Journal of Circuits, Systems, and Computers
Vol. 27, No. 12 (2018) 1850185 (22 pages)

DOI: 10.1142/S0218126618501852

World Scientific
www.worldscientific.com

Robust Transductive Support Vector Machine for Multi-View Classification*

Yanchao Li†, Yongli Wang‡, Junlong Zhou§ and Xiaohui Jiang¶

School of Computer Science and Engineering,
Nanjing University of Science and Technology,
Xuanwu District, Nanjing, Jiangsu 210094, P. R. China
†*leeyc.gm@gmail.com*
‡*yongliwang@njust.edu.cn*
§*jlzhou@njust.edu.cn*
¶*jxhchina@gmail.com*

Received 4 August 2017
Accepted 1 January 2018
Published 14 March 2018

Semi-Supervised Learning (SSL) aims to improve the performance of models trained with a small set of labeled data and a large collection of unlabeled data. Learning multi-view representations from different perspectives of data has proved to be very effectively for improving generalization performance. However, existing semi-supervised multi-view learning methods tend to ignore the specific difficulty of different unlabeled examples, such as the outliers and noise, leading to error-prone classification. To address this problem, this paper proposes Robust Transductive Support Vector Machine (RTSVM) that introduces the margin distribution into TSVM, which is robust to the outliers and noise. Specifically, the first-order (margin mean) and second-order statistics (margin variance) are regularized into TSVM, which try to achieve strong generalization performance. Then, we impose a global similarity constraint between distinct RTSVMs each trained from one view of the data. Moreover, our algorithm runs with fast convergence by using concave–convex procedure. Finally, we validate our proposed method on a variety of multi-view datasets, and the experimental results demonstrate that our proposed algorithm is effective. By exploring large amount of unlabeled examples and being robust to the outliers and noise among different views, the generalization performance of our method show the superiority to single-view learning and other semi-supervised multi-view learning methods.

Keywords: Semi-supervised learning; multi-view learning; transductive support vector machines; margin distribution; classification.

*This paper was recommended by Regional Editor Tongquan Wei.
‡Corresponding author.

1. Introduction

In recent years, it is quite easy to get a large number of unlabeled data in many practical tasks. However, labeled ones are fairly expensive because they require human effort. For example, large-scale data (e.g., text, image and video) are uploaded in social networks (i.e., Facebook, Twitter and Weibo), where large amount of data are unlabeled. Therefore, Semi-Supervised Learning (SSL),[1,2] which aims to use both labeled (particularly few) and unlabeled examples to improve model performance. A large number of SSL methods jointly optimize two training objective functions: the supervised loss over labeled data and the unsupervised loss over both labeled and unlabeled data such as graph-based SSL algorithms.[3,4] SSL propagates their limited label information to unlabeled examples and it mainly follows the clustering or manifold assumption. Clustering assumption methods[5,6] assume that the examples of different classes are from several well-separated clusters, and the decision boundary falls into the low density area in the feature space. Most manifold-based methods[2,7] assume that there is a low-dimensional manifold structure embedded in the data space. SSL algorithms have been successfully applied to the fields of natural language process, multimedia and real-time systems.[8,9]

Training data could be represented with multiple views. For example, social media information could be described as text, visual and audio views. Conventional machine learning algorithms, such as support vector machines (SVMs), TSVMs, logistic regression and kernel machines, concatenate all multiple views into one single view to adapt into the learning setting. However, this concatenation usually causes over-fitting in the case of a small size training examples and is not physically meaningful because each view has a special statistical property.[10] Alternatively, multi-view learning methods exploit multiple representations of data from different perspectives to improve performance. Co-training[11] and Canonical Correlation Analysis (CCA)[12] are two representative techniques in early studies of multi-view learning. The main idea of Co-training is that it trains two classifiers separately on two sufficient and redundant views. In addition, Bickel *et al.*[13] and Kumar *et al.*[14] advanced Co-training for data clustering and designed effective algorithms for multi-view data. CCA tries to learn the projections from two-views via maximizing the correlation between them. Besides, many other CCA-based approaches[15–17] have been proposed. Moreover, Multiple Kernel Learning (MKL)[18,19] has been widely applied into multi-view data because kernels in MKL naturally correspond to different views. However, these methods tend to ignore the specific difficulty of different unlabeled examples, such as the outliers and noise, leading to error-prone classification.

In this paper, we try to address the specific difficulty of different unlabeled examples. TSVMs[20–23] are typical SSL approaches and they assume the unlabeled examples as the data to be measured. The basic idea is to determine the margin classification boundary in all the training and testing sets. Compared with the

inductive method, TSVM has achieved good performance, especially for few labeled training sets. Inspired by recent theoretical results[24]: they proved that the margin distribution is more vital to better generalization performance of AdaBoost, rather than maximizing the margin. Later, Zhang and Zhou[25] found that the margin distribution also, played a key role in generalization performance of SVM. There are several research works like Gary and Roth[26] and Pelckmans *et al.*[27] who introduced margin distribution into SVM and proposed the margin distribution optimization algorithms. Consequently, Aiolli *et al.*[28] proposed a kernel method for optimizing the margin distribution. In our method, we introduce the margin distribution into TSVM, which is more robust to outliers and noise (RTSVM). To be specific, the first-order (margin mean) and second-order statistics (margin variance) are regularized into TSVM, which try to achieve strong generalization performance by maximizing the margin mean and minimizing the margin variance simultaneously. Then, we incorporate it into multi-view setting, which impose a global constraint that requires each RTSVM which assigns the same class label to each labeled and unlabeled data.

Motivation. Data from different perspectives could be collected in some applications, such as social networks and multi-source heterogeneous data. However, large amount of these data are unlabeled. Besides, there always exists the outliers and noise in these datasets. Most of the existing semi-supervised multi-view learning methods tend to ignore the difficulty of different unlabeled examples, such as outliers and noise, leading to error-prone classification. The classifiers trained from views that they always retain a maximum consensus on their predictions. By enforcing different classifiers trained from different views to agree on both labeled and unlabeled training examples, the structure learned from each view can reinforce one another. The outputs of two classifiers can be used individually and the voting or weighting scheme can also be applied to combine the classifier outputs to make predictions. The main contributions of this paper are summarized as follows:

- We propose Robust Transductive Support Vector Machine (RTSVM) method that introduce the margin distribution (margin mean and margin variance) into TSVM, which is robust to the outliers and noise.
- We present a Semi-Supervised Multi-View learning algorithm, which impose a global similarity constraint between distinct RTSVMs each trained from one view of the data. The optimization method is effective and converges quickly.
- By exploring large amount of unlabeled examples and being robust to outliers and noise among different views. The proposed method is competent to the tasks on multi-view datasets. Compared with other semi-supervised learning methods and multi-view methods, we achieve some state-of-the-art results.

The rest of this paper is organized as follows. Section 2 introduces some related works. We give the formulation and convergence of RTSVM method in Sec. 3. Section 4 extends RTSVM into multi-view learning and gives the specific algorithm. Extensive experimental results are provided and analyzed in Sec. 5. Finally, Sec. 6 concludes this work with future direction.

2. Related Work

Our work is related to traditional semi-supervised and multi-view learning approaches. Co-training[11] is a well-known semi-supervised learning paradigm. When it was proposed at first, they recommended that it worked under two-view setting, which should be sufficient and redundant for each other. For example, the description of a web page can be represented as words in each page, and also be the words occurring in hyperlinks to that pages. Later, it extended to co-EM,[29] which did not consider the confidence when they labeled the unlabeled examples at each iteration. In contrast to the co-training setting, which normally uses two classifiers, Zhou and Li proposed Tri-training,[30] which extended Co-training to consider using three classifiers. Moreover, several methods were proposed to train classifiers on different views based on so-called co-regularization criterion, which were used to minimize the differences of decision values from the classifiers on different views. For example, Sindhwani *et al.*[31] presented a method which learned a multi-view classifier from partially labeled data using a view consensus-based regularization term. Similarly, Collins and Singer[32] proposed a co-boost approach that optimized an objective function by maximizing the agreement between each classifier. In addition, Yu *et al.*[33] proposed a Bayesian co-training framework which defined a multi-view kernel for semi-supervised learning with Gaussian Processes. More recently, Xu *et al.*[34] proposed an effective approach called co-labeling to solve the multi-view weakly labeled learning problem.

