IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 2, 2021, accepted December 19, 2021, date of publication December 31, 2021, date of current version February 7, 2022.

*Digital Object Identifier 10.1109/ACCESS.2021.3139929*

# LP-MLTSVM: Laplacian Multi-Label Twin Support Vector Machine for Semi-Supervised Classification

**FARHAD GHAREBAGHI**[1] **AND ALI AMIRI**[2]

[1]Department of Computer Engineering, Faculty of Computer and Information Technology Engineering, Islamic Azad University, Qazvin Branch, Qazvin 15195-34199, Iran

[2]Department of Computer Engineering, University of Zanjan, Zanjan 45371-38111, Iran

Corresponding author: Farhad Gharebaghi (farhad.gharebaghi@iauz.ac.ir)

**ABSTRACT** In the machine learning jargon, multi-label classification refers to a task where multiple mutually non-exclusive class labels are assigned to a single instance. Generally, the lack of sufficient labeled training data demanded by a classification task is met by an approach known as semi-supervised learning. This type of learning extracts the decision rules of classification by utilizing both labeled and unlabeled data. Regarding multi-label data, however, current semi-supervised learning methods are unable to classify them accurately. Therefore, with the goal of generalizing the state-of-the-art semi-supervised approaches to multi-label data, this paper proposes a novel two-stage method for multi-label semi-supervised classification. The first stage determines the label(s) of the unlabeled training data by means of a smooth graph constructed using the manifold regularization. In the second stage, thanks to the capability of the twin support vector machine to relax the requirement that hyperplanes should be parallel in classical SVM, we employ it to establish a multi-label classifier called LP-MLTSVM. In the experiments, this classifier is applied on benchmark datasets. The simulation results substantiate that compared to the existing multi-label classification algorithms, LP-MLTSVM shows superior performance in terms of the Hamming loss, average precision, coverage, ranking loss, and one-error metrics.

**INDEX TERMS** Multi-label classification, semi-supervised learning, smooth graph, Graph Laplacian, manifold regularization, twin support vector machine.

## I. INTRODUCTION

Classification is a well-known task in the field of machine learning. Traditionally, machine learning techniques are divided into supervised, unsupervised and semi-supervised.

In supervised learning, one or more labels are assigned to each given data point by the intervention of a supervisor [1]. In supervised learning techniques, a model is constructed and trained with features of train data to predict the class labels of unseen data. Classification and regression algorithms are supervised learning algorithms. In classification, the output of classifier is a discrete number from a predefined limited set [2], while in regression the output of regressor is a continuous value [1], [3]. Support vector machine (SVM) [4], twin support vector machine (TWSVM) [5] and neural networks [6] are well-known examples of supervised learning. These types of learning have been used in a wide range of applications like pattern recognition [7] and text categorization [8].

With respect to the number of class labels of each data point, the classification tasks are divided into single-label or multi-label. For instance, spam email filtering is a single-label classification problem where each instance has a single-label. In this problem, once the classifier learns the features of a spam email, it will be able to distinguish spam from non-spam emails [9].

Multiple-label learning paradigm involves a process in which several labels selected from a set of labels are assigned to a data instance. There are varieties of application that utilize multiple-label learning, music categorization [10] and image annotation [11], to name a few. In practice, music or an image can be associated with several topics.

The associate editor coordinating the review of this manuscript and approving it for publication was Juan Wang.

Learning based on multiple-label data has gained significant attention [12], recently. These efforts can be categorized to algorithm adaptation, problem transformation and ensemble method. Algorithm adaptation strategy extends single-label classifiers to the multi-label data classification. Rank-SVM [13], BPMLL [14], MLRBF [15], MLIBLR [16] and SS-MLLSTSVM [17], belong to this category. Problem transformation strategy involves transforming a multi-label problem to a set of single-label sub-problems. Then a sub-classifier is constructed through transforming the aforementioned single-label classifiers. Binary relevance (e.g., one-against-all) [18] and calibrated label ranking (e.g., one-against-one) [19] are features of this category. Ensemble strategy constructs a multi-label classifier by combining several single-label classifiers. Generally, this strategy provides the most efficient solution for multi-label problems. Boosting-based methods such as Adaboost [20] and an ensemble of classifier chains [3], [21] are examples of this strategy.

Unsupervised learning algorithms deal with unlabeled data [22]. These techniques seek to recognize a meaningful resemblance and pattern among data. Data clustering is a major goal of the unsupervised learning. Data which lay in a cluster bear a maximum similarity among themselves and a maximum dissimilarity with other clusters at the same time. K-means [23] and K-medoids [24] are two examples of this technique.

Semi-supervised learning (SSL) is conducted by combining the two aforementioned learning approaches (i.e., supervised and unsupervised learning) [25]. The principal idea behind the constructing a classifier by means of SSL is to exploit the unlabeled data which are excessive in number accompanied by a few available labeled data [26]. The SSL technique establishes a model in which not only are instance labels employed but also meaningful patterns among instances are utilized [27]. One approach to implement this type of learning is simply neglecting unlabeled data [28]. However, this approach may result in overfitting [29]. As already mentioned, the objective of SSL is to utilize all available data for the construction of the intended model [26]. Currently, there are varieties of applications that deal with the data showing multi-label and semi-supervised trait. Video surveillance [30], [31] and protein 3D structure prediction [32] are two well-known examples of such applications.

Generally, SSL is conducted through either inductive [33] or transductive [27] approaches. Taking into account that SSL consists of both labeled and unlabeled training data, two main goals need to be achieved. The first goal is to predict the label of test data and the second goal involves predicting label of those training data with no label. These two goals are reached by means of inductive and transductive learning, respectively. Denoting labeled data and unlabeled data are denoted by $\{(x_i, y_i)\}_{i=1}^{L}$ and $\{(x_j)\}_{j=L+1}^{L+U}$, respectively. Inductive SSL seeks to train a function$f : x \rightarrow y$ in such a way that this function shows better performance in predicting unseen data compared to the case where unlabeled data are solely exploited. Analogous to the supervised learning, an established way to assess performance of semi-supervised classifier is to use these test data $\{(x_k, y_k)\}_{k=1}^{m}$ that do not participate in training phase.

Transductive SSL trains a function $f : x^{L+U} \rightarrow y^{L+U}$ so that the label of unlabeled training instances can be predicted by means of this function. It is worth mentioning that function $f$ can only be employed for training data set and not to predict any data outside of this set.

Semi-supervised algorithms are grounded on the cluster assumption and the manifold assumption in leveraging unlabeled data. The cluster assumption implies that similar data have the same class label, so the decision boundary passes from a low-density region. According to the manifold assumption, data can be represented by a Graph Laplacian. This graph is utilized by a classifier that assigned the same label to similar instances. Almost all the SSL algorithms are based on either one or both assumptions. For example, maximum margin semi-supervised learning method such as transductive support vector machine (TSVM) [34] and semi-supervised SVM (S3VM) [35], LapSVM [36] and LapTSVM [37], leverage the cluster assumption. Graph-based semi-supervised classification methods like label propagation [38] and manifold regularization [39] adopt the manifold assumption.

Graph-based semi-supervised learning [40] schemes are mainly inductive, which makes them incompetent in practical applications, where predicting unseen instance is required. To tackle this incompetency, recent graph-based multi-label semi-supervised schemes follow transductive approach among which are image retrieval [41] and web spam identification [42] applications.

The promising performance of the MLTSVM [43] has led to its widespread application in the development of multi-label classifiers. MLTSVM, however, entails a number of drawbacks as follows. It ignores unlabeled data during the classifier construction process, and consequently it merely shows favorable performance in the cases where data samples have a limited number of labels. Moreover, this construction process involves solving quadratic programming problems, as a result of which the learning is decelerated. Therefore, we propose a novel two-stage method called Laplacian Multi-Label Twin Support Vector Machine (LP-MLTSVM) that addresses the aforementioned drawbacks by utilizing both labeled and unlabeled data. In its first stage, leveraging a Graph Laplacian an undirected weighted graph is constructed where training instances constitute its vertices and the weight of each edge reflects the similarity of its corresponding vertices. The second stage constructs sub-classifiers exploiting MLTSVM and manifold regularizer [44]. In other words, the major contributions of the proposed method to address shortcomings of the MLTSVM are:

- Utilizing both labeled and unlabeled data by predicting a label ( or labels ) for each unlabeled instance, and avoiding neglecting unlabeled instance.

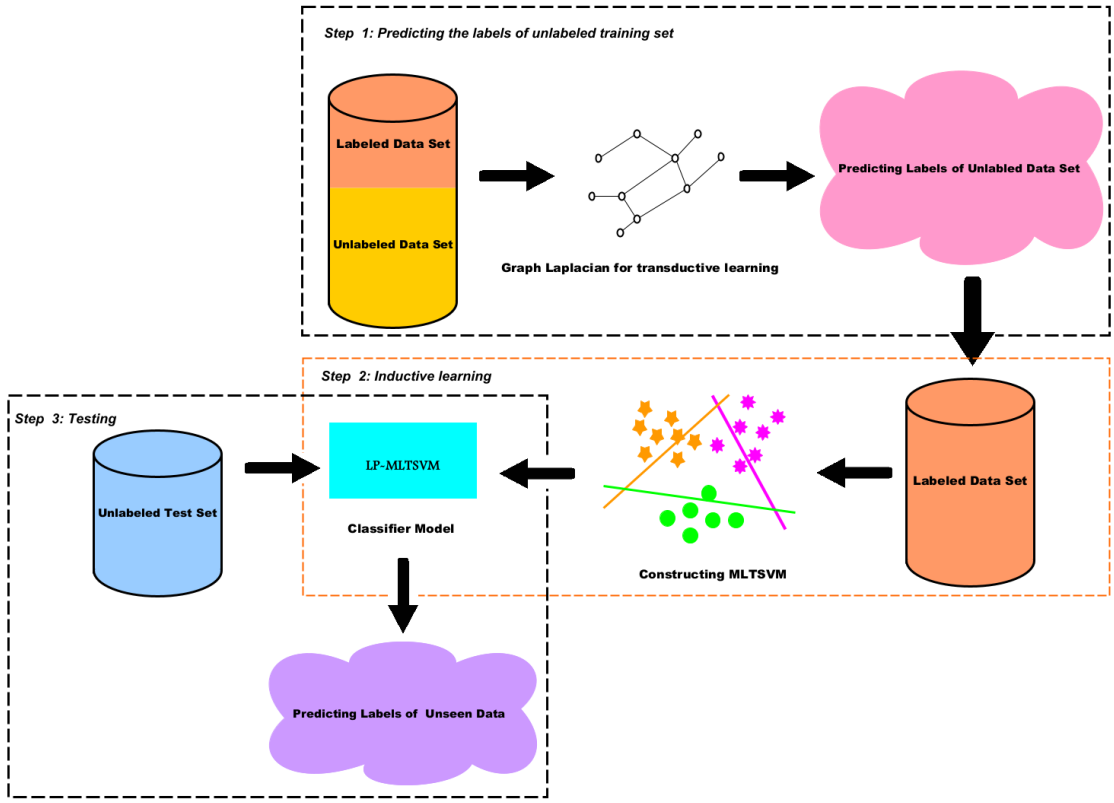


**FIGURE 1.** The schematic explanation of the porposed method.

- Constructing sub-classifiers exploiting MLTSVM and manifold regularizer.
- Employing successive over-relaxation (SOR) to enhance the learning speed.

