Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2021.013357
Article

Tech Science Press

A Novel Semi-Supervised Multi-Label Twin Support Vector Machine

Qing Ai[1,2,*], Yude Kang[1] and Anna Wang[2]

[1]School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China
[2]College of Information Science and Engineering, Northeastern University, Shenyang, 110819, China
[*]Corresponding Author: Qing Ai. Email: lyaiqing@126.com
Received: 04 August 2020; Accepted: 25 September 2020

Abstract: Multi-label learning is a meaningful supervised learning task in which each sample may belong to multiple labels simultaneously. Due to this characteristic, multi-label learning is more complicated and more difficult than multi-class classification learning. The multi-label twin support vector machine (MLTSVM) [1], which is an effective multi-label learning algorithm based on the twin support vector machine (TSVM), has been widely studied because of its good classification performance. To obtain good generalization performance, the MLTSVM often needs a large number of labelled samples. In practical engineering problems, it is very time consuming and difficult to obtain all labels of all samples for multi-label learning problems, so we can only obtain a large number of partially labelled and unlabelled samples and a small number of labelled samples. However, the MLTSVM can use only expensive labelled samples and ignores inexpensive partially labelled and unlabelled samples. Because of the MLTSVM's disadvantages, we propose an alternative novel semi-supervised multi-label twin support vector machine, named SS-MLTSVM, which can take full advantage of the geometric information of the edge distribution embedded in partially labelled and unlabelled samples by introducing a manifold regularization term into each sub-classifier and use the successive overrelaxation (SOR) method to speed up the solving process. Experimental results on several publicly available benchmark multi-label datasets show that, compared with the classical MLTSVM, our proposed SS-MLTSVM has better classification performance.

Keywords: Multi-label learning; semi-supervised learning; TSVM; MLTSVM

1 Introduction

Multi-label learning is a meaningful supervised learning task, wherein each sample may belong to multiple different labels simultaneously. In real life, many applications employ multi-label learning, including text classification [2,3], image annotation [4], bioinformatics [5], and so on [6]. Because the samples can have multiple labels simultaneously, multi-label learning is more complicated and more difficult than multi-class classification learning. At present, there are two kinds of methods to solve multi-label learning problems: problem transformation and algorithm adaptation. The problem transformation solves the multi-label learning problem by transforming it into one or more single-label problems, such as

binary relevance (BR) [7], classifier chains (CC) [8], label powerset (LP) [9], calibrated label ranking (CLR) [10], and random k-labelsets (RAKEL) [11]. The algorithm adaptation extends the existing single-label learning algorithm to handle multi-label learning problems, such as multi-label k-nearest neighbour (ML-kNN) [12], multi-label decision tree (ML-DT) [13], ranking support vector machine (Rank-SVM) [14], and collective multi-label classifier (CML) [15].

The TSVM [16], proposed by Jayadeva, is used to solve classification problems. It has been widely studied because of its good classification performance. Subsequent to its release, many improved algorithms have been proposed [17–29]. The aforementioned improved algorithms can solve only single-label learning problems, not multi-label learning problems. In 2016, Chen et al. extended the TSVM to solve multi-label learning problems and proposed a multi-label twin support vector machine (MLTSVM) [1]. Compared with other traditional multi-label classification algorithms, the MLTSVM has better generalization performance. Thereafter, many improved algorithms of MLTSVM have been proposed [30,31]. Hanifelou et al. introduced local information and structural information of samples into the MLTSVM and proposed the k-nearest neighbour-based MLTSVM with priority of labels (PKNN-MLTSVM) [30]. Meisam et al. proposed the structural least square twin support vector machine for multi-label learning (ML-SLSTSVM) [31], which proposed a least squares version of the MLTSVM and added structural information of samples.

The aforementioned improvements to the MLTSVM mainly focused on improving the generalization performance and learning speed. It is very time consuming and difficult to obtain all labels of all samples for multi-label learning problems; in fact, we can obtain only a large number of partially labelled and unlabelled samples and a small number of labelled samples. However, the MLTSVM and its improvements can use only expensive labelled samples and ignores inexpensive partially labelled and unlabelled samples. Because of this disadvantage, we propose a novel semi-supervised MLTSVM, named SS-MLTSVM, which can take full advantage of the geometric information of the edge distribution embedded in partially labelled and unlabelled samples by introducing a manifold regularization term into each sub-classifier and use the successive overrelaxation (SOR) method to increase the solving speed. Experimental results show that, compared with the MLTSVM, our SS-MLTSVM has better classification performance.

The structure of this paper is as follows: Section 2 introduces some related works, such as the TSVM and MLTSVM. In Section 3, the SS-MLTSVM is introduced in detail, including the linear model, nonlinear model, decision rules and training algorithm. The fourth section gives the experimental results of the proposed algorithm on the benchmark datasets. The fifth section is the conclusion.

2 Related Works

2.1 TSVM

For the binary classification problem, we suppose the training set is $T = \{(x_i, y_i)|i = 1, \ldots, m\}$, where $x_i \in R^n$ is the training sample and $y_i \in \{+1, -1\}$ is the label corresponding to the training sample x_i. We denote positive training samples by $A \in R^{m_1 \times n}$ and negative training samples by $B \in R^{m_2 \times n}$. $m = m_1 + m_2$ is the total number of training samples.

The TSVM is aimed to find two nonparallel hyperplanes:

$$f_1(x) : w_1^T x + b_1 = 0,\ f_2(x) : w_2^T x + b_2 = 0. \tag{1}$$

The original problem of TSVM is:

$$\begin{aligned} &\min_{w_1,b_1,\xi_2} \frac{1}{2}\|Aw_1 + e_1b_1\|^2 + c_1e_2^T\xi_2 \\ &s.t. \quad -(Bw_1 + e_2b_1) + \xi_2 \geq e_2,\ \xi_2 \geq 0 \end{aligned}, \tag{2}$$

$$\begin{aligned} &\min_{w_2,b_2,\xi_1} \frac{1}{2}\|Bw_2 + e_2b_2\|^2 + c_2e_1^T\xi_1 \\ &s.t. \quad Aw_2 + e_1b_2 + \xi_1 \geq e_1,\ \xi_1 \geq 0 \end{aligned}, \tag{3}$$

where c_1 and c_2 are the penalty parameters, ξ_1 and ξ_2 are the slack variables, and e_1 and e_2 are all 1 vector of the proper dimension.

By introducing the Lagrange multiplier, the dual problems of (2) and (3) can be obtained as follows:

$$\begin{aligned} &\max_{\alpha} \ e_2^T\alpha - \frac{1}{2}\alpha^T G\left(H^TH\right)^{-1}G^T\alpha \\ &s.t. \quad 0 \leq \alpha \leq c_1e_2 \end{aligned}, \tag{4}$$

$$\begin{aligned} &\max_{\gamma} \ e_1^T\gamma - \frac{1}{2}\gamma^T H\left(G^TG\right)^{-1}H^T\gamma \\ &s.t. \quad 0 \leq \gamma \leq c_2e_1 \end{aligned}, \tag{5}$$

where $H = [A \quad e_1]$, $G = [B \quad e_2]$, and α and γ are the Lagrange multipliers.

