International Journal of Machine Learning and Cybernetics (2020) 11:2489–2499
https://doi.org/10.1007/s13042-020-01134-2

ORIGINAL ARTICLE

Multi-view semi-supervised least squares twin support vector machines with manifold-preserving graph reduction

Xijiong Xie[1]

Received: 8 September 2019 / Accepted: 15 April 2020 / Published online: 15 May 2020

Abstract

Multi-view semi-supervised support vector machines consider learning with multi-view unlabeled data to boost the learning performance. However, they have several defects. They need to solve the quadratic programming problem and the time complexity is quite high. Moreover, when a large number of multi-view unlabeled examples exist, it can generate more outliers and noisy examples and influence the performance. Therefore, in this paper, we propose two novel multi-view semi-supervised support vector machines called multi-view Laplacian least squares twin support vector machine and its improved version with the manifold-preserving graph reduction which can enhance the robustness of the algorithm. They can reduce the time complexity by changing the constraints to a series of equality constraints and lead to a pair of linear equations. The linear multi-view Laplacian least squares twin support vector machine and its improved version with manifold-preserving graph reduction are further generalized to the nonlinear case via the kernel trick. Experimental results demonstrate that our proposed methods are effective.

Keywords Multi-view semi-supervised learning · Least squares twin support vector machines · Semi-supervised learning · Manifold-preserving graph reduction

1 Introduction

Support vector machine (SVM) has been widely investigated [1–4], which implements the structural risk minimization of statistical learning theory. In contrast with other classification algorithms such as artificial neural networks [5], SVM can gain a better generalization capability. In recent years, non-parallel hyperplane classifiers have emerged and attracted much attention of many researchers. Twin support vector machine (TSVM) [6] is a typical non-parallel hyperplane classifier which creates two non-parallel hyperplanes such that one of the hyperplanes is closer to one class and has a certain distance to the other. Although the scale of TSVM is smaller than SVM, it still need to solve two quadratic programming problems (QPPs). Least squares twin support vector machine (LSTSVM) [7–10] can make its learning speed faster than the one of TSVM.

In real application, collecting labeled examples spends most time and manual labor, while the collection of unlabeled examples may be relatively easy. Semi-supervised learning [11–14] was presented to handle this problem. When the unlabeled data are adopted rationally, it can outperform the performance of the counterpart supervised learning approach. There exist several semi-supervised learning methods of SVM and TSVM, such as transductive SVM [15], semi-supervised support vector machines [16], Laplacian support vector machines (LapSVM) [17] and Laplacian twin support vector machines (LapTSVM) [18].

Multi-view learning [19–23] is a hot spot in machine learning with good theoretical evidence and great successful practice. Multi-view learning leverages multiple feature sets to improve the generalization performance. For examples, images and videos, color information and texture information are two different kinds of features, which can serve as multi-view data. In web page classification, there exist two views for describing a given web page: the text content in itself and the anchor text linking to this web page. A noteworthy character of multi-view learning is that performance on a natural single view can be improved by using manually generated multiple views.

✉ Xijiong Xie
xjxie11@gmail.com

[1] The School of Information Science and Engineering, Ningbo University, Zhejiang 315211, China

The current multi-view learning methods can be classified into three major styles: co-training style algorithms, co-regularization style algorithms and margin consistency style algorithms. Co-training style algorithms are inspired by co-training [24] which is one of the earliest methods that the learners are trained alternately on two distinct views with confident labels for the unlabeled data. Representative algorithms are co-EM [25], co-testing [26] and robust co-training [27]. Co-regularization style algorithms fuse regularization terms of discriminant or regression function with the objective function. SVM-2K [28], sparse multi-view SVM [29], multi-view TSVM (MvTSVM) [30], multi-view privileged SVM (PSVM-2V) [31], multi-view LapSVM (MvLapSVM) [32] and multi-view LapTSVM (MvLapTSVM) [33] are typical algorithms. Farquhar et al. [28] provided a theoretical analysis for SVM-2K and reduce the Rademacher complexity [34] of the corresponding function class significantly. Sun and Shawe-Taylor [29] characterized the generalization error of sparse multi-view SVM for the margin bound and derived the empirical Rademacher complexity of the considered function class. Recently, Sun et al. [28] proposed several PAC-Bayes bounds for co-regularization style algorithms, which are the first application of PAC-Bayes theory under the supervised and semi-supervised multi-view learning framework. Margin-consistency style algorithms are recently proposed to make use of the latent consistency of classification results from multiple views [36–39]. They are realized under the framework of maximize entropy discrimination (MED), such as alternative multi-view maximum entropy discrimination (AMVMED) [37]. Chao et al. [40] proposed semi-supervised multi-view maximum entropy discrimination with expectation Laplacian regularization.

MvLapTSVM combines two views with the constraint of similarity between two distinct TSVM from two feature spaces. However, MvLapTSVM has some disadvantages as follows. It needs to solve a pair of QPPs and the time complexity is quite high. A large number of unlabeled examples can generate more outliers and noisy examples. This can effect the performance of the algorithm and increase the running time of the algorithm. In this paper, to overcome the above disadvantages, we propose two novel multi-view semi-supervised SVMs called multi-view Laplacian least squares twin support vector machine (MvLapLSTSVM) and its improved version with manifold-preserving graph reduction (MPGR) [41] (MvLapLSTSVM with MPGR). The MPGR is a sparse approximate method that uses only a subset of the examples and focuses on the strategy of selecting the representative and informative examples to form the sparse subset. It can eliminate outliers and noisy examples so can enhance the robustness of the relevant algorithm. Here we first search for two sparse subsets of two views by the MPGR, respectively. Then a new sparse subset is formed by the intersection of the two sparse subsets. This would enhance the robustness of the algorithm.

The contribution of this paper is concluded below:

(1) They combine two views by introducing a multi-view co-regularization term and leverage the manifold regularization to semi-supervised learning.
(2) They can reduce the time complexity by changing the constraints to a series of equality constraints and lead to a pair of linear equations based on the principle of LSTSVM.
(3) MvLapLSTSVM with MPGR can integrate the MPGR to select informative unlabel examples from a large number of unlabel examples. This strategy can enhance the robustness of the relevant algorithm MvLapLSTSVM.
(4) Experimental results validate the effectiveness of our proposed methods.

The remainder of this paper proceeds as follows. Section 2 reviews related work about LSTSVM, MvLapTSVM and MPGR. Section 3 thoroughly introduces our proposed methods MvLapLSTSVM and its improved version with MPGR. After reporting experimental results in Sect. 4, we give conclusions and future work in Sect. 5.

2 Related work

In this section, we briefly review LSTSVM, MvLapTSVM and MPGR.

2.1 LSTSVM

Suppose training examples belonging to classes 1 and -1 are represented by matrices A_+ and B_-, and the size of A_+ and B_- are $(m_1 \times d)$ and $(m_2 \times d)$, respectively. The central idea of LSTSVM [7] is to seek two non-parallel hyperplanes

$$w_1^\top x + b_1 = 0 \quad \text{and} \quad w_2^\top x + b_2 = 0 \tag{1}$$

around which the examples of the corresponding class get clustered. Define two matrices A, B and four vectors v_1, v_2, e_1, e_2,

$$A = (A_+, e_1),\ B = (B_-, e_2),\ v_1 = \begin{pmatrix} w_1 \\ b_1 \end{pmatrix},\ v_2 = \begin{pmatrix} w_2 \\ b_2 \end{pmatrix}. \tag{2}$$

For simplicity in the paper, e, e_1 and e_2 are vectors of ones of appropriate dimensions.

