, which can be used for the efficient classification of large-scale datasets.

III. Optimal Discriminant SVM

In this section, we first introduce the motivation of the proposed ODSVM. Then the optimization framework of ODSVM is presented. The optimization algorithms are proposed for both binary-class and multi-class ODSVM. Particularly, we propose an efficient solver for the large-scale QPP in multi-class ODSVM.

A. Motivation of ODSVM

In this subsection, we explain the motivation of the proposed ODSVM step by step with the following three parts.

Previous works and limitations: Previous research tried to combine SVM and feature extraction into a unified model, but they encountered difficulties in model design and optimization. Typically, MMPP [30] designs the following joint optimization model for feature learning and classification:

$$\min_{\mathbf{w},\mathbf{P}} \frac{1}{2}\|\mathbf{w}\|_2^2 + c\sum_{i=1}^{n}[1 - y_i\mathbf{w}^\top\mathbf{P}^\top\mathbf{x}_i]_+, \ \text{s.t.}\ \mathbf{P}^\top\mathbf{P} = \mathbf{I}. \quad (7)$$

However, this model will degrade to (1) by substitution $\hat{\mathbf{w}} = \mathbf{P}\mathbf{w}$. Thus, the optimal solution of MMPP at most achieves the same performance as SVM trained in the original space. The reason is that MMPP has no further constraints on the projection matrix $\mathbf{P}$. In contrast, MMLDF involves additional term $\sum_{i,j}\|\mathbf{P}^\top\mathbf{x}_i - \mathbf{P}^\top\mathbf{x}_j\|^2 G_{ij}$ and regularizer $\|\mathbf{P}\|_{2,1}$ based on (7) but discards the orthogonal constraint [31]. However, the convergence of the MMLDF cannot be ensured in theory, which probably leads to suboptimal or trivial solutions. Therefore, it is still challenging to design an effective joint learning model integrating feature extraction and SVM with the algorithm's convergence guarantee, which has been an open problem for the last decade. This is the primary problem to be solved in this paper.

Novel discriminative feature learning with graph: We first introduce a more effective scheme for learning discriminative representation. Graph is a powerful tool to reveal the underlying relation of data, and thus is widely applied in discriminative feature extraction methods. The typical methods are Laplacian Eigenmap (LE), its linearized version Locality Preserving Projection (LPP) and the aforementioned MMLDF, which leverage the following optimization criterion:

$$\min_{\boldsymbol{\pi}} \sum_{i,j=1}^{n}\|\boldsymbol{\pi}_i - \boldsymbol{\pi}_j\|^2 G_{ij}, \quad \text{s.t.} \sum_{i=1}^{n}\boldsymbol{\pi}_i\boldsymbol{\pi}_i^\top = \mathbf{I}, \quad (8)$$

where $[\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots, \boldsymbol{\pi}_n] \in \mathbb{R}^{d\times n}$ denotes the extracted features of $\mathbf{X}$, and $\boldsymbol{\pi}_i = \mathbf{P}^\top\mathbf{x}_i$ for LPP and MMLDF with learnable projection matrix $\mathbf{P}$. Matrix $\mathbf{G} \in \mathbb{R}^{n\times n}$ is the adjacency matrix indicating the neighbor relations, and $G_{ij} > 0$ means the i-th sample and j-th sample are neighbors. Basically, these methods can preserve the similarity of sample pairs defined by the graph, in which the discriminative structure is discovered. Despite their success, these methods tend to underuse the information of the original feature $\mathbf{X}$. Extremely, LE embeddings only rely on the Laplacian matrix of $\mathbf{G}$. The inherent reason is that these methods ignore the reconstruction property of the embeddings, which sometimes causes overfitting and loss of essential information. Consequently, the performance of these methods is limited. In contrast, PCA efficiently maintains the information from the original data by reconstruction error minimization. Therefore, it is vital to exploit the reconstruction information and graph similarity preservation simultaneously to address the above issues.

Let $\mathbf{Q} \in \mathbb{R}^{m\times d}$ be an orthogonal dictionary with constraint $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$. If $\mathbf{x}_i$ and $\mathbf{x}_j$ are neighbors, we can write

$$\begin{aligned}\|\boldsymbol{\pi}_i - \boldsymbol{\pi}_j\| &= \|\mathbf{Q}\boldsymbol{\pi}_i - \mathbf{Q}\boldsymbol{\pi}_j\| = \|\mathbf{Q}\boldsymbol{\pi}_i - \mathbf{x}_i + \mathbf{x}_i - \mathbf{Q}\boldsymbol{\pi}_j\| \\ &\leq \underbrace{\|\mathbf{x}_i - \mathbf{Q}\boldsymbol{\pi}_i\|}_{\text{self-reconstruction}} + \underbrace{\|\mathbf{x}_i - \mathbf{Q}\boldsymbol{\pi}_j\|}_{\text{neighbor-encoding}}, \quad (9)\end{aligned}$$

where the first equality follows from the orthogonality of $\mathbf{Q}$. At first glance, minimizing the right-hand side of inequality (9) indirectly suppresses the distance $\|\boldsymbol{\pi}_i - \boldsymbol{\pi}_j\|$, which enforces the alignment between two neighbors $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$. The first term facilitates self-reconstruction to extract the characteristics from the data, and the second term encodes the data with its neighbors' embeddings, which ensures the compactness of the representations. Motivated by this observation, we propose the following feature extraction framework:

$$\min_{\mathbf{Q},\{\boldsymbol{\pi}_i\}_{i=1}^n} \sum_{i,j=1}^{n} \|\mathbf{x}_i - \mathbf{Q}\boldsymbol{\pi}_j\|_p^q G_{ij}, \quad \text{s.t. } \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}, \tag{10}$$

where $p \geq 1$ and $q > 0$ can be selected flexibly to achieve different distance metrics. (10) naturally incorporates self-reconstruction and neighbor-encoding for graph similarity preservation. Compared with (8), it emphasizes the original data and thus tends to obtain a more informative representation. Interestingly, some existing methods can be formulated within the framework (10). Here we list some typical instances.

- PCA [2] and SPCA [37]: $p = q = 2$ and $G_{ij} = 1$ if $i = j$ and $G_{ij} = 0$ otherwise. Thus, they only focus on self-reconstruction and preserve the primary structure of data. The PCA projection is $\boldsymbol{\pi}_i = \mathbf{Q}^\top\mathbf{x}_i$, while SPCA uses $\boldsymbol{\pi}_i = \mathbf{P}^\top\mathbf{x}_i$ with sparse regularized projection matrix $\mathbf{P}$.
- LRPP_GRR [8]: $p = q = 2$ and $\boldsymbol{\pi}_j = \mathbf{P}^\top\mathbf{X}\boldsymbol{\beta}_j$, where $\boldsymbol{\beta}_j \in \mathbb{R}^n$ is the coefficient for low-rank representation. $\mathbf{G}$ is defined as the k-nearest neighbor (KNN) graph.
- RDR [7]: $\boldsymbol{\pi}_i = \mathbf{P}^\top\mathbf{x}_i$ and $\mathbf{G}$ is KNN graph. Different from the above methods, RDR employs $p = 2$ and $q = 1$ for the extra consideration of the robustness.

Therefore, (10) is more general and effective than the feature extraction schemes in MMPP and MMLDF. It is expected to integrate (10) into the joint learning framework of feature extraction and classification to improve its performance.

Optimal Discriminant SVM: Commonly, feature learning and classification are achieved via optimizing distinct objectives. Consequently, the learned features may be unsuitable for the classifier. The joint learning paradigm is required for the consistency of low-dimensional representation and the downstream classifier. Thus, we integrate the SVM classifier and (10) into a unified optimization framework as follows:

$$\begin{aligned}\min_{\mathbf{f},\mathbf{P},\mathbf{Q}} J(\mathbf{f},\mathbf{P},\mathbf{Q}) &= \frac{1}{2}\|\mathbf{f}\|^2 + c\sum_{i=1}^{n}\ell_H(\mathbf{f},\mathbf{P}^\top\mathbf{x}_i, y_i)\\ &\quad + \frac{\lambda}{2}\sum_{i,j=1}^{n}\|\mathbf{x}_i - \mathbf{Q}\mathbf{P}^\top\mathbf{x}_j\|_2^2 G_{ij} + \frac{\gamma}{2}\|\mathbf{P}\|_F^2,\\ \text{s.t. } &\mathbf{Q}^\top\mathbf{Q} = \mathbf{I},\end{aligned} \tag{11}$$

where we utilize a common setting $p = q = 2$ and $\boldsymbol{\pi}_i = \mathbf{P}^\top\mathbf{x}_i$. $\lambda, \gamma > 0$ are hyper-parameters, and $\ell_H(\cdot)$ could be either binary-class or multi-class hinge loss. Besides, the supervised graph is utilized to preserve the within-class similarity:

$$G_{ij} = \begin{cases}\exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{t}\right), & y_i = y_j,\\ 0, & \text{otherwise,}\end{cases} \quad (t > 0). \tag{12}$$

We illustrate the proposed ODSVM in Fig. 1. Intuitively, optimizing the second term of (11) tends to induce embeddings with a larger between-class margin and in turn improves the performance of SVM. Meanwhile, the minimization of the third term reduces the within-class discrepancy. By Fisher's criterion, the data separability is strengthened via the seamless collaboration of SVM and feature extraction. Thus, the most discriminative subspace is learned such that the corresponding SVM classifier performs the best in this subspace, and the classification performance is significantly improved.

B. *Optimization of Binary-Class ODSVM*

We begin with the basic ODSVM model for binary classification and discuss the optimization procedure. In this case, the optimization framework (11) becomes the following problem for binary-class ODSVM:

$$\begin{aligned}\min_{\mathbf{w},\mathbf{P},\mathbf{Q}} &\frac{1}{2}\|\mathbf{w}\|_2^2 + c\sum_{i=1}^{n}[1 - y_i\mathbf{w}^\top\mathbf{P}^\top\mathbf{x}_i]_+\\ &+ \frac{\lambda}{2}\sum_{i,j=1}^{n}\|\mathbf{x}_i - \mathbf{Q}\mathbf{P}^\top\mathbf{x}_j\|_2^2 G_{ij} + \frac{\gamma}{2}\|\mathbf{P}\|_F^2,\\ \text{s.t. } &\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}.\end{aligned} \tag{13}$$

We adopt an alternatively iterative approach to solve (13), which is described in the following optimization steps.

$\mathbf{w}$ *step:* We first fix $\mathbf{P}$ and $\mathbf{Q}$ to compute $\mathbf{w}$, and the optimization problem (13) becomes the following problem.

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 + c\sum_{i=1}^{n}[1 - y_i\mathbf{w}^\top\mathbf{P}^\top\mathbf{x}_i]_+, \tag{14}$$

which is a binary-class SVM with training data $\mathbf{P}^\top\mathbf{X}$. Therefore, this problem can be efficiently solved using LIBLINEAR.