The two hyperplanes can be obtained by solving the dual problems as follows:

$$v_1 = -\left(H^TH\right)^{-1}G^T\alpha,\ v_1 = \left[w_1^T \quad b_1\right]^T, \tag{6}$$

$$v_2 = -\left(G^TG\right)^{-1}H^T\gamma,\ v_2 = \left[w_2^T \quad b_2\right]^T. \tag{7}$$

2.2 MLTSVM

For the multi-label problem, we denote the training set as:

$$T = \{(x_i, y_i)|i = 1, \ldots, m\}, \tag{8}$$

where $x_i \in R^n$ is the training sample, $y_i = \left\{y_{i1}, \ldots, y_{ip}\right\}$ is the label set of the sample x_i, $y_{iq} \in \{1, \ldots, K\}(q = 1, \ldots, p)$, m is the total number of training samples and K is the total number of labels.

The MLTSVM seeks K hyperplanes:

$$f_k(x) : w_k^Tx + b_k = 0,\ k = 1, 2, \ldots, K. \tag{9}$$

We denote the samples belonging to the kth class by A_k and the other samples by B_k. The original problem for label k is:

$$\begin{aligned} &\min_{w_k,b_k,\epsilon_{B_k}} \frac{1}{2}\|A_kw_k + e_{A_k}b_k\|^2 + c_ke_{B_k}^T\xi_{B_k} + \frac{1}{2}\lambda_k(\|w_k\|^2 + b_k^2) \\ &s.t. \quad -(B_kw_k + e_{B_k}b_k) + \xi_{B_k} \geq e_{B_k},\ \xi_{B_k} \geq 0 \end{aligned}, \tag{10}$$

where c_k and λ_k are the penalty parameters, e_{A_k} and e_{B_k} are all 1 vector of the proper dimension, and ξ_{B_k} is the slack variable.

By introducing the Lagrange multiplier, the dual problems of (10) can be obtained as follows:

$$\begin{aligned} &\max_{\alpha_{B_k}} \ e_{B_k}^T \alpha_{B_k} - \frac{1}{2} \alpha_{B_k}^T G (H^T H + \lambda_k I_k)^{-1} G^T \alpha_{B_k} \\ &s.t. \quad 0 \le \alpha_{B_k} \le c_k \end{aligned}, \tag{11}$$

where $H = [A_k \quad e_{A_k}]$, $G = [B_k \quad e_{B_k}]$, I_k are the diagonal matrices of the proper dimension, and α_{B_k} is the Lagrange multiplier.

By solving the dual problem (11), we can obtain:

$$u_k = -(H^T H + \lambda_k I_k)^{-1} G^T \alpha_{B_k}, \ u_k = \left[w_k^T \quad b_k^T \right]^T. \tag{12}$$

2.3 ML-STSVM

Similar to the MLTSVM, the ML-STSVM also seeks K hyperplanes:

$$f_k(x) : w_k^T x + b_k = 0, \ k = 1, 2, \ldots, K. \tag{13}$$

The original problem for label k is:

$$\begin{aligned} &\min_{w_k, b_k, \epsilon_{B_k}} \frac{1}{2} \|A_k w_k + e_{A_k} b_k\|^2 + c_{k1} e_{B_k}^T \xi_{B_k} + \frac{1}{2} c_{k2} (\|w_k\|^2 + b_k^2) + \frac{1}{2} c_{k3} (w_k^T \Sigma_{A_k} w_k) \\ &s.t. \qquad -(B_k w_k + e_{B_k} b_k) + \xi_{B_k} \ge e_{B_k}, \ \xi_{B_k} \ge 0 \end{aligned}, \tag{14}$$

where $c_{ki} (i = 1, 2, 3)$ are the penalty parameters, ξ_{B_k} is the slack factor, e_{A_k} and e_{B_k} are all 1 vector of the proper dimension, and $\Sigma_{A_k} = \Sigma_1 + \ldots + \Sigma_i + \ldots + \Sigma_n$, Σ_i is the covariance matrix of the ith cluster in A_k.

The dual problem of (14) is:

$$\begin{aligned} &\max_{\alpha_{B_k}} \ e_{B_k}^T \alpha_{B_k} - \frac{1}{2} \alpha_{B_k}^T G (H^T H + c_{k2} I_k + c_{k2} J)^{-1} G^T \alpha_{B_k} \\ &s.t. \quad 0 \le \alpha_{B_k} \le c_{k1} \end{aligned}, \tag{15}$$

where $H = [A_k \quad e_{A_k}]$, $G = [B_k \quad e_{B_k}]$, and $J = \begin{bmatrix} \Sigma_{A_k} & 0 \\ 0 & 0 \end{bmatrix}$. I_k are the diagonal matrices of the proper dimension, and α_{B_k} is the Lagrange multiplier.

By solving the dual problem (15), we can obtain:

$$u_k = -(H^T H + c_{k2} I_k + c_{k3} J)^{-1} G^T \alpha_{B_k}, \ u_k = \left[w_k^T \quad b_k^T \right]^T. \tag{16}$$

3 SS-MLTSVM

For the semi-supervised multi-label problem, we define the training set as follows:

$$T = \{(x_i, y_i) | i = 1, \ldots, l\}, \tag{17}$$

where $x_i \in R^n$ is the training sample and $y_i = \{y_{i1}, \ldots, y_{iK}\}$ is the label matrix of the sample x_i.

$$y_{ip} = \begin{cases} +1, & if \ x_i \ belongs \ to \ the \ kth \ class, \\ -1 & if \ x_i \ doesn't \ belong \ to \ the \ kth \ class, \\ 0, & uncertain, \end{cases} \quad 1 \le p \le K. \tag{18}$$

3.1 Manifold Regularization Framework

The manifold regularization framework [32], proposed by Belkin et al., can effectively solve semi-supervised learning problems. The objective optimization function of the manifold regularization framework can be expressed as follows:

$$f^* = \arg\min_{f \in H_k} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_H \|f\|_H^2 + \gamma_M \|f\|_M^2, \tag{19}$$

where f is the decision function to be solved, V is the loss function on the labelled samples, the regularization term $\|f\|_H^2$ is used to control the complexity of the classifier, and the manifold regularization term $\|f\|_M^2$ reflects the internal manifold structure of the data distribution.

3.2 Linear SS-MLTSVM

Similar to the MLTSVM, for each label, the SS-MLTSVM seeks a hyperplane:

$$f_k(x) : w_k^T x + b_k = 0,\ k = 1, 2, \ldots, K. \tag{20}$$

For the kth label, we denote the samples that definitely belong to the kth class by A_k, i.e., $A_k = \{x_i | y_{ik} = +1\}$; the samples that definitely do not belong to the kth class by, i.e., $B_k = \{x_i | y_{ik} = -1\}$; and the samples that are uncertain of belonging to the kth class, by U_k, i.e., $U_k = \{x_i | y_{ik} = 0\}$; $T = A_k \cup B_k \cup U_k$.