The classifier is given by solving the following QPPs separately.

(LSTSVM1)

$$\begin{aligned} \min_{v_1,q_1} \quad & \frac{1}{2}(Av_1)^\top(Av_1) + \frac{c_1}{2}q_1^\top q_1 \\ \text{s.t.} \quad & -Bv_1 + q_1 = e_2, \end{aligned} \tag{3}$$

(LSTSVM2)

$$\begin{aligned} \min_{v_2,q_2} \quad & \frac{1}{2}(Bv_2)^\top(Bv_2) + \frac{d_1}{2}q_2^\top q_2 \\ \text{s.t.} \quad & Av_2 + q_2 = e_1, \end{aligned} \tag{4}$$

where c_1, d_1 are nonnegative parameters and q_1, q_2 are slack vectors of appropriate dimensions. The term $\frac{1}{2}(Av_1)^\top(Av_1)$ and constraint $-Bv_1 + q_1 = e_2$ aim to make $+1$ class hyperplane closest to the corresponding $+1$ class and as far as possible from the corresponding -1 class simultaneously. The term $\frac{1}{2}(Bv_2)^\top(Bv_2)$ and constraint $Av_2 + q_2 = e_1$ aim to make -1 class hyperplane closest to the corresponding -1 class and as far as possible from the corresponding $+ 1$ class simultaneously. Each of the above two QPPs can be converted to the explicit expression of LSTSVM.

(LSTSVM1)

$$\min_{v_1} \quad \frac{1}{2}(Av_1)^\top(Av_1) + \frac{1}{2}c_1(e_2 + Bv_1)^\top(e_2 + Bv_1), \tag{5}$$

(LSTSVM2)

$$\min_{v_2} \quad \frac{1}{2}(Bv_2)^\top(Bv_2) + \frac{1}{2}d_1(e_1 - Av_2)^\top(e_1 - Av_2). \tag{6}$$

The two non-parallel hyperplanes are obtained by solving the following two linear equations:

$$\begin{aligned} v_1 &= -\left(A^\top A + \frac{1}{c_1}B^\top B\right)^{-1} A^\top e_2, \\ v_2 &= \left(B^\top B + \frac{1}{d_1}A^\top A\right)^{-1} B^\top e_1. \end{aligned} \tag{7}$$

The label of a new example x is determined by the minimum of $|x^\top w_r + b_r|$ $(r = 1, 2)$ which are the perpendicular distances of x to the two hyperplanes given in (1).

2.2 MvLapTSVM

In this part, we introduce MvLapTSVM [33]. MvLapTSVM combines two views by introducing the constraint of similarity between two one-dimensional projections identifying two distinct TSVMs from two feature spaces. Here on view 1, positive examples are represented by $A_1^{'}$ and negative examples are represented by $B_1^{'}$. On view 2, positive examples are represented by $A_2^{'}$ and negative examples are represented by $B_2^{'}$. The optimization problems of linear MvLapTSVM can be written as

$$\begin{aligned} \min_{w_1,w_2,b_1,b_2,q_1,q_2,\eta} \quad & \frac{1}{2}\|A_1^{'}w_1 + e_1b_1\|^2 \\ & + \frac{1}{2}\|A_2^{'}w_2 + e_1b_2\|^2 + c_1e_2^\top q_1 + c_2e_2^\top q_2 \\ & + \frac{1}{2}c_3(\|w_1\|^2 + b_1^2 + \|w_2\|^2 + b_2^2) \\ & + \frac{1}{2}c_4[(w_1^\top M_1^{'\top} + e^\top b_1)L_1(M_1^{'}w_1 + eb_1) \\ & + (w_2^\top M_2^{'\top} + e^\top b_2)L_2(M_2^{'}w_2 + eb_2)] + De_1^\top\eta \\ \text{s.t.} \quad & |A_1^{'}w_1 + e_1b_1 - A_2^{'}w_2 - e_1b_2| \preceq \eta, \\ & - B_1^{'}w_1 - e_2b_1 + q_1 \succeq e_2, \\ & - B_2^{'}w_2 - e_2b_2 + q_2 \succeq e_2, \\ & q_1 \succeq 0,\ q_2 \succeq 0, \\ & \eta \succeq 0, \end{aligned} \tag{8}$$

$$\begin{aligned} \min_{w_3,w_4,b_3,b_4,q_3,q_4,\zeta} \quad & \frac{1}{2}\|B_1^{'}w_3 + e_2b_3\|^2 \\ & + \frac{1}{2}\|B_2^{'}w_4 + e_2b_4\|^2 + c_1e_1^\top q_3 + c_2e_1^\top q_4 \\ & + \frac{1}{2}c_3(\|w_3\|^2 + b_3^2 + \|w_4\|^2 + b_4^2) \\ & + \frac{1}{2}c_4[(w_3^\top M_1^{'\top} + e^\top b_3)L_1(M_1^{'}w_3 + eb_3) \\ & + (w_4^\top M_2^{'\top} + e^\top b_4)L_2(M_2^{'}w_4 + eb_4)] + He_2^\top\zeta \\ \text{s.t.} \quad & |B_1^{'}w_3 + e_2b_3 - B_2^{'}w_4 - e_2b_4| \preceq \zeta, \\ & - A_1^{'}w_3 - e_1b_3 + q_3 \succeq e_1, \\ & - A_2^{'}w_4 - e_1b_4 + q_4 \succeq e_1, \\ & q_3 \succeq 0,\ q_4 \succeq 0, \\ & \zeta \succeq 0, \end{aligned} \tag{9}$$

where $\|\cdot\|$ indicates the ℓ_2-norm, $M_1^{'}$ includes all of labeled data and unlabeled data from view 1. $M_2^{'}$ includes all of labeled data and unlabeled data from view 2. L_1 is the graph Laplacian of view 1 and L_2 is the graph Laplacian of view 2. w_1, b_1, w_2, b_2, w_3, b_3, w_4, b_4 are classifier parameters. c_1, c_2, c_3, c_4, D and H are nonnegative parameters. q_1, q_2, q_3, q_4, η and ζ are slack vectors of appropriate dimensions.

Define

$$A_1 = (A_1^{'},\ e_1),\ A_2 = (A_2^{'},\ e_1),\ B_1 = (B_1^{'},\ e_2),\ B_2 = (B_2^{'},\ e_2), \\ J_1 = (M_1^{'},\ e),\ J_2 = (M_2^{'},\ e),\ v_1 = \begin{pmatrix} w_1 \\ b_1 \end{pmatrix},\ v_2 = \begin{pmatrix} w_2 \\ b_2 \end{pmatrix}. \tag{10}$$