$\mathbf{P}$ *step:* $\mathbf{P}$ is optimized with fixing $\mathbf{w}$ and $\mathbf{Q}$. By introducing the slack variables $\boldsymbol{\xi}$, we can reformulate (13) as follows:

$$\begin{aligned}\min_{\boldsymbol{\xi},\mathbf{P}} \ &c\mathbf{e}^\top\boldsymbol{\xi} + \frac{\lambda}{2}\sum_{i,j=1}^{n}\|\mathbf{x}_i - \mathbf{Q}\mathbf{P}^\top\mathbf{x}_j\|_2^2 G_{ij} + \frac{\gamma}{2}\|\mathbf{P}\|_F^2\\ \text{s.t. } &y_i\mathbf{w}^\top\mathbf{P}^\top\mathbf{x}_i \geq 1 - \xi_i, \ \ \xi_i \geq 0.\end{aligned} \tag{15}$$

The second term in (15) can be reformulated as

$$\begin{aligned}&\sum_{i,j=1}^{n}\|\mathbf{x}_i - \mathbf{Q}\mathbf{P}^\top\mathbf{x}_j\|_2^2 G_{ij}\\ &= \sum_{i,j=1}^{n}\left(\mathbf{x}_i^\top\mathbf{x}_i - 2\mathbf{x}_i^\top\mathbf{Q}\mathbf{P}^\top\mathbf{x}_j + \mathbf{x}_j^\top\mathbf{P}\mathbf{Q}^\top\mathbf{Q}\mathbf{P}^\top\mathbf{x}_j\right)G_{ij}\\ &= \sum_{i=1}^{n}\left(\mathbf{x}_i^\top D_{ii}\mathbf{x}_i + \mathbf{x}_i^\top\mathbf{P}D_{ii}\mathbf{P}^\top\mathbf{x}_i\right) - 2\sum_{i,j=1}^{n}\mathbf{x}_i^\top\mathbf{Q}G_{ij}\mathbf{P}^\top\mathbf{x}_j\\ &= \mathrm{Tr}(\mathbf{X}\mathbf{D}\mathbf{X}^\top) + \mathrm{Tr}(\mathbf{P}^\top\mathbf{X}\mathbf{D}\mathbf{X}^\top\mathbf{P}) - 2\mathrm{Tr}(\mathbf{P}^\top\mathbf{X}\mathbf{G}\mathbf{X}^\top\mathbf{Q})\\ &= \mathrm{Tr}(\mathbf{P}^\top\mathbf{S}_{\mathrm{A}}\mathbf{P}) - 2\mathrm{Tr}(\mathbf{P}^\top\mathbf{S}_{\mathrm{B}}\mathbf{Q}) + \text{const},\end{aligned}$$

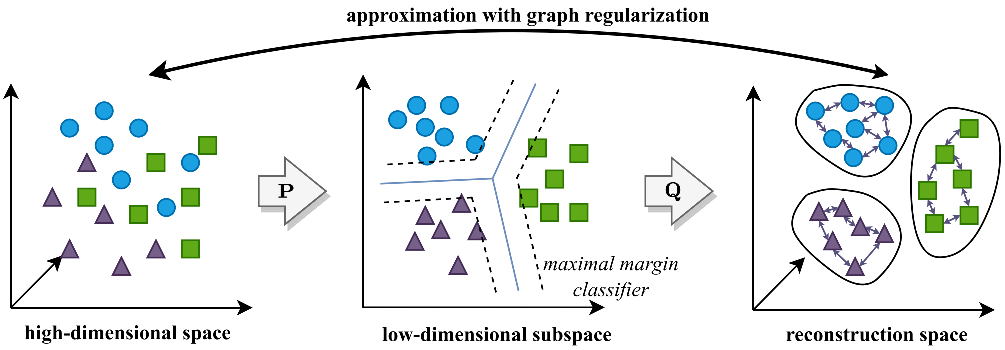


Fig. 1. The overview of the proposed ODSVM method. The projection matrix (as well as reconstruction matrix) and SVM classifier are learned jointly to induce the subspace such that SVM performs the best. That is, the embedding with maximal between-class margins is obtained, and the within-class distance is minimized via graph-regularized reconstruction for locality preservation, further enhancing the discriminability of data. Therefore, the derived subspace is the most suitable for SVM classification, and the performance of the SVM classifier is greatly improved.

where $\mathrm{Tr}(\cdot)$ is the trace operator, $\mathbf{D}$ is a diagonal matrix such that $D_{ii} = \sum_{j=1}^{n} G_{ij}$, and $\mathbf{S}_A = \mathbf{X}\mathbf{D}\mathbf{X}^\top$, $\mathbf{S}_B = \mathbf{X}\mathbf{G}\mathbf{X}^\top$. Therefore, problem (15) can be rewritten as follows:

$$\min_{\mathbf{P},\boldsymbol{\xi}} \frac{\lambda}{2}\mathrm{Tr}(\mathbf{P}^\top\mathbf{S}_{\mathrm{A}}\mathbf{P} - 2\mathbf{P}^\top\mathbf{S}_{\mathrm{B}}\mathbf{Q}) + \frac{\gamma}{2}\|\mathbf{P}\|_2^2 + \mathrm{c}\mathbf{e}^\top\boldsymbol{\xi}$$
$$\text{s.t. } \mathbf{w}^\top\mathbf{P}^\top\mathbf{z}_i \geq 1 - \xi_i,\ \ \xi_i \geq 0, \tag{16}$$

where $\mathbf{z}_i = y_i\mathbf{x}_i$. Since problem (16) is a convex problem with inequality constraints, we solve it with the duality method. The corresponding Lagrangian function of (16) is

$$L(\mathbf{P},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{\lambda}{2}\mathrm{Tr}(\mathbf{P}^\top\mathbf{S}_{\mathrm{A}}\mathbf{P} - 2\mathbf{P}^\top\mathbf{S}_{\mathrm{B}}\mathbf{Q}) + \frac{\gamma}{2}\|\mathbf{P}\|_{\mathrm{F}}^2$$
$$+ c\mathbf{e}^\top\boldsymbol{\xi} + \boldsymbol{\alpha}^\top(\mathbf{e} - \boldsymbol{\xi} - \mathbf{Z}^\top\mathbf{P}\mathbf{w}) - \boldsymbol{\beta}^\top\boldsymbol{\xi}, \tag{17}$$

where $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n]$, and $\boldsymbol{\alpha}, \boldsymbol{\beta}$ are Lagrange multipliers. The Karush-Kuhn-Tucker (KKT) conditions give

$$\nabla_{\mathbf{P}}L = \lambda(\mathbf{S}_A\mathbf{P} - \mathbf{S}_B\mathbf{Q}) + \gamma\mathbf{P} - \mathbf{Z}\boldsymbol{\alpha}\mathbf{w}^\top = \mathbf{0}, \tag{18}$$
$$\nabla_{\boldsymbol{\xi}}L = c\mathbf{e} - \boldsymbol{\alpha} - \boldsymbol{\beta} = 0. \tag{19}$$

Therefore, from (18) we can express the optimal $\mathbf{P}$ as

$$\mathbf{P} = (\lambda\mathbf{S}_A + \gamma\mathbf{I})^{-1}(\lambda\mathbf{S}_B\mathbf{Q} + \mathbf{Z}\boldsymbol{\alpha}\mathbf{w}^\top). \tag{20}$$

For convenience, we set $\mathbf{M} = (\lambda\mathbf{S}_A + \gamma\mathbf{I})^{-1}$. Although (20) gives a closed-form solution to compute $\mathbf{P}$, the Lagrange multiplier $\boldsymbol{\alpha}$ is still unknown. Therefore, we need to compute the optimal $\boldsymbol{\alpha}$ for $\mathbf{P}$ in (20) by solving the dual problem of (16). From (17), (19), (20), we can derive

$$\min_{\mathbf{P},\boldsymbol{\xi}} L(\mathbf{P},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})$$
$$= \mathbf{e}^\top\boldsymbol{\alpha} - \frac{1}{2}\mathrm{Tr}\left[\mathbf{P}^\top(\lambda\mathbf{S}_{\mathrm{B}}\mathbf{Q} + \mathbf{Z}\boldsymbol{\alpha}\mathbf{w}^\top)\right]$$
$$= \mathbf{e}^\top\boldsymbol{\alpha} - \frac{1}{2}\mathrm{Tr}\left[(\lambda\mathbf{S}_{\mathrm{B}}\mathbf{Q} + \mathbf{Z}\boldsymbol{\alpha}\mathbf{w}^\top)^\top\mathbf{M}(\lambda\mathbf{S}_{\mathrm{B}}\mathbf{Q} + \mathbf{Z}\boldsymbol{\alpha}\mathbf{w}^\top)\right]$$
$$= \mathbf{e}^\top\boldsymbol{\alpha} - \left(\lambda\boldsymbol{\alpha}^\top\mathbf{Z}^\top\mathbf{M}\mathbf{S}_B\mathbf{Q}\mathbf{w} + \frac{1}{2}\boldsymbol{\alpha}^\top\mathbf{Z}^\top\mathbf{M}\mathbf{Z}\boldsymbol{\alpha}\mathbf{w}^\top\mathbf{w}\right)$$
$$- \frac{\lambda^2}{2}\mathrm{Tr}(\mathbf{Q}^\top\mathbf{S}_{\mathrm{B}}^2\mathbf{Q})$$
$$= -\frac{\|\mathbf{w}\|_2^2}{2}\boldsymbol{\alpha}^\top\mathbf{Z}^\top\mathbf{M}\mathbf{Z}\boldsymbol{\alpha} + (\mathbf{e} - \lambda\mathbf{Z}^\top\mathbf{M}\mathbf{S}_B\mathbf{Q}\mathbf{w})^\top\boldsymbol{\alpha} + \text{const.}$$

Algorithm 1: Training Binary-Class ODSVM.

```
Input: Data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, labels $\mathbf{y} \in \{-1,1\}^n$, the hyperparameters $\{c, \lambda, \gamma, d\}$, and max iterations MAX_ITER.
Output: Projection matrix $\mathbf{P} \in \mathbb{R}^{m\times d}$, $\mathbf{Q} \in \mathbb{R}^{m\times d}$, weight vector $\mathbf{w} \in \mathbb{R}^d$.
$\mathbf{P} \leftarrow \mathtt{PCA}(\mathbf{X}, d)$, $l \leftarrow 0$.
while $l <$ MAX_ITER do
    $\mathbf{w} \leftarrow \mathtt{SVM}(\mathbf{y}, \mathbf{P}^\top\mathbf{X})$.
    Solve QPP (21) and obtain $\boldsymbol{\alpha}$.
    Update $\mathbf{P}$ via (20).
    $[\mathbf{U}, \boldsymbol{\Sigma}, \mathbf{V}] \leftarrow \mathtt{svd}(\mathbf{S}_B^\top\mathbf{P}, ''\mathtt{econ}'')$.
    $\mathbf{Q} \leftarrow \mathbf{U}\mathbf{V}^\top$.
    if the objective function (13) is unchanged then
        break.
    $l \leftarrow l + 1$.
    end if
end while
```

According to duality [38], the optimal $\boldsymbol{\alpha}$ is given by the following dual problem of (16).

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta}}\min_{\mathbf{P},\boldsymbol{\xi}} L(\mathbf{P},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}), \quad \text{s.t. } \alpha_i, \beta_i \geq 0$$
$$\Leftrightarrow \max_{\boldsymbol{\alpha}} -\frac{\|\mathbf{w}\|_2^2}{2}\boldsymbol{\alpha}^\top\mathbf{Z}^\top\mathbf{M}\mathbf{Z}\boldsymbol{\alpha} + (\mathbf{e} - \lambda\mathbf{Z}^\top\mathbf{M}\mathbf{S}_B\mathbf{Q}\mathbf{w})^\top\boldsymbol{\alpha},$$
$$\text{s.t. } 0 \leq \alpha_i \leq c. \tag{21}$$

This is a typical QPP that can be solved by successive over-relaxation (SOR) algorithm [33]. After the optimal $\boldsymbol{\alpha}$ is obtained, we can compute $\mathbf{P}$ with (20).

$\mathbf{Q}$ *step:* When fixing $\mathbf{P}$ and $\mathbf{w}$, we can reformulate the original problem as follows:

$$\max_{\mathbf{Q}} \mathrm{Tr}(\mathbf{P}^\top \mathbf{S}_{\mathrm{B}}\mathbf{Q}), \quad \text{s.t. } \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}, \tag{22}$$

which is an orthogonal Procrustes problem [37]. Thus, the optimal $\mathbf{Q}$ is given by singular value decomposition (SVD). Let the economy-sized SVD of $\mathbf{S}_B^\top\mathbf{P}$ be

$$\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top = \mathbf{S}_B^\top\mathbf{P}. \tag{23}$$

Then we can obtain the solution of (22) by

$$\mathbf{Q} = \mathbf{U}\mathbf{V}^\top. \tag{24}$$

For details of the algorithm for binary-class ODSVM, please see Algorithm 1.