To make full use of U_k, according to the manifold regularization framework, in our SS-MLTSVM, the loss function V is replaced by a square loss function and a Hinge loss function, namely:

$$V(x_i, y_i, f_k) = \left(\left(A_{i,\cdot} w_k\right) + b_k\right)^2 + \max\left(0, 1 - f_k\left(B_{i,\cdot}\right)\right). \tag{21}$$

The regularization term $\|f\|_H^2$ can be replaced by:

$$\|f_k\|_H^2 = \frac{1}{2}\left(\|w_k\|_2^2 + b_k^2\right). \tag{22}$$

The manifold regularization term $\|f\|_M^2$ can be expressed as:

$$\|f_k\|_{\mathrm{M}}^2 = \frac{1}{l^2} \sum_{i,j=1}^{l} W_{i,j} \left(f_k(x_i) - f_k\left(x_j\right)\right)^2 = f_k^T L f_k, \tag{23}$$

where $f_k = [f_k(x_1), \ldots, f_k(x_l)]^T = Tw_k + eb_k$. $L = D - W$ is the Laplace matrix, where W is defined as follows:

$$W_{i,j} = \begin{cases} \exp(-\left\|x_i - x_j\right\|^2 / 2\sigma^2), & \textit{if } x_i \textit{ and } x_j \textit{ are k nearest neighbor}, \\ 0, & \textit{otherwise}, \end{cases} \tag{24}$$

and D is defined as follows:

$$D_{i,i} = \sum_{j=1}^{l+u} W_{i,j}. \tag{25}$$

For the kth label, the original problem in the linear SS-MLTSVM is:

$$\begin{aligned} &\min \frac{1}{2}(A_k w_k + e_{A_k} b_k)^2 + c_{k1} e_{B_k}^T \xi_{B_k} + \frac{1}{2} c_{k2} \left(\|w_k\|^2 + b_k^2 \right) + \frac{1}{2} c_{k3} (T w_k + e b_k)^T L (T w_k + e b_k) \\ &s.t. \quad -\left(B_k w_k + e_{B_k} b_k \right) + \xi_{B_k} \geq e_{B_k}, \ \xi_{B_k} \geq 0 \end{aligned}, \tag{26}$$

where $c_{ki} (i = 1, 2, 3)$ are the penalty parameters, ξ_{B_k} is the slack factor, e_{A_k}, e_{B_k} and e are all 1 vector of the proper dimension, and L is the Laplace matrix.

The Lagrange function of (26) is as follows:

$$\begin{aligned} L(\Theta) = &\frac{1}{2}(A_k w_k + e_{A_k} b_k)^2 + c_{k1} e_{B_k}^T \xi_{B_k} + \frac{1}{2} c_{k2} \left(\|w_k\|^2 + b_k^2 \right) + \frac{1}{2} c_{k3} (T w_k + e b_k)^T L (T w_k + e b_k) \\ &- \alpha_{B_k}^T \left(-(B_k w_k + e_{B_k} b_k) + \xi_{B_k} - e_{B_k} \right) - \beta_{B_k}^T \xi_{B_k} \end{aligned}, \tag{27}$$

where $\Theta = \left\{ w_k, b_k, \xi_k, \alpha_{B_k}, \beta_{B_k} \right\}$. $\alpha_{B_k} \geq 0$ and $\beta_{B_k} \geq 0$ are Lagrange multipliers. Using KKT theory, we can obtain:

$$\frac{\partial L}{\partial w_k} = A_k^T (A_k w_k + e_{A_k} b_k) + c_{k2} w_k + c_{k3} T^T L (T w_k + e b_k) + B_k^T \alpha_{B_k} = 0, \tag{28}$$

$$\frac{\partial L}{\partial b_k} = e_{A_k}^T (A_k w_k + e_{A_k} b_k) + c_{k2} b_k + c_{k3} e^T L (T w_k + e b_k) + e_{B_k}^T \alpha_{B_k} = 0, \tag{29}$$

$$\frac{\partial L}{\partial \xi_k} = c_{k1} e_{B_k} - \alpha_{B_k} - \beta_{B_k} = 0. \tag{30}$$

According to (28)–(30), we can obtain the dual problem of (26) as follows:

$$\begin{aligned} &\max_\alpha \ e_{B_k}^T \alpha_{B_k} - \frac{1}{2} \alpha_{B_k}^T G \left(H^T H + c_{k2} I_k + c_{k3} J^T L J \right)^{-1} G^T \alpha_{B_k} \\ &s.t. \quad 0 \leq \alpha_{B_k} \leq c_{k1} e_{B_k} \end{aligned}, \tag{31}$$

where $H = [A_k \quad e_{A_k}]$, $G = [B_k \quad e_{B_k}]$, and $J = [T \quad e]$.

For the kth label, the hyperplane can be obtained by solving the dual problem as follows:

$$v_k = -\left(H^T H + c_{k2} I_k + c_{k3} J^T L J \right)^{-1} G^T \alpha_{B_k}, \ v_k = \left[w_k^T \quad b_k^T \right]^T. \tag{32}$$

3.3 Nonlinear SS-MLTSVM

In this section, using the kernel-generated surfaces, we extend the linear SS-MLTSVM to the nonlinear case. For each label, the nonlinear SS-MLTSVM seeks the following hyperplanes:

$$f_k(x) : K(x^T, T^T) w_k + b_k = 0, \ k = 1, 2, \ldots, K, \tag{33}$$

where $K(\cdot, \cdot)$ is a kernel function. Similar to the linear case, the regularization term $\|f\|_H^2$ and the manifold regularization term $\|f\|_M^2$ in (19) can be, respectively, expressed as:

$$\|f_k\|_H^2 = \frac{1}{2} \left(w_k^T K(T, T^T) w_k + b_k^2 \right), \tag{34}$$

$$\|f_k\|_M^2 = f_k^T L f_k = \left(K(T, T^T) w_k + e b_k \right)^T L \left(K(T, T^T) w_k + e b_k \right). \tag{35}$$

The original problem of the nonlinear SS-MLTSVM is as follows:

$$\begin{aligned}\min\ & \frac{1}{2}\left(K\left(A_k,T^T\right)w_k+e_{A_k}b_k\right)^2+c_{k1}e_{B_k}^T\xi_{B_k}+\frac{1}{2}c_{k2}\left(w_k^TK\left(T,T^T\right)w_k+b_k^2\right)\\&+\frac{1}{2}c_{k3}\left(K\left(T,T^T\right)w_k+eb_k\right)^TL\left(K\left(T,T^T\right)w_k+eb_k\right)\\ s.t.\ & -\left(K\left(B_k,T^T\right)w_k+e_{B_k}b_k\right)+\xi_{B_k}\geq e_{B_k},\xi_{B_k}\geq 0\end{aligned}. \tag{36}$$