There are several multi-view methods proposed for SSL scenario.[35–37] CCA[12] is a typical approach for two views learning, which respectively projects the examples into a common subspace and maximizes the cross-correlation between two views. To deal with multiple views scenario, multi-view CCA was proposed to improve the performance of learning classifiers.[38] In addition, Kernel CCA (KCCA)[15] had been proved to be an effective preprocessing step that improved the performance of classification algorithm such as SVM. Then, Farquhar *et al.*[39] proposed a two-stage learning joint KCCA and SVM named SVM-2K, which gave experimental and theoretical results on some multi-view datasets. Moreover, Diethe *et al.*[40] extended Fisher's Discriminant Analysis (FDA) into the latent subspace spanned by multi-view data. Since the latent subspace is valuable for inferring another view from the observation view, Quadrianto and Lampert[41] and Zhai *et al.*[42] presented multi-view metric learning by constructing embedding projections into a shared subspace from multi-view data. Besides, Sindhwani *et al.*[43,44] proposed a co-regularization approach

for semi-supervised learning with multiple views, then they constructed a single RKHS with manifold regularization that led to major empirical improvements on semi-supervised tasks. In addition, Li *et al.*[45] proposed two-view TSVM, which achieved good performance when the number of labeled examples was small. Sun[46] proposed Laplacian support vector machines for multi-view classification. However, some of these methods cannot take good advantage of much unlabeled examples. Besides, these methods tend to ignore the difficulty of different unlabeled examples, such as the outliers and noise, leading to error-prone classification. Different from them, our proposed method can explore large amount of unlabeled examples and being robust to outliers and noise among different views.

3. Robust Transductive Support Vector Machine

In this section, we give the formulation of RTSVM. Then, we introduce the Concave-Convex Procedure (CCCP), which is utilized to solve nonconvex optimization problem of RTSVM. Finally, we analyze the convergence of RTSVM.

3.1. *Formulation*

Given dataset $\mathcal{S} = \mathcal{L} \cup \mathcal{U}$, where $\mathcal{L} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\} \in \mathcal{X} \times \mathcal{Y}$ is the labeled dataset and $\mathcal{U} = (\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, \ldots, \mathbf{x}_{l+u}) \in \mathcal{X}$ is the unlabeled test dataset. $\mathcal{Y} = \{-1, +1\}$; Formally, the dataset could be represented as matrix $\mathbf{X}$ and $\mathbf{y}$. The matrix of $\mathbf{X}$ could be denoted as $[\mathbf{x}^1, \ldots, \mathbf{x}^l]$, which each of column $\mathbf{x}^i$ is $[x^i_1, \ldots, x^i_d]$. $\mathbf{y}$ could be denoted as $[y_1, \ldots, y_l]^T$. Meanwhile, we define that $\mathbf{Y}$ is a $l \times l$ diagonal matrix, which each of diagonal elements is y_i.

According to Refs. 22 and 47, the margin of example $(\mathbf{x}_i, y_i)$ is defined as

$$p_i = y_i \mathbf{w}^T \phi(\mathbf{x_i}) \quad i = 1, \ldots, l. \tag{1}$$

Thus, the margin mean $\bar{\rho}$ and margin variance $\hat{\rho}$ are defined as

$$\bar{\rho} = \frac{1}{l} \sum_{i=1}^{l} y_i \mathbf{w}^T \phi(\mathbf{x_i}) = \frac{1}{l} (\mathbf{Xy})^T \mathbf{w}, \tag{2}$$

$$\begin{aligned} \hat{\rho} &= \sum_{i=1}^{l} \sum_{j=1}^{l} (y_i \mathbf{w}^T \phi(\mathbf{x_i}) - y_j \mathbf{w}^T \phi(\mathbf{x_j}))^2 \\ &= \frac{2}{l^2} (l \mathbf{w}^T \mathbf{X} \mathbf{X}^T \mathbf{w} - \mathbf{w}^T \mathbf{X} \mathbf{y} \mathbf{y}^T \mathbf{X}^T \mathbf{w}). \end{aligned} \tag{3}$$

RTSVM extends the margin distribution that is the first-order and second-order statistics into TSVM as regularization, which tries to achieve strong generalization performance by maximizing the margin mean and minimizing the margin variance

simultaneously. The minimization problem of RTSVM could be denoted as

$$\begin{aligned} &\frac{1}{2}\mathbf{w}^T\mathbf{w} + \lambda_1\hat{\rho} - \lambda_2\bar{\rho} + C\sum_{i=1}^{l}\xi_i + C^*\sum_{i=l+1}^{l+u}\xi_i \\ &\text{subject to}: y_i\mathbf{w}^T\phi(\mathbf{x_i}) \geq 1-\xi_i, \quad i=1,\ldots,l \\ &\qquad\qquad |\mathbf{w}^T\phi(\mathbf{x_i})| \geq 1-\xi_i, \quad i=l+1,\ldots,l+u, \end{aligned} \tag{4}$$

where $\phi(\mathbf{x})$ is a feature mapping of $\mathbf{x}$ induced by a kernel k, such as, $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$.[22] C, C^* and $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_{l+u}]^T$ are parameters of TSVM. The λ_1 and λ_2 are the parameters for trading-off the margin variance, margin mean and model complexity. In order to use CCCP to solve this optimization problem, we could reformulate the minimization equation as

$$\begin{aligned} \ell(\mathbf{w}) = &\frac{1}{2}\mathbf{w}^T\mathbf{w} + \lambda_1\hat{\rho} - \lambda_2\bar{\rho} + C\sum_{i=1}^{l}\mathbf{H_1}(y_i\mathbf{w}^T\phi(\mathbf{x_i})) \\ &+ C^*\sum_{i=l+1}^{l+u}\mathbf{H_1}(|\mathbf{w}^T\phi(\mathbf{x_i})|), \end{aligned} \tag{5}$$

where $\mathbf{H_1}(\cdot) = \max(0, 1-\cdot)$ is the Hinge loss function.

3.2. *The concave–convex procedure for RTSVM*

As is revealed in Fig. 1, in order to divide the cost function $\ell(\mathbf{w})$ of RTSVM into the sum of convex part $\ell_{\text{vex}}(\mathbf{w})$ and a concave part $\ell_{\text{cav}}(\mathbf{w})$, we denote the Ramp loss as the following:

$$R_s(z) = H_1(z) - H_s(z). \tag{6}$$

The $H_s(z)$ can be formulated as $\max(0, s-z)$, and the Hinge loss $H_1(z)$ is illustrated as the following:

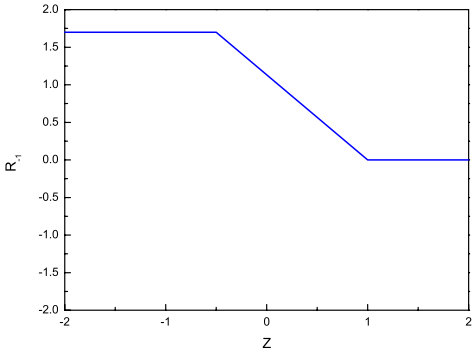
$$H_1(z) = \max(0, 1-z) = \min\xi \quad s.t.\ \xi \geq 0, \xi \geq 1-z. \tag{7}$$

Thus, Eq. (5) can be simplified as the following:

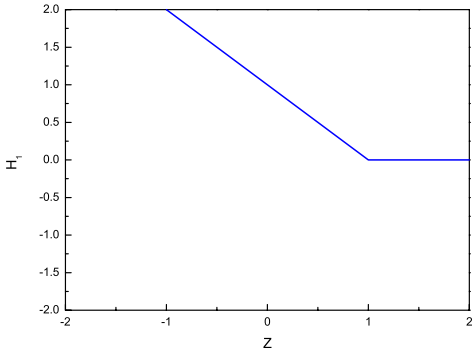
$$\begin{aligned} \ell^s_{\text{vex}}(\mathbf{w}) + \frac{\partial \ell^s_{\text{cav}}(\mathbf{w})}{\partial \mathbf{w}} \cdot \mathbf{w} &= \ell^s_{\text{vex}}(\mathbf{w}) + \left(\sum_{i=l+1}^{l+2u} y_i\beta_i\frac{\partial \mathbf{w}^T\phi(\mathbf{x}_i)}{\partial \mathbf{w}}\right)\cdot\mathbf{w} \\ &= \ell^s_{\text{vex}}(\mathbf{w}) + \sum_{i=l+1}^{l+2u}\beta_i y_i \mathbf{w}\cdot\phi(\mathbf{x}_i), \end{aligned} \tag{8}$$

where the notation we introduce is the same as [23]