C. Optimization of Multi-Class ODSVM

Applying the loss of CSSVM [34], it is easy to obtain the optimization model of multi-class ODSVM from the proposed unified framework (11):

$$\begin{aligned}\min_{\mathbf{W},\mathbf{P},\mathbf{Q}} &\frac{1}{2}\|\mathbf{W}\|_F^2 + c\sum_{i=1}^{n}[1 - \mathbf{w}_{y_i}^\top\mathbf{P}^\top\mathbf{x}_i + \max_{j\neq y_i}\mathbf{w}_j^\top\mathbf{P}^\top\mathbf{x}_i]_+ \\ &+\frac{\lambda}{2}\sum_{i,j=1}^{n}\|\mathbf{x}_i - \mathbf{Q}\mathbf{P}^\top\mathbf{x}_j\|_2^2 G_{ij} + \frac{\gamma}{2}\|\mathbf{P}\|_F^2, \\ &\text{s.t. } \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}.\end{aligned} \tag{25}$$

We adopt the same optimization strategy with binary-class ODSVM to solve (25). Note that we can optimize $\mathbf{W}$ and $\mathbf{Q}$ similarly with binary-class ODSVM. In each step, the optimal $\mathbf{W}$ is obtained by CS-SVM trained with $\mathbf{P}^\top\mathbf{X}$, and the optimal $\mathbf{Q}$ is still obtained through solving the Procrustes problem (22). Therefore, we primarily discuss the optimization of $\mathbf{P}$ in this section. When fixing $\mathbf{W}$ and $\mathbf{Q}$, problem (25) can be reformulated via introducing slack variables as follows:

$$\begin{aligned}\min_{\mathbf{P},\boldsymbol{\xi}} &\frac{\lambda}{2}\mathrm{Tr}(\mathbf{P}^\top\mathbf{S}_{\mathrm{A}}\mathbf{P} - 2\mathbf{P}^\top\mathbf{S}_{\mathrm{B}}\mathbf{Q}) + \frac{\gamma}{2}\|\mathbf{P}\|_{\mathrm{F}}^2 + c\mathbf{e}^\top\boldsymbol{\xi}, \\ &\text{s.t. } \mathbf{w}_{y_i}^\top\mathbf{P}^\top\mathbf{x}_i + \delta_{ij} - \mathbf{w}_j^\top\mathbf{P}^\top\mathbf{x}_i \geq 1 - \xi_i,\end{aligned} \tag{26}$$

where δ_{ij} is defined as

$$\delta_{ij} = \begin{cases} 1, & j = y_i, \\ 0, & \text{otherwise.}\end{cases} \tag{27}$$

Therefore, we use $\boldsymbol{\Delta} = \{\delta_{ij}\} \in \mathbb{R}^{n\times\kappa}$ to represent the one-hot label matrix. The Lagrangian function of is

$$\begin{aligned}L(\mathbf{P},\boldsymbol{\xi},\mathbf{A}) = &\frac{\lambda}{2}\mathrm{Tr}(\mathbf{P}^\top\mathbf{S}_{\mathrm{A}}\mathbf{P} - 2\mathbf{P}^\top\mathbf{S}_{\mathrm{B}}\mathbf{Q}) + \frac{\gamma}{2}\|\mathbf{P}\|_{\mathrm{F}}^2 \\ &+ c\mathbf{e}^\top\boldsymbol{\xi} + \sum_{i=1}^{n}\sum_{j=1}^{\kappa}A_{ij}(1 - \xi_i - \mathbf{w}_{y_i}^\top\mathbf{P}^\top\mathbf{x}_i \\ &+ \mathbf{w}_j^\top\mathbf{P}^\top\mathbf{x}_i - \delta_{ij}),\end{aligned} \tag{28}$$

where $A_{ij} \geq 0$ indicates the Lagrange multiplier. According to KKT conditions, we have

$$\nabla_{\xi_i}L = \sum_{j=1}^{\kappa}A_{ij} - c = 0 \Rightarrow \sum_{j=1}^{\kappa}A_{ij} = c, \tag{29}$$

$$\begin{aligned}\nabla_{\mathbf{P}}L = &\lambda(\mathbf{S}_A\mathbf{P} - \mathbf{S}_B\mathbf{Q}) + \gamma\mathbf{P} \\ &- \sum_{i=1}^{n}\mathbf{x}_i\mathbf{w}_{y_i}^\top\sum_{j=1}^{\kappa}A_{ij} + \sum_{i=1}^{n}\sum_{j=1}^{\kappa}\mathbf{x}_iA_{ij}\mathbf{w}_j^\top = \mathbf{0}.\end{aligned} \tag{30}$$

Notice that $\mathbf{w}_{y_i} = \mathbf{W}\boldsymbol{\delta}^{(i)}$, where $\boldsymbol{\delta}^{(i)} \in \mathbb{R}^\kappa$ denotes the i-th row of $\boldsymbol{\Delta}$. Since $\sum_{i,j}\mathbf{x}_iA_{ij}\mathbf{w}_j^\top = \mathbf{X}\mathbf{A}\mathbf{W}^\top$, we can derive

$$\mathbf{P} = (\lambda\mathbf{S}_A + \gamma\mathbf{I})^{-1}(\lambda\mathbf{S}_B\mathbf{Q} + \mathbf{X}(c\boldsymbol{\Delta} - \mathbf{A})\mathbf{W}^\top). \tag{31}$$

Similar to the binary-class case, we should compute the optimal $\mathbf{A}$ by solving the dual problem. Let $\hat{\mathbf{A}} = c\boldsymbol{\Delta} - \mathbf{A}$, and we can obtain the dual problem of

$$\begin{aligned}\max_{\hat{\mathbf{A}}} &-\frac{1}{2}\mathrm{Tr}(\hat{\mathbf{A}}^\top\mathbf{X}^\top\mathbf{M}\mathbf{X}\hat{\mathbf{A}}\mathbf{W}^\top\mathbf{W}) \\ &- \lambda\mathrm{Tr}(\hat{\mathbf{A}}^\top\mathbf{X}^\top\mathbf{M}\mathbf{S}_{\mathrm{B}}\mathbf{Q}\mathbf{W}) + \mathrm{Tr}(\hat{\mathbf{A}}^\top\boldsymbol{\Delta}), \\ &\text{s.t. } c\delta_{ij} - \hat{A}_{ij} \geq 0, \quad \sum_{j=1}^{\kappa}\hat{A}_{ij} = 0.\end{aligned} \tag{32}$$

For simplification, let $\mathbf{H} = \lambda\mathbf{X}^\top\mathbf{M}\mathbf{S}_B\mathbf{Q}\mathbf{W} - \boldsymbol{\Delta}$, $\boldsymbol{\Phi} = \mathbf{W}^\top\mathbf{W}$ and $\mathbf{K} = \mathbf{X}^\top\mathbf{M}\mathbf{X}$, the above problem can be rewritten as follows:

$$\begin{aligned}\min_{\hat{\mathbf{A}}} &\frac{1}{2}\mathrm{Tr}(\hat{\mathbf{A}}^\top\mathbf{K}\hat{\mathbf{A}}\boldsymbol{\Phi}) + \mathrm{Tr}(\hat{\mathbf{A}}^\top\mathbf{H}), \\ &\text{s.t. } c\boldsymbol{\Delta} \geq \hat{\mathbf{A}}, \quad \hat{\mathbf{A}}\mathbf{e} = \mathbf{0}.\end{aligned} \tag{33}$$

Problem (33) is equivalent to a very large-scale QPP with $n\kappa \times n\kappa$ matrix. It is challenging to store such a large matrix, and it is also time-consuming to directly solve this QPP. To reduce the memory and computation burden, we adopt a block coordinate descent strategy to solve problem (33). That is, the i-th row of $\mathbf{A}$ is optimized and other rows are fixed in each step. The main procedure is shown in Algorithm 2. Therefore, we will obtain the following reduced sub-problem.

$$\begin{aligned}\min_{\boldsymbol{\alpha}^{(i)}} &\frac{1}{2}K_{ii}(\boldsymbol{\alpha}^{(i)})^\top\boldsymbol{\Phi}\boldsymbol{\alpha}^{(i)} + \boldsymbol{\tau}_i^\top\boldsymbol{\alpha}^{(i)} \\ &\text{s.t. } \mathbf{e}^\top\boldsymbol{\alpha}^{(i)} = 0, \quad \boldsymbol{\alpha}^{(i)} \leq c\boldsymbol{\delta}^{(i)},\end{aligned} \tag{35}$$

where $\boldsymbol{\alpha}^{(i)} \in \mathbb{R}^\kappa$ is the i-th row of $\hat{\mathbf{A}}$, and $\boldsymbol{\tau}_i$ is defined as

$$\boldsymbol{\tau}_i = \mathbf{h}^{(i)} + \sum_{j=1, j\neq i}^{n}K_{ij}\boldsymbol{\Phi}\boldsymbol{\alpha}^{(j)}, \tag{36}$$

where $\mathbf{h}^{(i)}$ denotes the i-th row of $\mathbf{H}$.

For simplification of notations, we rewrite (35) as follows in the below analysis.

$$\begin{aligned}\min_{\boldsymbol{\alpha}} &\frac{1}{2}K_{ii}\boldsymbol{\alpha}^\top\boldsymbol{\Phi}\boldsymbol{\alpha} + \boldsymbol{\tau}^\top\boldsymbol{\alpha} \\ &\text{s.t. } \mathbf{e}^\top\boldsymbol{\alpha} = 0, \quad \boldsymbol{\alpha} \leq c\boldsymbol{\delta}^{(i)}.\end{aligned} \tag{37}$$

Algorithm 2: Block Coordinate Descent Procedure Solving (33).

```
Input: Matrices $\mathbf{K} \in \mathbb{R}^{n\times n}, \mathbf{H} \in \mathbb{R}^{n\times\kappa}, \mathbf{\Delta} \in \{0,1\}^{n\times\kappa}$,
and iteration limits $T \in \mathbb{Z}_+$.
Output: Matrix $\hat{\mathbf{A}} \in \mathbb{R}^{n\times\kappa}$.
while $t < T$ do
    $[r(1), r(2), \dots, r(n)] = \mathtt{randperm}([1, 2, \dots, n])$.
    for $i = [1, 2, \dots, n]$ do
        Compute $\boldsymbol{\tau}_{r(i)}$ by (36).
        Update $\boldsymbol{\alpha}_{r(i)}$ via Algorithm 3.
    end for
    $t \leftarrow t + 1$
end while
```

Algorithm 3: SMO Algorithm Solving (37).

```
Input: Matrix $\mathbf{\Phi} \in \mathbb{R}^{\kappa\times\kappa}$, vector $\boldsymbol{\tau}, \boldsymbol{\delta}^{(i)} \in \mathbb{R}^{\kappa}$, scalar
$c > 0, K_{ii} \geq 0$, and the tolerance $\epsilon$.
Output: Vector $\boldsymbol{\alpha} \in \mathbb{R}^{\kappa}$.
Initialize with $\boldsymbol{\alpha} \leftarrow \mathbf{0}$.
if $K_{ii} = 0$ or (49) is satisfied then
    return$\boldsymbol{\alpha}$.
end if
while $\rho_{\max}(\boldsymbol{\alpha}) \geq \rho_{\min}(\boldsymbol{\alpha}) + \epsilon$ do
    Select two index $r, s \in \{1, 2, \dots, \kappa\}$.
    $q \leftarrow K_{ii}(\phi_{rr} + \phi_{ss} - 2\phi_{rs})$.
    $p \leftarrow K_{ii}(\mathbf{\Phi}_r - \mathbf{\Phi}_s)^\top \boldsymbol{\alpha} + \tau_r - \tau_s$.
    Solve the one-dimensional QPP:
        $\min_{t\in\mathbb{R}} \frac{1}{2}qt^2 + pt, \quad \text{s.t. } \alpha_s - c\delta_{is} \leq t \leq c\delta_{ir} - \alpha_r.$   (34)
    Update $\alpha_r \leftarrow \alpha_r + t$, and $\alpha_s \leftarrow \alpha_s - t$.
end while
```