The Lagrange function of (36) is as follows:

$$\begin{aligned}L(\Theta)=&\frac{1}{2}\left(K\left(A_k,T^T\right)w_k+e_{A_k}b_k\right)^2+c_{k1}e_{B_k}^T\xi_{B_k}+\frac{1}{2}c_{k2}\left(w_k^TK\left(T,T^T\right)w_k+b_k^2\right)\\&+\frac{1}{2}c_{k3}\left(K\left(T,T^T\right)w_k+eb_k\right)^TL\left(K\left(T,T^T\right)w_k+eb_k\right)\\&-\alpha_{B_k}^T\left(-\left(K(B_k,T^T)w_k+e_{B_k}b_k\right)+\xi_{B_k}-e_{B_k}\right)-\beta_{B_k}^T\xi_{B_k}\end{aligned}. \tag{37}$$

According to KKT theory, we can obtain:

$$\begin{aligned}\frac{\partial L}{\partial w_k}=&K\left(A_k,T^T\right)^T\left(K\left(A_k,T^T\right)w_k+e_{A_k}b_k\right)+c_{k2}K\left(T,T^T\right)w_k\\&+c_{k3}K\left(T,T^T\right)^TL\left(K\left(T,T^T\right)w_k+eb_k\right)+K\left(B_k,T^T\right)^T\alpha_{B_k}=0\end{aligned}, \tag{38}$$

$$\frac{\partial L}{\partial b_k}=e_{A_k}^T\left(K\left(A_k,T^T\right)w_k+e_{A_k}b_k\right)+c_{k2}b_k+c_{k3}e^TL\left(K\left(T,T^T\right)w_k+eb_k\right)+e_{B_k}^T\alpha_{B_k}=0, \tag{39}$$

$$\frac{\partial L}{\partial \xi_k}=c_{k1}e_{B_k}-\alpha_{B_k}-\beta_{B_k}=0. \tag{40}$$

According to (38)–(40), we can obtain the dual problem of (32) as follows:

$$\begin{aligned}\max_{\alpha}\ & e_{B_k}^T\alpha_{B_k}-\frac{1}{2}\alpha_{B_k}^TG\left(H^TH+c_{k2}I_k+c_{k3}J^TLJ\right)^{-1}G^T\alpha_{B_k}\\ s.t.\ & 0\leq\alpha_{B_k}\leq c_{k1}e_{B_k}\end{aligned}, \tag{41}$$

where $H=\left[K(A_k,T^T)\quad e_{A_k}\right]$, $G=\left[K(B_k,T^T)\quad e_{B_k}\right]$, $J=\left[K(T,T^T)\quad e\right]$ and $I_k=\begin{bmatrix}K(T,T^T) & 0\\ 0 & 1\end{bmatrix}$.

By solving the dual problem, the hyperplane of the kth label can be obtained as follows:

$$v_k=-\left(H^TH+c_{k2}I_k+c_{k3}J^TLJ\right)^{-1}G^T\alpha_{B_k},\ v_k=\left[w_k^T\quad b_k^T\right]^T. \tag{42}$$

3.4 Decision Function

In this subsection, we present the decision function of our SS-MLTSVM. For a new sample x, as mentioned above, if the sample x is proximal enough to a hyperplane, it can be assigned to the corresponding class. In other words, if the distance $d_k(x)$ between x and the kth hyperplane

$$d_k(x)=\frac{\left|w_k^Tx+b_k\right|}{\|w_k\|},\quad k=1,\ldots,K, \tag{43}$$

is less than or equal to the given value Δ_k, $k = 1, \ldots, K$, then the sample x is assigned to the kth class. To choose the proper Δ_k, we apply the strategy in the MLTSVM, which is a simple and effective method, i.e., we set $\Delta_k = \Delta = \min_{p=1,\ldots,K}\left(\frac{1}{\|w_p\|}\right)$, $k = 1, \ldots, K$.

3.5 *Fast Solvers*

In this subsection, we use SOR to solve the dual problems (31) and (41) efficiently [33]. For convenience, we set $Q = G(H^T H + c_{k2} I_k + c_{k3} J^T L J)^{-1} G^T$. The dual problems (31) and (41) can be uniformly rewritten as:

$$\begin{aligned} &\max_{\alpha} \; e_{B_k}^T \alpha_k - \frac{1}{2}\alpha_k^T Q \alpha_k \\ &s.t. \quad 0 \le \alpha_k \le c_{k1} e_{B_k} \end{aligned}, \tag{44}$$

Algorithm 1: The SOR for optimization problem (44)

INPUT:

penalty parameter c_{k1}, relaxation factor $t \in (0, 2)$, and matrix Q.

OUTPUT:

The optimal solution α_k in (44).

Step 1: Initialize the iteration variable $i = 0$ and start with any α^0.

Step 2: Decompose $Q = S + M + S'$, where M is a diagonal matrix and S is a strict lower triangular matrix.

Step 3: While $\|\alpha^{i+1} - \alpha^i\| < \sigma$

Calculate $\alpha^{i+1} = \alpha^i + t\Delta_\alpha$, where $\Delta_\alpha = -M^{-1}\left(Q\alpha^i - e + S'(\alpha^{i+1} - \alpha^i)\right)$.

Project $\alpha^{i+1} \in [0, c_{k1}]$.

End while.

Algorithm 1 only updates one variable α^{i+1} in each iteration, which can effectively reduce the complexity of the algorithm and speed up the learning process.

4 Experiments

In this section, we present the classification results of backpropagation for multi-label learning (BPMLL) [34], ML-kNN, Rank-SVM, MLTSVM and our SS-MLTSVM on the benchmark datasets. All the algorithms are implemented in MATLAB (R2017b), and the experimental environment is an Intel Core i3 processor and 4G RAM. In the experiments, we use five common datasets for multi-label learning, including flags, birds, emotions, yeast and scene (see Tab. 1). To verify the classification performance of our SS-MLTSVM, we choose 50% of the dataset as labelled samples and the remaining samples as unlabelled samples.

The parameters of the algorithm have an important impact on the classification performance. We use 10-fold cross-validation to select the appropriate parameters for each algorithm. For BPMLL, the number of hidden neurons is set to 20% of the input dimension, and the number of training epochs is 100. For the ML-kNN, the number of nearest neighbours is set to 5. For the Rank-SVM, the penalty parameter c is

selected from $\{2^{-6},\ldots,2^{0},\ldots,2^{6}\}$. For the MLTSVM, we select penalty parametersc_k and λ_k from $\{2^{-8},\ldots,2^{0},\ldots,2^{8}\}$. For the SS-MLTSVM, we select penalty parameters c_{k1}, c_{k2}, and c_{k3} from $\{2^{-8},\ldots,2^{0},\ldots,2^{8}\}$.

Table 1: Detailed description of the datasets

Datasets	Domain	Unlabelled Instance	Labelled Instance	Feature	Label
Flags	image	97	97	19	7
Birds	audio	322	323	260	19
Emotions	music	296	297	72	6
Scene	image	1203	1204	294	6
Yeast	biology	1235	1236	103	14

4.1 Evaluation Criteria

In the experiments, we use five popular metrics to evaluate the multi-label classifiers, which are Hamming loss, average precision, coverage, one_error and ranking loss. Next, we introduce these five evaluation metrics in detail.