Fig 1 exhibits the steps of the proposed method. The simulation results show that compared to the existing multi-label classification algorithms, LP-MLTSVM shows superior performance in terms of the Hamming loss, average precision, coverage, ranking loss and one-error.

The rest of this paper is structured as follows: section II presents the related works. The proposed scheme is elaborated in section III. Section IV is devoted to results and discussion. Finally, section V concludes the paper.

## II. RELATED WORKS

Since the proposed method is based on SVM, in this section we briefly introduce the SVM and some extensions.

### A. SUPPORT VECTOR MACHINE (SVM)

Assume $\mathcal{T}_c$ is a set of training instances of a binary classifier defined as

$$\mathcal{T}_c = \left\{ \left( \mathcal{X}^i, \mathcal{Y}_i \right), \quad (i = 1, \ldots, m) \right\}, \tag{1}$$

where $\mathcal{X}^i \epsilon \Re$ and $\mathcal{Y}_i \epsilon \{-1, 1\}$. Let $m1$ denote samples labeled by $+1$ and similarly $m2$ represent samples with -1 label. Accordingly, matrix $A_{m_1 \times n}$ and $B_{m_2 \times n}$ are constructed such that $i - th$ row represents a data sample labeled as $+1$ and $-1$, respectively and we have $m = m1 + m2$.

Original SVM [4] separated the two sets of data denoted by$A$ and $B$ through establishing two parallel hyperplanes. These two hyperplanes are obtained using the following equations.

$$Aw^T + eb = +1, \tag{2}$$
$$Bw^T + eb = -1, \tag{3}$$

where $b \epsilon \Re$, $w \epsilon \Re^n$, and $e$ is appropriate vector of ones. This approach seeks to establish a trade-off between misclassification error and the decision margin. This trade-off can be formulated as the following optimization problem:

$$\begin{aligned} &\min_{(w,b,\varepsilon)} \frac{1}{2} w^T w + C \sum_{i=1}^{m} \varepsilon_i \\ &s.t: \mathcal{Y}_i \left( w^T x^i + b \right) \varepsilon_i \geq \mathbf{1}, \\ &\varepsilon_i \geq 0, \quad (i = 1, \ldots, m) \end{aligned} \tag{4}$$

where $\varepsilon_i$ and $C$ are slack variable and a user defined penalty factor, respectively. The Wolf dual of (4) can be expressed as

$$\begin{aligned} &max \ -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \mathcal{Y}_i \mathcal{Y}_j \left( x^i \right)^T x^j \alpha_i \alpha_j + \sum_{j=1}^{m} \alpha_j \\ &s.t: \sum_{i=1}^{m} \mathcal{Y}_i \alpha_i = 0 \\ &0 \leq \alpha_i \leq C, \quad (i = 1, \ldots, m) \end{aligned} \tag{5}$$

Let $\alpha^* = (\alpha_1^*, \ldots, \alpha_m^*)$ be the optimal solution of (5). Karush–Kuhn–Tucker (KKT) conditions yield a hyperplane represented as $x^T w^* + eb^*$, where

$$w^* = \sum_{i=1}^{m} \alpha_i^* \mathcal{Y}_i x^i, \tag{6}$$

$$b^* = (\mathcal{Y}_i - \sum_{i=1}^{N_{SV}} \alpha_i^* \mathcal{Y}_i) * \frac{1}{N_{SV}}. \tag{7}$$

Here $x^i$ and $\alpha_i$ are data samples used as support vectors and their respective multipliers. $N_{SV}$ denotes the number of support vectors holding $0 < \alpha_i < C$. A new sample is labeled as $+1$ or $-1$ according to the proposed classifier.

### B. TWIN SUPPORT VECTOR MACHINE (TWSVM)

TWSVM [5] has been proposed to classify binary data. It relaxes the requirement that the hyperplanes should be parallel in the classical SVM. It establishes two non-parallel hyperplanes in such a way that each of them has the highest possible distance from a class and is the lowest possible distance from another one. Unlike standard SVM, which solves a large dimensional quadratic programming problem (QPP), TWSVM arranges to solve two reduced dimensional QPPs, and accordingly it performs faster than SVM. Indeed, each of these reduced problems follows the relations of the standard SVM where each data sample can only appear in the constraint of just one problem. Those two hyperplanes are

$$Aw_+^T + eb_+ = +1, \tag{8}$$

$$Bw_-^T + eb_- = -1. \tag{9}$$

where $w_+, w_- \epsilon \mathbb{R}^n$ and $b_+, b_- \epsilon \mathbb{R}^n$ also as well as $e$ represent a vector of ones with a proper size. Construction of hyperplane leveraged by TWSVM involves solving the following optimization problems.

$$\begin{aligned} &min\ \frac{1}{2}\|Aw_+ + e_+b_+\|_2^2 + C_1 e_-^T \varepsilon \\ &s.t:\ -(Bw_+ + e_-b_+) + \varepsilon \geq e_-, \quad \varepsilon \geq 0 \end{aligned} \tag{10}$$

$$\begin{aligned} &min\ \frac{1}{2}\|Bw_- + e_-b_-\|_2^2 + C_2 e_+^T \eta \\ &s.t:\ (Aw_- + e_+b_-) + \eta \geq e_+, \quad \eta \geq 0 \end{aligned} \tag{11}$$

where $C_1, C_2 \geq 0$ are the error penalty, and $e_-, e_+$ are vectors of ones with a proper size. Applying some algebra on (10) and (11) yields

$$\begin{aligned} &max\ e_-^T\alpha - \frac{1}{2}\alpha^T G \left(H^T H\right)^{-1} G^T \alpha \\ &s.t: 0 \leq \alpha \leq C_1 e_- \end{aligned} \tag{12}$$

$$\begin{aligned} &max\ e_+^T\beta - \frac{1}{2}\beta^T P \left(Q^T Q\right)^{-1} P^T \beta \\ &s.t: 0 \leq \beta \leq C_2 e_+ \end{aligned} \tag{13}$$

where $C_1, C_2$ and $e_-, e_+$ are Lagrange multipliers and the followings relations are yielded $G = \left[B\ e_-\right]$, $H = \left[A\ e_+\right]$, $P = \left[A\ e_+\right]$ and $Q = \left[B\ e_-\right]$.

The non-parallel hyperplanes (8), (9) are obtained from the solutions $\alpha$, $\beta$ of (12), (13) by

$$v_1 = -\left(H^T H\right)^{-1} G^T \alpha \quad where\ v_1 = \left[w_+^T b_+\right]^T, \tag{14}$$

$$v_2 = -\left(Q^T Q\right)^{-1} P^T \beta \quad where\ v_2 = \left[w_-^T b_-\right]^T. \tag{15}$$

Depending on which of two hyperplanes it is close to, a new data point is labeled as +1 or -1, that is

$$f(x) = arg\ min\ d_{\pm}(x), \tag{16}$$

$$d_{\pm}(x) = |\ w_{\pm}^T x\ + b_{\pm}\ |\ . \tag{17}$$

where $|\ .\ |$ gives the distance of point $x$ from planes $w_{\pm}^T x + b_{\pm}$.

### C. MULTI-LABEL TWSVM (MLTWSVM)

Let us consider a multi-label classification problem in $\mathfrak{R}^n$ consisting $K$ classes [45]. Suppose $\mathcal{X}_{L+i} \epsilon \mathfrak{R}$, $(i = 1, \ldots, U)$ and $\mathcal{Y}$ are input samples and label set, respectively. A multi-label task involves constructing a decision function $h(.) : \mathcal{X} \rightarrow 2^{\mathcal{Y}}$ according to the training samples, which is

$$\mathcal{T}_c = \{(x_{L+i}\, y_i) \mid 1 \leq i \leq U\ ,\ 1 \leq j \leq K\}, \tag{18}$$

$x_{L+i} \epsilon \mathcal{X}$ denotes training samples and $y_i \epsilon \mathcal{Y}$ is the label associated with class expressed as $y_j = [y_{j1}, \ldots, y_{jk}, \ldots, y_{jK}]^T$ where

$$y_{jk} = \begin{cases} +1, & if\ x_i\ belings\ to\ k-th\ class; \\ -1, & if\ x_i\ not\ belings\ to\ k-th\ class; \\ 0, & Otherwise. \end{cases} \tag{19}$$

In this problem, $K$ non-parallel hyperplanes represented as $f_k(x) = w_k^T + b_k = 0$ are established such that $K - th$ hyperplane is as close as possible to the samples of class $K$ while it is as far as possible from samples of other classes. Here, $w_k$ and $b_k$ are normal vector and bias terms of $K - th$ hyperplane, respectively. The classifier $h(x) \subseteq \mathcal{Y}$ is associated with an unseen instance $x \epsilon \mathcal{X}_{L+i}$ predicting its label.

### D. MANIFOLD REGULARIZATION (MR)

Regularization is an overfitting suppression technique [46]. Recently it is employed in SSL problems [47].