Algorithm 4: Training Multi-Class ODSVM.

```
Input: Sample matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, one-hot encoded label
matrix $\mathbf{\Delta} \in \{0,1\}^{n\times\kappa}$, the hyperparameters $\{c, \lambda, \gamma, d\}$,
and max iterations MAX_ITER.
Output: Projection matrix $\mathbf{P} \in \mathbb{R}^{m\times d}$, $\mathbf{Q} \in \mathbb{R}^{m\times d}$, weight
matrix $\mathbf{W} \in \mathbb{R}^{d\times k}$.
$\mathbf{P} \leftarrow \mathtt{PCA}(\mathbf{X}, d)$, $l \leftarrow 0$.
while $l <$ MAX_ITER do
    $\mathbf{W} \leftarrow \mathtt{CSSVM}(\mathbf{\Delta}, \mathbf{P}^\top\mathbf{X})$.
    Solve QPP (33) via Algorithm 2 to update $\hat{\mathbf{A}}$.
    Update $\mathbf{P}$ via (31).
    $[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] \leftarrow \mathtt{svd}(\mathbf{S}_B^\top\mathbf{P}, \text{"econ"})$.
    $\mathbf{Q} \leftarrow \mathbf{U}\mathbf{V}^\top$.
    if the objective function (25) is unchanged then
        break.
    end if
    $l \leftarrow l + 1$.
end while
```

In this paper, we solve this problem via an SMO algorithm, and the detailed procedure is described in Algorithm 3. The algorithm first initializes a feasible $\boldsymbol{\alpha}$ with a zero vector. In each iteration, only two elements of $\boldsymbol{\alpha}$ are optimized and others are fixed. Suppose that α_r and α_s are selected and updated to $\alpha_r + t$ and $\alpha_s - t$, respectively, which makes the equality constraint $\mathbf{e}^\top\boldsymbol{\alpha} = 0$ always holds. With the other elements of $\boldsymbol{\alpha}$ fixed, problem (37) can be transformed into a one-dimensional QPP with respect to variable t with only box constraint, which can be solved efficiently in $O(1)$ time.

In Algorithm 3, the objective function monotonically decreases in each iteration, which implies that the proposed SMO algorithm will finally converge to the global optimum of QPP (37). However, we still need a termination criterion to check whether the optimum is achieved or not, and a strategy for variable selection in each iteration. To this end, we further analyze the optimality of $\boldsymbol{\alpha}$ via the following theorem, and reveal the stopping condition of Algorithm 3.

Theorem 1: Define $\rho_{\max}(\boldsymbol{\alpha})$ and $\rho_{\min}(\boldsymbol{\alpha})$ as

$$\rho_{\max}(\boldsymbol{\alpha}) = \max_j K_{ii}\mathbf{\Phi}_j^\top\boldsymbol{\alpha} + \tau_j, \tag{38}$$

$$\rho_{\min}(\boldsymbol{\alpha}) = \min_{j:\alpha_j < c\delta_{ij}} K_{ii}\Phi_j^\top\boldsymbol{\alpha} + \tau_j. \tag{39}$$

For a feasible solution $\tilde{\boldsymbol{\alpha}}$ such that $\tilde{\alpha}_j \leq c\delta_{ij}$ and $\mathbf{e}^\top\tilde{\boldsymbol{\alpha}} = 0$, it is the optimal solution if and only if $\rho_{\max}(\tilde{\boldsymbol{\alpha}}) = \rho_{\min}(\tilde{\boldsymbol{\alpha}})$.

Proof: The KKT conditions are the necessary and sufficient conditions for the optimal solution to problem (37), which can be concluded as existing $\boldsymbol{\mu} \in \mathbb{R}^{\kappa}$ and $\rho \in \mathbb{R}$ such that,

$$K_{ii}\mathbf{\Phi}\boldsymbol{\alpha} + \boldsymbol{\tau} + \boldsymbol{\mu} - \rho\mathbf{e} = 0, \tag{40}$$

$$\mu_j \geq 0, \tag{41}$$

$$\mu_j(\alpha_j - c\delta_{ij}) = 0, \tag{42}$$

$$\alpha_j \leq c\delta_{ij}, \quad \mathbf{e}^\top\boldsymbol{\alpha} = 0. \tag{43}$$

Equation (40) is equivalent to $\boldsymbol{\mu} = \rho\mathbf{e} - \boldsymbol{\tau} - K_{ii}\mathbf{\Phi}\boldsymbol{\alpha}$. Therefore, given $\tilde{\boldsymbol{\alpha}}$ such that $\tilde{\alpha}_j \leq c\delta_{ij}$ and $\mathbf{e}^\top\tilde{\boldsymbol{\alpha}} = 0$, the KKT conditions can be reformulated as

$$\rho - \tau_j - K_{ii}\mathbf{\Phi}_j^\top\tilde{\boldsymbol{\alpha}} \geq 0, \tag{44}$$

$$(\rho - \tau_j - K_{ii}\mathbf{\Phi}_j^\top\tilde{\boldsymbol{\alpha}})(\tilde{\alpha}_j - c\delta_{ij}) = 0. \tag{45}$$

If $\tilde{\boldsymbol{\alpha}}$ is the optimal, then (44) suggests that

$$\rho \geq \tau_j + K_{ii}\mathbf{\Phi}_j^\top\tilde{\boldsymbol{\alpha}}, \quad \forall j = 1, 2, \dots, \kappa, \tag{46}$$

which means $\rho \geq \rho_{\max}(\tilde{\boldsymbol{\alpha}})$. According to (45), since $\tilde{\alpha}_j < c\delta_{ij}$ implies $\rho = \tau_j + K_{ii}\mathbf{\Phi}_j^\top\tilde{\boldsymbol{\alpha}}$, we can derive

$$\rho_{\min}(\tilde{\boldsymbol{\alpha}}) = \min_{j:a_j < c\delta_{ij}} K_{ii}\Phi_j^\top\boldsymbol{\alpha} + \tau_j = \rho \geq \rho_{\max}(\tilde{\boldsymbol{\alpha}}). \tag{47}$$

On the other hand, it is apparent that $\rho_{\max}(\tilde{\boldsymbol{\alpha}}) \geq \rho_{\min}(\tilde{\boldsymbol{\alpha}})$, so it imposes that $\rho_{\max}(\tilde{\boldsymbol{\alpha}}) = \rho_{\min}(\tilde{\boldsymbol{\alpha}})$.

If $\rho_{\max}(\tilde{\boldsymbol{\alpha}}) = \rho_{\min}(\tilde{\boldsymbol{\alpha}})$, we easily verify that $\exists\rho = \rho_{\max}(\tilde{\boldsymbol{\alpha}})$ such that both (44) and (45) hold. Therefore, the KKT conditions are satisfied and $\tilde{\boldsymbol{\alpha}}$ is the optimal solution. □

Theorem 1 gives the sufficient and necessary condition of the optimal solution, and thus it provides the stopping criterion of

Algorithm 3. That is, the optimality is reached if $\rho_{\max}(\boldsymbol{\alpha}) = \rho_{\min}(\boldsymbol{\alpha})$, or in practice,

$$\rho_{\max}(\boldsymbol{\alpha}) - \rho_{\min}(\boldsymbol{\alpha}) < \epsilon, \tag{48}$$

for a predefined tolerance parameter $\epsilon > 0$. Therefore, in each iteration, we update $\boldsymbol{\alpha}$ and then check (48) until it is satisfied, which indicates that the optimal solution is obtained. It should be noted that the evaluation of $\rho_{\max}$ and $\rho_{\min}$ takes time of $O(\kappa^2)$, which is the most time-consuming operation in Algorithm 3. Fortunately, it is possible to avoid such computation under modest conditions and obtain the result within $O(1)$ time instead. The following proposition reveals the conditions in which problem (37) has an all-zero solution.

Proposition 1: If $K_{ii} = 0$, or the y_i-th element of $\boldsymbol{\tau}$ is the largest, i.e.,

$$\tau_{y_i} = \max_j \tau_j, \tag{49}$$

then the optimal solution of (37) is $\boldsymbol{\alpha}^* = \mathbf{0}$.

Proof: Notice that if $K_{ii} = 0$ or (49) is satisfied, we have

$$\rho_{\max}(\mathbf{0}) = \max_j \tau_j = \tau_{y_i}, \tag{50}$$

$$\rho_{\min}(\mathbf{0}) = \max_{\delta_{ij} > 0} \tau_j = \tau_{y_i}. \tag{51}$$

Thus we have $\rho_{\max}(\mathbf{0}) = \rho_{\min}(\mathbf{0})$. According to Theorem 1, $\boldsymbol{\alpha} = \mathbf{0}$ is the optimal solution of problem (37). □

Proposition 1 demonstrates that the optimal solution can be directly obtained in some special cases. Therefore, we check whether the conditions in Proposition 1 are satisfied before the iterative procedure in Algorithm 3. If it satisfies, the algorithm will directly output $\boldsymbol{\alpha} = \mathbf{0}$ without iterations, which avoids redundant computation and further accelerates the algorithm.

Finally, we study the variable selection strategy in each iteration. A trivial way is to pick two indices r and s randomly. However, it is inefficient and usually causes slow convergence. Inspired by Theorem 1, we perform the following heuristic approach to determine which variables to optimize:

$$\begin{aligned} r &= \arg \max_{j=1,2,\ldots,k} K_{ii} \boldsymbol{\Phi}_j^\top \boldsymbol{\alpha} + \tau_j, \\ s &\in \mathcal{S}, \end{aligned} \tag{52}$$

where $\mathcal{S} = \{j = 1, 2, \ldots, k : \alpha_j < c\delta_{ij}\}$, and s is randomly picked from $\mathcal{S}$. The above strategy aims to minimize the gap between $\rho_{\max}(\boldsymbol{\alpha})$ and $\rho_{\min}(\boldsymbol{\alpha})$ as far as possible. The experiments also suggest that the proposed heuristic strategy is efficient. See Section V-B4 for details. We summarize the optimization procedure for the multi-class ODSVM in Algorithm 4.

IV. Theoretical Analysis

In this section, we proceed to the theoretical analysis of the convergence and efficiency of ODSVM.

A. Convergence Analysis

Unlike the existing methods, our method has a strict theoretical guarantee of global convergence. We begin with the following theorem that demonstrates the convergence of objective function throughout the training process.

Theorem 2: Algorithms 1 and 4 decrease the objective function monotonically, and thus, the algorithms will converge to a local minimum within finite iterations.

Proof: In the l-th iteration, since $\mathbf{w}$ (or $\mathbf{W}$) is obtained through SVM trained with $\mathbf{P}^\top\mathbf{X}$, we have $J(\mathbf{f}_{l+1}, \mathbf{P}_l, \mathbf{Q}_l) \leq J(\mathbf{f}_l, \mathbf{P}_l, \mathbf{Q}_l)$. For the $\mathbf{P}$ step, the duality theory implies that $J(\mathbf{f}_{l+1}, \mathbf{P}_{l+1}, \mathbf{Q}_l) \leq J(\mathbf{f}_{l+1}, \mathbf{P}_l, \mathbf{Q}_l)$. For the $\mathbf{Q}$ step, the Procrustes problem is solved and hence $J(\mathbf{f}_{l+1}, \mathbf{P}_{l+1}, \mathbf{Q}_{l+1}) \leq J(\mathbf{f}_{l+1}, \mathbf{P}_{l+1}, \mathbf{Q}_l)$. These inequalities show that $J(\mathbf{f}_{l+1}, \mathbf{P}_{l+1}, \mathbf{Q}_{l+1}) \leq J(\mathbf{f}_l, \mathbf{P}_l, \mathbf{Q}_l)$, which suggests the objective function decreases monotonically in each iteration. Since the objective function is bounded, i.e., $J \geq 0$, the algorithms converge to a limit value, which is the local minimum, within sufficient iterations. □

The above theorem intuitively suggests that the algorithms terminate within finite iterations. The numerical experiments also verified that the convergence is reached in a few number of iterations. Furthermore, we prove the convergence of iterative sequence to the local minimizer through the accumulation point. It is demonstrated by the following theorem.