Therefore, the dual optimization formulation is

$$\begin{aligned}\min_{\xi_1,\xi_2,\alpha_1,\alpha_2}\ &\frac{1}{2}\xi_1^\top(A_1^\top A_1+c_3I+c_4J_1^\top L_1J_1)^{-1}\xi_1\\&+\frac{1}{2}\xi_2^\top(A_2^\top A_2+c_3I+c_4J_2^\top L_2J_2)^{-1}\xi_2\\&-(\alpha_1+\alpha_2)^\top e_2\\\text{s.t. }&\xi_1=A_1^\top(\beta_2-\beta_1)-B_1^\top\alpha_1,\\&\xi_2=A_2^\top(\beta_1-\beta_2)-B_2^\top\alpha_2,\\&0\le\beta_1,\beta_2,\beta_1+\beta_2\le De_1,\\&0\le\alpha_{1/2}\le c_{1/2}e_2,\end{aligned}\tag{11}$$

$$\begin{aligned}\min_{\rho_1,\rho_2,\omega_1,\omega_2}\ &\frac{1}{2}\rho_1^\top(B_1^\top B_1+c_3I+c_4J_1^\top L_1J_1)^{-1}\rho_1\\&+\frac{1}{2}\rho_2^\top(B_2^\top B_2+c_3I+c_4J_2^\top L_2J_2)^{-1}\rho_2\\&-(\omega_1+\omega_2)^\top e_1\\\text{s.t. }&\rho_1=B_1^\top(\gamma_2-\gamma_1)-A_1^\top\omega_1,\\&\rho_2=B_2^\top(\gamma_1-\gamma_2)-A_2^\top\omega_2,\\&0\le\gamma_1,\gamma_2,\gamma_1+\gamma_2\le He_2,\\&0\le\omega_{1/2}\le c_{1/2}e_1,\end{aligned}\tag{12}$$

where $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma_1, \gamma_2, \omega_1$ and ω_2 are the vectors of nonnegative Lagrange multipliers.

$$v_1=(A_1^\top A_1+c_3I+c_4J_1^\top L_1J_1)^{-1}[A_1^\top(\beta_2-\beta_1)-B_1^\top\alpha_1],\tag{13}$$

$$v_2=(A_2^\top A_2+c_3I+c_4J_2^\top L_2J_2)^{-1}[A_2^\top(\beta_1-\beta_2)-B_2^\top\alpha_2].\tag{14}$$

The augmented vectors $u_1=\begin{pmatrix}w_3\\b_3\end{pmatrix}$, $u_2=\begin{pmatrix}w_4\\b_4\end{pmatrix}$ are given by

$$u_1=(B_1^\top B_1+c_3I+c_4J_1^\top L_1J_1)^{-1}[B_1^\top(\gamma_2-\gamma_1)-A_1^\top\omega_1],\tag{15}$$

$$u_2=(B_2^\top B_2+c_3I+c_4J_2^\top L_2J_2)^{-1}[B_2^\top(\gamma_1-\gamma_2)-A_2^\top\omega_2].\tag{16}$$

For an example x with x_1' and x_2', if $\frac{1}{2}(|x_1^\top v_1|+|x_2^\top v_2|)\le\frac{1}{2}(|x_1^\top u_1|+|x_2^\top u_2|)$, where $x_1=(x_1',1)$ and $x_2=(x_2',1)$, it is classified to class +1, otherwise class −1.

2.3 MPGR

In this section, we briefly introduce the manifold-preserving graph reduction algorithm [41].

MPGR is an efficient graph reduction algorithm based on the manifold assumption. A sparse graph with manifold-preserving properties means that a point outside of it should have a high connectivity with a point to be reserved. Suppose there is a graph G composed of m vertices and the sparsity level or the number of vertices in the desired sparse graphs. The manifold-preserving sparse graphs G_s are those sparse graph candidates G_c which have a high space connectivity with G. The value of space connectivity is as follows:

$$\frac{1}{m-s}\sum_{i=s+1}^{m}\left(\max_{j=1,\ldots,s}W_{ij}\right),\tag{17}$$

where s is the number of vertices to be retained, and W is the weight matrix of G. For subset selection, a point which is closer to surrounding points should be selected since it contains more important information. This conforms to MPGR in which the point with a large degree will be preferred. The degree $d(p)$ is defined as

$$d(p)=\sum_{p-q}w_{pq},\tag{18}$$

where $p-q$ means that the point p is connected with the point q and w_{pq} is their corresponding weight. If two vertices are not linked, their weight would be zero. Due to its simplicity, $d(p)$ is generally considered as a criterion to construct sparse graphs. A bigger $d(p)$ means the point p contains more information. Namely, the point p is more likely to be selected into the sparse graphs. In a word, the subset constructed by MPGR is high representative and maintains a good global manifold structure of the original data distribution. Suppose that E is the maximum number of edges linked to a point in the original graph. The time complexity of the algorithm is less than $O(Ems)$. This can eliminate the outlier and noise examples, and enhance the robustness of the algorithm.

3 Our proposed method

3.1 MvLapLSTSVM with MPGR

Now we introduce our methods multi-view Laplacian least squares twin support vector machine and its improved version with MPGR. We begin to construct two optimization problems for linear MvLapLSTSVM which can be written as

$$\begin{aligned}\min_{v_1,v_2,q_1,q_2}\ &\frac{1}{2}\|A_1v_1\|^2+\frac{1}{2}\|A_2v_2\|^2+\frac{D}{2}\|A_1v_1-A_2v_2\|^2\\&+c_1(q_1^\top q_1+q_2^\top q_2)+\frac{c_2}{2}(v_1^\top J_1^\top L_1J_1v_1+v_2^\top J_2^\top L_2J_2v_2)\\\text{s.t. }&-B_1v_1+q_1=e_2,\\&-B_2v_2+q_2=e_2,\end{aligned}\tag{19}$$

$$\begin{aligned}\min_{u_1,u_2,k_1,k_2} \ & \frac{1}{2}\|B_1u_1\|^2+\frac{1}{2}\|B_2u_2\|^2+\frac{H}{2}\|B_1u_1-B_2u_2\|^2\\ & +d_1(k_1^\top k_1+k_2^\top k_2)\\ & +\frac{d_2}{2}(u_1^\top J_1^\top L_1J_1u_1+u_2^\top J_2^\top L_2J_2u_2)\\ \text{s.t.} \ & -A_1u_1+k_1=e_1,\\ & -A_2v_2+k_2=e_1,\end{aligned} \tag{20}$$

where $\|\cdot\|$ indicates the ℓ_2-norm, v_1, v_2, u_1, u_2 are classifier parameters. c_1, c_2, d_1, d_2, D, H are nonnegative parameters and q_1, q_2, k_1, k_2 are slack vectors of appropriate dimensions. The term $\frac{1}{2}\|A_1v_1\|^2+\frac{1}{2}\|A_2v_2\|^2$ and constraints $-B_1v_1+q_1=e_2$, $-B_2v_2+q_2=e_2$ aim to make $+1$ class hyperplanes of two views closest to the corresponding $+1$ class and as far as possible from the corresponding -1 class simultaneously. The term $\frac{1}{2}\|B_1u_1\|_2^2+\frac{1}{2}\|B_2u_2\|^2$ and constraints $-A_1u_1+k_1=e_1$, $-A_2v_2+k_2=e_1$ aim to make -1 class hyperplanes of two views closest to the corresponding -1 class and as far as possible from the corresponding $+1$ class simultaneously. The multi-view regularization term $\frac{D}{2}\|A_1v_1-A_2v_2\|^2$ and $\frac{H}{2}\|B_1u_1-B_2u_2\|^2$ can be understood as minimizing the difference of function values between two views. The last terms $\frac{c_2}{2}(v_1^\top J_1^\top L_1J_1v_1+v_2^\top J_2^\top L_2J_2v_2)$ and $\frac{d_2}{2}(u_1^\top J_1^\top L_1J_1u_1+u_2^\top J_2^\top L_2J_2u_2)$ are manifold regularization terms which are similar to the ones of MvLapTSVM.