Let $(x_{L+1} \ldots, x_{L+U})$ be the set of mixture of labeled and unlabeled data where $x_{L+i} \epsilon \mathfrak{R}^n$, $i = 1, \ldots, U$ and labels are distributed according to, pdf, $P_x$. In MR, labels are distributed according to Riemannian manifold [39] in a way that more geometrically close data resemble more to their respective label. Generally, data are stored in a matrix $\mathcal{M} \epsilon \mathfrak{R}^{(L+U)\times n}$. Applying MR requires representing training sample as a weighted graph where these samples are its vertices [43]. The adjacency matrix of this graph is denoted as $w_{ij}$ where its elements capture the similarity of the training samples. Accordingly, the elements of $i - th$ row and $j - th$ column are calculated as

$$w_{ij} = \begin{cases} exp(-\|x_i - x_j\|_2^2 / 2\sigma, & if\ x_i, x_j\ are\ neighbor; \\ 0, & Otherwise, \end{cases} \tag{20}$$

where $\|x_i - x_j\|_2^2$ denotes the Euclidean distance between sample $x_i$ and $x_j$. Also, $\sigma$ is known as the bandwidth parameter that controls the decreasing rate of the weight.

Graph Laplacian $L$ is obtained as $L = D - W$, where $D$ and $W$ are two $(L + U) \times (L + U)$ matrices. $W$ is the symmetry matrix which can be constructed according to (20). $D$ is an orthogonal matrix whose elements are defined as $D_{ii} = \sum_{i=1}^{l+u} w_{ij}$.

In the Graph Laplacian, nodes are sorted in such a way that at first, labeled nodes are located followed by the unlabeled ones. This matrix can be divided into sub matrices as follows:

$$L = \begin{bmatrix} L_{ll} & L_{lu} \\ L_{ul} & L_{uu} \end{bmatrix}. \tag{21}$$

According to a Graph Laplacian $L$, a prediction function $f$ can be articulated as:

$$f(x_i) = y_i, \tag{22}$$

$$f(x_j) = \frac{\sum_{k=1}^{l+u} w_{jk} f(x_k)}{\sum_{k=1}^{l+u} w_{jk}}, \quad j = l+1 \cdots l+u. \tag{23}$$

In this function, the values of the labeled instances are the respective label of these instances. However, the value of each unlabeled instance is the average of its neighbor's weight, which can be obtained from the following optimization problem:

$$\begin{aligned} &min \sum_{i,j=1}^{l+u} w_{ij}(f(x_i) - f(x_j))^2 \\ &s.t: f(x_i) = y_i \quad for\ i = 1 \cdots l. \end{aligned} \tag{24}$$

The function $f$ generates values in the range $[-1, 1]$. By defining the threshold value as 0, the generated values of this function can be mapped to the labels of unlabeled instances. The optimization problem (24) can be reduced to:

$$\frac{1}{2} \sum_{i,j=1}^{l+u} w_{i,j}(f(x_i) - f(x_j))^2 = f^T L f. \tag{25}$$

Solving this problem yields:

$$f_l = y_l, \tag{26}$$

$$f_u = -L_{uu}^{-1} L_{ul} y_l. \tag{27}$$

For the sake of clarification, an example is provided. Let the graph of Fig 2 be a training data whose nodes are numbered from left to right. Among these nodes are only those numbered as 1 and 7 which have label $+1$ and $-1$, respectively, and other nodes are unlabeled. By moving nodes that are associated with labeled instance to the front of this graph, the order of nodes is arranged as (1, 7, 2, 3, 4, 5, and 6). Applying (26) and (27) on matrices $L, D$ and $W$yields $f_u = (\frac{2}{3}, \frac{1}{3}, 0, \frac{-1}{3}, \frac{-2}{3})^T$ where $y_l = (1, -1)^T$.

Setting the threshold value as 0, the labels of unlabeled instances are determined. Accordingly, the labels of nodes 2 and 3 will be $+1$ and nodes 5 and 6 are labeled with $-1$.

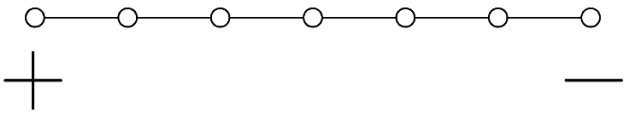

**FIGURE 2.** Chain graph comprised of seven nodes among which only nodes 1 and 7 are labeled $+1$ and $-1$, respectively. The other nodes do not have any label.

The aforementioned example shows that at the first stage of the proposed algorithm, the label of the unlabeled training data are predicted, leveraging a Graph Laplacian. In the second phase utilizing TWSVM a multi-label semi-supervised classifier called LP-MLTSVM is established through which the labels of the unseen data can be predicted.

### E. SS-MLLSTSVM

Semi-supervised multi-label least square (SS-MLLSTSVM) [17] employs the concept of the least square to increase the learning speed of the MLTSVM. Unlike MLTSVM, it uses manifold regularization to fully utilize both labeled and unlabeled data. SS-MLLSTSVM can be expressed as the following optimization problem:

$$\begin{aligned} min\ &\frac{1}{2}\left(A_k w_k + e_{A_k} b_k\right)^2 + \frac{1}{2} c_{k2}\left(\|w_k\|^2 + b_k^2\right) \\ &+ \frac{1}{2} c_{k1} \xi_{B_k}^T \xi_{B_k} \\ &+ \frac{1}{2} c_{k3} \left(Tw_k + eb_k\right)^T L \left(Tw_k + eb_k\right), \\ s.t:\ &-\left(B_k w_k + e_{B_k} b_k\right) + \xi_{B_k} = e_{B_k}, \end{aligned} \tag{28}$$

where $c_{ki}$ are the penalty parameters, $\xi_{B_k}$ is the salck variable, and $L$ is the Laplace matrix. The samples belonging to the $k$th class are denoted by $A_k$ and the samples not belonging to the $k$th class are shown by $B_k$. Like the linear case, the linear SS-MLLSTSVM is extended to the nonlinear case using approximate kernel generating surface. The optimization problem of nonlinear SS-MLLSTSVM is as follows:

$$\begin{aligned} min\ &\frac{1}{2}\left(K\left(A_k, T^T\right) w_k + e_{A_k} b_k\right)^2 + \frac{1}{2} c_{k1} \xi_{B_k}^T \xi_{B_k} \\ &+ \frac{1}{2} c_{k3} \left(K\left(T, T^T\right) w_k + eb_k\right)^T \\ &\times L \left(K\left(T, T^T\right) w_k + eb_k\right) \\ &+ \frac{1}{2} c_{k2}\left(\|w_k\|^2 + b_k^2\right), \\ s.t:\ &-\left(K\left(B_k, T^T\right) w_k + e_{B_k} b_k\right) + \xi_{B_k} = e_{B_k}, \end{aligned} \tag{29}$$

where $K(.,.)$ is a suitable kernel function.

## III. LAPLACIAN MULTI-LABEL TWIN SUPPORT VECTOR MACHINE FOR SEMI-SUPERVISED CLASSIFICATION

TWSVM supposes that each sample can take only one label [5]. However, samples can have multiple labels. This paper extends TWSVM to multi-label problems.

### A. LINEAR LP-MLTSVM

Inspired by TWSVM, the proposed scheme employs square loss function and Hing loss function as:

$$V_k = (x_i, y_i, f_k) = ((A_k.w_k) + b_k)^2 + max\,(0, 1 - f_k(A_k))\,, \tag{30}$$

$$V_{\overline{k}} = (x_i, y_j, f_{\overline{k}}) = \left((A_{\overline{k}}.w_{\overline{k}}) + b_{\overline{k}}\right)^2 + max\left(0, 1 - f_{\overline{k}}(A_{\overline{k}})\right). \tag{31}$$

$A_k$ represents data samples with label $k$ and $A_{\overline{k}}$ are other data samples. The decision functions of these problems can be expressed as

$$f_k(x) = (w_k.x) + b_k, \tag{32}$$

$$f_{\overline{k}}(x) = (w_{\overline{k}}.x) + b_{\overline{k}}. \tag{33}$$

Also the regularization terms are defined as

$$\|f_k\|^2_{\mathcal{H}} = \frac{1}{2}\left(\|w_k\|_2^2 + b_k^2\right), \tag{34}$$

$$\|f_{\overline{k}}\|^2_{\mathcal{H}} = \frac{1}{2}\left(\|w_{\overline{k}}\|_2^2 + b_{\overline{k}}^2\right). \tag{35}$$

Accordingly manifold regularizations are

$$\|f_k\|^2_{\mathcal{M}} = \frac{1}{(L+U)^2}\sum_{i,j=1}^{L+U} w_{ij}\left(f_k(x_i) - f_k(x_j)\right)^2 = f_k^T L f_k \tag{36}$$

$$\|f_{\overline{k}}\|^2_{\mathcal{M}} = \frac{1}{(L+U)^2}\sum_{i,j=1}^{L+U} w_{ij}\left(f_{\overline{k}}(x_i) - f_{\overline{k}}(x_j)\right)^2 = f_{\overline{k}}^T L f_{\overline{k}}. \tag{37}$$

where $w_{ij}$ is the element of data adjacency matrix in which higher similarity between $x_i, x_j$ will lead to a larger $w_{ij}$. Also $L$ is Graph Laplacian as

$$f_k = [f_k(x_1), \ldots, f_k(x_{L+U})]^T = Mw_k + eb_k, \tag{38}$$

$$f_{\overline{k}} = \left[f_{\overline{k}}(x_1), \ldots, f_{\overline{k}}(x_{L+U})\right]^T = Mw_{\overline{k}} + eb_{\overline{k}}. \tag{39}$$

where $M \epsilon \Re^{(L+U)\times n}$ consists of all labeled and unlabeled data, and e vectors of ones with an appropriate size.

For kernel function $k(.,.)$, which is associated with a reproducing kernel Hilbert space $\mathcal{H}_k$, the decision function can be obtained by minimizing

$$f^* = arg\ \min_{f\epsilon\mathcal{H}} \sum_{i=1}^{L+U} V(x_i, y_i, f) + \gamma_{\mathcal{H}}\|f\|^2_{\mathcal{H}} + \gamma_{\mathcal{M}}\|f\|^2_{\mathcal{M}}. \tag{40}$$

where $f$ is an unknown decision function, $V$ represents some loss functions on the labeled data, and $\gamma_{\mathcal{H}}$ is the weight of $\|f\|^2_{\mathcal{H}}$ that controls the complexity of $f$ in reproducing the Kernel Hilbert space, $\gamma_{\mathcal{M}}$ is the weight of $\|f\|^2_{\mathcal{M}}$ and controls the complexity of the function in the intrinsic geometry of marginal distribution, and $\|f\|^2_{\mathcal{M}}$ is able to penalize $f$ along the Riemann manifold $\mathcal{M}$.