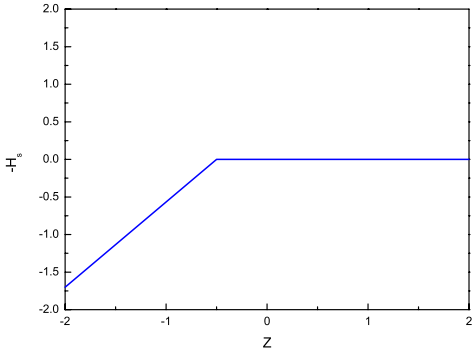
$$\beta_i = y_i\frac{\partial \ell^s_{\text{cav}}(\mathbf{w})}{\partial \mathbf{w}^T\phi(\mathbf{x}_i)} = \begin{cases} C^* & \text{if } y_i\ \mathbf{w}^T\phi(\mathbf{x}_i) \leq s \text{ and } i \geq l+1, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$



(a) The Ramp loss $R_s(t)$



(b) The Convex Hinge loss $H_I(t)$



(c) The Concave loss function $H_s(t) = \max(0, s - t)$

Fig. 1. The illusion of Ramp loss function: (a) can be decomposed into the sum of (b) and (c): The Ramp loss function $R_s(t) = \min(1 - s, \max(0, 1 - t)) = H_1(t) - H_s(t)$, where s controls the wideness of clipped Ramp loss.

Besides, the cost $\ell(\mathbf{w}^t)$ can be decreased by each iteration

$$\begin{aligned} \ell_{\text{vex}}(\mathbf{w}^{t+1}) + \ell'_{\text{cav}}(\mathbf{w}^t) \cdot \mathbf{w}^{t+1} &\leq \ell_{\text{vex}}(\mathbf{w}^t) + \ell'_{\text{cav}}(\mathbf{w}^t) \cdot \mathbf{w}^t \ell_{\text{cav}}(\mathbf{w}^{t+1}) \\ &\leq \ell_{\text{cav}}(\mathbf{w}^t) + \ell'_{\text{cav}}(\mathbf{w}^t) \cdot (\mathbf{w}^{t+1} - \mathbf{w}^t). \end{aligned} \tag{10}$$

The more convergence details of CCCP can be found in this work.[48]

4. RTSVM for Multi-View Learning

In this section, we propose a RTSVM for multi-view learning method (RTSVM-MV), which jointly exploits unlabeled examples from RTSVMs among multiple views. Then, we give the optimization method and specific algorithm for our proposed model. Finally, we analyze the complexity of algorithm. Following Farquhar *et al.*,[39] which proposed a supervised learning algorithm called SVM-2K, combining the idea

of KCCA with SVM. An SVM can be thought as projecting the feature to a one-dimensional (1D) space followed by thresholding, after which SVM-2K forces the constraint of consensus with two views on this 1D space. Formally, the constraint can be written as

$$|f_\theta^{v_1}(\mathbf{x}_i^{v_1}) - f_\theta^{v_2}(\mathbf{x}_i^{v_2})| \leq \tau_i + \epsilon, \tag{11}$$

where τ_i is a variable that imposes consensus between the two views, and ϵ is a slack variable. In multi-view embedding, we conduct the embedding for multiple features simultaneously while considering the consistency and complement of different views. By reformulating Eq. (8) as the following quadratic programming problem. The notation † denotes views v_1 and v_2.

$$\begin{aligned}\ell(\mathbf{W}_\dagger) = &\frac{1}{2}\mathbf{w}_\dagger^T\mathbf{w}_\dagger + \frac{2\lambda_1}{l^2}\mathbf{w}_\dagger^T(l\mathbf{X}_\dagger\mathbf{X}_\dagger{}^T - \mathbf{X}_\dagger \mathrm{y}_\dagger \mathrm{y}_\dagger{}^T\mathbf{X}_\dagger^T)\mathbf{w}_\dagger - \frac{\lambda_2}{l}(\mathbf{X}_\dagger\mathbf{y}_\dagger)^T\mathbf{w}_\dagger \\ &+ C_\dagger\sum_{i=1}^{l}\xi_{i\dagger} + C_\dagger^*\sum_{i=l+1}^{l+2u}\xi_{i\dagger} + \sum_{i=l+1}^{l+2u}\beta_{i\dagger}y_i\mathbf{w}_\dagger^T\phi(\mathbf{x}_{i\dagger}) \\ \text{subject to}: &\ y_i\mathbf{w}_\dagger^T\phi(\mathbf{x}_{i\dagger}) \geq 1 - \xi_{i\dagger} \quad i = 1,\ldots,l+2u, \\ &\ \xi_{i\dagger} \geq 0, \quad i = 1,\ldots,l+2u.\end{aligned} \tag{12}$$

Li *et al.*[45] proposed a two-view TSVM, which was especially useful for both toy and real-life datasets. However, it decreases the performance when the training data has some outliers or noise. Here, we consider a simple scenario, two-view RTSVM. In this setting, when two view representations of one data are available, we maximize the overall consensus of the predictions while training classifiers from each view. The proposed method is extended by the framework of Farquhar *et al.*[39] however, it takes good advantage of unlabeled examples, which is very pervasive in social and real life. This two-view learning achieves a better performance than single view learning, which is demonstrated in the below experimental results. The objective problem can be written as

$$\begin{aligned}&\min_{\mathbf{w}_\dagger,\boldsymbol{\xi}_\dagger} \ell(\mathbf{w}^{v_1}) + \ell(\mathbf{W}^{v_2}) + \lambda_3\sum_{i=1}^{l+2u}\tau_i \\ &\text{subject to}: y_i\mathbf{w}_\dagger^T\phi(\mathbf{x}_{i\dagger}) \geq 1 - \xi_{i\dagger} \quad i = 1,\ldots,l+2u, \\ &|\mathbf{w}^{v_1}\phi(\mathbf{x}_i^{v_1}) - \mathbf{w}^{v_2}\phi(\mathbf{x}_i^{v_2})| \leq \tau_i + \epsilon \quad i = 1,\ldots,l+2u, \\ &\tau_i \geq 0, \quad i = 1,\ldots,l+2u, \\ &\xi_{i\dagger} \geq 0, \quad i = 1,\ldots,l+2u,\end{aligned} \tag{13}$$

where $\mathbf{w}^{v_1}$ and $\mathbf{w}^{v_2}$ are the weight of the first and second RTSVM respectively, and σ is the weight of hybrid decision function. Thus, the final decision function is

$$f(\mathbf{z}) = \sigma f^{v_1}(\boldsymbol{z}^{v_1}) + (1-\sigma)f^{v_2}(\boldsymbol{z}^{v_2}). \tag{14}$$

4.1. *Optimization and algorithm for RTSVM-MV*

As mentioned before, we use CCCP to solve the above optimization problems. Fortunately, inspired by the Representer Theorem in Schlkopf and Smola,[49] the following Theorem states for Eq. (12) can be spanned by $\mathbf{x}_i, 1 \leq i \leq l$.

Theorem 1. *The optimal solution* $\mathbf{w}_*$ *for the problem Eq.* (12) *admits a representation of the form*

$$\mathbf{w}_* = \sum_{i=1}^{l} \alpha_i \phi(\mathbf{x}_i) = \mathbf{X}\boldsymbol{\alpha}, \tag{15}$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_l]^T$ *are the coefficients.*