We denote the total number of samples by l and the total number of labels by K. Y_i and $\overline{Y_i}$ represent the relevant label set and irrelevant label set of sample x_i, respectively. The function $f(x,y)$ returns a confidence of y being the right label of sample x, and the function $rank(x,y,f)$ returns a descending rank of $f(x,y)$ for any $y \in \{y_1,\ldots,y_K\}$.

4.1.1 Hamming Loss

The evaluation criteria are used to measure the proportion of labels that are wrongly classified.

$$Hamming\_loss(H\_L) = \frac{1}{l}\sum_{i=1}^{l}\frac{1}{K}|h(x_i)\Delta Y_i|, \tag{45}$$

where $h(x_i)$ is the predicted label set of sample x_i.

4.1.2 Coverage

The evaluation criteria are used to measure how many steps we need to go down the ranked label list to contain all true labels of a sample.

$$Coverage = \frac{1}{l}\sum_{i=1}^{l}\max_{y\in Y_i} rank\,(x_i,y,f) - 1. \tag{46}$$

4.1.3 One_Error

The evaluation criteria are used to measure the proportion of samples whose label with the highest prediction probability is not in the true label set.

$$One\_error(O\_E) = \frac{1}{l}\sum_{i=1}^{l} H(x_i), \tag{47}$$

where

$$H(x_i) = \begin{cases} 0 & if\ \arg\max f(x_i, y) \in Y_i \\ 1 & otherwise \end{cases}. \tag{48}$$

4.1.4 Ranking Loss

The evaluation criteria are used to measure the proportion of label pairs that are ordered reversely.

$$Ranking\_loss(R\_L) = \frac{1}{l}\sum_{i=1}^{l}\left(\frac{1}{|Y_i||\overline{Y_i}|}\left|\left\{\left(y^{'}y^{''}\right)|f(x_i, y') \le f\left(x_i, y^{''}\right),\ y' \in Y_i,\ y^{''} \in \overline{Y_i}\right\}\right|\right). \tag{49}$$

4.1.5 Average Precision

The evaluation criteria are used to measure the proportion of labels ranked above a particular label $y \in Y_i$.

$$Average\_precision(A\_P) = \frac{1}{l}\sum_{i=1}^{l}\left(\frac{1}{|Y_i|}\sum_{y\in Y_i}\frac{|\{(y' \in Y_i)|rank(x_i, y', f) \le rank(x_i, y, f)\}|}{rank(x_i, y, f)}\right). \tag{50}$$

4.2 Results

We show the average precision, coverage, Hamming loss, one_error and ranking loss of each algorithm on the benchmark datasets in Tabs. 2–6. From Tabs. 2 and 3, we can observe that, for average precision and coverage, our SS-MLTSVM is superior to the other algorithms for each dataset, while for Hamming loss, one_error and ranking loss, no algorithm is superior to any other algorithm on all datasets. Therefore, for Hamming loss, one_error and ranking loss, we proceed to use the Friedman test to evaluate each algorithm statistically. The Friedman statistics is as follows:

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right], \tag{51}$$

where $R_j = \frac{1}{N}\sum_i r_i^j$, r_i^j represents the rank of the jth algorithm on the ith dataset. Because χ_F^2 is undesirably conservative, we apply the better statistic

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1) - \chi_F^2} \sim F(k-1, (k-1)(N-1)), \tag{52}$$

where k is the number of algorithms and N is the number of datasets.

We can obtain $\chi_F^2(H\_L) = 7.84$, $\chi_F^2(O\_E) = 9.76$, $\chi_F^2(R\_L) = 9.92$, and $F_F(H\_L) = 2.58$, $F_F(O\_E) = 3.81$, $F_F(R\_L) = 3.94$. When the significance level is $\alpha = 0.10$, $F(4, 16) = 2.33$. Because $F_F(H\_L) = 2.58$, $F_F(O\_E) = 3.81$ and $F_F(R\_L) = 3.94$ are larger than the critical values, 5 algorithms have significant differences for the three metrics. We list the rank of the different algorithms in light of Hamming loss, one_error and ranking loss in Tabs. 7–9. From Tabs. 7–9, we can see that the average rank of our SS-MLTSVM is better than other algorithms; thus, the SS-MLTSVM has better classification performance.

Table 2: Average precision of algorithms on the benchmark datasets

	BPMLL	ML-kNN	Rank-SVM	MLTSVM	SS-MLTSVM
Flags	0.785639 ± 0.051239	0.793221 ± 0.008711	0.795623 ± 0.003876	0.777998 ± 0.009513	**0.800176 ± 0.008125**
Birds	0.358771 ± 0.010801	0.421682 ± 0.008798	0.406008 ± 0.003114	0.404525 ± 0.005299	**0.449760 ± 0.004186**
Emotions	0.581655 ± 0.030349	0.697028 ± 0.004776	0.709426 ± 0.005503	0.757375 ± 0.005520	**0.811084 ± 0.006444**
Scene	0.453430 ± 0.020162	0.842465 ± 0.003841	0.830496 ± 0.002475	0.827238 ± 0.002951	**0.845295 ± 0.004920**
Yeast	0.720021 ± 0.023814	0.731047 ± 0.001198	0.730825 ± 0.000957	0.731715 ± 0.000454	**0.744821 ± 0.001117**

Table 3: Coverage of algorithms on the benchmark datasets

	BPMLL	ML-kNN	Rank-SVM	MLTSVM	SS-MLTSVM
Flags	3.875556 ± 0.663135	3.777000 ± 0.031977	3.797440 ± 0.024735	3.837778 ± 0.082848	**3.771110 ± 0.054257**
Birds	3.923201 ± 0.073242	4.101629 ± 0.052476	3.889508 ± 0.029002	3.482169 ± 0.028734	**3.133523 ± 0.063058**
Emotions	2.868046 ± 0.294812	2.245276 ± 0.034788	2.191575 ± 0.032059	2.044092 ± 0.035888	**1.842989 ± 0.057524**
Scene	2.171729 ± 0.141061	0.555427 ± 0.008191	0.598790 ± 0.007469	0.612847 ± 0.011717	**0.523698 ± 0.014979**
Yeast	6.341873 ± 0.342050	6.456955 ± 0.022583	6.231757 ± 0.010173	6.987227 ± 0.024749	**6.222335 ± 0.042419**