Substituting (30), (31), (34) and (35) into (40), the primal problems of linear LP-MLTSVM can be written as

$$\begin{aligned}\min_{(w_k,b_k,\xi)} &\left\{\frac{1}{2}\|A_k w_k + e_k b_k\|_2^2 + c_k e_{\overline{k}}^T \xi\right.\\ &+ \frac{1}{2}\lambda_k(\|w_k\|_2^2 + b_k)\\ &\left.+ \gamma_k(w_k^T M^T + e_k^T b_k)L(Mw_k + e_k b_k)\right\}\\ s.t:\ &-(A_{\overline{k}} w_k + e_k b_k) + \xi \geq e_{\overline{k}}, \xi \geq 0,\\ &k = 1, 2, \ldots, K.\end{aligned} \tag{41}$$

The Lagrangian corresponding to the problem (41) is given by

$$\begin{aligned}L(\Theta) = &\frac{1}{2}(A_k w_k + e_k b_k)^T(A_k w_k + e_k b_k) + c_k e_{\overline{k}}^T \xi\\ &+\frac{1}{2}\lambda_k(\|w_k\|_2^2 + b_k^2)\\ &+\frac{1}{2}\gamma_k(w_k^T M^T + e_k^T b_k)L(Mw_k + e_k b_k)\\ &-\alpha_k^T(-(A_{\overline{k}} w_k + e_{\overline{k}} b_k) + \xi - e_{\overline{k}}) - \beta_k^T \xi.\end{aligned} \tag{42}$$

Where $\Theta = \{w_k, b_k, \xi, \alpha_k, \beta_k\}$, $\alpha_k = (\alpha_{k_1}, \ldots, \alpha_{k_{m_1}})^T$, $\beta_k = (\beta_{k_1}, \ldots, \beta_{k_{m_1}})^T$ are the Lagrange multipliers for class $k$. The dual problem can be formulated as

$$\begin{aligned}&max\ L(\theta)\\ &s.t:\ \underset{(w_k,b_k,\xi)}{L(\theta)} = 0,\ \alpha_k, \beta_k \geq 0.\end{aligned} \tag{43}$$

From (43), we obtain

$$\begin{aligned}\underset{w_k}{\nabla L} = &A_k^T(A_k w_k + e_k b_k) + \lambda_k w_k\\ &+\gamma_k M^T L(Mw_k + e_k b_k) + A_{\overline{k}}^T \alpha_k = 0,\end{aligned} \tag{44}$$

$$\begin{aligned}\underset{b_k}{\nabla L} = &e_k^T(A_k w_k + e_k b_k) + \lambda_k b_k\\ &+\gamma_k e_k^T L(Mw_k + e_k b_k) + e_{\overline{k}}^T \alpha_k = 0,\end{aligned} \tag{45}$$

$$\underset{\xi}{\nabla L} = c_k e_{\overline{k}} - a_k - \beta_k = 0. \tag{46}$$

Since $\beta_k \geq 0$, (46) turns out to be $0 \leq \alpha_k \leq c_k e_{\overline{k}}$. Next, combining (44) and (45) leads to

$$\begin{aligned}&\begin{pmatrix}A_k^T\\ e_k^T\end{pmatrix}[A_k\ e_k]\begin{pmatrix}w_k\\ b_k\end{pmatrix} + \lambda_k\begin{pmatrix}w_k\\ b_k\end{pmatrix}\\ &+\gamma_k\begin{pmatrix}M^T\\ e^T\end{pmatrix}L\,[M\ e_k]\begin{pmatrix}w_k\\ b_k\end{pmatrix} + \alpha_k\begin{pmatrix}A_{\overline{k}}^T\\ e_{\overline{k}}^T\end{pmatrix} = 0.\end{aligned} \tag{47}$$

Let $H = [A_k\ \ e_k]$, $J = [M\ \ e_k]$, $G = [A_{\overline{k}}\ \ e_{\overline{k}}]$ and $V_k = [w_k^T\ e_k^T]^T$.

Equation (44) can be rewritten as

$$\begin{aligned}&(H^T H + \lambda_k I + \gamma_k J^T L J)V_k + G^T \alpha_k = 0\\ &i.e: V_k = -(H^T H + \lambda_k I + \gamma_k J^T L J)^{-1}(G^T \alpha_k).\end{aligned} \tag{48}$$

where $I$ is an identity matrix of appropriate dimensions. Substituting (48) into (43), we obtain the Wolf dual of (41) as follows:

$$\begin{aligned}&max\ e_{\overline{k}}^T \alpha_k - \frac{1}{2}G(H^T H + \lambda_k I + \gamma_k J^T L J)^{-1}(G^T \alpha_k).\\ &s.t: 0 \leq \alpha_k \leq c_k e_{\overline{k}}\end{aligned} \tag{49}$$

Problem presented as (49) is a convex optimization problem that can be solved by quadratic programming technique to yield $\alpha$. It is relaxed as

$$Q = G(H^T H + \lambda_k I + \gamma_k J^T L J)^{-1} G^T, \tag{50}$$

$$max\ e_{\overline{k}}^T \alpha_k - \frac{1}{2}\alpha_k^T Q \alpha_k.$$

$$s.t: 0 \le \alpha_k \le c_k e_{\overline{k}} \tag{51}$$

Predicting a label for an unseen data $x \epsilon \Re^n$involves using

$$f(x) = arg \underset{k=1\ldots K}{min}\, d_k(x). \tag{52}$$

where

$$\underset{k=1\ldots K}{d_k(x)} = \frac{\mid w_k^T + b_k \mid}{\|w_k\|} < \Delta_k. \tag{53}$$

If a new sample $x$ is close enough to the proximal hyperplane of $k$, its corresponding label is assigned to this sample.

We can summarize the steps of the proposed successive over relaxation [48] to be employed in LP-MLTSVM in Algorithm-1.

This algorithm obtains $\alpha$ used in LP-MLTSVM classifier as presented in algorithm-2.

**Algorithm 1** The Successive Over-Relaxation Algorithm

**Input:**

The penalty parameter $C_k$, the relaxation factor $\omega \epsilon (0, 2)$
And the matrix $Q$ is defined by (50).

**Output:**

The optimal solution $\alpha_k$ for the problem.

1: Initialize iterate $i = 0$ and start with any $\alpha_k^0 \epsilon \Re^{L+U}$
2: Split $Q = L+D+L'$, where $L$ is the strictly lower triangular matrix and $D$ is the diagonal matrix
3: while $\left(\|\alpha^{i+1} - \alpha^i\| < 10^{-6}\right)$do
4: Compute $\alpha^{i+1} = \alpha^i + \omega\Delta_\alpha$, where $\Delta_\alpha$is given by
$\Delta_\alpha = -D^i\left(Q\alpha^i - e + L^T(\alpha^{i+1} - \alpha^i)\right)$.
5: Project $\alpha^{i+1}$to the feasible range $[0.C_k]$.
6: end while.

**Algorithm 2** LP-MLTSVM Algorithm

**Input:**

Labeled data $L \epsilon \{(x_i, y_i)\}_{i=1}^l$,unlabeled data $U \epsilon \left\{x_j\right\}_{j=l+1}^{l+u}$.

**Output:**

Label assigned to the unlabeled data.

1: Construct graph according to similarity function $w_{ij} = \frac{exp(-\|x_i - x_j\|_2^2)}{2\sigma}$.
2: Compute diagonal matrix $D_{ii} = \sum_{i=1}^{l+u} w_{ij}$.
3: Compute Laplacian matrix $L = D - W$.
4: Apply $L$ according to (41).
5: Obtain $\alpha$ according to SOR function.
6: Construct $K$ non-parallel proximal hyperplane.

### B. NONLINER LP-MLTSVM

Now we extend the linear LP-MLTSVM to the nonlinear case. Like linear case, the cost function of the errors $V_k = (x_i, y_j, f_k)$ and $V_{\overline{k}} = (x_i, y_j, f_{\overline{k}})$ can be expressed as (32) and (33). The decision function can be written as $f_k(x) = (w_k.\phi(x)) + b_k$, $f_{\overline{k}}(x) = (w_{\overline{k}}.\phi(x)) + b_{\overline{k}}$, where $\phi(x)$ is a nonlinear mapping from low dimensional space to a higher dimensional Hilbert space $\mathcal{H}$. According to Hilbert space theory, $w_k$ and $w_{\overline{k}}$ can be expressed as $w_k = \sum_{i=1}^{L+U} \phi(x_i)\lambda_k = \phi(M)\lambda_k$, $w_{\overline{k}} = \sum_{i=1}^{L+U} \phi(x_i)\lambda_{\overline{k}} = \phi(M)\lambda_{\overline{k}}$. For the nonlinear case, LP-MLTSVM construct K approximate kernel generating surfaces. These kernel generated surfaces are:

$$K(X^T, M)\lambda_k + b_k = 0, \tag{54}$$

$$K(X^T, M)\lambda_{\overline{k}} + b_{\overline{k}} = 0. \tag{55}$$

where $K(.,.)$ is a suitable kernel function defined as $K(x_i, x_j) = (\phi(x_i), \phi(x_j))$. By means of the kernel matrix $K$ and relevant coefficient, $\lambda_k$,$\lambda_{\overline{k}}$ the regularization term $\|f_k\|_{\mathcal{H}}^2$ and $\|f_{\overline{k}}\|_{\mathcal{H}}^2$ can be expressed as

$$\|f_k\|_{\mathcal{H}}^2 = \frac{1}{2}(\lambda_k^T K \lambda_k + b_k^2), \tag{56}$$

$$\|f_{\overline{k}}\|_{\mathcal{H}}^2 = \frac{1}{2}(\lambda_{\overline{k}}^T K \lambda_{\overline{k}} + b_{\overline{k}}^2). \tag{57}$$