The proof process can be seen in Appendix A. According to Theorem 1, we have

$$\begin{aligned} \mathbf{X}^T\mathbf{w} &= \mathbf{X}^T\mathbf{X}\alpha = \boldsymbol{K}\boldsymbol{\alpha}, \\ \mathbf{w}^T\mathbf{w} &= \boldsymbol{\alpha}^T\mathbf{X}^T\boldsymbol{X}\boldsymbol{\alpha} = \boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{\alpha}, \end{aligned} \tag{16}$$

where $\mathbf{K} = \mathbf{X}^T\mathbf{X}$ is the kernel matrix. Let $\mathbf{K_{:i}}$ denote the ith column of $\mathbf{K}$, then one view RTSVM can be cast as

$$\begin{aligned} &\min_{\alpha,\xi} \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Q}\alpha + \mathbf{q}^T\boldsymbol{\alpha} + C\sum_{i=1}^{l}\xi_i + C^*\sum_{i=l+1}^{l+2u}\xi_i + \sum_{i=l+1}^{l+2u}\beta_i y_i \boldsymbol{\alpha}^T\mathbf{K}_{:i}, \\ &\text{subject to}: y_i\boldsymbol{\alpha}^T\mathbf{K}_{:i} \geq 1 - \xi_i \quad i = 1, \ldots, l+2u, \\ &\qquad\qquad \xi_i \geq 0, \quad i = 1, \ldots, l+2u \end{aligned} \tag{17}$$

where $\mathbf{Q} = 4\lambda_1(l\mathbf{K}^T\mathbf{K} - (\mathbf{K}\mathbf{y})(\mathbf{K}\mathbf{y})^T)/l^2 + \mathbf{K}$ and $\mathbf{q} = -\lambda_2\mathbf{K}\mathbf{y}/l$. Following Farquhar *et al.*,[39] which can be seen as the global optimization of two distinct SVMs, one in each of the two feature spaces. As in our method, then with the usual 1-norm RTSVM constraints, the objective problem can be written as

$$\begin{aligned} &\min_{\mathbf{w}^{v_1/v_2},\boldsymbol{\xi}^{v_1/v_2},\boldsymbol{\tau}} \ell(\mathbf{w}^{v_1}) + \ell(\mathbf{w}^{v_2}) + \lambda_3\sum_{i=1}^{l+2u}\tau_i \\ &\text{subject to} : y_i\boldsymbol{\alpha}^T\mathbf{K}_{:i}^{v_1/v_2} \geq 1 - \xi_i^{v_1/v_2} \quad i = 1, \ldots, l+2u, \\ &\qquad\qquad \xi_i \geq 0, \quad i = 1, \ldots, l+2u, \\ &\qquad\qquad |\boldsymbol{\alpha}^T\mathbf{K}_{:i}^{v_1} - \boldsymbol{\alpha}^T\mathbf{K}_{:i}^{v_2}| \leq \tau_i + \epsilon \quad i = 1, \ldots, l+2u, \\ &\qquad\qquad \tau_i \geq 0, \quad i = 1, \ldots, l+2u, \\ &\qquad\qquad \frac{1}{u}\sum_{i=l+1}^{l+u}\boldsymbol{\alpha}^T\mathbf{K}_{:i} = \frac{1}{l}\sum_{i=1}^{l} y_i. \end{aligned} \tag{18}$$

Augmented Lagrangian is a method for solving constrained optimization problems.[50] It reformulates a constrained optimization problem into an unconstrained one by adding Lagrange multipliers and an extra penalty term for each constraint to the original objective function. We can denote the equality constraints

$\frac{1}{u}\sum_{i=l+1}^{l+u}\boldsymbol{\alpha}^T\mathbf{K}_{:i} = \frac{1}{l}\sum_{i=1}^{l} y_i$ as h_1, and rewrite the minimization problem Eq. (18) into the augmented Lagrangian form as

$$\begin{aligned}\min_{\boldsymbol{\alpha},\boldsymbol{\xi}} \quad & \left[\frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Q}\alpha + \mathbf{q}^T\boldsymbol{\alpha} + C\sum_{i=1}^{l}\xi_i + C^*\sum_{i=l+1}^{l+2u}\xi_i + \sum_{i=l+1}^{l+2u}\beta_i y_i \boldsymbol{\alpha}^T\mathbf{K}_{:i}\right]^{v_1} \\ & + \left[\frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Q}\alpha + \mathbf{q}^T\boldsymbol{\alpha} + C\sum_{i=1}^{l}\xi_i + C^*\sum_{i=l+1}^{l+2u}\xi_i + \sum_{i=l+1}^{l+2u}\beta_i y_i \boldsymbol{\alpha}^T\mathbf{K}_{:i}\right]^{v_2} \\ & + \lambda_3\sum_{i=1}^{l+2u}\tau_i - \lambda_4 h_1 + \frac{u}{2}||h_1||^2 \end{aligned} \tag{19}$$

$$\begin{aligned}\text{subject to} \ : y_i\boldsymbol{\alpha}^T\mathbf{K}_{:i}^{v_1/v_2} &\geq 1 - \xi_i^{v_1/v_2} \quad i = 1,\ldots,l+2u, \\ \xi_i &\geq 0, \quad i = 1,\ldots,l+2u, \\ |\boldsymbol{\alpha}^T\mathbf{K}_{:i}^{v_1} - \boldsymbol{\alpha}^T\mathbf{K}_{:i}^{v_2}| &\leq \tau_i + \epsilon \quad i = 1,\ldots,l+2u, \\ \tau_i &\geq 0, \quad i = 1,\ldots,l+2u.\end{aligned}$$

Algorithm 1. The proposed RTSVM-MV Algorithm

Input: Labeled and unlabeled data of two views
Initialize: $\boldsymbol{\lambda}, \mu, C, C^*, \boldsymbol{\alpha}^{v_1/v_2}, \boldsymbol{\beta}^{v_1/v_2}$
repeat
  Solve the following sub-problem.
  repeat
    Solve the minimization problem Eq. (19) with fixed $\boldsymbol{\lambda}^k$ and μ^k.
    Compute $f_{\mathbf{w}}^{v_1(t+1)}$ and $f_{\mathbf{w}}^{v_2(t+1)}$ with the solution of Eq. (19)
    Compute $\boldsymbol{\beta}^{v_1(t+1)}$ and $\boldsymbol{\beta}^{v_2(t+1)}$ by Eq. (9) with the value of $f_{\mathbf{w}}^{v_1(t+1)}$ and $f_{\mathbf{w}}^{v_2(t+1)}$.
    Update the lower and upper bounds of $\boldsymbol{\alpha}^{v_1(t+1)}$ and $\boldsymbol{\alpha}^{v_2(t+1)}$ by solution of Eq. (17).
  until $\boldsymbol{\beta}^{v_1(t+1)} = \boldsymbol{\beta}^{v_1(t)}$ and $\boldsymbol{\beta}^{v_2(t+1)} = \boldsymbol{\beta}^{v_2(t)}$
  Update the Lagrange multiplier $\boldsymbol{\lambda}$ by
  $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \mu^k\mathbf{h}^{\mathbf{k}}$
  Update the penalty parameter μ by
  $\mu^{k+1} = \phi\mu^k$
until $||\mathbf{h}^{\mathbf{k}}|| \leq \epsilon$
Return: The decision functions corresponding to two views calculated by Eqs. (20) and (21).

As inspired by ,[51,52] Eq. (19) can be efficiently solved by the augmented Lagrangian method. Thus, two views can be calculated by the following:

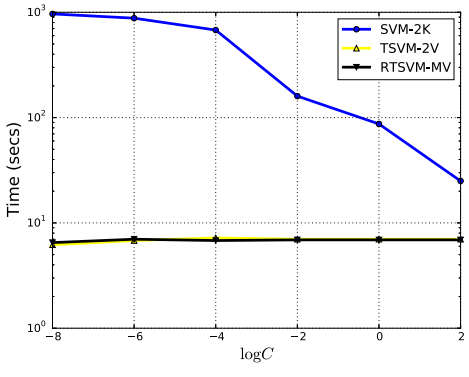
$$f^{v_1}(\mathbf{z}^{v_1}) = \text{sgn}(\mathbf{w}^T\phi(\mathbf{z})) = \text{sgn}\left(\sum_{i=1}^{l}\alpha_i^{v_1}k^{v_1}(\mathbf{x}_i,\mathbf{z})\right), \tag{20}$$

$$f^{v_2}(\mathbf{z}^{v_2}) = \text{sgn}(\mathbf{w}^T\phi(\mathbf{z})) = \text{sgn}\left(\sum_{i=1}^{l}\alpha_i^{v_2}k^{v_2}(\mathbf{x}_i,\mathbf{z})\right). \tag{21}$$

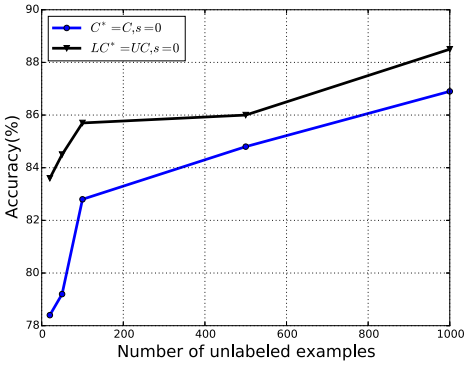
The hybrid decision function can be rewritten as the following equation from the above two classifiers

$$f(\mathbf{z}) = \sigma f^{v_1}(\boldsymbol{z}^{v_1}) + (1-\sigma)f^{v_2}(\boldsymbol{z}^{v_2}), \tag{22}$$

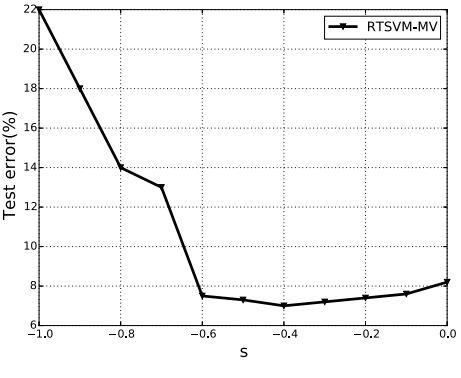
where $0 \leq \sigma \geq 1$. The pseudo-code of RTSVM-MV can be seen in Algorithm 1.