Theorem 3: The sequence $\{(\mathbf{f}_l, \mathbf{P}_l, \mathbf{Q}_l)\}$ generated by Algorithms 1 or 4 has accumulation point. Denote $(\widehat{\mathbf{f}}, \widehat{\mathbf{P}}, \widehat{\mathbf{Q}})$ the accumulation point of the above sequence, then it is a local minimizer of problem (11).

The proof of Theorem 3 is provided in the Appendix, available online, which utilizes a similar approach presented in [39]. This theorem implies the global convergence to the local minimum, i.e., the iterative sequence generated by the proposed algorithms always converges to a local minimum point regardless of the initialization. This offers rigorous assurance of the algorithm's effectiveness and further guarantees the stability of the proposed method. It is a crucial property that distinguishes the proposed ODSVM from other methods simply combining SVM and feature extraction, as they may fail to converge and obtain a suboptimal performance.

B. Complexity Analysis

In this section, the time complexity is theoretically analyzed to show the efficiency of the proposed algorithm. Suppose the QPP solvers in the $\mathbf{w}$-step (or $\mathbf{W}$-step) and $\mathbf{P}$-step terminate if the change of objective function is less than ϵ. In binary-class ODSVM, matrices $\mathbf{Z}^\top\mathbf{M}\mathbf{Z}$ and $\mathbf{Z}^\top\mathbf{M}\mathbf{S}_B$ can be precomputed since it does not change during iteration, and the corresponding time complexity is $O(mn(m+n) + m^3)$. In $\mathbf{w}$-step, the computational bottleneck appears in solving QPP (2), which is solved with the dual coordinate descent method in LIBLINEAR [36]. Owing to the linear convergence rate of this solver [40], the time complexity of $\mathbf{w}$-step complexity is $O(\log(1/\epsilon)dn)$. In $\mathbf{P}$-step, we apply SOR algorithm, which also converges linearly [33], to solve the QPP (21). In each iteration of SOR, $O(n^2)$ operations are involved. Hence, the

$\mathbf{P}$-step complexity is $O(\log(1/\epsilon)n^2 + nmd)$. Finally, $\mathbf{Q}$-step involves the SVD of $\mathbf{S}_B^\top\mathbf{P}$, which takes $O(md^2)$. Therefore, the total time complexity of binary-class ODSVM is $O(mn(m+n) + m^3 + L(\log(1/\epsilon)(dn + n^2) + nmd + md^2))$, where L is the maximum iteration number of the outer loop. In general, L is significantly small according to empirical studies (see Section IV-A for details).

The optimization of multi-class ODSVM is more complicated than the binary-class counterpart, since the corresponding QPPs are larger and involve equality constraints. The computation of $\mathbf{Q}$ still takes time of $O(md^2)$. Similarly, matrices $\mathbf{K} = \mathbf{X}^\top\mathbf{M}\mathbf{X}$ and $\mathbf{X}^\top\mathbf{M}\mathbf{S}_B$ are precomputed with $O(mn(m+n) + m^3)$. In $\mathbf{W}$-step, LIBLINEAR employs the sequential dual method (SDM) to solve QPP, requiring time of $O(L_1 kdn)$ [35], where L_1 denotes the number of iterations of SDM. $\mathbf{P}$-step requires $O(nmd)$ time to compute the matrix $\mathbf{H}$. In Algorithm 2, The most time-consuming operation is the computation of $\boldsymbol{\tau}_i$ with complexity $O(nk^2)$. It should be noted that Algorithm 3 has a time complexity of $O(L_2k^2)$ (the worst case), which is considerably lower than $O(nk^2)$, especially for a large sample size. To justify this, it has been demonstrated by [41] that SMO-type methods offer nonasymptotic convergence rate of $O(1/\epsilon)$, i.e., $L_2 \leq O(1/\epsilon)$. Nevertheless, the asymptotic linear convergence rate is also established by [42], indicating that the convergence can be reached within $L_2 \leq O(\log(1/\epsilon))$ steps if the initialization is appropriate. Overall, it takes $O(Tn^2k^2)$ time to solve QPP (33), where T is the iteration number of the outer loop of Algorithm 2. The selection of T depends on the balance between precision and time. Empirically, it is appropriate to use $T \in [3, 5]$ to reach satisfactory performance.

V. Experiments

In this section, we conducted numerical experiments on 13 real-world datasets to show the effectiveness of the proposed ODSVM. Its classification performance is also compared with the related methods.

A. Experiment Setup

To evaluate the performance of the proposed ODSVM, we performed the experiments on 13 real-world datasets, where Musk, Heart, Sonar, Adult, DNA, Giseete, Epsilon, Web, and SmallNORB (S-NORB for short) are available on the LIBSVM website and UCI repository (https://archive.ics.uci.edu/ml). BinaryAlpha (B-Alpha for short) dataset includes the images of digits "0" through "9" and capital letters "A" through "Z" with binary features. CHG dataset records the grayscale images of different hand gestures [43]. OrganMNIST3D and NoduleMNIST3D (OM3D and NM3D for short) is a part of Medical MNIST (MedMNIST) datasets [44], [45], [46], which is composed of different CT images. The statistical properties of these datasets are summarized in Table I.

TABLE I
Profile of the Datasets Used in the Experiments

Dataset	#classes	#features	#samples
Sonar[1]	2	60	208
Heart[1]	2	13	270
Adult[1]	2	123	2265
Web[1]	2	300	4000
Musk[2]	2	166	7074
Epsilon[1]	2	2000	5000
Gisette[1]	2	5000	6000
NoduleMNIST3D (NM3D)[5]	2	21952	1633
BinaryAlpha (B-Alpha)[3]	36	320	1404
DNA[1]	3	180	3186
CHG[4]	9	8000	900
OrganMNIST3D (OM3D)[5]	11	21952	1743
SmallNORB (S-NORB)[1]	5	2048	24300

The proposed ODSVM was compared with other related feature extraction methods, including PCA [2], LPP [6], RDR [7] with SVM as classifier and MMPP [29], [30] and MMLDF [31]. The low-dimensional representations learned by the above methods are classified with SVM. To evaluate the impact of the classifier, we also classify these low-dimensional representations with the k-Nearest Neighbors (KNN) classifier [47], a non-parametric algorithm, and study its performance. The adjacency matrix $\mathbf{G}$ is constructed by (12). We randomly split each dataset into the training set and testing set with a proportion of 6:4, and repeated the experiments 10 times to compute the average accuracy as well as the standard deviation. In addition, following [48], we performed a paired t-test to test the statistical significance of ODSVM.

In terms of implementation, we use LIBLINEAR to perform both binary-class and multi-class SVM (CS-SVM) classification for its high efficiency. Besides, to reduce the computation burden, PCA pre-processing is applied to CHG, OM3D, and NM3D datasets. we preserve 98% of energy and drop the components corresponding to the small eigenvalues. The experiments were run on a PC (CPU: Intel Core i5, 2.70GHz; RAM: 16GB; OS: 64-bits Windows11) with MATLAB R2021b.

B. Experimental Results

1) Classification Performance: Table II shows the classification performance of the tested methods on each dataset. As shown in the table, we can observe that the proposed ODSVM achieves the best classification accuracy in all datasets. Moreover, it is obvious that most of the p-values are smaller than 0.05, which indicates that ODSVM exhibits higher performance than other compared methods with statistical significance. Fig. 2 depicts the classification accuracy versus the number of dimensions on Adult, BinaryAlpha, and CHG datasets. Overall, ODSVM achieves the highest accuracies for different numbers of dimensions, and we have the following observations.

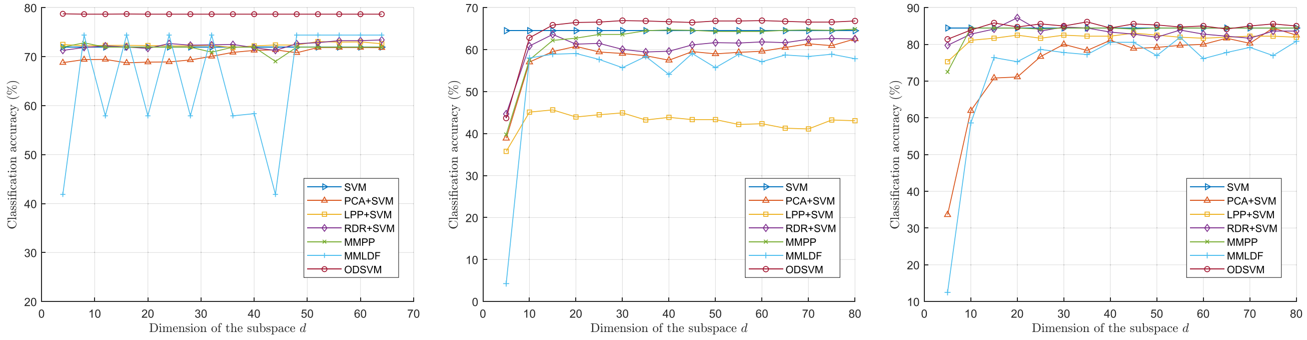
- Note that the curves of MMLDF are fluctuating, which indicates that the performance of MMLDF is sensitive to the number of dimensions. In contrast, the curve of

[1] https://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/
[2] http://archive.ics.uci.edu/ml
[3] https://cs.nyu.edu/∼roweis/data/
[4] http://mi.eng.cam.ac.uk/tkk22
[5] https://medmnist.com/

TABLE II
THE PERFORMANCE OF DIFFERENT FEATURE EXTRACTION METHODS WITH **SVM** AS CLASSIFIER

Dataset	Original	PCA [2]	LPP [6]	RDR [7]	MMPP [30]	MMLDF [31]	ODSVM
Heart	81.57±2.13 (0.0612)	80.92±2.78 (0.0218)	82.78±3.54 (0.2789)	82.50±2.97 (0.4048)	81.57±2.13 (0.0612)	79.81±2.71 (0.0371)	**83.06**±3.58 –
Sonar	73.73±4.34 (0.1379)	71.57±6.22 (0.1868)	71.57±4.54 (0.0220)	75.06±4.63 (0.6075)	73.73±4.34 (0.1379)	60.48±4.59 (1.8e-5)	**75.78**±4.59 –
Adult	71.88±1.46 (1.0e-5)	72.01±1.24 (1.0e-5)	72.01±1.12 (1.4e-5)	73.66±1.64 (0.0004)	71.84±1.39 (8.4e-6)	74.87±1.46 (0.0015)	**77.83**±1.97 –
Musk	92.27±0.15 (3.0e-5)	91.81±1.84 (6.3e-6)	91.72±0.43 (0.0007)	92.45±0.15 (0.0108)	91.96±0.43 (0.0022)	83.85±0.60 (1.0e-12)	**92.72**±0.28 –
Web	94.96±0.82 (0.0327)	90.98±4.08 (0.0106)	95.00±0.73 (0.0004)	94.99±0.57 (0.0082)	95.16±0.53 (0.0081)	92.33±2.61 (0.0049)	**95.63**±0.69 –
Epsilon	81.00±0.99 (0.0016)	78.17±0.81 (1.4e-6)	81.54±1.05 (0.8164)	81.46±1.04 (0.5675)	81.00±0.99 (0.0016)	80.75±0.51 (0.0107)	**81.58**±0.82 –
Gisette	94.60±0.50 (3.5e-5)	89.53±2.08 (9.2e-6)	95.81±0.38 (0.6188)	95.13±1.42 (0.1377)	94.66±0.57 (0.0017)	94.88±0.37 (2.5e-5)	**95.89**±0.26 –
NM3D	79.56±1.20 (0.0020)	79.23±2.24 (0.0057)	79.14±2.13 (0.0002)	82.16±0.86 (0.5076)	79.56±1.27 (0.0018)	79.00±2.37 (0.0002)	**82.51**±1.79 –
DNA	90.51±1.13 (0.0001)	64.34±1.94 (2.9e-11)	90.69±1.08 (0.0089)	92.55±1.12 (0.0224)	90.55±1.12 (9.0e-5)	83.19±2.07 (1.9e-6)	**93.04**±1.34 –
B-Alpha	63.40±2.01 (3.4e-5)	62.81±1.82 (3.7e-6)	40.66±1.57 (1.1e-11)	62.14±2.12 (3.8e-5)	63.45±2.09 (5.2e-5)	60.43±1.96 (1.1e-6)	**67.12**±1.46 –
CHG	86.78±2.02 (0.0371)	86.28±2.31 (0.0301)	85.14±2.32 (0.0005)	86.25±1.91 (0.0136)	86.56±1.91 (0.0052)	84.17±1.29 (0.0001)	**87.75**±2.09 –
OM3D	73.06±1.23 (6.5e-6)	73.06±1.23 (6.5e-6)	75.93±1.29 (1.8e-4)	77.47±1.78 (0.0023)	73.10±1.05 (5.5e-6)	75.67±1.70 (0.0091)	**78.38**±1.87 –
S-NORB	95.25±0.26 (8.6e-5)	72.30±0.87 (5.9e-13)	95.24±0.21 (3.9e-5)	95.77±0.22 (0.0003)	96.40±0.15 (0.1446)	93.24±0.26 (2.3e-8)	**96.67**±0.53 –

The average accuracy, standard deviation, and the p-value of the statistical significance test are shown. “Original” denotes the performance of the classifier without feature extraction. The best results are highlighted in boldface.