For manifold regularization, on the basis of $f_k = [f_k(x_1), \ldots, f_k(x_{L+U})]^T = K\lambda_k + e_k b_k$ and $f_{\overline{k}} = [f_{\overline{k}}(x_1), \ldots, f_{\overline{k}}(x_{L+U})]^T = K\lambda_{\overline{k}} + e_{\overline{k}} b_{\overline{k}}$, $\|f_k\|_{\mathcal{M}}^2$ and $\|f_{\overline{k}}\|_{\mathcal{M}}^2$ can be written as

$$\|f_k\|_{\mathcal{M}}^2 = f_k^T L f_k = (\lambda_k^T K + e_k^T b_k) L (K\lambda_k + e_k b_k), \tag{58}$$

$$\|f_{\overline{k}}\|_{\mathcal{M}}^2 = f_{\overline{k}}^T L f_{\overline{k}} = (\lambda_{\overline{k}}^T K + e_{\overline{k}}^T b_{\overline{k}}) L (K\lambda_{\overline{k}} + e_{\overline{k}} b_{\overline{k}}). \tag{59}$$

Thus, the nonlinear optimization problems are expressed as

$$\underset{(\lambda_k, b_k, \xi)}{min}\ \frac{1}{2}\|K(A_k, M^T)\lambda_k + e_k b_k\|^2 + c_k e_{\overline{k}}^T \xi + \frac{1}{2}\gamma_k(\lambda_k^T K \lambda_k + b_k^2) + \frac{1}{2}\eta_k(\lambda_k^T K + e_k^T b_k) L (\lambda_k K + e_k b_k)$$

$$s.t:\ -(K(A_{\overline{k}}, M^T)\lambda_{\overline{k}} + e_{\overline{k}} b_k) + \xi \ge e_{\overline{k}}, \quad \xi \ge 0. \tag{60}$$

Define the Lagrangian corresponding to the problem (60) as

$$L(\theta) = \frac{1}{2}\|K(A_k, M^T)\lambda_k + e_k b_k\|^2 + c_k e_{\overline{k}}^T \xi + \frac{1}{2}\gamma_k(\lambda_k^T K \lambda_k + b_k^2) + \frac{1}{2}\eta_k(\lambda_k^T K + e_k^T b_k) L (\lambda_k K + e_k b_k) = \alpha_k^T(-(K(A_{\overline{k}}, M^T)\lambda_k + e_{\overline{k}} b_k) + \xi - e_{\overline{k}} - \beta_k^T \xi \tag{61}$$

where $\theta = \{\lambda_k, b_k, \xi, \alpha_k, \beta_k\}$.

The dual problem can be formulated as

$$\underset{(\theta)}{max}\ L(\theta)$$

$$s.t: \underset{(\lambda_k, b_k, \xi)}{\nabla L(\theta)} = 0, \quad \alpha_k, \beta_k \ge 0. \tag{62}$$

**TABLE 1.** Specification of the synthetic datasets.

| Datasets | Number of Sample | Instance | | | Number of Label |
|---|---|---|---|---|---|
| | | Relevant | Irrelevant | Redundant | |
| Hyperspheres (HS1) | 400 | 15 | 5 | 0 | 5 |
| Hyperspheres (HS2) | 600 | 35 | 10 | 5 | 10 |
| Hypercubes (HC1) | 400 | 15 | 5 | 0 | 5 |
| Hypercubes (HC2) | 600 | 35 | 10 | 5 | 10 |

From (62), we obtain

$$\nabla_{\lambda_k} L = K(A_k, M^T)^T(K(A_k, M^T)\lambda_k + e_k b_k) + \gamma_k K\lambda_k + \eta_k KL(K\lambda_k + e_k b_k) + K(A_{\bar{k}}, M^T)^T\alpha_k = 0, \quad (63)$$

$$\nabla_{b_k} L = e_k^T(K(A_k, M^T)\lambda_k + e_k b_k + \gamma_k b_k + \eta_k e^T L(K\lambda_k + e_k b_k) + e_{\bar{k}}^T\alpha_k = 0, \quad (64)$$

$$\nabla_{\xi} L = c_k e_{\bar{k}} + \alpha_k - \beta_k = 0. \quad (65)$$

Combining (63) and (64) leads to

$$\begin{bmatrix} K(A_k, M^T)^T \\ e_k^T \end{bmatrix}\begin{bmatrix} K(A_k, M^T)\ e_k \end{bmatrix}\begin{bmatrix} \lambda_k \\ b_k \end{bmatrix} + \gamma_k \begin{bmatrix} K & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \lambda_k \\ b_k \end{bmatrix} + \eta_k \begin{bmatrix} K \\ e^T \end{bmatrix} L \begin{bmatrix} K\ e_k \end{bmatrix}\begin{bmatrix} \lambda_k \\ b_k \end{bmatrix} + \begin{bmatrix} K(A_{\bar{k}}, M^T)^T \\ e_k^T \end{bmatrix}\alpha_k = 0. \quad (66)$$

Let

$$H_\phi = \begin{bmatrix} K(A_k, M^T)\ e_k \end{bmatrix}, \quad O_\phi = \begin{bmatrix} K & 0 \\ 0 & 1 \end{bmatrix}, \quad J_\phi = \begin{bmatrix} K\ e_k \end{bmatrix}, \quad G_\phi = \begin{bmatrix} K(A_{\bar{k}}, M^T)^T\ e_{\bar{k}} \end{bmatrix}. \quad (67)$$

and the augmented vector $\rho_k = [\lambda_k, b_k]^T$, can be rewritten as

$$H_\phi^T H_\phi \rho_k + \gamma_k O_\phi \rho_k + \eta_k J_\phi^T L J_\phi \rho_k + G_\phi^T \alpha_k = 0$$
$$i.e: \rho_k = -(H_\phi^T H_\phi + \gamma_k O_\phi + \eta_k J_\phi^T L J_\phi)^{-1}(G_\phi^T \alpha_k). \quad (68)$$

So the Wolf dual problem (60) is formulated as follows:

$$max\ e_{\bar{k}}^T\alpha_k - \frac{1}{2}(\alpha_k^T G_\phi)(H_\phi^T H_\phi + \gamma_k O_\phi + \eta_k J_\phi^T L J_\phi)^{-1}(G_\phi^T \alpha_k)$$
$$s.t: 0 \le \alpha_k \le c_k e_{\bar{k}}. \quad (69)$$

Once vector $\rho_k$ is obtained from (68), a new data point $x \epsilon \Re^n$ might be assigned to class $K$, in a manner similar to the linear case.

## IV. EXPERIMENTAL RESULTS

This section is devoted to the evaluation of the proposed scheme. We compare it with MLTSVM, Rnak-SVM, BPMLL and SS-MLLSTSVM using synthetic and real datasets. All synthetic data are generated by Mldatagen according to the some predefined parameters such as (hyperspheres or hypercubes), number of relevant, irrelevant and redundant features, number of instance and number of labels. We add 5% noises to labels of each instance to make the learning task more arduous.Table 1 summarizes the specifications of the synthetic datasets. Also, the real datasets Emotion, Yeast, Scene, Medical, Flags and Birds are widely used for evaluating multi-label learning methods. All real-world datasets summarized in Table 2. It is worth mentioning that these datasets are obtained from UCI [49].

**TABLE 2.** Specification of the real datasets.

| Datasets | Field | Instance | Features | Number of Label |
|---|---|---|---|---|
| Emotion | Music | 593 | 72 | 6 |
| Scene | Image | 2407 | 294 | 6 |
| Medical | Medicine | 978 | 1449 | 45 |
| Yeast | Bioinformatics | 2471 | 103 | 14 |
| Flags | Image | 194 | 19 | 7 |
| Birds | Audio | 645 | 260 | 19 |

RBF kernel expressed as $w_{ij} = \frac{exp(-\|x_i - x_j\|_2^2)}{2\sigma}$ is utilized to evaluate the proposed scheme. Its only parameter is $\sigma$.

Moreover, parameters $C, \lambda, \gamma$ which are used by this scheme need to be determined optimally. 10-fold cross validation is employed to determine the parameters. It selects the best value of these parameters by assigning a range of values to each of them from $\{2^i \mid i = -5, \ldots, 5\}$. It should be mentioned that this procedure is applied to all data sets and the best parameter values are obtained according to Table 3. For the Rank-SVM, the kernel function parameter and penalty parameter C are selected from $\{2^{-6}, \ldots, 2^0, \ldots, 2^6\}$. For the BPMLL, the number of hidden neurons is {5%, 10%, . . . , 25%} of the number of input neurons, the training epochs is set to be 100. For the SS-MLLSTSVM, the penalty parameters and regularization parameters are selected from $\{2^{-6}, \ldots, 2^0, \ldots, 2^6\}$.

Our algorithm code is written in MATLAB 2013 on a PC with an Intel Core I5 processor with 2GB RAM. Hamming loss, Average precision, coverage, ranking loss one-error and one-error are the comparison metrics.

### A. HAMMING loss (Hloss)

Hamming loss calculates the number of times that an instance-label pair is misclassified between the predicted

**TABLE 3.** Specification of the best parameter values.

| Datasets | MLTSVM | | | | | LP-MLTSVM | | |
|---|---|---|---|---|---|---|---|---|
| | Sigma | $\gamma$ | $\lambda$ | $C$ | Ave-pre | $\lambda$ | C | Ave-pre |
| Emotions | 0.7±0.41 | 2.1±1.88 | 0.55±0.41 | 0.45±0.11 | 73.27±2.13 | 2.5±3.2 | 0.8±0.27 | 72.38±2.32 |
| Yeast | 0.25±0.00 | 2.6±1.34 | 1.3±0.96 | 0.25±0.00 | 70.01±0.77 | 1.65±0.78 | 0.5±0.00 | 69.81±0.84 |
| Scene | 0.3±0.11 | 1.2±0.45 | 1.65±0.78 | 0.80±0.27 | 76.46±0.74 | 2.0±0.0 | 1.0±0.0 | 76.01±1.24 |
| Medical | 0.25±0.0 | 1±0.0 | 1.8±0.45 | 0.25±0.0 | 79.83±2.06 | 0.85±0.7 | 0.55±0.27 | 76.49±2.89 |
| Flags | 0.6 ±0.32 | 2.30 ±1.35 | 0.57 ±0.40 | 0.43 ±0.15 | 72.14 ±1.93 | 2.30 ±3.51 | 0.8 ±0.23 | 71.40 ±2.50 |
| Birds | 0.29 ±0.02 | 2.8 ±1.19 | 1.8 ±0.73 | 0.34 ±0.02 | 73.02 ±0.77 | 1.42 ±0.62 | 0.43 ±0.05 | 66.45 ±0.3 |
| HS1 | 0.71 ±0.38 | 2.7 ±1.52 | 0.59 ±0.38 | 0.41 ±0.18 | 71.04 ±0.63 | 2.49 ±3.42 | 0.82 ±0.23 | 72.46 ±0.2 |
| HS2 | 0.66 ±0.41 | 2.47 ±1.06 | 0.49 ±0.63 | 0.41 ±0.17 | 73.16 ±1.52 | 2.43 ±2.01 | 0.81 ±0.24 | 71.11 ±2.4 |
| HC1 | 0.23 ±0.14 | 1.25 ±0.66 | 1.82 ±0.61 | 0.72 ±0.25 | 75.03 ±0.53 | 2.50 ±0.11 | 2.40 ±0.003 | 74.82 ±1.03 |
| HC2 | 0.68 ±0.42 | 2.51 ±1.24 | 0.51 ±0.72 | 0.43 ±0.19 | 74.23 ±1.32 | 2.13 ±2.15 | 0.85 ±0.35 | 74.16 ±1.9 |