The above optimization problems can be written as another form

$$\begin{aligned}\min_{v_1,v_2,q_1,q_2} \ & \frac{1}{2}\begin{pmatrix}v_1\\v_2\end{pmatrix}^\top\begin{pmatrix}(1+D)A_1^\top A_1+c_2J_1^\top L_1J_1 & -DA_1^\top A_2\\ -DA_2^\top A_1 & (1+D)A_2^\top A_2+c_2J_2^\top L_2J_2\end{pmatrix}\begin{pmatrix}v_1\\v_2\end{pmatrix}\\ & +c_1\begin{pmatrix}q_1\\q_2\end{pmatrix}^\top\begin{pmatrix}q_1\\q_2\end{pmatrix}\\ \text{s.t.} \ & -\begin{pmatrix}B_1 & 0\\ 0 & B_2\end{pmatrix}\begin{pmatrix}v_1\\v_2\end{pmatrix}+\begin{pmatrix}q_1\\q_2\end{pmatrix}=e_2,\end{aligned} \tag{21}$$

$$\begin{aligned}\min_{u_1,u_2,k_1,k_2} \ & \frac{1}{2}\begin{pmatrix}u_1\\u_2\end{pmatrix}^\top\begin{pmatrix}(1+H)B_1^\top B_1+d_2J_1^\top L_1J_1 & -HB_1^\top B_2\\ -HB_2^\top B_1 & (1+H)B_2^\top B_2+d_2J_2^\top L_2J_2\end{pmatrix}\begin{pmatrix}u_1\\u_2\end{pmatrix}\\ & +d_1\begin{pmatrix}k_1\\k_2\end{pmatrix}^\top\begin{pmatrix}k_1\\k_2\end{pmatrix}\\ \text{s.t.} \ & -\begin{pmatrix}A_1 & 0\\ 0 & A_2\end{pmatrix}\begin{pmatrix}u_1\\u_2\end{pmatrix}+\begin{pmatrix}k_1\\k_2\end{pmatrix}=e_1.\end{aligned} \tag{22}$$

Let

$$\begin{aligned}& v=\begin{pmatrix}v_1\\v_2\end{pmatrix}, u=\begin{pmatrix}u_1\\u_2\end{pmatrix},\\ & E=\begin{pmatrix}(1+D)A_1^\top A_1+c_2J_1^\top L_1J_1 & -DA_1^\top A_2\\ -DA_2^\top A_1 & (1+D)A_2^\top A_2+c_2J_2^\top L_2J_2\end{pmatrix}, \quad F=\begin{pmatrix}B_1 & 0\\ 0 & B_2\end{pmatrix},\\ & M=\begin{pmatrix}(1+H)B_1^\top B_1+d_2J_1^\top L_1J_1 & -HB_1^\top B_2\\ -HB_2^\top B_1 & (1+H)B_2^\top B_2+d_2J_1^\top L_1J_1\end{pmatrix}, \quad N=\begin{pmatrix}A_1 & 0\\ 0 & A_2\end{pmatrix},\\ & q=\begin{pmatrix}q_1\\q_2\end{pmatrix}, k=\begin{pmatrix}k_1\\k_2\end{pmatrix}.\end{aligned} \tag{23}$$

They can be written as

$$\begin{aligned}\min_{v,q} \ & \tfrac{1}{2}v^\top Ev+\tfrac{c_1}{2}q^\top q+c_3||v||^2\\ \text{s.t.} \ & -Fv+q=e_2,\end{aligned} \tag{24}$$

$$\begin{aligned}\min_{u,k} \ & \tfrac{1}{2}u^\top Mu+\tfrac{d_1}{2}k^\top k+d_3||u||^2\\ \text{s.t.} \ & -Nu+k=e_1.\end{aligned} \tag{25}$$

Simultaneously, we draw the structural risk minimization into the above optimization problems. The Lagrangian of the optimization problem (24) is given by

$$L = \frac{1}{2}v^\top Ev + \frac{c_1}{2}(Fv + e)^\top(Fv + e) + c_3||v||^2. \tag{26}$$

We take partial derivatives of the above equation and let them be zero

$$\frac{\partial L}{\partial v} = Ev + c_1F^\top Fv + c_1F^\top e + c_3v = 0, \tag{27}$$

From the above equations, we obtain

$$v = -\left(\frac{E}{c_1} + F^\top F + \frac{c_3}{c_1}I\right)^{-1} F^\top e_2. \tag{28}$$

Applying the same techniques to (25), we obtain

$$u = -\left(\frac{M}{d_1} + N^\top N + \frac{d_3}{d_1}I\right)^{-1} N^\top e_1. \tag{29}$$

Then we integrate the MPGR to select two sparse subsets of the unlabeled examples corresponding to the two views, respectively. Then a new sparse subset is formed by the intersection of the two sparse subsets. The procedure of the whole algorithm is in Algorithm 1.

Algorithm 1 Multi-view Laplacian Least Squares Twin Support Vector Machines with Manifold-preserving Graph Reduction Algorithm

```
Input: labeled two view data $l_1$, $l_2$, and unlabeled two view data $u_1$, $u_2$, model parameters ($c_1$, $c_2$, $c_3$, $d_1$, $d_2$, $d_3$, $D$,$H$).
Use the MPGR on unlabeled data of view 1 $u_1$ to generate the sparse subset $T_1$ and use the MPGR on unlabeled data of view 2 $u_2$ to generate the sparse subset $T_2$.
Obtain the common sparse subset $T$ which is the intersection of $T_1$ and $T_2$.
Obtain $E$, $M$, $F$ and $N$ using (23).
Determine parameters of two hyperplanes for each view by solving the linear equation (28) and (29), respectively.
Output: For a test example $x = (x_1^\top\ 1\ x_2^\top\ 1)^\top$, if $|x^\top v| \le |x^\top u|$, it is classified to class +1, otherwise class −1.
```

3.2 Kernel MvLapLSTSVM with MPGR

In this part, we extend MvLapLSTSVM with MPGR to the nonlinear case. The kernel-generated hyperplanes are:

Table 1 Datasets

Name	Attributes	Instances	Classes
Ionosphere	34	351	2
Handwritten digits	649	2000	10
Caltech-101	3766	9146	101
Corel	89	68,040	99

$$\begin{aligned} &K\{x_1^\top, C_1^\top\}w_1 + b_1 = 0, \quad K\{x_2^\top, C_2^\top\}w_2 + b_2 = 0,\\ &K\{x_1^\top, C_1^\top\}w_3 + b_3 = 0, \quad K\{x_2^\top, C_2^\top\}w_4 + b_4 = 0, \end{aligned} \tag{30}$$

where K is a chosen kernel function which is defined by $K\{x_i, x_j\} = (\Phi(x_i), \Phi(x_j))$. $\Phi(\cdot)$ is a nonlinear mapping from a low-dimensional feature space to a high-dimensional feature space. C_1 and C_2 denote training examples from view 1 and training examples from view 2 respectively, that is, $C_1 = (A_1^{'\top}, B_1^{'\top})^\top, C_2 = (A_2^{'\top}, B_2^{'\top})^\top$. We define

$$\begin{aligned} &G_1 = (K\{A_1^{'}, C_1^\top\},\ e), H_1 = (K\{B_1^{'}, C_1^\top\},\ e),\\ &G_2 = (K\{A_2^{'}, C_2^\top\},\ e), H_2 = (K\{B_2^{'}, C_2^\top\},\ e),\\ &v_1 = \begin{pmatrix} w_1 \\ b_1 \end{pmatrix},\ v_2 = \begin{pmatrix} w_2 \\ b_2 \end{pmatrix},\ u_1 = \begin{pmatrix} w_3 \\ b_3 \end{pmatrix},\ u_2 = \begin{pmatrix} w_4 \\ b_4 \end{pmatrix}. \end{aligned} \tag{31}$$