(a) Adult (b) BinaryAlpha (c) CHG

Fig. 2. The classification accuracy versus the dimension of the subspace. The curves of SVM represent the baseline performance that is obtained without dimensionality reduction.

ODSVM is smoother than those of others, which suggests the strong stability and robustness of ODSVM.

- The accuracy of MMPP is almost the same as SVM, and it implies that MMPP cannot significantly enhance the performance of SVM, while ODSVM can always reach higher classification accuracy than SVM.
- When d is small, the accuracy of ODSVM increases as d gets larger. Note that ODSVM reaches the best performance when $d \approx \kappa$, and the larger d $(d \gg \kappa)$ brings no significant increase in the accuracy, which means that ODSVM can obtain promising performance with only a few projections. Therefore, in most high-dimensional datasets, i.e., $\kappa \ll m$, we can find a relatively small d to achieve superb performance and reduce the dimension of data significantly at the same time.

Moreover, we performed k-nearest neighbor (KNN) classifier with the low-dimensional features learned by different methods on each dataset. The results are shown in Table III, where the embeddings generated by ODSVM are classified by both KNN (ODSVM+KNN) and SVM (ODSVM+SVM). Table III shows that ODSVM+KNN usually achieves better performance than other methods using the KNN classifier, which suggests

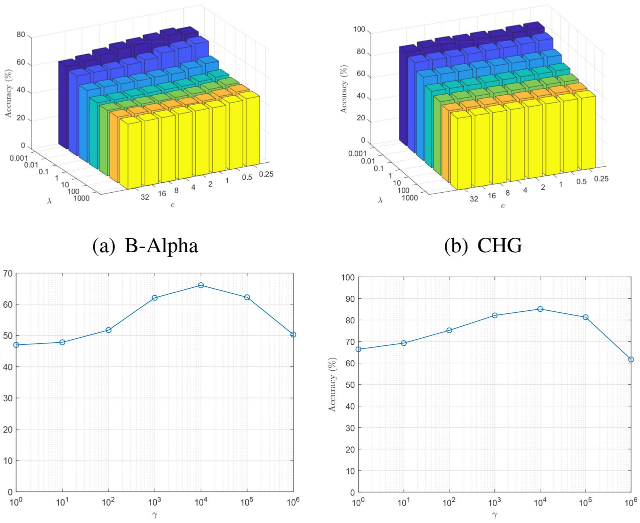
TABLE III
THE PERFORMANCE OF DIFFERENT FEATURE EXTRACTION METHODS WITH **KNN** CLASSIFIER

Dataset	Original	PCA [2]	LPP [6]	RDR [7]	MMPP [30]	MMLDF [31]	ODSVM+KNN	ODSVM+SVM
Heart	73.79±3.90 (4.5e-5)	73.98±4.12 (2.4e-5)	75.09±4.35 (3.7e-5)	75.37±2.62 (2.5e-5)	77.22±3.56 (0.0011)	70.37±9.42 (0.0038)	79.54±1.57 (0.0047)	**83.06**±3.58 –
Sonar	69.64±6.28 (0.0005)	70.48±7.07 (0.0009)	70.36±4.51 (0.0006)	74.94±5.52 (0.5657)	73.25±3.76 (0.0786)	70.12±3.80 (0.0031)	**76.14**±3.80 (0.7548)	75.78±4.59 –
Adult	75.91±1.97 (0.0489)	74.43±1.35 (0.0015)	73.04±1.84 (0.0001)	74.91±1.36 (0.0039)	75.49±0.70 (0.0062)	74.85±1.86 (0.0049)	76.56±1.10 (0.0436)	**77.83**±1.97 –
Musk	91.47±0.64 (0.0002)	89.40±0.81 (6.1e-7)	92.12±0.44 (0.0031)	92.65±0.39 (0.5777)	90.77±0.60 (3.6e-6)	92.48±0.51 (0.2494)	**93.01**±0.38 (0.1059)	92.72±0.28 –
Web	91.49±1.12 (1.5e-5)	90.84±1.11 (9.6e-7)	94.18±0.68 (0.0004)	94.03±0.84 (3.9e-5)	92.90±0.75 (0.0001)	93.24±0.61 (8.1e-7)	94.05±0.68 (0.0001)	**95.63**±0.69 –
Epsilon	56.46±0.95 (3.1e-11)	58.76±1.21 (6.3e-10)	58.56±0.92 (3.4e-12)	60.77±0.96 (1.5e-10)	59.82±1.32 (7.7e-10)	59.94±1.10 (9.2e-13)	78.58±0.74 (0.0002)	**81.58**±0.82 –
Gisette	95.53±0.45 (0.0628)	95.74±0.22 (0.2989)	**97.55**±0.19 (6.4e-7)	97.46±0.18 (2.2e-7)	95.43±0.34 (0.0252)	92.27±1.15 (5.0e-6)	96.54±0.39 (0.0016)	95.89±0.26 –
NM3D	77.61±1.26 (4.4e-6)	75.97±1.27 (2.0e-6)	77.53±1.23 (7.8e-7)	78.04±1.33 (0.0001)	76.81±0.98 (2.5e-5)	77.20±1.07 (3.1e-5)	79.04±1.33 (0.0012)	**82.51**±1.79 –
DNA	70.85±1.38 (1.1e-10)	81.44±1.47 (6.6e-9)	83.85±1.68 (3.1e-9)	84.49±1.44 (5.6e-8)	80.26±1.72 (7.1e-8)	77.06±1.32 (4.3e-10)	91.52±1.35 (0.0002)	**93.04**±1.34 –
B-Alpha	60.34±1.35 (3.5e-7)	62.51±1.68 (4.9e-5)	47.85±2.34 (3.1e-9)	62.90±2.12 (2.1e-5)	64.20±2.31 (0.0012)	63.29±1.60 (2.2e-5)	64.82±2.22 (0.0054)	**67.12**±1.46 –
CHG	82.08±2.01 (7.2e-5)	82.47±1.78 (1.1e-5)	85.03±1.49 (0.0007)	81.58±2.28 (2.5e-5)	84.16±2.18 (0.0006)	81.28±2.15 (8.0e-6)	85.53±3.02 (0.0137)	**87.75**±2.09 –
OM3D	71.25±1.31 (2.6e-7)	77.37±0.95 (0.0397)	74.28±2.03 (0.0002)	74.29±2.13 (0.0004)	76.74±1.48 (0.0321)	75.54±2.24 (0.0163)	76.87±1.47 (0.0358)	**78.37**±1.87 –
S-NORB	99.02±0.14 (1.1e-7)	99.03±0.10 (2.0e-7)	99.55±0.08 (3.9e-8)	**99.60**±0.07 (2.5e-8)	97.96±0.16 (4.2e-5)	91.74±1.35 (8.5e-6)	99.43±0.08 (6.3e-8)	96.67±0.53 –

The average accuracy, standard deviation, and the p-value of the statistical significance test are shown. "Original" denotes the performance of the classifier without feature extraction. The best results are highlighted in boldface.

that ODSVM can greatly enhance the discriminability of data to improve the performance of the KNN classifier. Note that ODSVM (with SVM as the classifier) still performs better than ODSVM+KNN on 10 out of 13 datasets with $p < 0.05$, which indicates that the learned representations of ODSVM are most suitable for SVM classification. It is an interesting observation that the performance of the KNN classifier with other methods is inferior on the Epsilon dataset (around 60%), but ODSVM+KNN achieves outstanding performance (78.58%). It implies that ODSVM can significantly improve data separability and enhance the performance of classifiers. In addition, we see that the KNN classifier usually performs better than SVM on the S-NORB dataset. The reason may be that the dataset is large enough and sufficiently dense, making it suitable for the nearest neighbor classifiers.

2) Parameter Study: In this section, we studied the influence of the hyperparameters on the performance of ODSVM. ODSVM contains three parameters to balance different penalty terms, i.e., c, λ, and γ. In the experiments, γ is selected from $\{10^i : i = 0, 1, \dots, 6\}$, λ from $\{10^i : i = -3, -2, \dots, 3\}$, and c from $\{2^i : i = -2, -1, \dots, 5\}$, respectively. The experimental results on B-Alpha and CHG datasets are illustrated in Fig. 3.



(a) B-Alpha

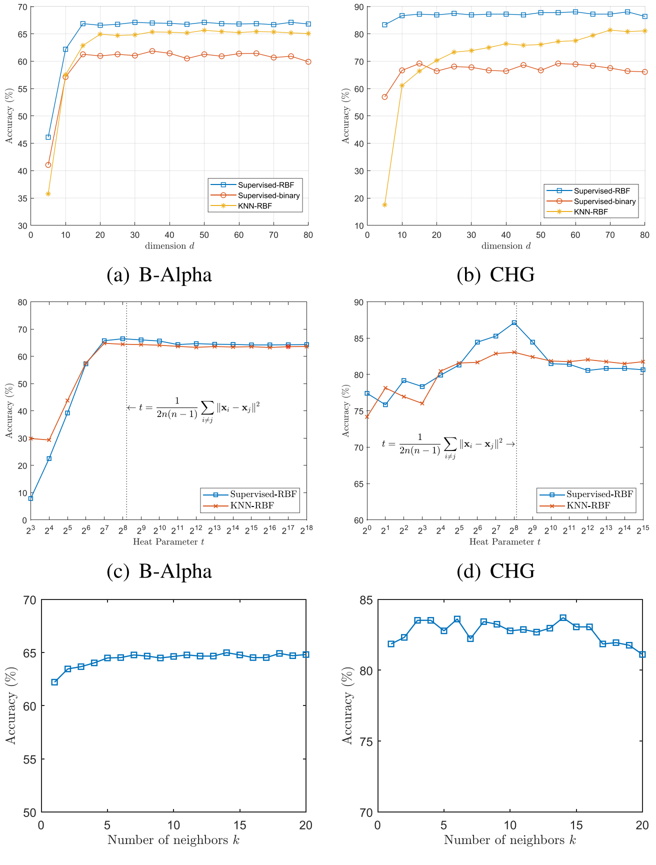
(b) CHG

(c) B-Alpha

(d) CHG

Fig. 3. The accuracy versus (a), (b) different c and λ, (c), (d) different γ.

As shown in Fig. 3(a) and (b), one can find that the large c as well as the large λ degrade the performance. It turns out that the optimal c and λ lies on intervals $[0.25, 1]$ and $[10^{-3}, 1]$, respectively. Parameter γ also affects the performance of ODSVM to a certain extent. From Fig. 3(c) and (d), we observe that ODSVM usually achieves the best accuracy when $\gamma \in [10^3, 10^5]$, which implies that the regularization of projection matrix $\mathbf{P}$ can improve the generalization ability of the proposed method.



(a) B-Alpha (b) CHG

(c) B-Alpha (d) CHG

(e) B-Alpha (f) CHG

Fig. 4. Impact of (a), (b) different graph construction $\mathbf{G}$, (c), (d) the heat parameter for RBF-Supervised and RBF-KNN graphs, and (e), (f) the number of the nearest neighbors k for the RBF-KNN graph.