Table 4: Hamming loss of algorithms on the benchmark datasets

	BPMLL	ML-kNN	Rank-SVM	MLTSVM	SS-MLTSVM
Flags	0.347143 ± 0.028855	0.329460 ± 0.009443	0.306206 ± 0.006918	0.315238 ± 0.009461	**0.301361 ± 0.012386**
Birds	0.067793 ± 0.001566	**0.055723 ± 0.000044**	0.106989 ± 0.003470	0.097220 ± 0.001389	0.087929 ± 0.001621
Emotions	0.374502 ± 0.040605	0.263362 ± 0.003004	0.264391 ± 0.003745	0.222902 ± 0.003495	**0.206513 ± 0.007170**
Scene	0.285942 ± 0.019638	0.151188 ± 0.003112	**0.090790 ± 0.000977**	0.184711 ± 0.00128	0.147302 ± 0.001591
Yeast	0.216659 ± 0.016479	0.212532 ± 0.000808	0.212260 ± 0.001056	0.210753 ± 0.001176	**0.209740 ± 0.001138**

Table 5: One_error of algorithms on the benchmark datasets

	BPMLL	ML-kNN	Rank-SVM	MLTSVM	SS-MLTSVM
Flags	0.262222 ± 0.012525	0.246806 ± 0.029941	0.195583 ± 0.021652	0.238889 ± 0.022773	**0.161905 ± 0.023462**
Birds	0.765142 ± 0.019694	**0.633331 ± 0.012700**	0.642681 ± 0.004973	0.733791 ± 0.012932	0.686058 ± 0.012041
Emotions	0.634943 ± 0.051561	0.412047 ± 0.014014	0.383724 ± 0.012444	0.296793 ± 0.006964	**0.232759 ± 0.007655**
Scene	0.820441 ± 0.026645	0.269452 ± 0.003841	0.278494 ± 0.004690	0.287614 ± 0.004593	**0.252672 ± 0.009250**
Yeast	0.240158 ± 0.036489	0.265893 ± 0.003882	0.241793 ± 0.002361	0.249052 ± 0.003871	**0.232493 ± 0.003846**

Table 6: Ranking loss of algorithms on the benchmark datasets

	BPMLL	ML-kNN	Rank-SVM	MLTSVM	SS-MLTSVM
Flags	0.229981 ± 0.043959	0.228139 ± 0.008245	**0.216063 ± 0.00325**	0.239685 ± 0.010365	0.229881 ± 0.009013
Birds	0.313953 ± 0.006586	0.318532 ± 0.004400	0.302169 ± 0.004040	0.292203 ± 0.002815	**0.250932 ± 0.004507**
Emotions	0.392612 ± 0.039955	0.266880 ± 0.004724	0.253129 ± 0.005824	0.216821 ± 0.006205	**0.169383 ± 0.008775**
Scene	0.418993 ± 0.028879	0.098806 ± 0.001565	0.105142 ± 0.001448	0.124898 ± 0.002293	**0.085024 ± 0.002948**
Yeast	0.185699 ± 0.001101	0.181432 ± 0.000784	**0.171814 ± 0.000470**	0.217908 ± 0.000920	0.174228 ± 0.021022

Table 7: Ranks of algorithms in light of Hamming loss

	BPMLL	ML-kNN	Rank-SVM	MLTSVM	SS-MLTSVM
Flags	5	4	2	3	**1**
Birds	2	**1**	5	4	3
Emotions	5	3	4	2	**1**
Scene	5	3	**1**	4	2
Yeast	5	4	3	2	**1**
Average	4.4	3	3	3	**1.6**

Table 8: Ranks of algorithms in light of one_error

	BPMLL	ML-kNN	Rank-SVM	MLTSVM	SS-MLTSVM
Flags	5	4	2	3	**1**
Birds	5	**1**	2	4	3
Emotions	5	4	3	2	**1**
Scene	5	2	3	4	**1**
Yeast	2	5	3	4	**1**
Average	4.4	3.2	2.6	3.4	**1.4**

Table 9: Ranks of algorithms in light of ranking loss

	BPMLL	ML-kNN	Rank-SVM	MLTSVM	SS-MLTSVM
Flags	4	2	**1**	5	3
Birds	4	5	3	2	**1**
Emotions	5	4	3	2	**1**
Scene	5	2	3	4	**1**
Yeast	4	3	**1**	5	2
Average	4.4	3.2	2.2	3.6	**1.6**

From the above analysis, we can conclude that our SS-MLTSVM is superior to the other algorithms for all metrics.

We show the learning time of different algorithms on the benchmark datasets in Tab. 10. From Tab. 10, we can observe that our SS-MLTSVM has a lower learning speed than the MLTSVM. This is mainly because our SS-MLTSVM adds a manifold regularization term that needs to solve the Laplace matrix with whole samples. Even so, our SS-MLTSVM still has great advantages compared with the Rank-SVM and BPMLL.

Table 10: Learning time of algorithms on the benchmark datasets

	BPMLL	ML-kNN	Rank-SVM	MLTSVM	SS-MLTSVM
Flags	4.808817	0.050400	0.359997	**0.050300**	0.074584
Birds	16.868832	**0.139376**	4.273635	0.187632	1.566527
Emotions	17.929608	**0.147162**	1.282045	0.384163	1.153655
Scene	154.982600	**1.913366**	5.794408	4.145783	12.102076
Yeast	69.403602	**2.093822**	49.447880	19.581926	38.792532
Average	52.798700	0.868840	12.231600	**4.869937**	10.737873

4.3 Sensitivity Analysis

In this subsection, we investigate the effect of the size of unlabelled samples on the classification performance. In Figs. 1–5, we show the classification performance of our SS-MLTSVM and the MLTSVM on different datasets for different sizes of unlabelled samples.

From Figs. 1–5, we can observe that the classification performance of the SS-MLTSVM is better than that of the MLTSVM in all cases. With the increase of unlabelled samples, the advantages of the SS-MLTSVM become increasingly obvious, because, with the increase of in unlabelled samples, the SS-MLTSVM can make full use of the embedded geometric information, construct a more reasonable classifier, and further improve the classification performance.

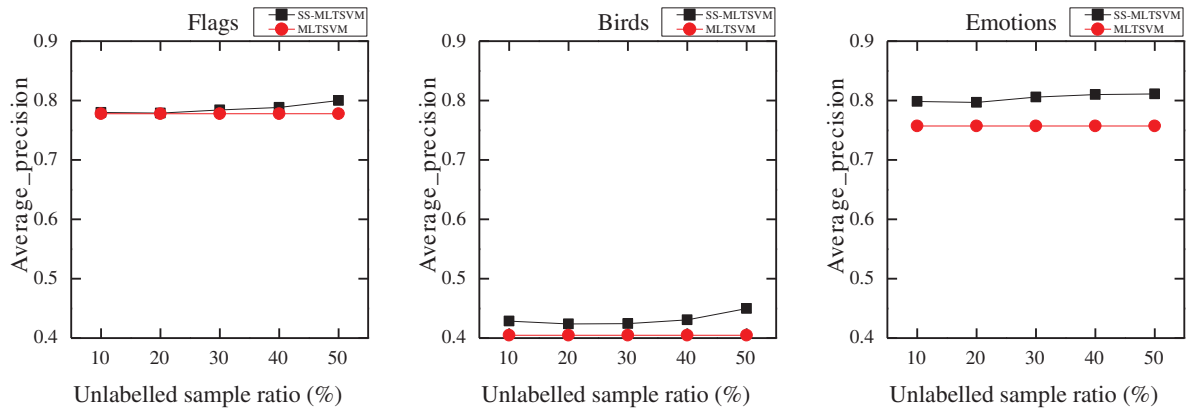

Figure 1: Average precision of the SS-MLTSVM and MLTSVM for unlabelled samples of different sizes