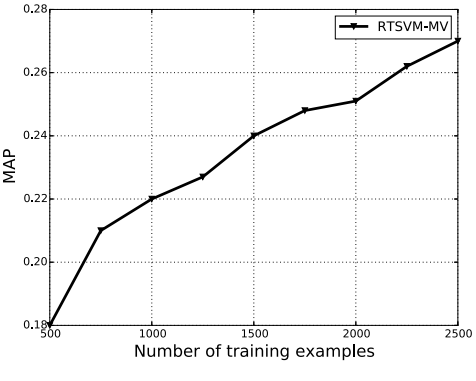
(a) Impact of C.



(b) Impact of C*.



(c) Impact of s.



(d) Impact of different number of labeled examples.

Fig. 2. Parameters study of our proposed method.

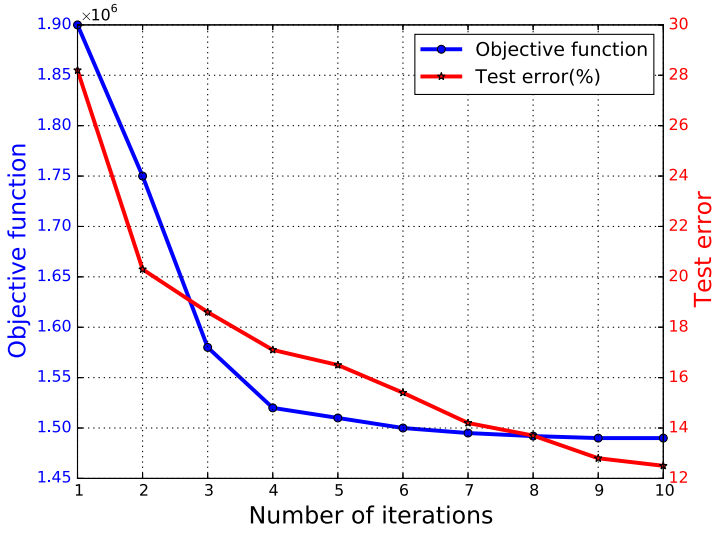


Fig. 3. Convergence study.

Algorithm 1 summarizes the two-view RTSVM algorithm. The detailed convergence analysis of the Lagrange multiplier iteration, which corresponds to the outer loop of Algorithm 1 can be found in Ref. 50. Also the convergence of the CCCP procedure is discussed in Ref. 53. In our experiments, we set the maximum number of Lagrange multiplier iterations to seven. Because we observe that the algorithm converges before reaching the maximum number of iterations in most cases.

4.2. *Complexity*

As empirical study of TSVM,[23] one uses $l + 2u$ variables to solve a series of quadratic optimization problems. Thus, the space complexity of two view RTSVM is $O((l + 2u)^2$. Here, we assume each data examples $\boldsymbol{x}_i$ has $\bar{d}$ nonzero elements, so the time complexity os RTSVM-MV is $O((l + 2u)\bar{d})$. When RTSVM-MV uses Gaussian kernel, then each $g_{ij} = \exp(\gamma||x_i - x_j||^2)$ requires $O(\bar{d})$ computation time. The advantage of our algorithm is that it only needs a small amount of iterations to reach the minimum when solving optimization problems of RTSVM-MV, which can be seen the experimental results in Fig. 3.

5. Experimental Study

In this section, we introduce the small and large-scale datasets, which have at least two views, and give the evaluation metrics. Then, we compare our algorithm with baselines and a number of related state-of-the-art approaches. Finally, we show the

empirical results on these datasets, including the effect of different number of labeled examples and parameters study for our method.

5.1. *Datasets*

We choose WebKB course, a few benchmark and UCI[a] multi-view datasets for the task of our proposed method and related state-of-the-art approaches. The WebKB course dataset has two views (pages view and links view) and contains 1,051 examples, each corresponding to a web page. There are 230 positive examples. In page view and links view, we use 66 attributes and 5 attributes, respectively. The Ads dataset has five views. The task is to predict whether an example, corresponding to an image on the web, is used for advertisement or not. This dataset contains 983 examples, among which there are 138 positive examples. The UCI Ads dataset has more than two views. We denote the Ads12 dataset uses the url view and origurl view, Ads13 use url view and destination-url view, while Ads23 uses origurl view and destination-url view. In addition, we also choose another two datasets, which contain news articles collected from the BBC. There are five topics (business, entertainment, politics, sports and technology) from the BBC dataset and five classes (athletics, cricket, football, rugby and tennis) from BBCSport, where each contains 2,225 and 737 sports news documents, respectively. The details of these datasets are summarized in Tables 1 and 2.

For large-scale multi-view datasets, we adopt the dataset from the NUS-WIDE,[b] a popular dataset for cross-modal retrieval, which contains 269,648 images downloaded from Flickr that has been manually annotated, with several tags (2–5 on

Table 1. Summarization of the Datasets (The $d1/d2/d3$, p and n are the dimension, positive and negative examples of dataset).

Datasets	view1	view2	view3	$d1/d2/d3$	p	n
WebKB	page	link	—	3000/1840	230	821
Ads	image url	dest url	alt	457/472/111	459	2820

Table 2. Summarization of the Datasets (The $d1/d2, c, l, u$ and t are the dimension of two views, numbers of classes, labeled training examples, unlabeled training examples and test examples, respectively.).

Datasets	$d1$	$d2$	c	l	u	t
BBC	4,817	4,818	5	10	1,104	1,111
BBCSport	2,306	2,307	5	10	360	367

[a] http://archive.ics.uci.edu/ml/datasets.html.
[b] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm.

average) per image. There are six kinds of global visual features in this collection, namely color histogram, color auto-correlogram and block-wise color moments, which are all color-based features. The other features are based on texture technique, which are edge direction histogram and wavelet texture. In addition, one last set of visual features are based on the SIFT technique.[54] In our experiments, visual features including 73-dimension edge direction histogram (EDH) and 128-dimension wavelet texture (WT) are used as well as a new set of features called Decaf which is based on caffe.[55] In this experiment, we choose 1,000 labeled examples and other training examples as the unlabeled examples.

5.2. *Baselines and evaluation setup*

To validate the effectiveness of our method, we compare it with baselines and a few number of related state-of-the-art approaches, which are enumerated as follows:

- **SVM:** One of the most influential classification algorithms as the baseline algorithm. For small and large-scale datasets, Linear kernel is employed in our methods.
- **SVM-2K:** Farquhar *et al.*[39] proposed a two-stage learning joint KCCA and SVM, which had demonstrated that it can be possible to leverage the correlation between the two views to improve classification accuracy.
- **TSVM:** CCCP for Large-Scale TSVMs was proposed by Collobert *et al.*[23]. It can efficiently solve large-scale datasets that have few labeled examples together with a large collection of unlabeled examples.
- **TSVM-2V:** Li *et al.*[45] proposed an extension of the existing two-view supervised learning algorithm into a semi-supervised setting, which is able to take advantage of unlabeled examples among multiple views.
- **CoLapSVM:** Sindhwani *et al.*[43] proposed a Co-Regularization approach for semi-supervised learning with multiple views.
- **CoMR:** Sindhwani and Rosenberg[44] constructed a single RKHS with manifold regularization that led to major empirical improvements on semi-supervised tasks.
- **MvLapSVM:** Sun[46] proposed multi-view Laplacian SVMs for semi-supervised learning under the multi-view scenario.