**TABLE 4.** Performance comparison of the proposed method in terms of the Hloss.

| Datasets | LP-MLTSVM ( mean ± std ) | MLTSVM ( mean ± std ) | Rank-SVM ( mean ± std ) | BPMLL ( mean ± std ) | SS-MLLSTSVM ( mean ± std ) |
|---|---|---|---|---|---|
| Emotions | 0.2388 ± 0.0205 | 0.2369 ± 0.0315 | 0.2632 ± 0.0032 | 0.3130 ± 0.0104 | 0.2412 ± 0.0038 |
| Yeast | 0.2439 ± 0.0104 | 0.2772 ± 0.0049 | 0.2541 ± 0.0027 | 0.2632 ± 0.0017 | 0.2332 ± 0.0120 |
| Scene | 0.1441 ± 0.0078 | 0.1401 ± 0.068 | 0.1392 ± 0.0009 | 0.2130 ± 0.0163 | 0.2492 ± 0.0033 |
| Medical | 0.0135 ± 0.0010 | 0.0149 ± 0.0013 | 0.0134 ± 0.0019 | 0.0331 ± 0.0021 | 0.0129 ± 0.0777 |
| Flags | 0.1534 ± 0.0202 | 0.2141 ± 0.0011 | 0.2314 ± 0.0069 | 0.2317 ± 0.0041 | 0.1433 ± 0.0030 |
| Birds | 0.1314 ± 0.0101 | 0.3327 ± 0.0104 | 0.3151 ± 0.0088 | 0.4231 ± 0.0024 | 0.1519 ± 0.0701 |
| HS1 | 0.2415 ± 0.0015 | 0.2113 ± 0.0108 | 0.2715 ± 0.0010 | 0.3237 ± 0.0108 | 0.2422 ± 0.0037 |
| HS2 | 0.2514 ± 0.0111 | 0.2613 ± 0.0610 | 0.2621 ± 0.0039 | 0.2651 ± 0.0115 | 0.2532 ± 0.0041 |
| HC1 | 0.2413 ± 0.0074 | 0.2441 ± 0.0116 | 0.2515 ± 0.0016 | 0.2671 ± 0.0020 | 0.2451 ± 0.0119 |
| HC2 | 0.2511 ± 0.0071 | 0.2621 ± 0.0044 | 0.2728 ± 0.0033 | 0.3119 ± 0.0303 | 0.2510 ± 0.0072 |

**TABLE 5.** Performance comparison of the proposed method in terms of the Avepre.

| Datasets | LP-MLTSVM ( mean ± std ) | MLTSVM ( mean ± std ) | Rank-SVM ( mean ± std ) | BPMLL ( mean ± std ) | SS-MLLSTSVM ( mean ± std ) |
|---|---|---|---|---|---|
| Emotions | 0.7327±0.0213 | 0.7238 ± 0.0232 | 0.7132 ± 0.0055 | 0.5931 ± 0.0033 | 0.7422 ± 0.0029 |
| Yeast | 0.7001± 0.0077 | 0.6981± 0.0084 | 0.6817 ± 0.0013 | 0.6619 ± 0.0049 | 0.6983 ± 0.0019 |
| Scene | 0.7646 ± 0.0741 | 0.7601 ± 0.0124 | 0.7521 ± 0.0024 | 0.4371 ± 0.0214 | 0.7762 ± 0.0141 |
| Medical | 0.7983 ± 0.0206 | 0.7649 ± 0.0289 | 0.7638 ± 0.0011 | 0.4130 ± 0.0310 | 0.7845 ± 0.0072 |
| Flags | 0.7335 ± 0.0112 | 0.7241 ± 0.0041 | 0.7041 ± 0.0015 | 0.6138 ± 0.0202 | 0.7342 ± 0.0052 |
| Birds | 0.7114 ± 0.0028 | 0.6815 ± 0.0102 | 0.6713 ± 0.0039 | 0.6514 ± 0.0108 | 0.7014 ± 0.0108 |
| HS1 | 0.7016 ± 0.0118 | 0.6928 ± 0.0214 | 0.6821 ± 0.0091 | 0.6611 ± 0.0125 | 0.7010 ± 0.0061 |
| HS2 | 0.7441 ± 0.0030 | 0.7315 ± 0.0205 | 0.7219 ± 0.0018 | 0.6241 ± 0.0063 | 0.7409 ± 0.0025 |
| HC1 | 0.7419 ± 0.0410 | 0.7361 ± 0.0118 | 0.7221 ± 0.0054 | 0.6115 ± 0.0042 | 0.7511 ± 0.0071 |
| HC2 | 0.7221 ± 0.0330 | 0.7109 ± 0.0207 | 0.6672 ± 0.0061 | 0.6061 ± 0.0011 | 0.7209 ± 0.0090 |

label set $h(x)$ and the ground-truth set $\mathcal{Y}$:

$$Hloss \downarrow = \frac{1}{p}\sum_{i=1}^{p}\frac{1}{K} \mid h(x_i)\Delta y_i \mid \epsilon [0,1]\,. \tag{70}$$

where $\Delta$ stands for the symmetric difference of the two sets. The small value of Hloss shows the better performance of a scheme. Table 4 and Fig 3 represent the performance of the proposed scheme terms of Hamming loss.

### B. AVERAGE PRECISION (Avepre)

Average precision evaluates the average fraction of labels ranked above a particular label $y\epsilon y_i$ (71), as shown at the bottom of the next page.

The large value of this metric approves the better performance of the scheme. The performance of the proposed scheme in terms of Avepre is demonstrated in Table 5 and Fig 4.

**TABLE 6.** Performance comparison of the proposed method in terms of the Cov.

| Datasets | LP-MLTSVM ( mean ± std ) | MLTSVM ( mean ± std ) | Rank-SVM ( mean ± std ) | BPMLL ( mean ± std ) | SS-MLLSTSVM ( mean ± std ) |
|---|---|---|---|---|---|
| Emotions | 2.3164 ± 0.1864 | 2.3853 ± 0.1930 | 2.5338 ± 0.0411 | 2.7531 ± 0.8120 | 2.4510 ± 0.0856 |
| Yeast | 7.6414 ± 0.2019 | 7.3072 ± 0.1995 | 7.2782 ± 0.0209 | 7.2912 ± 0.0350 | 7.5441 ± 0.0214 |
| Scene | 0.9141 ± 0.02741 | 0.8748 ± 0.0662 | 0.9214 ± 0.0531 | 2.3450 ± 0.0038 | 1.1941 ± 0.0237 |
| Medical | 4.6452± 0.6973 | 5.6281 ± 0.0988 | 4.3120 ± 0.0024 | 7.3410 ± 0.0420 | 4.5621 ± 0.0225 |
| Flags | 2.2130 ± 0.1112 | 2.4142 ± 0.0215 | 2.6142 ± 0.0141 | 2.8214 ± 0.0335 | 2.2014 ± 0.0208 |
| Birds | 7.8210 ± 0.0518 | 7.4011 ± 0.0414 | 7.3112 ± 0.0821 | 7.1185 ± 0.0442 | 7.7110 ± 0.0528 |
| HS1 | 1.1201 ± 0.0338 | 3.1910 ± 0.0122 | 3.7102 ± 0.0038 | 3.4180 ± 0.0332 | 1.2101 ± 0.0116 |
| HS2 | 3.5119 ± 0.1102 | 5.1121 ± 0.0519 | 5.1830 ± 0.0325 | 5.5310 ± 0.0069 | 3.3560 ± 0.0142 |
| HC1 | 1.2419 ± 0.0252 | 3.1513 ± 0.0448 | 3.2610 ± 0.0442 | 3.5101 ± 0.0449 | 1.3920 ± 0.0337 |
| HC2 | 3.7211 ± 0.0470 | 5.3329 ± 0.220 | 5.0199 ± 0.0602 | 5.7302 ± 0.0113 | 3.8102 ± 0.0405 |