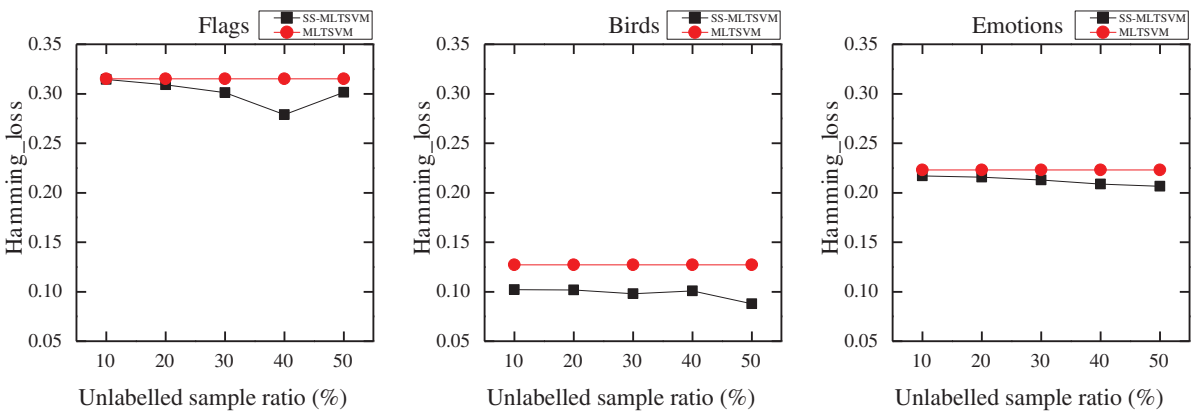

Figure 2: Hamming loss of the SS-MLTSVM and MLTSVM for unlabelled samples of different sizes

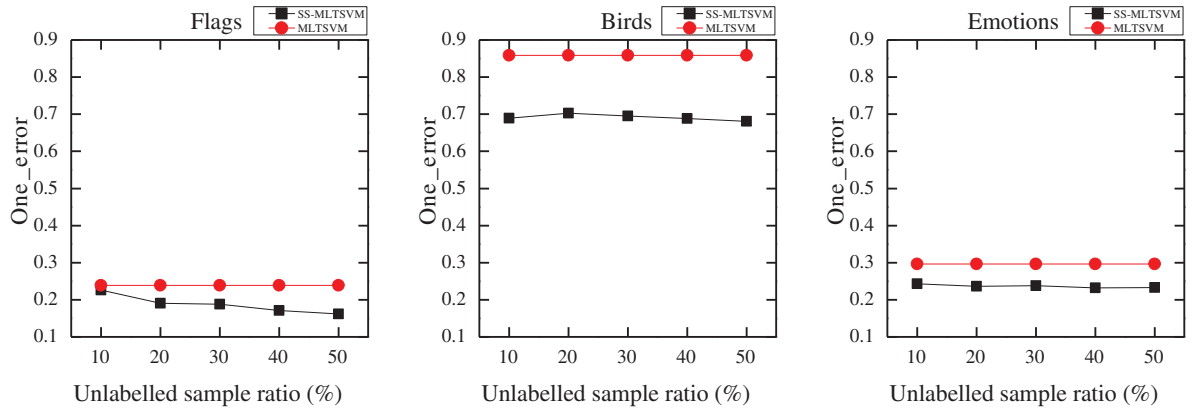

Figure 3: One_error of the SS-MLTSVM and MLTSVM for unlabelled samples of different sizes

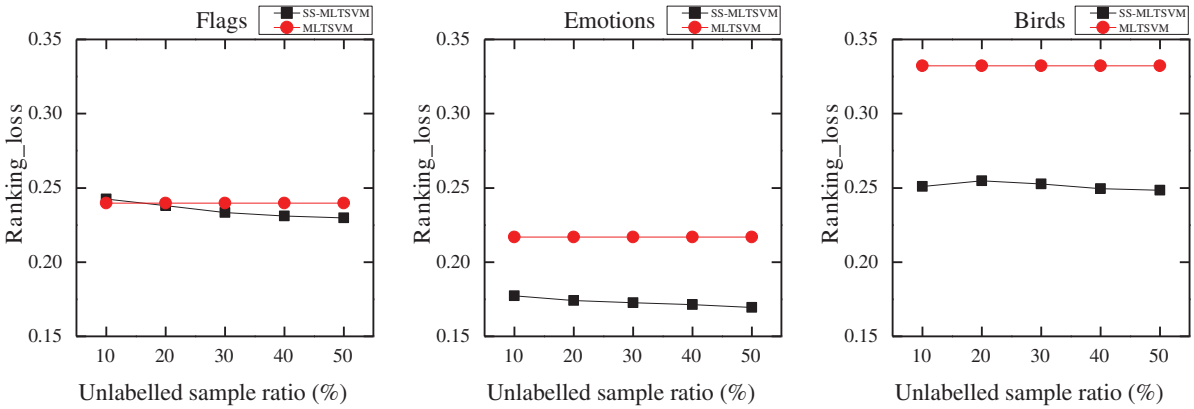

Figure 4: Ranking loss of the SS-MLTSVM and MLTSVM for unlabelled samples of different sizes

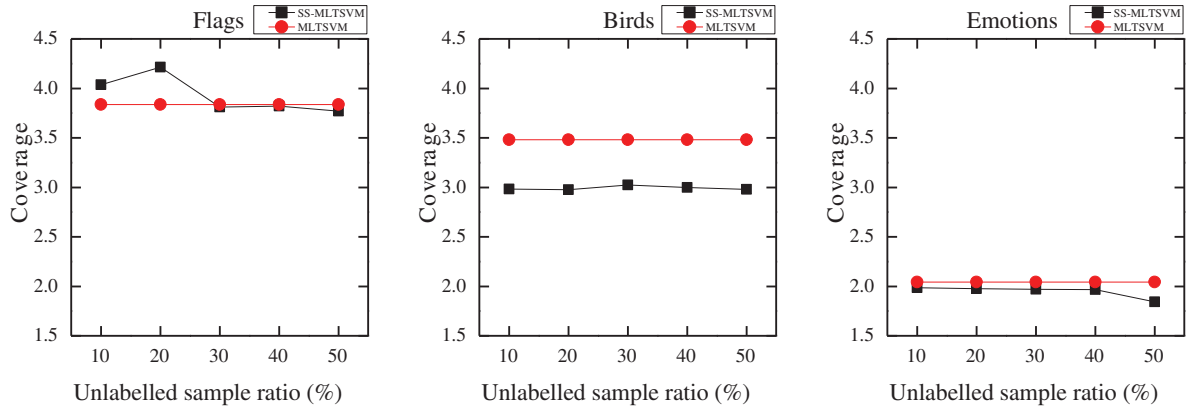

Figure 5: Coverage of the SS-MLTSVM and MLTSVM for unlabelled samples of different sizes

5 Conclusion

In this paper, a novel SS-MLTSVM is proposed to solve semi-supervised multi-label classification problems. By introducing the manifold regularization term into the MLTSVM, we construct a more reasonable classifier and use SOR to speed up learning. Theoretical analysis and experimental results show that, compared with the existing multi-label classifiers, the SS-MLTSVM can take full advantage of

the geometric information embedded in partially labelled and unlabelled samples and effectively solve semi-supervised multi-label classification problems. It should be pointed out that our SS-MLTSVM does not consider the correlation among labels; however, the correlation among labels is very valuable to improve the generalization performance. Therefore, more effective methods of obtaining correlation among labels should be addressed in the future.