Here, we introduce some package tools, which was used in our experiments. For small-scale datasets, LIBSVM[c] and SVM$^{\text{light}}$[d] are employed. For large-scale datasets,

[c] http://www.csie.ntu.edu.tw/ cjlin/libsvm/.
[d] http://svmlight.joachims.org/.

due to high computational load of LIBSVM and SVM$^{\text{light}}$, efficient LIBLINEAR,[e] UniverSVM[f] and TSVM[g] are served as baselines instead. Since these methods involve kernels, the width of RBF kernel is set from $\{2^{-2}\delta, \ldots, 2^{2}\delta\}$, where δ is the average distance between instances from inductive SVM.[2] We use $UC^* = LC$ heuristic as empirical study of Ref. 23 and set s of Symmetric Hinge loss at intervals of 0.1 from -1.0 to 0. The empirical value of regularization parameter $\log C$[56] is selected from $\{-8, -6, \ldots, 0, 2\}$, and the regularization parameters λ_1, λ_2 are selected from the set of $\{2^{-8}, \ldots, 2^{-2}\}$ by 10-fold cross-validation. All tests are conducted on a machine equipped with a dual Xeon X5650 CPUs (6 cores each 2 hyper-threading) with 64 GB of RAM and a 1 TB dedicated disk.

In our below experiments, data are partitioned into two parts: the training data and the testing data. The training data is used for model estimation while the test data is utilized to test the performance of our methods. When reporting the performance of two-view baselines on single view datasets, we use the different regularizations or kernels .[43] While reporting the performance of single-view methods on hybrid view datasets, we concatenate the input feature vectors from each view to form a large feature set. On each dataset, we perform a 10-fold cross-validation. The experiments repeat for 30 times and report the accuracy performance. We use t-test at 95% significance level to make pairwise comparison. It will be demonstrated in Secs. 3 to 4 about the classification and standard deviation for running various baselines and related semi-supervised multi-view learning methods.

5.3. *Comparing with various baselines on small-scale datasets*

In this section, we compare our method with baselines and a number of related state-of-the-art approaches on small-scale datasets. Following the experimental studies in Tables 3–6, we observe that the proposed method shows the best performance among all of the compared methods in terms of the different number of labeled examples, single-view and hybrid views on both the WebKB, BBC, BBCSport and Ads datasets. To be specific, we conduct our experiments on WebKB dataset with 20 and 60 labeled examples and see that our method obtains a strong performance than other methods. 92.1% and 93.7% of average classification accuracy can be achieved on Hybrid views of WebKB dataset with 20 and 60 labeled examples, respectively. The same trend is that our algorithm still achieves best performance on BBC and BBCSport datasets with 10 labeled examples. It is not very surprising to see that RTSVM-MV provides good performance since the robustness to the outliers and noise. And we also achieve the best result on UCI Ads dataset with 20 labeled examples, which the average classification accuracy is 92.6% on hybrid views.

[e] http://www.csie.ntu.edu.tw/cjlin/linlinear/.
[f] http://mloss.org/software/view/19/.
[g] http://www.kyb.tuebingen.mpg.de/bs/people/fabee/transduction.html.

Table 3. Experimental results (mean ± standard deviation) on the WebKB dataset. Results in boldface are better than the others.

	20 Labeled			60 Labeled		
Methods	Page	Link	Hybrid	Page	Link	Hybrid
SVM	73.8 ± 10.8	75.2 ± 9.8	76.4 ± 9.5	77.6 ± 8.4	77.9 ± 9.2	79.2 ± 8.6
SVM-2K	89.0 ± 5.6	89.2 ± 8.9	90.6 ± 8.7	89.7 ± 5.8	89.5 ± 8.8	91.2 ± 7.9
TSVM	88.1 ± 7.9	90.2 ± 7.7	91.0 ± 7.8	88.5 ± 7.5	90.6 ± 6.9	91.3 ± 7.6
TSVM-2V	89.2 ± 5.3	91.8 ± 2.5	93.1 ± 2.9	89.4 ± 5.5	92.0 ± 2.6	93.6 ± 3.1
CoLapSVM	90.3 ± 5.8	81.5 ± 10.7	88.6 ± 7.6	90.7 ± 5.9	83.9 ± 8.8	89.4 ± 7.9
CoMR	90.6 ± 5.5	82.3 ± 9.5	90.2 ± 5.3	90.6 ± 6.1	85.2 ± 6.3	89.8 ± 4.1
MvLapSVM	89.7 ± 7.8	89.6 ± 9.0	90.4 ± 6.2	90.5 ± 6.4	85.0 ± 7.1	90.3 ± 3.8
RTSVM-MV	**91.2 ± 5.6**	**92.1 ± 5.8**	**92.2 ± 5.9**	91.3 ± 6.0	**92.3 ± 4.7**	**93.7 ± 2.9**

Table 4. Experimental results (mean ± standard deviation) on the BBC dataset. Results in boldface are better than the others.

	BBC(10 Labeled)			BBCSport(10 Labeled)		
Methods	View1	View2	View1+2	View1	View2	View1+2
SVM	69.2 ± 3.9	66.9 ± 4.3	78.3 ± 3.5	75.2 ± 3.4	70.2 ± 2.5	81.3 ± 3.7
SVM-2K	69.8 ± 4.1	67.3 ± 4.5	79.2 ± 3.6	76.0 ± 3.5	71.1 ± 1.9	82.4 ± 4.2
TSVM	73.2 ± 5.5	68.0 ± 3.3	78.1 ± 3.9	75.8 ± 5.7	66.5 ± 3.4	79.2 ± 3.6
TSVM-2V	73.7 ± 4.9	68.6 ± 3.5	78.6 ± 4.2	76.3 ± 5.8	67.2 ± 3.8	80.0 ± 3.5
CoLapSVM	77.4 ± 3.2	75.2 ± 4.8	83.2 ± 2.9	74.6 ± 3.5	70.9 ± 2.5	81.7 ± 3.6
CoMR	77.8 ± 3.3	75.5 ± 4.9	84.0 ± 3.0	75.2 ± 3.7	72.3 ± 2.6	83.0 ± 2.8
MvLapSVM	80.2 ± 2.6	77.3 ± 3.9	85.2 ± 3.2	81.3 ± 1.7	79.2 ± 1.8	85.1 ± 2.4
RTSVM-MV	**83.0 ± 1.7**	**80.6 ± 4.2**	**88.1 ± 2.9**	**83.8 ± 2.0**	**82.4 ± 2.3**	**87.5 ± 3.8**

Compared with SVM, the proposed RTSVM-MV method takes the advantage of exploiting the unlabeled examples. While comparing with TSVMs, the proposed RTSVM-MV method shows the role of margin distribution in generalization performance of classifier and improves the performance of models on multi-view datasets. Moreover, our proposed method shows superior to other semi-supervised

Table 5. Experimental results (mean ± standard deviation) on the Ads dataset. Results in boldface are better than the others.

	UCI Ads(20 Labeled)		
Method	Image URL	Destination URL	Hybrid
SVM	85.3 ± 2.1	87.9 ± 1.8	88.5 ± 2.0
SVM-2K	82.2 ± 10.8	89.4 ± 0.7	90.6 ± 0.9
TSVM	86.9 ± 3.0	88.3 ± 1.5	89.8 ± 2.1
TSVM-2V	88.0 ± 5.9	89.7 ± 6.2	90.6 ± 5.8
CoLapSVM	86.8 ± 1.7	88.1 ± 1.3	88.5 ± 1.4
CoMR	87.2 ± 1.6	89.7 ± 1.1	90.9 ± 1.2
MvLapSVM	87.8 ± 3.0	90.1 ± 4.8	90.5 ± 2.7
RTSVM-MV	**90.2 ± 1.8**	92.3 ± 2.0	**92.6 ± 2.1**

Table 6. Experimental results (mean ± standard deviation) on the Ads dataset. Results in boldface are better than the others.

	UCI Ads(20 Labeled)		
Method	Destination URL	Alt	Hybrid
SVM	88.3 ± 1.8	86.7 ± 1.2	89.0 ± 1.7
SVM-2K	90.6 ± 1.0	86.9 ± 1.1	90.8 ± 1.2
TSVM	89.5 ± 1.4	87.3 ± 0.9	90.2 ± 1.8
TSVM-2V	90.7 ± 1.5	85.9 ± 1.7	90.1 ± 2.0
CoLapSVM	88.6 ± 1.3	88.0 ± 0.6	89.8 ± 1.7
CoMR	90.8 ± 0.9	88.4 ± 1.0	90.1 ± 1.9
MvLapSVM	89.6 ± 1.5	88.2 ± 1.1	90.3 ± 1.6
RTSVM-MV	$\mathbf{91.4 \pm 0.7}$	88.6 ± 1.2	91.8 ± 0.9

multi-view methods, which indicates that RTSVM-MV could exploit large amount of unlabeled examples and being robust to outliers and noise among different views. Therefore, the results of experimental study demonstrate that our proposed method can help to improve classification accuracy in semi-supervised multi-view learning.