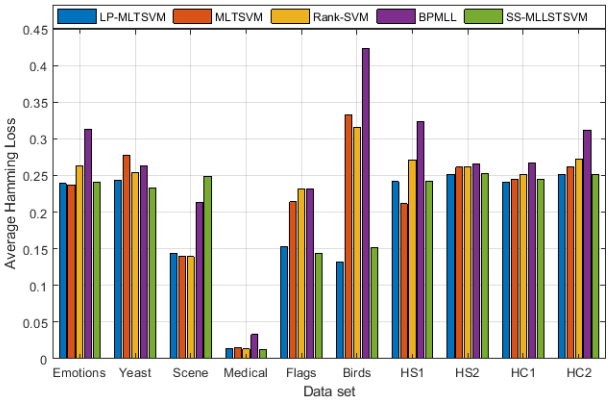


**FIGURE 3.** Comparison of proposed scheme based on Hloss.

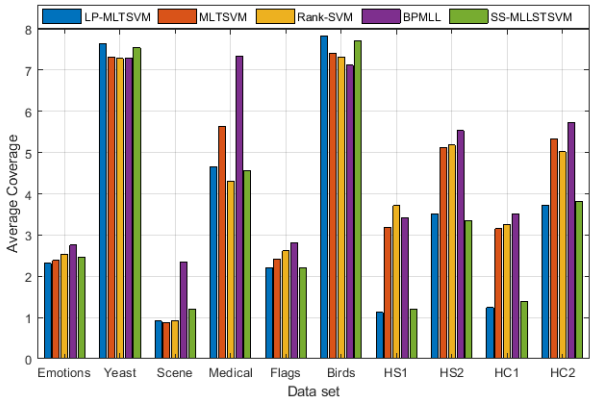


**FIGURE 5.** Comparison of proposed scheme based on Cov.

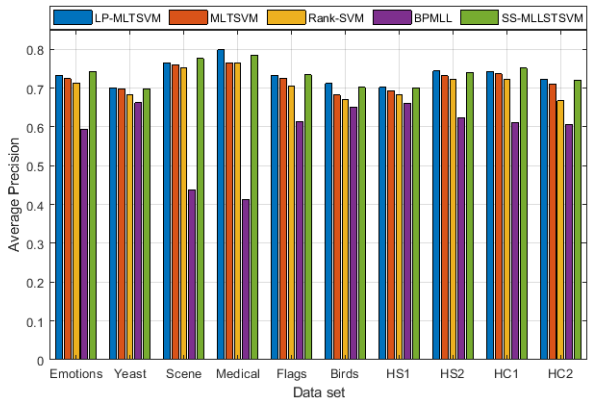


**FIGURE 4.** Comparison of the proposed scheme based on Avepre.

### C. COVERAGE (Cov)

Coverage evaluates how far we need, on average, to go down the ranked list of labels in order to cover all the possible labels of the instance:

$$Cov \downarrow = \frac{1}{p}\sum_{i=1}^{p}\left(\underset{y\epsilon y_i}{Max}\, rank_f(x_i, y) - 1\right)\epsilon\,[0, K-1]\,. \tag{72}$$

The smaller this metric is, the higher the performance of the algorithm will be. Table 6 and Fig 5 represent the performance of the algorithm in terms of this metric.

### D. RANKING LOSS (Rloss)

Ranking loss evaluates the average fraction of label pairs that are reversely ordered. Let $\bar{y}$ be the complementary set of $y$ in $\mathcal{Y}$, so we have

$$\begin{aligned} Rloss \downarrow \\ = \frac{1}{p}\sum_{i=1}^{p} \\ \times\left(\frac{1}{\mid y_i \mid \mid \overline{y_i} \mid}\left|\left\{(y', y'') \mid f_{y'\epsilon y''}(x_i) \leq f_{y'\epsilon y''}(x_i)\right\}\right|\right)\epsilon\,[0, 1]\,. \end{aligned} \tag{73}$$

$$Avepre \uparrow = \frac{1}{p}\sum_{i=1}^{p}\left(\frac{1}{\mid y_i \mid}\sum_{y\epsilon y_i}\frac{\mid \{y'\epsilon y_i \mid rank_f(x_i, y') \leq rank_f(x_i, y)\} \mid}{rank_f(x_i, y)}\right)\epsilon\,[0, 1]\,. \tag{71}$$

**TABLE 7.** Performance comparison of the proposed method in terms of the Rloss.

| Datasets | LP-MLTSVM ( mean ± std ) | MLTSVM ( mean ± std ) | Rank-SVM ( mean ± std ) | BPMLL ( mean ± std ) | SS-MLLSTSVM ( mean ± std ) |
|---|---|---|---|---|---|
| Emotions | 0.4065 ± 0.0245 | 0.4118 ± 0.0367 | 0.4530 ± 0.0053 | 0.5240 ± 0.0392 | 0.4016 ± 0.0049 |
| Yeast | 0.4031 ± 0.0078 | 0.4164 ± 0.0052 | 0.3971 ± 0.0047 | 0.4931 ± 0.0022 | 0.4139 ± 0.0036 |
| Scene | 0.3149 ± 0.0169 | 0.2775 ± 0.0117 | 0.1823 ± 0.0013 | 0.3819 ± 0.0530 | 0.3038 ± 0.0068 |
| Medical | 0.1689± 0.0212 | 0.2234 ± 0.0365 | 0.1563 ± 0.0067 | 0.2739 ± 0.0243 | 0.1566 ± 0.0043 |
| Flags | 0.4024 ± 0.0018 | 0.3524 ± 0.0066 | 0.1951 ± 0.0080 | 0.4512 ± 0.0159 | 0.4119 ± 0.0072 |
| Birds | 0.1272 ± 0.0109 | 0.2023 ± 0.0220 | 0.1419 ± 0.0071 | 0.2243 ± 0.0240 | 0.1342 ± 0.0040 |
| HS1 | 0.2411 ± 0.090 | 0.2320 ± 0.0104 | 0.2421 ± 0.0083 | 0.3772 ± 0.0410 | 0.2372 ± 0.0019 |
| HS2 | 0.4219 ± 0.0401 | 0.3763 ± 0.0112 | 0.4302 ± 0.0037 | 0.4717 ± 0.330 | 0.4315 ± 0.0044 |
| HC1 | 0.2433 ± 0.0331 | 0.2461 ± 0.0206 | 0.2919 ± 0.0022 | 0.3807 ± 0.0105 | 0.2452 ± 0.0063 |
| HC2 | 0.4110 ± 0.0152 | 0.4371 ± 0.0327 | 0.4217 ± 0.0051 | 0.4555 ± 0.0148 | 0.4271 ± 0.0064 |

**TABLE 8.** Performance comparison of the proposed method in terms of the Oerr.

| Datasets | LP-MLTSVM ( mean ± std ) | MLTSVM ( mean ± std ) | Rank-SVM ( mean ± std ) | BPMLL ( mean ± std ) | SS-MLLSTSVM ( mean ± std ) |
|---|---|---|---|---|---|
| Emotions | 0.1085 ± 0.0380 | 0.1141 ± 0.0449 | 0.1831 ± 0.0125 | 0.2320 ± 0.0362 | 0.1095 ± 0.0366 |
| Yeast | 0.0544 ± 0.0103 | 0.0381± 0.0080 | 0.0380 ± 0.0022 | 0.0561 ± 0.0581 | 0.1133 ± 0.0307 |
| Scene | 0.2327 ± 0.0207 | 0.1798 ±0.0101 | 0.2530 ± 0.0032 | 0.2821 ± 0.0272 | 0.1460 ± 0.0119 |
| Medical | 0.1164 ± 0.0241 | 0.1651 ± 0.0360 | 0.1662 ± 0.0012 | 0.2031 ± 0.0584 | 0.1239 ± 0.0087 |
| Flags | 0.1116 ± 0.0145 | 0.1610 ± 0.0128 | 0.2015 ± 0.0036 | 0.2510 ± 0.0369 | 0.1340 ± 0.0149 |
| Birds | 0.1030 ± 0.0308 | 0.1412 ± 0.0408 | 0.1415 ± 0.0061 | 0.1972 ± 0.0221 | 0.1016 ± 0.0146 |
| HS1 | 0.1006 ± 0.0205 | 0.1319 ± 0.0328 | 0.1977 ± 0.0033 | 0.2315 ± 0.0421 | 0.1230 ± 0.0116 |
| HS2 | 0.1471 ± 0.0180 | 0.1931 ± 0.0154 | 0.2216 ± 0.0020 | 0.2611 ± 0.0339 | 0.1433 ± 0.302 |
| HC1 | 0.1233 ± 0.0177 | 0.1514 ± 0.0501 | 0.1960 ± 0.0071 | 0.2011 ± 0.0111 | 0.1131 ± 0.0202 |
| HC2 | 0.1430 ± 0.0331 | 0.1338 ± 0.0244 | 0.2319 ± 0.0050 | 0.2360 ± 0.0237 | 0.1433 ± 0.0339 |

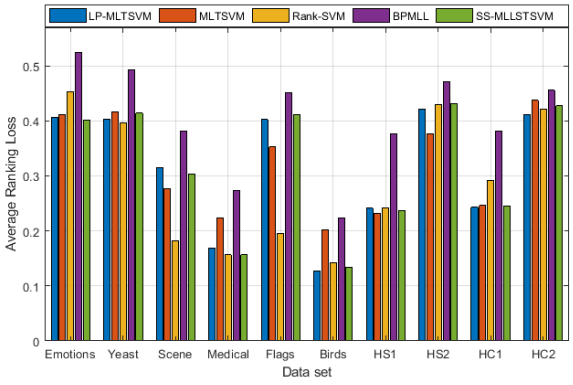


**FIGURE 6.** Comparison of proposed scheme based on Rloss.

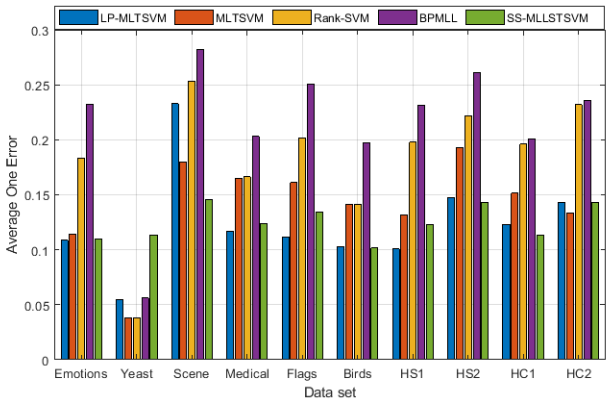


**FIGURE 7.** Comparison of proposed scheme based on Oerr.

The smallest value of this metric shows the best performance of the algorithm. Table 7 and Fig 6 represent the performance of the algorithm in terms of this metric.

### E. ONE-ERROR (Oerr)

One-error evaluates the number of times the top-ranked label is not in the set of proper labels of the instance

$$Oerr \downarrow = \frac{1}{p}\sum_{i=1}^{p} H(x_i)\epsilon\,[0,1]$$
$$H(x_i) = \begin{cases} 0 & if\ arg\ max\ f_y(x_i)\epsilon y_i \\ 1 & Otherwise. \end{cases} \quad (74)$$

The smallest value of this metric shows the best performance of the algorithm. Table 8 and Fig 7 illustrate the performance of the algorithm in terms of this metric.