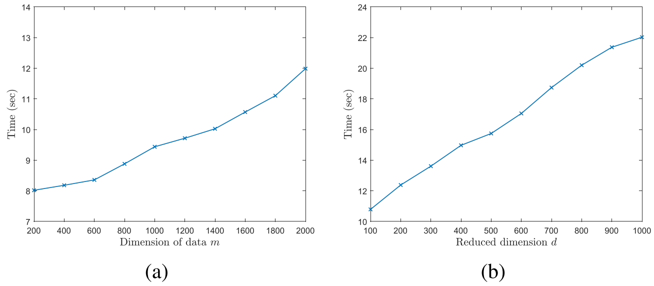
3) Graph Construction: The setting of graph matrix $\mathbf{G}$ is also an essential factor affecting the classification performance. Therefore, we compared three commonly used graph settings, including (12), which is denoted as **"Supervised-RBF"**, and the following two graphs:

$$G_{ij} = \begin{cases} 1, & y_i = y_j, \\ 0, & \text{otherwise.} \end{cases} \qquad (\textbf{Supervised-binary})$$

$$G_{ij} = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{t}\right), & \mathcal{N}_k(\mathbf{x}_i, \mathbf{x}_j) = 1, \\ 0, & \text{otherwise.} \end{cases} \qquad (\textbf{KNN-RBF})$$

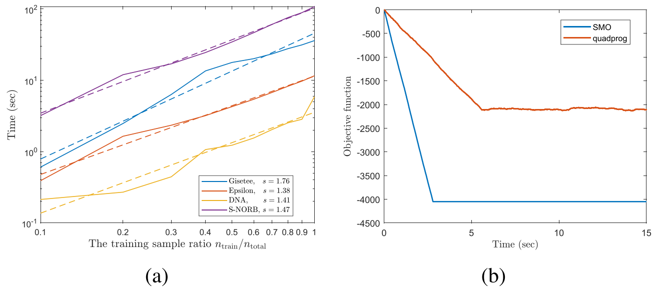
where $\mathcal{N}_k(\mathbf{x}_i, \mathbf{x}_j) = 1$ means that $\mathbf{x}_i$ is KNN of $\mathbf{x}_j$, or $\mathbf{x}_j$ is the KNN of $\mathbf{x}_i$. Then we study two hyperparameters in the graph construction mentioned above: the heat parameter t of the RBF-based graphs and the number of nearest neighbors k.

Fig. 4(a) and (b) show the performance of ODSVM learned with different graphs. Apparently, "Supervised-RBF" always achieves the best accuracy, which means it can fully drive the best performance of ODSVM. This is why we use this graph in the classification tasks. In contrast, "Supervised-binary" obtains relatively unstable results and fails to perform with better accuracy than the others. We observe that "KNN-RBF" does not use the supervised information of the within-class similarity, but still, with only neighborhood information, achieves higher accuracy than "Supervised-binary". This phenomenon might come from the adaptivity of the RBF graph, which can better capture the similarity of samples and thus bring better performance.



(a) (b)

Fig. 5. The training time versus (a) the dimension of data m, fixing $d = 16$ and (b) the reduced dimension d, fixing $m = 2048$ on the S-NORB dataset.



(a) (b)

Fig. 6. (a) The training time versus the sample size, where both x-axis and y-axis are log-scaled. (b) The objective function of QPP (33) versus runtime with of different solvers.

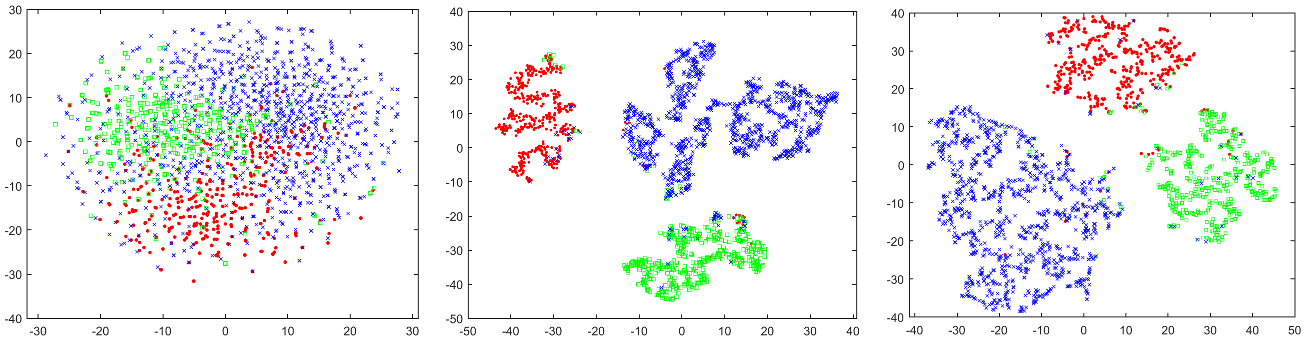
TABLE IV
THE PERFORMANCE COMPARISON, INCLUDING ACCURACY (%) AND EXECUTING TIME (SEC) OF ODSVM ON MULTI-CLASS DATASETS

Datasets	ODSVM (quadprog)	ODSVM (SMO)	Speedup
	ACC (%), Time (s)	ACC (%), Time (s)	
DNA	92.76, 9.0854	**93.04**, **0.4610**	19.70x
B-Alpha	66.99, 9.4787	**67.12**, **1.6632**	5.70x
CHG	86.14, 2.3284	**87.75**, **1.1022**	2.11x
OM3D	77.50, 7.4440	**78.38**, **1.6580**	4.49x

As shown in Fig. 4(c) and (d), for both Supervised-RBF and KNN-RBF graphs, ODSVM's performance is poor when t is small. As t increases, the accuracy improves significantly and achieves a highest value rapidly. After that, the accuracy drops slightly with the increase of t. Hence, there exists a moderate t that derives the top accuracy. In the experiments, we found that the following value is nearly optimal:

$$t = \frac{1}{2n(n-1)}\sum_{i\neq j}\|\mathbf{x}_i - \mathbf{x}_j\|^2, \tag{53}$$

where $\{\mathbf{x}_i\}_{i=1}^n$ is the training set. The reason is that (53) makes the value $\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/t)$ relatively robust to the scale of the data. In this way, the constructed graph is informative to illustrate the similarity between samples. Besides, We also observe that KNN-RBF is relatively robust to the setting of t.



(a) Original data (b) ODSVM embedding (c) Reconstruction data

Fig. 7. T-SNE visualization on DNA dataset. The data separability is greatly improved in the subspace obtained by ODSVM.

Although its highest performance is not as good as Supervised-RBF, it may perform better than Supervised-RBF when t is extremely large.

We illustrate the performance of ODSVM versus k in Fig. 4(e) and (f) for KNN-RBF graphs. It can be observed that the accuracy increases as k grows at the beginning on both CHG and B-Alpha datasets. When k is large, the ODSVM's performance tends to be stable on the B-Alpha dataset, but it drops modestly on the CHG dataset. The cause of slight performance degradation may be that larger k introduces more dissimilar data. In practical application, a large k is not recommended, as it may involve irrelevant samples and also increase the computational cost. The optimal k should be a small but sufficient number to capture the local structure of the data. In our experiments, we found that such k is around $\lfloor \log_2 n \rfloor$ for most datasets.

4) Efficiency Evaluation: We conducted the experiments to investigate the efficiency of the proposed method. Fig. 5 illustrates the influence of data dimension m, and the reduced dimension d on the training time. Intuitively, the time roughly exhibits linear correlation w.r.t both m and d. This phenomenon coincides with the theoretical result in Section IV-B, which indicates that the iteration time is linearly dependent on m and d, respectively. Although the precomputation of $\mathbf{X}^\top \mathbf{M}\mathbf{X}$ and $\mathbf{X}^\top \mathbf{M}\mathbf{S}_B$ requires the time of $O(mn(m+n)+m^3)$, its influence on overall computational time is insignificant compared to the time required for iteration procedure when m varies in a modest interval. However, the precomputation process might become the computational bottleneck if m is sufficiently large. This impact is also observable in Fig. 5(a) for $m > 1400$, where the growth rate of the curve tends to increase slightly. Thus, we recommend preprocessing data with PCA or downsampling to alleviate this problem.

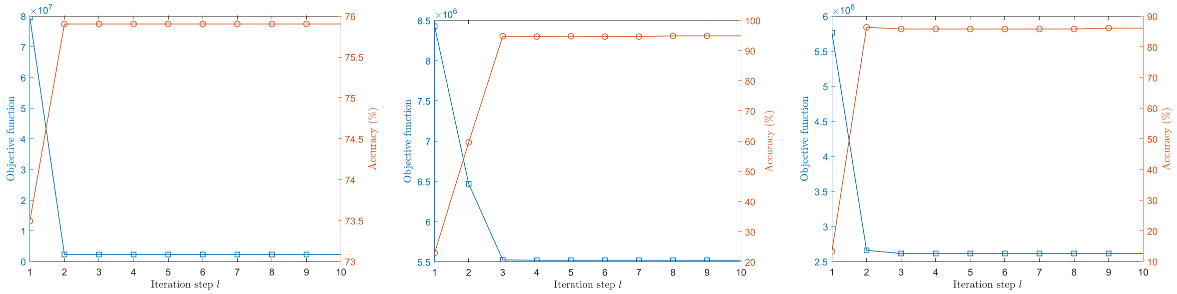
The influence of sample size on the training time is examined on different datasets. The results are shown in Fig. 6(a), where the axes are log-scaled. Assuming $\mathsf{Time} = O(n^s)$, or $\log(\mathsf{Time}) \approx s\log(n) + \mathsf{Bias}$, we can establish the linear correlation between time and sample size in the log-scaled space. Therefore, s represents the slope of the curves in Fig. 6(a). For instance, we can see $s = 1.47$ for the S-NORB dataset and its training time $\mathsf{Time} = O(n^{1.47})$. It is observed from Fig. 6(a) that $s < 2$ for both binary-class and multi-class datasets. The theoretical results in Section IV-B suggest that $s = 2$ is the worst case. Nevertheless, the pruning operation based on Proposition 1 significantly reduces the practical time, which verifies the efficiency of the proposed algorithms.

We further evaluated the performance of the SMO algorithm in multi-class ODSVM, and MATLAB built-in function "quadprog" serves as a baseline, which applies the interior point algorithm to solve QPP. We plot the variation of the objective function (33) during the optimization of SMO and "quadprog" in Fig. 6(b). It is shown that the proposed SMO algorithm achieves the minimum quickly, whereas "quadprog" is slower and fails to reach the optimal solution. This result demonstrates the superiority of the proposed SMO algorithm.

Additionally, we compared the execution time of ODSVM with "quadprog" and SMO as the solver of problem (37), respectively. The results are shown in Table IV. It is obvious that the computation time is significantly reduced via using the SMO algorithm instead of "quadprog", and ODSVM with SMO also obtains higher accuracy than ODSVM with "quadprog", indicating that the SMO algorithm will achieve a more precise solution than "quadprog", which fits the results in Fig. 6(b). The reason for this phenomenon might be that "quadprog" usually produces an interior point solution which may be suboptimal with a loss of precision. In contrast, the proposed SMO algorithm always yields a solution that strictly satisfies the KKT condition, and the optimality is rigorously guaranteed. In short, the experimental results verify the high efficiency and effectiveness of the proposed SMO algorithm.

5) Feature Visualization: To intuitively show the effect of ODSVM, we visualized the original high-dimensional features, the embedding obtained by ODSVM, and the reconstruction data via T-SNE algorithm [49]. The results are illustrated in Fig. 7. It is evident that the original data shown in Fig. 7(a) is scattered and different classes overlap heavily. In contrast, the learned embedding and reconstruction data (Fig. 7(b) and (c), respectively) of ODSVM are more compact, and have smaller within-class scatters and more significant margins between classes than the original data. Therefore, the class separability of data is prominently improved in the subspace learned by ODSVM.