Funding Statement: This research was funded in part by the Natural Science Foundation of Liaoning Province in China (2020-MS-281, 20180551048, 20170520248) and in part by the Talent Cultivation Project of the University of Science and Technology Liaoning in China (2018RC05).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] W. J. Chen, Y. H. Shao and C. N. Li, "MLTSVM: A novel twin support vector machine to multi-label learning," *Pattern Recognition*, vol. 52, pp. 61–74, 2016.

[2] H. Peng, J. Li and S. Wang, "Hierarchical taxonomy-aware and attentional graph capsule RCNNs for large-scale multi-label text classification," *IEEE Transactions on Knowledge and Data Engineering*, to be published.

[3] T. Wang, L. Liu and N. Liu, "A multi-label text classification method via dynamic semantic representation model and deep neural network," *Applied Intelligence*, vol. 50, no. 8, pp. 2339–2351, 2020.

[4] Y. Liu, K. Wen and Q. Gao, "SVM based multi-label learning with missing labels for image annotation," *Pattern Recognition*, vol. 78, pp. 307–317, 2018.

[5] Y. Guo, F. Chung and G. Li, "Multi-label bioinformatics data classification with ensemble embedded feature selection," *IEEE Access*, vol. 7, pp. 103863–103875, 2019.

[6] K. Yang, C. She, W. Zhang, J. Yao and S. Long, "Multi-label learning based on transfer learning and label correlation," *Computers, Materials & Continua*, vol. 61, no. 1, pp. 155–169, 2019.

[7] M. R. Boutell, J. Luo, X. Shen and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[8] J. Read, B. Pfahringer, G. Holmes and E. Frank, "Classifier chains for multi-label classification," *Machine Learning*, vol. 85, no. 3, pp. 333–359, 2011.

[9] E. A. Cherman, M. C. Monard and J. Metz, "Multi-label problem transformation methods: A case study," *CLEI Electronic Journal*, vol. 14, no. 1, pp. 1–10, 2011.

[10] J. Fürnkranz, E. Hüllermeier, E. L. Mencía and K. Brinker, "Multi-label classification via calibrated label ranking," *Machine Learning*, vol. 73, no. 2, pp. 133–153, 2008.

[11] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *Proc. of the 18th European Conf. on Machine Learning*, Warsaw, Poland, pp. 406–417, 2007.

[12] M. L. Zhang and Z. H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[13] A. Clare and R. D. King, "Knowledge discovery in multi-label phenotype data, " in *European Conf. on Principles of Data Mining and Knowledge Discovery*, Freiburg, Germany, 42–53, 2001.

[14] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," *Advances in Neural Information Processing Systems*, vol. 14, pp. 681–687, 2001.

[15] N. Ghamrawi and A. McCallum, "Collective multi-label classification, " in *Int. Conf. on Information and Knowledge Management*, Bremen, Germany, 195–200, 2005.

[16] Jayadeva, R. Khemchandani and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 905–910, 2007.

[17] H. Wang and Z. Zhou, "An improved rough margin-based v-twin bounded support vector machine," *Knowledge-Based Systems*, vol. 128, no. 15, pp. 125–138, 2017.

[18] A. Mir and J. A. Nasiri, "KNN-based least squares twin support vector machine for pattern classification," *Applied Intelligence*, vol. 48, no. 12, pp. 4551–4564, 2018.

[19] X. Peng, "A ν-twin support vector machine (ν-TSVM) classifier and its geometric algorithms," *Information Sciences*, vol. 180, no. 20, pp. 3863–3875, 2010.

[20] X. Chen, J. Yang and Q. Ye, "Recursive projection twin support vector machine via within-class variance minimization," *Pattern Recognition*, vol. 44, no. 10–11, pp. 2643–2655, 2011.

[21] Q. Ai, A. Wang, Y. Wang and H. Sun, "An improved Twin-KSVC with its applications," *Neural Computing and Applications*, vol. 31, no. 10, pp. 6615–6624, 2019.

[22] F. Xie and Y. Xu, "An efficient regularized K-nearest neighbor structural twin support vector machine," *Applied Intelligence*, vol. 49, no. 12, pp. 4258–4275, 2019.

[23] Z. Qi, Y. Tian and Y. Shi, "Laplacian twin support vector machine for semi-supervised classification," *Neural Networks*, vol. 35, pp. 46–53, 2012.

[24] J. Xie, K. Hone, W. Xie, X. Gao, Y. Shi *et al.,* "Extending twin support vector machine classifier for multi-category classification problems," *Intelligent Data Analysis*, vol. 17, no. 4, pp. 649–664, 2013.

[25] H. Gu, "A directed acyclic graph algorithm for multi-class classification based on twin support vector machine," *Journal of Information and Computational Science*, vol. 11, no. 18, pp. 6529–6536, 2014.

[26] Y. Xu, R. Guo and L. Wang, "A twin multi-class classification support vector machine," *Cognitive Computation*, vol. 5, no. 4, pp. 580–588, 2013.

[27] C. Angulo, X. Parra and C. Andreu, "K-SVCR: A support vector machine for multi-class classification," *Neurocomputing*, vol. 55, no. 1–2, pp. 57–77, 2003.

[28] Z. X. Yang, Y. H. Shao and X. S. Zhang, "Multiple birth support vector machine for multi-class classification," *Neural Computing and Applications*, vol. 22, no. S1, pp. 153–161, 2013.

[29] Q. Ai, A. Wang, Y. Wang and H. Sun, "Improvements on twin-hypersphere support vector machine using local density information," *Progress in Artificial Intelligence*, vol. 7, no. 3, pp. 167–175, 2018.

[30] Z. Hanifelou, P. Adibi and S. A. Monadjemi, "KNN-based multi-label twin support vector machine with priority of labels," *Neurocomputing*, vol. 322, pp. 177–186, 2018.

[31] M. AzadManjiri, A. Amiri and S. A. Saleh, "ML-SLSTSVM: A new structural least square twin support vector machine for multi-label learning," *Pattern Analysis and Applications*, vol. 23, no. 1, pp. 295–308, 2020.

[32] M. Belkin, P. Niyogi and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

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

[34] M. L. Zhang and Z. H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006.