Then the optimization problems for kernel MvLapLSTSVM with MPGR are written as same as the linear MvLapLSTSVM with MPGR

$$\begin{aligned} \min_{v_1,v_2,q_1,q_2}\ & \tfrac{1}{2}\|G_1v_1\|^2 + \tfrac{1}{2}\|G_2v_2\|^2 + \tfrac{D}{2}\|G_1v_1 - G_2v_2\|^2 \\ & + c_1(q_1^\top q_1 + q_2^\top q_2) + c_2(v_1^\top J_1^\top L_1J_1v_1 + v_2^\top J_2^\top L_2J_2v_2) \\ \text{s.t.}\ & -H_1v_1 + q_1 = e_2, \\ & -H_2v_2 + q_2 = e_2, \end{aligned} \tag{32}$$

$$\begin{aligned} \min_{u_1,u_2,k_1,k_2}\ & \tfrac{1}{2}\|H_1u_1\|^2 + \tfrac{1}{2}\|H_2u_2\|^2 + \tfrac{H}{2}\|H_1u_1 - H_2u_2\|^2 \\ & + d_1(k_1^\top k_1 + k_2^\top k_2) + d_2(u_1^\top J_1^\top L_1J_1u_1 + u_2^\top J_2^\top L_2J_2u_2) \\ \text{s.t.}\ & -G_1u_1 + k_1 = e_1, \\ & -G_2v_2 + k_2 = e_1, \end{aligned} \tag{33}$$

where v_1, v_2, u_1, u_2 are classifier parameters. c_1, c_2, d_1, d_2, D, H are nonnegative parameters, and q_1, q_2, k_1, k_2 are slack vectors of appropriate dimensions. Then we integrate the MPGR to select two sparse subsets of the unlabeled examples corresponding to the two views.

Suppose a new example x has two views x_1 and x_2. If $\frac{1}{2}|K\{x_1^\top, C_1^\top\}w_1 + b_1 + K\{x_2^\top, C_2^\top\}w_2 + b_2| \le \frac{1}{2}|K\{x_1^\top, C_1^\top\}w_3$ $+b_3 + K\{x_2^\top, C_2^\top\}w_4 + b_4|$, it is classified to class +1, otherwise class −1.

Above all, MvLapLSTSVM with MPGR needs to solve two small-scale linear equations and also contains the time complexity $O(Emu)$ of MPGR while MvLapTSVM solves a pairs of QPPs (time complexity $O((u + l)^3)$, u is the number

Table 2 Classification performance and running time (%,s) on ionosphere

Percentage	Method			
	MvLapSVM	MvLapTSVM	MvLapLSTSVM	MvLapLSTSVM with MPGR
10	82.35(6.50) (2.67*10^{3})	83.81(7.08) (3.44*10^{3})	**85.88**(4.68) (3.76*10^{2})	**85.88**(4.68) (3.14*10^{2})
20	83.14(7.02) (4.11*10^{3})	84.71(4.47) (4.51*10^{3})	**85.88**(4.68) (5.41*10^{2})	**85.88**(5.48) (5.40*10^{2})
30	83.53(4.92) (1.11*10^{4})	83.92(4.25) (4.33*10^{3})	84.31(6.04) (4.10*10^{2})	**85.10**(5.30) (6.06*10^{2})
40	82.75(5.08) (1.50*10^{4})	84.71(2.15) (4.44*10^{3})	84.31(6.04) (1.13*10^{3})	**85.88**(4.88) (1.04*10^{3})
50	82.75(8.01) (2.08*10^{4})	85.88(4.88) (2.61*10^{3})	**86.67**(3.77) (7.52*10^{2})	85.10(5.30) (7.34*10^{2})
60	82.35(6.93) (1.22*10^{4})	**86.67**(1.64) (2.63*10^{3})	**86.67**(5.44) (2.90*10^{2})	86.27(6.20) (6.49*10^{2})
70	84.71(7.12) (3.79*10^{3})	**85.49**(3.56) (3.14*10^{3})	**85.49**(2.97) (1.26*10^{3})	85.10(6.88) (1.50*10^{3})
80	81.18(6.59) (1.48*10^{4})	83.53(6.88) (3.16*10^{3})	83.92(3.77) (6.64*10^{2})	**85.88**(6.11) (7.38*10^{2})
90	81.18(6.59) (1.50*10^{4})	84.71(3.77) (3.16*10^{3})	83.92(4.88) (1.03*10^{3})	**85.49**(5.30) (9.84*10^{2})
100	81.57(6.29)(3.17*10^{4})	85.49(5.11) (4.69*10^{3})	**85.88**(3.22) (1.02*10^{3})	**85.88**(4.70) (7.06*10^{2})
Method	SVM-2K	MvTSVM	AMVMED	PSVM-2V
Accuracy	82.35(6.93) (1.37*10^{4})	78.04(9.55) (4.43*10^{3})	82.35(4.80) (1.09*10^{3})	79.22(9.96) (3.67*10^{4})

Bold values indicate the best accuracy

of unlabeled dataset and l is the number of labeled dataset) and MvLapSVM solves a large QPP (time complexity $O((u+l)^3)$). Thus MvLapLSTSVM with MPGR can reduce the time complexity largely.

4 Experiments

In this section, we evaluate our proposed methods MvLapLSTSVM with MPGR and MvLapLSTSVM on four real-world datasets. Three datasets ionosphere, handwritten digits and corel come from UCI Machine Learning Repository[1]. Another one is caltech-101[2] [42]. Specific information about the four datasets is listed in Table 1.

4.1 Dataset description and experimental setting

- *Ionosphere* The dataset incorporates 351 examples (225 positive examples and 126 negative examples). The positive examples are those radar returns which show some type of structure in the ionosphere while the negative examples are those that do not and their signals pass through the ionosphere.

In our experimental setting, we design the original data and the resultant data by PCA as two views. We adopt ten-fold cross-validation to choose the optimal parameters in the area $[2^{-7}, 2^{7}]$ with exponential growth 1 and get the results by operating these methods for five times. We divide the whole dataset to 300 training examples and 51 test examples. We select 80 examples as labeled examples and the others as unlabeled examples in the 300 examples. We set the input number of MPGR from 10% to 100% of 220 unlabeled examples for each view and intersect the sparse subsets of two views. Linear kernel is chosen for the dataset. Multi-view semi-supervised learning methods MvLapSVM and MvLapTSVM, and multi-view supervised learning methods SVM-2K, MvTSVM, PSVM-2V and AMVMED are used for comparison.

- *Handwritten digits* This dataset contains features of handwritten digits ($0 \sim 9$) extracted from a collection of Dutch utility maps. It contains 2000 examples (200 examples per class) with view 1 being the 76 Fourier coefficients and view 2 being the 64 Karhunen-Love coefficients of each image. Because our proposed methods are designed for multi-view binary classification while handwritten digit dataset contains 10 classes. We choose three pairs (0,8), (3,5) and (2,7) to evaluate all involved methods for the experiment. We use 50 examples as label examples and 150 unlabel examples, and 200 examples for testing. We adopt ten-fold cross-validation to choose the optimal parameters in the area $[2^{-10}, 2^{10}]$ with exponential growth 1. We set the input number of MPGR as 150 for each view and intersect the sparse subsets of two views.
- *Caltech-101* The caltech-101 dataset contains 9146 images in total which owns 101 different object categories, as well as an additional background/clutter category. Each object category contains between 40 and 800 images on average. Among them, we randomly select two classes named 'BACKGROUND-Google' and 'Faces' for

[1] https://archive.ics.uci.edu/ml/index.php.