5.4. *Comparing with various baselines on large-scale datasets*

In this section, we compare our method with baselines and a few related state-of-the-art approaches on large-scale datasets. We choose three two-view combinations (EDH+WT, EDH+Decaf and WT+Decaf) from three evaluated visual features. To be specific, we conduct our experiments on NUS-WIDE dataset with 1,000 labeled examples and see that our method obtains a strong performance than other methods. We use the well-known evaluation metric, Mean Average Precision (MAP), which has been widely used for information retrieval and classification tasks. From experimental results in Table 7, it can be seen that our method (RTSVM-MV) always performs better than SVM-2K, CoMR and MvLapSVM. For different multi-view settings, we also see that RTSVM-MV has a good performance when using the views include Decaf, which means that it has strong representative. 4.97%, 21.32% and 21.51% of MAP can be achieved on

Table 7. Experimental results (Mean Average Precision) on the NUS-WIDE dataset. Results in boldface are better than the others.

	NUS-WIDE Dataset(1000 Labeled)		
Method	EDH+WT	EDH+Decaf	WT+Decaf
SVM-2K	4.23%	5.85%	5.75%
CoMR	4.31%	12.28%	13.68%
MvLapSVM	4.28%	7.96%	9.82%
RTSVM-MV	**4.97%**	**21.32%**	**21.51%**

three two-view combinations, respectively. For large-scale multi-view learning, it is not very surprising to see that RTSVM-MV provides good performance since the robustness to the outliers and noise. Therefore, the results of experimental study demonstrate that our proposed method can help to improve classification accuracy in semi-supervised multi-view learning.

5.5. *Parameters sensitivity study*

In this section, we study how hyperparameters C, C^*, s and number of labeled training examples affect the performance of our proposed algorithm.

Impact of C values: We study the effect of parameter C with RTSVM-MV on WebKB dataset. As empirical value from these works,[25,56] which the regularization parameter $\log C$ are selected from $\{-8,-6,-4,-2,0,2\}$. We observe that SVM-2K costs much computation time, thus TSVM-2V as well as RTSVM-MV cost similar time and much less than SVM-2K. Moreover, the general trend is that with the $\log C$ increasing, the computation time decreases. The computation time with different $\log C$ values are reported in Fig. 2(a).

Impact of C^* values: The hyperparameters C and C^* of RTSVM-MV have the same effect with TSVM, which are trading-off margin size against misclassifying training examples.[20] Inspired by large-scale TSVM,[23] we compare $UC^* = LC$ under $s = 0$ for RTSVM-MV and a small value C^* that approximates the value C slowly until it reaches $C^* = C$. Following the experimental results in Fig. 2(b), the performance of using heuristic $UC^* = LC$ is better than using $C^* = C$ via increasing the number of unlabeled examples on BBC dataset.

Impact of s values: The Symmetric Hinge loss plays an important role in our method. We study the impact of hyperparameter s by varying from $\{-1,-0.8,-0.6,-0.4,-0.2,0\}$. These observations highlight the importance of the parameter s of the loss function by showing the best test error over different choices of s on Ads dataset. Following the experimental results in Fig. 2(c), it shows that increasing s values of RTSVM-MV will result in decreased testing error and remain steady.

Impact of different number of labeled training examples: To study the effect of different number of labeled training examples, we set the number of labeled training examples to $\{500,750,1000,\ldots,2000,2250,2500\}$ on NUS-WIDE dataset and measure the MAP performance for each set of training examples. We observe that with increasing the labeled training examples, the performance of RTSVM-MV undergo an increasing trend. The results are shown in Fig. 2(d).

5.6. *Convergence study*

In this section, we experimentally validate its convergence and study the speed of convergence of RTSVM-MV. It is our advantage that it could reach convergence of

algorithm only after a small amount of iterations. We note that with increasing the number of iterations, the value of objective function and test error remains steady. The convergence rate on BBC dataset is shown in Fig. 3. Following the experiments, which show that the optimization algorithm is effective and converges quickly.

6. Conclusion

This paper presents a novel semi-supervised multi-view method. We introduce the margin distribution that is the first-order (margin mean) and second-order statistics (margin variance) into transductive support vector machines (TSVMs). It tries to achieve strong generalization performance by maximizing the margin mean and minimizing the margin variance simultaneously, which is robust to the outliers and noise. RTSVMs trained from different views that they always retain a maximum consensus on their predictions and the structure learned from each view can reinforce one another. Our method is based on transduction that it is able to make better predictions with fewer labeled examples. The experimental study on benchmark and UCI datasets shows that the proposed method is superior to the previous semi-supervised multi-view learning methods. Besides, our algorithm runs with good convergence. Moreover, it is able to address the complexity conditions by using kernel methods. Finally, it will be practical to generalize this algorithm to real applications in future.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work is supported in part by the National Natural Science Foundation of China under Grant 61170035 and 61272420, Six talent peaks project in Jiangsu Province (Grant No. 2014 WLW-004), the Fundamental Research Funds for the Central Universities (Grant No. 30920130112006), Jiangsu Province special funds for transformation of science and technology achievement (Grant No. BA2013047), The Research Innovation Program for College Graduates of Jiangsu Province (Grant No. KYLX15_0376).

Appendix A. Proof of Theorem 1

Proof. $\mathbf{w}^*$ can be decomposed into a part that lives in the span of $\phi(\boldsymbol{x}_i)$ and an orthogonal part, i.e.,

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i \phi(\boldsymbol{x}_i) + \boldsymbol{\epsilon} = \mathbf{X}\boldsymbol{\alpha} + \boldsymbol{\epsilon}, \tag{A.1}$$

where $\boldsymbol{\epsilon}$ satisfies $\phi(\boldsymbol{x}_k) + \boldsymbol{\epsilon} = 0$ for all k, so, $\mathbf{X}^T\boldsymbol{\epsilon} = 0$, thus,

$$\mathbf{X}^T\mathbf{w} = \mathbf{X}^T(\mathbf{X}^T\boldsymbol{\alpha} + \boldsymbol{\epsilon}) = \mathbf{X}^T\mathbf{X}\boldsymbol{\alpha}. \tag{A.2}$$

Similarly, we can derive

$$\mathbf{w}^T\mathbf{w} = (\mathbf{X}^T\boldsymbol{\alpha} + \boldsymbol{\epsilon})^T(\mathbf{X}^T\boldsymbol{\alpha} + \boldsymbol{\epsilon}) = \boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\boldsymbol{\alpha} \geq \boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\alpha}, \tag{A.3}$$

only if the equation is equal, which means $\boldsymbol{\epsilon} = 0$. Hence, the problem of Eq. (19) admits a representation of the form Eq. (22) with $\mathbf{w}^*$. □