6) Convergence Analysis: In order to verify the theoretical analysis in Section IV-A, we plotted the convergence curves of the proposed algorithms on Sonar, DNA, and CHG datasets in Fig. 8. It is clear that the proposed ODSVM algorithm



(a) Sonar (b) DNA (c) CHG

Fig. 8. Convergence of binary-class ODSVM on Sonar dataset (a), and multi-class ODSVM on DNA dataset (b) and CHG dataset (c), respectively.

monotonically decreases the objective function in the iteration until convergence. Moreover, the curves show that the objective function reaches the local minimum points within a few iterations, highlighting the algorithms' efficient convergence. During the iteration, the accuracy of ODSVM monotonically increases and eventually achieves a steady value, which further indicates the stability of the proposed ODSVM.

VI. Conclusion

For the last decade, it has been a great challenge to integrate SVM and feature extraction into a unified model to meet the essential requirements of model effectiveness and algorithm convergence. In this paper, we tackle this tough problem by devising a novel joint learning method, namely ODSVM, which integrates graph-regularized discriminative feature extraction and SVM seamlessly as a joint learning framework. The efficient algorithms for binary-class and multi-class ODSVM are proposed, and the block coordinate descent approach incorporating the SMO solver is designed to solve the large-scale QPP efficiently. The establishment of global convergence towards the local minimum makes ODSVM significantly distinctive from previous works. To the best of our knowledge, ODSVM is the first method with such a rigorous convergence guarantee that implements SVM collaborated with discriminative feature extraction. Therefore, the proposed ODSVM is capable of achieving superior classification performance, which is validated in the experimental studies.

Beyond the scope of this paper, we give some potential directions for future studies. First, more characteristics and applications of the framework (10) could be explored, leading to novel schemes of representation learning. For example, it is possible to extend (10) into the domain generalization tasks, where the domain adaption can be illustrated with a well-designed cross-domain graph. The development of the nonlinear extension of ODSVM is also encouraged to improve its capacity, which can be used for nonlinear feature learning and classification on more complicated datasets. Future directions may involve deep learning and kernel methods leveraging ODSVM's optimization strategies and theories. Lastly, since a computational bottleneck may arise when dealing with large-scale matrix multiplication and inversion, further enhancement of ODSVM's efficiency for Big Data could be deemed a worthy research objective in the future.

References

[1] M. Verleysen and D. François, "The curse of dimensionality in data mining and time series prediction," in *Proc. Int. Work-Conf. Artif. Neural Netw.*, Springer, 2005, pp. 758–770.

[2] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.

[3] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, Feb. 2001.

[4] F. Zhu et al., "Neighborhood linear discriminant analysis," *Pattern Recognit.*, vol. 123, 2022, Art. no. 108422.

[5] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[6] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using Laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, Mar. 2005.

[7] Z. Lai, D. Mo, W. K. Wong, Y. Xu, D. Miao, and D. Zhang, "Robust discriminant regression for feature extraction," *IEEE Trans. Cybern.*, vol. 48, no. 8, pp. 2472–2484, Aug. 2018.

[8] J. Wen, N. Han, X. Fang, L. Fei, K. Yan, and S. Zhan, "Low-rank preserving projection via graph regularized reconstruction," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1279–1291, Apr. 2019.

[9] L. Hu, W. Zhang, and Z. Dai, "Joint sparse locality-aware regression for robust discriminative learning," *IEEE Trans. Cybern.*, vol. 52, no. 11, pp. 12245–12258, Nov. 2022.

[10] F. Nie, X. Zhao, R. Wang, and X. Li, "Adaptive maximum entropy graph-guided fast locality discriminant analysis," *IEEE Trans. Cybern.*, vol. 53, no. 6, pp. 3574–3587, Jun. 2023.

[11] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[12] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. Cambridge, MA, USA: MIT Press, 2018.

[13] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.

[14] V. Blanco, J. Puerto, and A. M. Rodriguez-Chia, "On l_p-support vector machines and multidimensional kernels," *J. Mach. Learn. Res.*, vol. 21, no. 14, pp. 1–29, 2020.

[15] H. Wang, Y. Shao, S. Zhou, C. Zhang, and N. Xiu, "Support vector machine classifier via soft $L_{0/1}$-margin loss," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 7253–7265, Oct. 2022.

[16] R. Khemchandani, S. Chandra, and A. Jayadeva, "Twin support vector machines for pattern classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 905–910, May 2007.

[17] Y. Tian, Z. Qi, X. Ju, Y. Shi, and X. Liu, "Nonparallel support vector machines for pattern classification," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1067–1079, Jul. 2014.

[18] L. Liu, M. Chu, R. Gong, and L. Zhang, "An improved nonparallel support vector machine," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 11, pp. 5129–5143, Nov. 2021.

[19] H. Yan et al., "Robust distance metric optimization driven GEPSVM classifier for pattern classification," *Pattern Recognit.*, vol. 129, 2022, Art. no. 108779.

[20] I. Kotsia and I. Pitas, "Facial expression recognition in image sequences using geometric deformation features and support vector machines," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 172–187, Jan. 2007.

[21] T. Ni, F.-L. Chung, and S. Wang, "Support vector machine with manifold regularization and partially labeling privacy protection," *Inf. Sci.*, vol. 294, pp. 390–407, 2015.

[22] H.-H. Tsai and Y.-C. Chang, "Facial expression recognition using a combination of multiple facial features and support vector machine," *Soft Comput.*, vol. 22, no. 13, pp. 4389–4405, 2018.

[23] M. Goudjil et al., "A novel active learning method using SVM for text classification," *Int. J. Automat. Comput.*, vol. 15, no. 3, pp. 290–298, 2018.

[24] D. Han, N. Zhao, and P. Shi, "Gear fault feature extraction and diagnosis method under different load excitation based on emd, PSO-SVM and fractal box dimension," *J. Mech. Sci. Technol.*, vol. 33, no. 2, pp. 487–494, 2019.

[25] H. Li, T. Jiang, and K. Zhang, "Efficient and robust feature extraction by maximum margin criterion," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 157–165, Jan. 2006.

[26] F. Wang, B. Zhao, and C. Zhang, "Unsupervised large margin discriminative projection," *IEEE Trans. Neural Netw.*, vol. 22, no. 9, pp. 1446–1456, Sep. 2011.

[27] S. Moon and H. Qi, "Hybrid dimensionality reduction method based on support vector machine and independent component analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 5, pp. 749–761, May 2012.

[28] S. Yang et al., "Unsupervised maximum margin feature selection via $L_{2,1}$-norm minimization," *Neural Comput. Appl.*, vol. 21, no. 7, pp. 1791–1799, 2012.

[29] D. Xie, F. Nie, and Q. Gao, "On the optimal solution to maximum margin projection pursuit," *Multimedia Tools Appl.*, vol. 79, no. 47, pp. 35 441–35 461, 2020.

[30] S. Nikitidis, A. Tefas, and I. Pitas, "Maximum margin projection subspace learning for visual data analysis," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4413–4425, Oct. 2014.

[31] C. Li, Q. Liu, W. Dong, F. Wei, X. Zhang, and L. Yang, "Max-margin-based discriminative feature learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2768–2775, Dec. 2016.

[32] J. C. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Advances in Kernel Methods - Support Vector Learning, Tech. Rep. MSR-TR-98-14, 1998.

[33] O. L. Mangasarian and D. R. Musicant, "Successive overrelaxation for support vector machines," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1032–1037, Sep. 1999.

[34] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *J. Mach. Learn. Res.*, vol. 2, no. Dec, pp. 265–292, 2001.

[35] S. S. Keerthi et al., "A sequential dual method for large scale multi-class linear SVMs," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2008, pp. 408–416.

[36] R.-E. Fan et al., "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, 2008.

[37] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graphical Statist.*, vol. 15, no. 2, pp. 265–286, 2006.

[38] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[39] Y. Shen, H. Xu, and X. Liu, "An alternating minimization method for robust principal component analysis," *Optim. Methods Softw.*, vol. 34, no. 6, pp. 1251–1276, 2019.

[40] C.-J. Hsieh et al., "A dual coordinate descent method for large-scale linear SVM," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 408–415.

[41] A. Beck, "The 2-coordinate descent method for solving double-sided simplex constrained minimization problems," *J. Optim. Theory Appl.*, vol. 162, pp. 892–919, 2014.

[42] P.-H. Chen, R.-E. Fan, and C.-J. Lin, "A study on SMO-type decomposition methods for support vector machines," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 893–908, Jul. 2006.

[43] T.-K. Kim and R. Cipolla, "Canonical correlation analysis of video volume tensors for action categorization and detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 8, pp. 1415–1428, Aug. 2009.

[44] J. Yang et al., "MedMNIST v2—A large-scale lightweight benchmark for 2D and 3D biomedical image classification," *Sci. Data*, vol. 10, no. 1, 2023, Art. no. 41.

[45] X. Xu et al., "Efficient multiple organ localization in CT image using 3D region proposal network," *IEEE Trans. Med. Imag.*, vol. 38, no. 8, pp. 1885–1898, Aug. 2019.

[46] S. G. Armato III et al., "The lung image database consortium (LIDC) and image database resource initiative (IDRI): A completed reference database of lung nodules on CT scans," *Med. Phys.*, vol. 38, no. 2, pp. 915–931, 2011.

[47] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.

[48] C. Wang et al., "Robust capped L_1-norm twin support vector machine," *Neural Netw.*, vol. 114, pp. 47–59, 2019.

[49] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[50] I. Söderkvist, "Perturbation analysis of the orthogonal procrustes problem," *BIT Numer. Math.*, vol. 33, pp. 687–694, 1993.

Junhong Zhang is currently working toward the PhD degree with Shenzhen University, Shenzhen, China. He is with the College of Computer Science and Software Engineering, Shenzhen University. His research interests include machine learning, pattern recognition, and large-scale kernel methods.

Zhihui Lai (Member, IEEE) received the BS degree in mathematics from South China Normal University, the MS degree from Jinan University, and the PhD degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology (NUST), China, in 2002, 2007 and 2011, respectively. He has been a research associate, postdoctoral fellow, and research fellow with The Hong Kong Polytechnic University. His research interests include face recognition, image processing and content-based image retrieval, pattern recognition, compressive sense, human vision modelization, and applications in the fields of intelligent robot research. He has published more than 200 scientific articles. Now he is a professor with Shenzhen University and an associate editor of *International Journal of Machine Learning and Cybernetics*. For more information including all papers and related codes, the readers are referred to the website (http://www.scholat.com/laizhihui).

Heng Kong received the BS and MD degrees from Chongqing Medical University, the MS degree from Guangzhou Medical University, and the PhD degree from Southern Medical University, China, in 2000, 2005 and 2008, respectively. She works as a visiting scholar in Cancer Center of Georgia Reagent University at Augusta in USA in 2014-2016. She is a professor and director in department of thyroid and breast surgery, BaoAn Central Hospital of Shenzhen (the fifth affiliated Hospital of Shenzhen University), Guangdong province. She is also doing basic and clinic research associated breast and thyroid cancer. Her research interests include gene therapy, immunotherapy, early diagnosis and prognosis analysis of breast cancer, and tumor image processing and recognition using machine learning and artificial intelligent methods.

Jian Yang received the PhD degree from the Nanjing University of Science and Technology (NJUST), Nanjing, China, in 2002, with a focus on pattern recognition and intelligence systems. In 2003, he was a post-doctoral researcher with the University of Zaragoza, Aragon, Spain. From 2004 to 2006, he was a postdoctoral fellow with the Hong Kong Polytechnic University, Hong Kong, China. From 2006 to 2007, he was a postdoctoral fellow with the New Jersey Institute of Technology, NJ, USA. He is currently a Chang-Jiang professor with the NJUST. He is the author of more than 200 scientific articles in pattern recognition and computer vision. His articles have been cited more than 38,000 times in the Scholar Google. He is a fellow of International Association for Pattern Recognition (IAPR). He currently is/was an associate editor of *Pattern Recognition*, *Pattern Recognition Letters*, *IEEE Transactions on Neural Networks and Learning Systems*, and *Neurocomputing*.