[2] http://www.vision.caltech.edu/Image_Datasets/Caltech101.

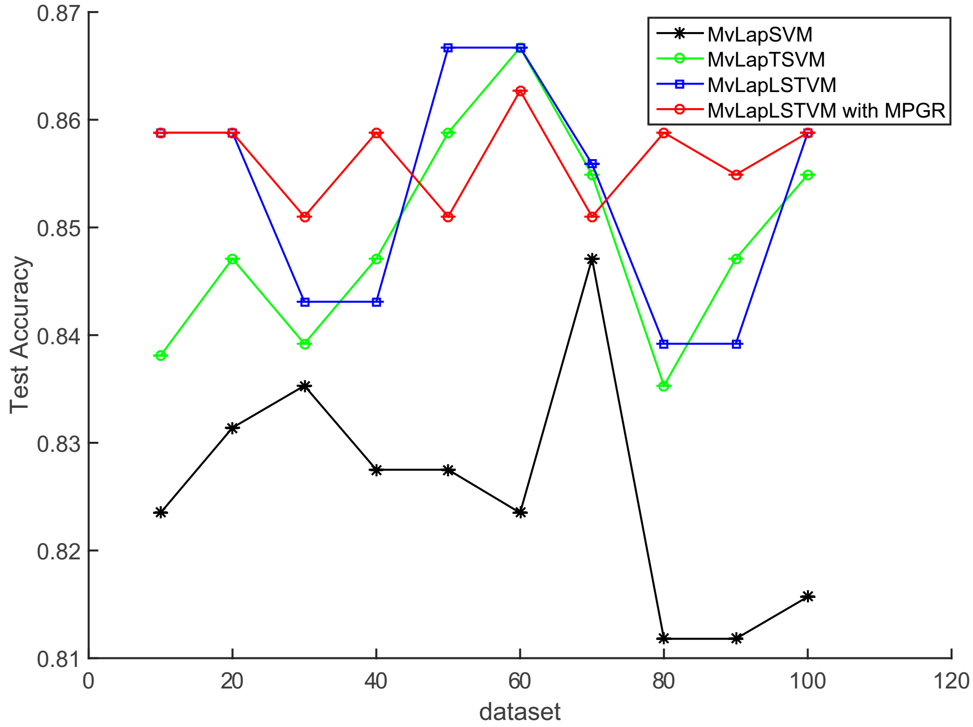

Fig. 1 Classification accuracies on ionosphere

multi-view binary classification. We adopt 80 examples as label examples and 520 unlabel examples, and 302 examples for testing. We adopt ten-fold cross-validation to choose the optimal parameters in the area $[2^{-5}, 2^{5}]$ with exponential growth 1. We set the input number of MPGR as 250 for each view and intersect the sparse subsets of two views. The other experimental setting is as same as the above experiment.

Table 3 Classification performance and running time (%,s) on handwritten digits

Method	Digit pair		
	(0,8)	(2,7)	(3,5)
SVM-2K	97.70(2.25)	97.90(3.03)	94.10(1.52)
MvTSVM	98.50(0.01)	98.90(0.74)	94.80(0.67)
PSVM-2V	98.30(1.60)	98.70(0.27)	**95.30**(2.20)
AMVMED	97.70(1.20)	98.60(0.74)	93.80(3.40)
MvLapSVM	97.00(1.46)	96.50(2.21)	95.10(3.23)
MvLapTSVM	98.30(3.25)	98.00(1.90)	94.10(0.35)
MvLapLSTSVM	98.60(0.82)	98.50(0.61)	94.80(2.20)
MvLapLSTSVM with MPGR	**99.00**(0.71)	**99.10**(0.65)	95.10(1.98)
SVM-2K	$3.54*10^{4}$	$3.25*10^{4}$	$3.88*10^{4}$
MvTSVM	$1.05*10^{4}$	$7.22*10^{3}$	$7.90*10^{3}$
PSVM-2V	$7.62*10^{4}$	$7.34*10^{4}$	$7.60*10^{4}$
AMVMED	$1.66*10^{3}$	$1.32*10^{3}$	$1.95*10^{3}$
MvLapSVM	$6.76*10^{3}$	$1.04*10^{4}$	$6.81*10^{3}$
MvLapTSVM	$3.75*10^{4}$	$3.75*10^{4}$	$3.74*10^{4}$
MvLapLSTSVM	$2.56*10^{3}$	$2.36*10^{3}$	$2.80*10^{3}$
MvLapLSTSVM with MPGR	$1.89*10^{3}$	$2.20*10^{3}$	$2.20*10^{3}$

Bold values indicate the best accuracy

- *Corel* This dataset has four views from a corel image collection. We choose two classes and two views for multi-view binary classification. We use 40 examples as label examples and 120 unlabel examples, and 62 examples for testing. We adopt 10-fold cross-validation to choose the optimal parameters in the area $[2^{-5}, 2^{5}]$ with exponential growth 1. We set the input number of MPGR as 120 for each view and intersect the sparse subsets of two views. The other experimental setting is as same as above experiments.

4.2 Experimental analysis

From the experimental results in Tables 2, 3 and Figs. 1, 2, we can find that our method MvLapLSTSVM with MPGR performs better than our method MvLapLSTSVM, MvLapSVM and MvLapTSVM in most cases. Our methods MvLapLSTSVM with MPGR and MvLapLSTSVM perform far better than SVM-2K, MvTSVM, PSVM-2V and AMVMED. However, PSVM-2V performs best in the pair (3,5). The performance of MvLapLSTSVM is just a