References

1. X. Zhu and A. B. Goldberg, Introduction to semi-supervised learning, *Synth. Lect. Artif. Intell. Mach. Learn.* **3** (2009) 1–130.
2. O. Chapelle *et al.*, Semi-supervised learning, *IEEE Transactions on Neural Networks* **20** (2009) 542.
3. A. Blum and S. Chawla, Learning from labeled and unlabeled data using graph mincuts (2001).
4. Z. Yang, W. Cohen and R. Salakhutdinov, Revisiting semi-supervised learning with graph embeddings, arXiv:1603.08861.
5. Y. Li, Y. Wang, X. Jiang and Z. Dong, Teaching-to-learn and learning-to-teach for few labeled classification, *2016 Int. Conf. Advanced Cloud and Big Data (CBD)* (IEEE, 2016), pp. 271–276.
6. O. Chapelle and A. Zien, Semi-supervised classification by low density separation, *AISTATS*, 2005, pp. 57–64.
7. M. Belkin, P. Niyogi and V. Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, *J. Mach. Learn. Res.* **7** (2006) 2399–2434.
8. J. Zhou and T. Wei, Stochastic thermal-aware real-time task scheduling with considerations of soft errors, *J. Syst. Softw.* **102** (2015) 123–133.
9. J. Zhou, T. Wei, M. Chen, J. Yan, X. S. Hu and Y. Ma, Thermal-aware task scheduling for energy minimization in heterogeneous real-time mpsoc systems, *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **35** (2016) 1269–1282.
10. C. Xu, D. Tao and C. Xu, A survey on multi-view learning, arXiv:1304.5634.
11. A. Blum and T. Mitchell, Combining labeled and unlabeled data with co-training, *Proc. Eleventh Annual Conf. Computational Learning Theory* (ACM, 1998), pp. 92–100.
12. H. Hotelling, Relations between two sets of variates, *Biometrika* **28** (1936) 321–377.
13. S. Bickel and T. Scheffer, Multi-view clustering, *ICDM*, Vol. 4 (2004), pp. 19–26.
14. A. Kumar, P. Rai and H. Daume, Co-regularized multi-view spectral clustering, *Advances in Neural Information Processing Systems* (2011), pp. 1413–1421.
15. S. Akaho, A kernel method for canonical correlation analysis, arXiv:cs/0609071.
16. W. Zheng, X. Zhou, C. Zou and L. Zhao, Facial expression recognition using kernel canonical correlation analysis (kcca), *IEEE Trans. Neural Netw.* **17** (2006) 233–238.
17. K. Chaudhuri, S. M. Kakade, K. Livescu and K. Sridharan, Multi-view clustering via canonical correlation analysis, *Proc. 26th Annual Int. Conf. Machine Learning* (ACM, 2009), pp. 129–136.
18. Z. Xu, R. Jin, H. Yang, I. King and M. R. Lyu, Simple and efficient multiple kernel learning by group lasso, *Proc. 27th Int. Conf. Machine Learning (ICML-10)*, 2010, pp. 1175–1182.

19. N. Subrahmanya and Y. C. Shin, Sparse multiple kernel learning for signal processing applications, *IEEE Trans. Pattern Anal. Mach. Intell.* **32** (2010) 788–798.
20. T. Joachims, Transductive inference for text classification using support vector machines, *ICML* Vol. 99 (1999), pp. 200–209.
21. T. Joachims, Making large scale svm learning practical, Technical Report, Universität Dortmund (1999).
22. V. Vapnik, *The Nature of Statistical Learning Theory* (Springer Science & Business Media, NewYork, 2013).
23. R. Collobert, F. Sinz, J. Weston and L. Bottou, Large scale transductive svms, *J. Mach. Learn. Res.* **7** (2006) 1687–1712.
24. W. Gao and Z.-H. Zhou, On the doubt about margin explanation of boosting, *Artif. Intell.* **203** (2013) 1–18.
25. T. Zhang and Z.-H. Zhou, Large margin distribution machine, *Proc. 20th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining* (ACM, 2014), pp. 313–322.
26. A. Garg and D. Roth, Margin distribution and learning algorithms, *Proc. Fifteenth Int. Conf. Machine Learning (ICML)*, 2003, pp. 210–217.
27. K. Pelckmans, J. Suykens and B. D. Moor, A risk minimization principle for a class of parzen estimators, *Adv Neural Information Processing Systems*, 2008, pp. 1137–1144.
28. F. Aiolli, G. Da San Martino and A. Sperduti, A kernel method for the optimization of the margin distribution, *Int. Conf. Artificial Neural Networks* (Springer, 2008), pp. 305–314.
29. K. Nigam and R. Ghani, Analyzing the effectiveness and applicability of co-training, *Proc. Ninth Int. Conf. Information and Knowledge Management* (ACM, 2000), pp. 86–93.
30. Z.-H. Zhou and M. Li, Tri-training: Exploiting unlabeled data using three classifiers, *IEEE Trans. Knowl Data Eng.* **17** (2005) 1529–1541.
31. V. Sindhwani, P. Niyogi and M. Belkin, A co-regularization approach to semi-supervised learning with multiple views, *Proc. ICML Workshop on Learning with Multiple Views*, 2005, pp. 74–79.
32. M. Collins and Y. Singer, Unsupervised models for named entity classification, *Proc. Joint SIGDAT Conf Empirical Methods in Natural Language Processing and Very Large Corpora* (Citeseer, 1999), pp. 100–110.
33. S. Yu, B. Krishnapuram, R. Rosales and R. B. Rao, Bayesian co-training, *J. Mach. Learn. Res.* **12** (2011) 2649–2680.
34. X. Xu, W. Li, D. Xu and I. W. Tsang, Co-labeling for multi-view weakly labeled learning, *IEEE Trans. Pattern Anal. Mach. Intell.* **38** (2016) 1113–1125.
35. J. Liu, Y. Jiang, Z. Li, Z.-H. Zhou and H. Lu, Partially shared latent factor learning with multiview data, *IEEE Trans. Neural Netw. Learn. Syst.* **26** (2015) 1233–1246.
36. C. Christoudias, R. Urtasun and T. Darrell, Multi-view learning in the presence of view disagreement, arXiv:1206.3242.
37. M. Kan, S. Shan, H. Zhang, S. Lao and X. Chen, Multi-view discriminant analysis, *European Conf. Computer Vision* (Springer, 2012), pp. 808–821.
38. J. Zhou, X. S. Hu, Y. Ma and T. Wei, Balancing lifetime and soft-error reliability to improve system availability, *Design Automation Conf. (ASP-DAC), 2016 21st Asia and South Pacific* (IEEE, 2016), pp. 685–690.
39. J. Farquhar, D. Hardoon, H. Meng, J. S. Shawe-taylor and S. Szedmak, Two view learning: Svm-2k, theory and practice, *Advances in Neural Information Processing Systems*, 2005, pp. 355–362.

40. T. Diethe, D. R. Hardoon and J. Shawe-Taylor, Multiview fisher discriminant analysis, *NIPS Workshop on Learning from Multiple Sources* (MIT Press, Canada, 2008).
41. N. Quadrianto and C. H. Lampert, Learning multi-view neighborhood preserving projections, *Proc. 28th Int. Conf. Machine Learning (ICML-11)*, 2011, pp. 425–432.
42. D. Zhai, H. Chang, S. Shan, X. Chen and W. Gao, Multiview metric learning with global consistency and local smoothness, *ACM Trans. Intell. Syst. Technol. (TIST)* **3** (2012) 53.
43. V. Sindhwani, P. Niyogi and M. Belkin, Beyond the point cloud: From transductive to semi-supervised learning, *Proc. 22nd Int. Conf. Machine Learning* (ACM, 2005), pp. 824–831.
44. V. Sindhwani and D. S. Rosenberg, An rkhs for multi-view learning and manifold co-regularization, *Proc. 25th Int. Conf. on Machine Learning* (ACM, 2008), pp. 976–983.
45. G. Li, S. C. Hoi and K. Chang, Two-view transductive support vector machines, *SDM* (SIAM, 2010), pp. 235–244.
46. S. Sun, Multi-view laplacian support vector machines, *Int. Conf. Advanced Data Mining and Applications* (Springer, 2011), pp. 209–222.
47. C. Cortes and V. Vapnik, Support-vector networks, *Mach. Learn.* **20**(3) (1995) 273–297.
48. I. E.-H. Yen, N. Peng, P.-W. Wang and S.-D. Lin, On convergence rate of concave-convex procedure, *Proc. NIPS 2012 Optimization Workshop* (MIT Press, USA, 2012).
49. B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond* (MIT press, US, 2002).
50. D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods* (Academic press, US, 2014).
51. G.-X. Yuan, C.-H. Ho and C.-J. Lin, Recent advances of large-scale linear classification, *Proc. IEEE* **100** (2012) 2584–2603.
52. G.-X. Yuan, K.-W. Chang, C.-J. Hsieh and C.-J. Lin, A comparison of optimization methods and software for large-scale l1-regularized linear classification, *J. Mach. Learn. Res.* **11** (2010) 3183–3234.
53. A. L. Yuille, A. Rangarajan and A. Yuille, The concave-convex procedure (cccp), *Adv. Neural Inform. Process. Syst.* **2** (2002) 1033–1040.
54. D. G. Lowe, Object recognition from local scale-invariant features, *The Proc. Seventh IEEE Int. Conf. Computer vision, 1999*, Vol. 2 (IEEE, 1999), pp. 1150–1157.
55. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama and T. Darrell, Caffe: Convolutional architecture for fast feature embedding, *Proc. 22nd ACM Int. Conf. Multimedia* (ACM, 2014), pp. 675–678.
56. S. S. Keerthi and C.-J. Lin, Asymptotic behaviors of support vector machines with gaussian kernel, *Neural Comput.* **15** (2003) 1667–1689.