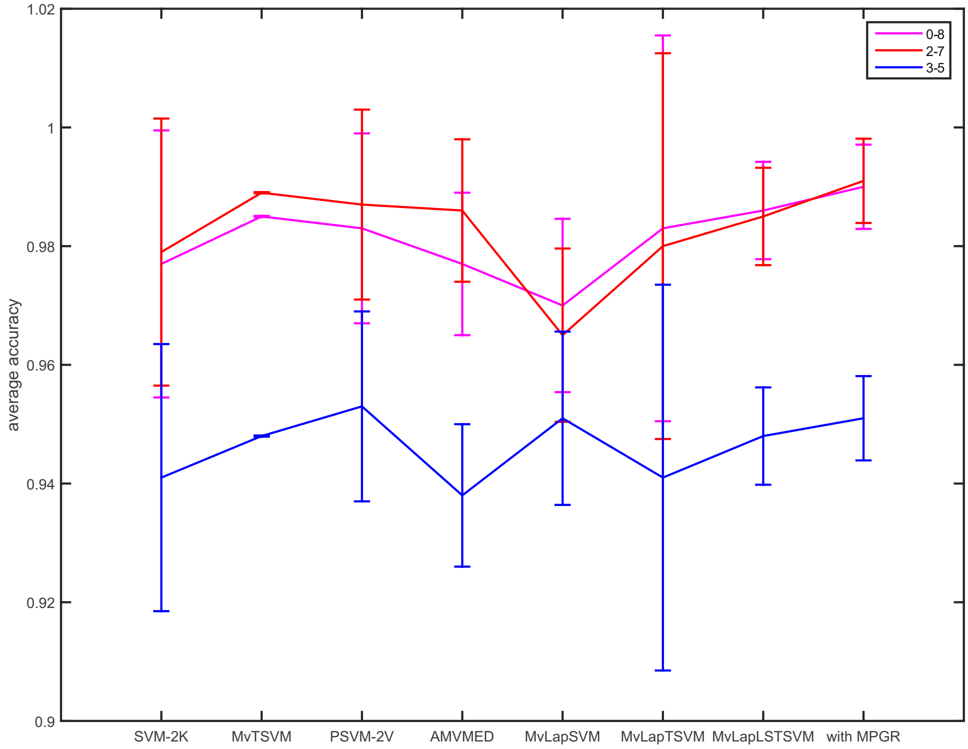

Fig. 2 Classification accuracies on handwritten digits

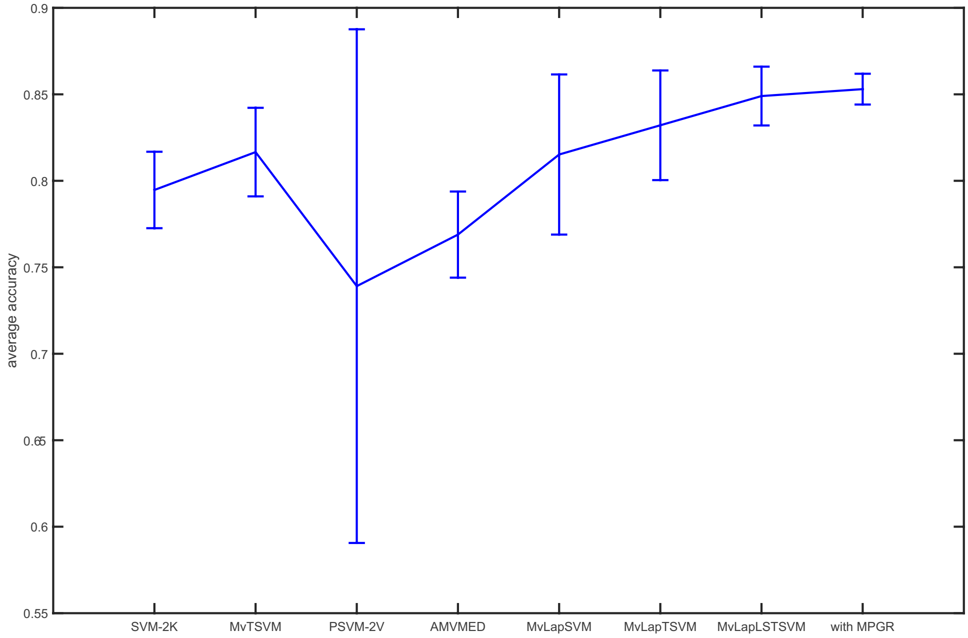

Fig. 3 Classification accuracies on caltech-101

little worse than the one of MvLapSVM and MvLapLSTSVM with MPGR in the pair (3,5). The reason may be that multi-view learning methods based on SVM MvLapSVM and PSVM-2V can effectively leverage the information of two views in the pair (3,5).

From the experimental results in Tables 4, 5 and Figs. 3, 4, we can find that MvLapLSTSVM with MPGR is superior to all the other methods. Our method MvLapLSTSVM peforms a little worse than MvLapLSTSVM with MPGR. From all experimental results, we find that the running time of MvLapLSTSVM and MvLapLSTSVM with MPGR is less than the one of MvLapSVM, MvLapTSVM, SVM-2K, MvTSVM and close to the one of AMVMED.

In short, we can conclude that our methods MvLapLSTSVM with MPGR and MvLapLSTSVM perform better than MvLapSVM, MvLapTSVM, SVM-2K, MvTSVM,

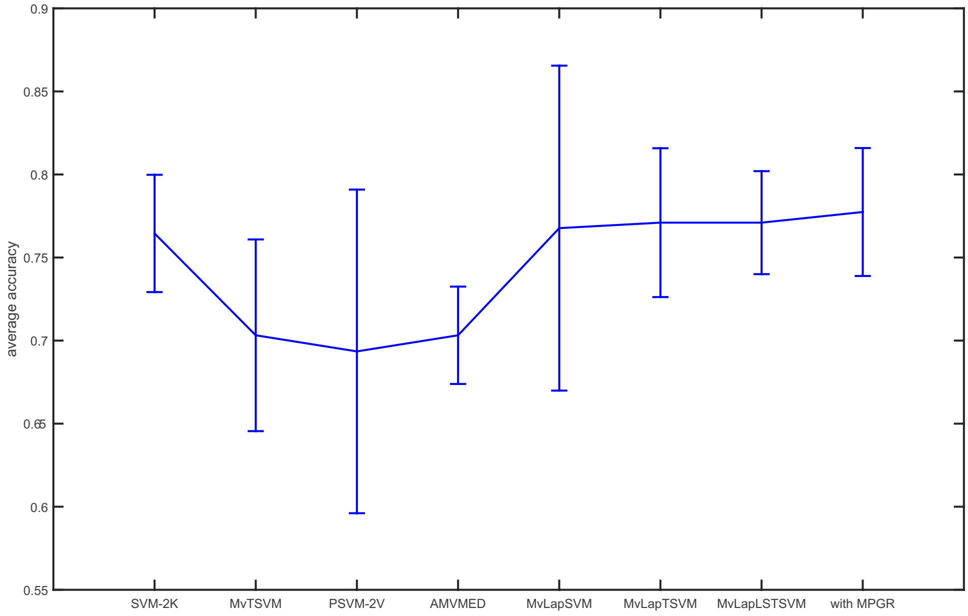

Fig. 4 Classification accuracies on corel

Table 4 Classification performance and running time (%,s) on caltech-101

Method	SVM-2K	MvTSVM	AMVMED	PSVM-2V
Accuracy	79.47(2.21) $6.37*10^3$	81.66(2.56) $1.59*10^3$	73.91(14.85) $8.82*10^2$	76.89(2.49) $1.60*10^4$
Method	MvLapSVM	MvLapTSVM	MvLapLSTSVM	MvLapLSTSVM with MPGR
Accuracy	81.52(4.63) $1.63*10^4$	83.21(3.17) $2.78*10^4$	84.90(1.70) $1.16*10^4$	**85.30**(0.89) $6.07*10^3$

Bold values indicate the best accuracy

Table 5 Classification performance and running time (%,s) on corel

Method	SVM-2K	MvTSVM	AMVMED	PSVM-2V
Accuracy	76.45(3.53) $4.79*10^3$	70.32(5.77) $3.42*10^3$	69.35(9.74) $7.94*10^2$	70.32(2.93) $1.11*10^4$
Method	MvLapSVM	MvLapTSVM	MvLapLSTSVM	MvLapLSTSVM with MPGR
Accuracy	76.77(9.78) $6.42*10^2$	77.10(4.48) $1.52*10^3$	77.10(3.10) $1.83*10^2$	**77.74**(3.85) $1.81*10^2$

Bold values indicate the best accuracy

PSVM-2V and AMVMED for almost all the cases. They own the less time complexity. MvLapLSTSVM with MPGR can eliminate outliers and noisy examples in unlabeled examples so can enhance the robustness of the relevant algorithm MvLapLSTSVM.

5 Conclusion and future work

In this paper, two novel multi-view semi-supervised support vector machines were proposed. MvLapLSTSVM and MvLapLSTSVM with MPGR combine two views by introducing multi-view co-regularization terms. They leverage the manifold regularization to the semi-supervised learning. They can reduce time complexity by solving a pair of linear equation problems. MvLapLSTSVM with MPGR uses the manifold-preserving graph reduction to select a representative and informative unlabeled subset, and can improve the robustness of the relevant algorithm MvLapLSTSVM. Experimental results on multiple real-world datasets indicate that MvLapLSTSVM with MPGR is superior to SVM-2K, AMVMED, PSVM-2V, MvTSVM, MvLapSVM, MvLapTSVM and our method MvLapLSTSVM. Our method MvLapLSTSVM is next to MvLapLSTSVM with MPGR. It would be interesting for future work to exploit other ways which select the informative and representative subsets from unlabeled examples to multi-view semi-supervised learning.

Acknowledgements This work is supported by Ningbo University talent project 421703670 as well as programs sponsored by K.C. Wong Magna Fund in Ningbo University. It is also supported by the Research Foundation of Education Department of Zhejiang Province under Project Y201635608, and the Zhejiang Provincial Natural Science Foundation of China under Project LQ18F020001, the Natural Science Foundation of Ningbo city of Zhejiang Province of China under Project 2018A610155, the Open Project Program of the State Key Lab of CAD&CG in Zhejiang University under Project A1815 and NSFC 61906101.

References

1. Shawe-Taylor J, Sun S (2011) A review of optimization methodologies in support vector machines. Neurocomputing 74(17):3609–3618
2. Vapnik V (1995) The nature of statistical learning theory. Springer, New York
3. Christianini N, Shawe-Taylor J (2002) An introduction to support vector machines. Cambridge University Press, Cambridge
4. Scholkopf B, Smola A (2003) Learning with kernels. MIT Press, Cambridge
5. Ripley B (1996) Pattern recognition and neural networks. Cambridge University Press, Cambridge
6. Jayadeva R, Khemchandani S Chandra (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 7:905–910
7. Kumar M, Gopal M (2009) Least squares twin support vector machines for pattern classification. Expert Syst Appl 36:7535–7543
8. Gao S, Ye Q, Ye N (2011) 1-Norm least squares twin support vector machines. Neurocomputing 74:3590–3597
9. Mei B, Xu Y (2019) Multi-task least squares twin support vector machine for classification. Neurocomputing 338(2019):26–33. https://doi.org/10.1016/j.neucom.2018.12.079
10. Xu Y, Chen M (2016) Li G (2015) Least squares twin support vector machine with Universum data for classification. Int J Syst Sci 47(15):3637–3645
11. Chapelle O, Schölkopf B, Zien A (2010) Semi-supervised learning. MIT Press, Massachusetts
12. Zhu X (2008) Semi-supervised learning literature survey. Technical Report 1530, Department of Computer Sciences University of Wisconsin Madison
13. Zhu X, Ghahramani Z, Lafferty J (2006) Semi-supervised learning using Gaussian fields and harmonic functions. In: Proceedings of the 20th international conference on machine learning, pp 912–919
14. Zhou Z, Zhan D, Yang Q (2007) Semi-supervised learning with very few labeled training example. In: Proceedings of the 22nd AAAI conference on artificial intelligence, pp 675–680
15. Joachims T (1999) Transductive inference for text classification using support vector machines. In: Proceedings of the 16th international conference on machine learning, pp 200-209
16. Bennett K, Demiriz A (1999) Semi-supervised support vector machines. Adv Neural Inf Process Syst 11:368–374
17. Melacci S, Beklin M (2011) Laplacian support vector machines trained in the primal. J Mach Learn Res 12:1149–1184
18. Qi Z, Tian Y, Shi Y (2012) Laplacian twin support vector machine for semi-supervised classification. Neural Netw 35:46–53
19. Zhao J, Xie X, Xu X, Sun S (2017) Multi-view learning overview: recent progress and new challenges. Inf Fus 38:43–54
20. Sun S, Xie X, Dong C (2019) Multiview learning with generalized eigenvalue proximal support vector machines. IEEE Trans Cybern 49:688–697
21. Yin J, Sun S (2019) Multi-view uncorrelated locality preserving projection. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2019.2944664
22. Xie X, Sun S (2019) Multi-view support vector machines with the consensus and complementary information. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2019.2933511
23. Sun X, Sun S, Yin M, Yang H (2019) Hybrid neural conditional random fields for multi-view sequence labelling. Knowl-Based Syst. https://doi.org/10.1016/j.knosys.2019.105151
24. Blum A, Mitchell T (1998) Combining labeled and unlabeled data with co-training. In: Proceedings of the 11th annual conference on computational learning theory, pp 92–100
25. Nigam K, Ghani R (2000) Analyzing the effectiveness and applicability of co-training. In: Proceedings of the 9th international conference on information and knowledge management, pp 86–93
26. Muslea I, Minton S, Knoblock C (2006) Active learning with multiple views. J Artif Intell Res 27:203–233
27. Sun S, Jin F (2011) Robust co-training. Int J Pattern Recognit Artif Intell 25:1113–1126
28. Farquhar J, Hardoon D, Shawe-Taylor J, Szedmak S (2006) Two view learning: SVM-2K, theory and practice. Adv Neural Inf Process Syst 18:355–362
29. Sun S, Shawe-Taylor J (2010) Sparse semi-supervised learning using conjugate functions. J Mach Learn Res 11:2423–2455
30. Xie X, Sun S (2015) Multi-view twin support vector machines. Intell Data Anal 19:701–712
31. Tang J, Tian Y, Zhang P, Liu X (2018) Multiview privileged support vector machines. IEEE Trans Neural Netw Learn Syst 29:3463–3477
32. Sun S (2011) Multi-view Laplacian support vector machines. Lect Notes Comput Sci 7121:209–222
33. Xie X, Sun S (2014) Multi-view Laplacian twin support vector machines. Appl Intell 41:1059–1068
34. Bartlett P, Mendelson S (2002) Rademacher and Gaussian complexities: risk bounds and structural results. J Mach Learn Res 3:463–482
35. Sun S, Shawe-Taylor J, Mao L (2017) PAC-Bayes analysis of multi-view learning. Inf Fus 35:117–131
36. Sun S, Chao G (2013) Multi-view maximum entropy discrimination. In: Proceedings of the 23rd international joint conference on artificial intelligence, pp 1706–1712
37. Chao G, Sun S (2016) Alternative multi-view maximum entropy discrimination. IEEE Trans Neural Netw Learn Syst 27:1445–1456
38. Mao L, Sun S (2016) Soft margin consistency based scalable multi-view maximum entropy discrimination. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence, pp 1839-1845
39. Chao G, Sun S (2016) Consensus and complementarity based maximun entropy discrimination for multi-view classification. Inf Sci 367:296–310
40. Chao G, Sun S (2018) Semi-supervised multi-view maximum entropy discrimination with expectation Laplacian regularization. Inf Fus 45:296–306
41. Sun S, Hussain Z, Shawe-Taylor J (2014) Manifold-preserving graph reduction for sparse semi-supervised learning. Neurocomputing 124:13–21
42. Li F, Rob F, Pietro P (2007) Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. Comput Vis Image Underst 106:59–70