### F. DISCUSSION

LP-MLTSVM algorithm suffers from a high time complexity in the training phase, because of its need to construct Graph Laplacian for determining labels of the unlabeled instances. This shortcoming, however, does not make this approach

**TABLE 9.** Rank of the five considered algorithms for Hloss.

| | LP-MLTSVM | MLTSVM | Rank-SVM | BPMLL | SS-MLLSTSVM |
|---|---|---|---|---|---|
| Emotions | 2 | 1 | 4 | 5 | 3 |
| Yeast | 2 | 5 | 3 | 4 | 1 |
| Scene | 3 | 2 | 1 | 4 | 5 |
| Medical | 3 | 4 | 2 | 5 | 1 |
| Flags | 2 | 3 | 4 | 5 | 1 |
| Birds | 1 | 4 | 3 | 5 | 2 |
| HS1 | 1 | 3 | 4 | 5 | 2 |
| HS2 | 1 | 3 | 4 | 5 | 2 |
| HC1 | 1 | 3 | 4 | 5 | 2 |
| HC2 | 2 | 3 | 4 | 5 | 1 |
| Average Rank | **1.8** | 3.1 | 3.3 | 4.8 | 2 |
| Average Result | **0.1910** | 0.2195 | 0.2274 | 0.2645 | 0.2023 |

**TABLE 10.** Rank of the five considered algorithms for Avepre.

| | LP-MLTSVM | MLTSVM | Rank-SVM | BPMLL | SS-MLLSTSVM |
|---|---|---|---|---|---|
| Emotions | 2 | 3 | 4 | 5 | 1 |
| Yeast | 1 | 3 | 4 | 5 | 2 |
| Scene | 2 | 3 | 4 | 5 | 1 |
| Medical | 1 | 3 | 4 | 5 | 2 |
| Flags | 2 | 3 | 4 | 5 | 1 |
| Birds | 1 | 3 | 4 | 5 | 2 |
| HS1 | 1 | 3 | 4 | 5 | 2 |
| HS2 | 1 | 3 | 4 | 5 | 2 |
| HC1 | 2 | 3 | 4 | 5 | 1 |
| HC2 | 1 | 3 | 4 | 5 | 2 |
| Average Rank | **1.4** | 3 | 4 | 5 | 1.6 |
| Average Result | 0.7350 | 0.7224 | 0.7080 | 0.5873 | **0.7351** |

**TABLE 11.** Rank of the five considered algorithms for Cov.

| | LP-MLTSVM | MLTSVM | Rank-SVM | BPMLL | SS-MLLSTSVM |
|---|---|---|---|---|---|
| Emotions | 1 | 2 | 4 | 5 | 3 |
| Yeast | 5 | 3 | 1 | 2 | 4 |
| Scene | 2 | 1 | 3 | 5 | 4 |
| Medical | 3 | 4 | 1 | 5 | 2 |
| Flags | 2 | 3 | 4 | 5 | 1 |
| Birds | 5 | 3 | 2 | 1 | 4 |
| HS1 | 1 | 3 | 5 | 4 | 2 |
| HS2 | 2 | 3 | 4 | 5 | 1 |
| HC1 | 1 | 3 | 4 | 5 | 2 |
| HC2 | 1 | 4 | 3 | 5 | 2 |
| Average Rank | **2.3** | 2.9 | 3.1 | 4.2 | 2.5 |
| Average Result | **3.5146** | 4.2798 | 4.2144 | 4.7859 | 3.5432 |

impractical as the learning phase needs to be conducted only once.

According to the results provided in Table 4, we can observe that in the proposed algorithm the established decision boundary only passes the low-density region of feature space and does not meet the unlabeled data instances. It is imperative that the weights assigned to the edges of the graph be smooth without any abrupt changes, since the weights reflect the similarity between instances. Equation (20) implies that if two instances are connected through a high weight edge, these two instances share the same labels.

Construction of a competent graph depends on a deep insight from the problem domain as well as defining

**TABLE 12.** Rank of the five considered algorithms for Rloss.

| | LP-MLTSVM | MLTSVM | Rank-SVM | BPMLL | SS-MLLSTSVM |
|---|---|---|---|---|---|
| Emotions | 2 | 3 | 4 | 5 | 1 |
| Yeast | 2 | 4 | 1 | 5 | 3 |
| Scene | 4 | 2 | 1 | 5 | 3 |
| Medical | 3 | 4 | 1 | 5 | 2 |
| Flags | 3 | 2 | 1 | 5 | 4 |
| Birds | 1 | 4 | 3 | 5 | 2 |
| HS1 | 3 | 1 | 4 | 5 | 2 |
| HS2 | 2 | 1 | 3 | 4 | 5 |
| HC1 | 1 | 3 | 4 | 5 | 2 |
| HC2 | 1 | 4 | 2 | 5 | 3 |
| Average Rank | **2.2** | 2.8 | 2.4 | 4.9 | 2.7 |
| Average Result | 0.3140 | 0.3175 | **0.2912** | 0.4033 | 0.3163 |

**TABLE 13.** Rank of the five considered algorithms for Oerr.

| | LP-MLTSVM | MLTSVM | Rank-SVM | BPMLL | SS-MLLSTSVM |
|---|---|---|---|---|---|
| Emotions | 1 | 3 | 4 | 5 | 2 |
| Yeast | 3 | 2 | 1 | 4 | 5 |
| Scene | 3 | 2 | 4 | 5 | 1 |
| Medical | 1 | 3 | 4 | 5 | 2 |
| Flags | 1 | 3 | 4 | 5 | 2 |
| Birds | 2 | 3 | 4 | 5 | 1 |
| HS1 | 1 | 3 | 4 | 5 | 2 |
| HS2 | 2 | 3 | 4 | 5 | 1 |
| HC1 | 2 | 3 | 4 | 5 | 1 |
| HC2 | 2 | 1 | 5 | 4 | 3 |
| Average Rank | **1.8** | 2.6 | 3.8 | 4.8 | 2 |
| Average Result | **0.1241** | 0.1409 | 0.1830 | 0.2151 | 0.1251 |

**TABLE 14.** p-value of Hloss according Bonferroni-Dunn analysis.

| Datasets | LP-MLTSVM | MLTSVM | Rank-SVM | BPMLL | SS-MLLSTSVM |
|---|---|---|---|---|---|
| LP-MLTSVM | 1 | 1 | 0.7516 | 0.1267 | 1 |
| MLTSVM | 1 | 1 | 1 | 1 | 1 |
| Rank-SVM | 0.7516 | 1 | 1 | 1 | 1 |
| BPMLL | 0.1267 | 1 | 1 | 1 | 0.2771 |
| SS-MLLSTSVM | 1 | 1 | 1 | 0.2771 | 1 |

**TABLE 15.** p-value of Avepre according Bonferroni-Dunn analysis.

| Datasets | LP-MLTSVM | MLTSVM | Rank-SVM | BPMLL | SS-MLLSTSVM |
|---|---|---|---|---|---|
| LP-MLTSVM | 1 | 1 | 1 | 0.0001 | 1 |
| MLTSVM | 1 | 1 | 1 | 0.0024 | 1 |
| Rank-SVM | 1 | 1 | 1 | 0.0330 | 1 |
| BPMLL | 0.0001 | 0.0024 | 0.0330 | 1 | 0.0001 |
| SS-MLLSTSVM | 1 | 1 | 1 | 0.0001 | 1 |

appropriate distance functions and parameters employed in Table 3. A label predictor function needs to be designed in a way that

1) Euclidean distance is not considered.
2) Whole graph is smooth.

In the proposed algorithm, these two criteria are satisfied by employing MR as indicated in (58), (59), which results a higher accuracy in Table 4.

We list the rank of different multi-label classification methods in terms of Hamming loss, average precision, coverage,

**TABLE 16.** p-value of Cov according Bonferroni-Dunn analysis.

| Datasets | LP-MLTSVM | MLTSVM | Rank-SVM | BPMLL | SS-MLLSTSVM |
|---|---|---|---|---|---|
| LP-MLTSVM | 1 | 1 | 1 | 1 | 1 |
| MLTSVM | 1 | 1 | 1 | 1 | 1 |
| Rank-SVM | 1 | 1 | 1 | 1 | 1 |
| BPMLL | 1 | 1 | 1 | 1 | 1 |
| SS-MLLSTSVM | 1 | 1 | 1 | 1 | 1 |

**TABLE 17.** p-value of Rloss according Bonferroni-Dunn analysis.

| Datasets | LP-MLTSVM | MLTSVM | Rank-SVM | BPMLL | SS-MLLSTSVM |
|---|---|---|---|---|---|
| LP-MLTSVM | 1 | 1 | 1 | 0.6566 | 1 |
| MLTSVM | 1 | 1 | 1 | 0.7772 | 1 |
| Rank-SVM | 1 | 1 | 1 | 0.3696 | 1 |
| BPMLL | 0.6566 | 0.7772 | 0.3696 | 1 | 1 |
| SS-MLLSTSVM | 1 | 1 | 1 | 1 | 1 |

**TABLE 18.** p-value of Oerr according Bonferroni-Dunn analysis.

| Datasets | LP-MLTSVM | MLTSVM | Rank-SVM | BPMLL | SS-MLLSTSVM |
|---|---|---|---|---|---|
| LP-MLTSVM | 1 | 1 | 0.0832 | 0.0026 | 1 |
| MLTSVM | 1 | 1 | 1 | 0.0911 | 1 |
| Rank-SVM | 0.0832 | 1 | 1 | 1 | 0.1137 |
| BPMLL | 0.0026 | 0.0911 | 1 | 1 | 0.0039 |
| SS-MLLSTSVM | 1 | 1 | 0.1137 | 0.0039 | 1 |

ranking loss and one-error in Table 9 to 13, respectively. We can observe that, the proposed LP-MLTSVM outperforms other algorithms in all metrics.

According to the average results provided in Table 9 to 13, except average precision and ranking loss the proposed method demonstrates a higher performance.

We conduct Bonferroni-Dunn analysis to determine if there is a significant difference between the proposed approach and the compared ones in terms of the comparison metrics. Table 14 to 18 presents the result of this analysis according which we can conclude that except coverage metric the proposed approach outperforms significantly.

## V. CONCLUSION

This paper aimed to leverage a large number of unlabeled data along with a limited number of labeled data for increasing classifier's precision. The main reason for employing semi-supervised learning stems from this fact that the number of available data is generally limited; on the other hand unlabeled data is prevalent. Accordingly, inspired from TWSVM this paper proposes a semi-supervised learning scheme called LP-MLTSVM, for the classification of multi-label data.

This scheme provides a classification model with a significant degree of precision which can more precisely classify training data compared to previous works. The proposed scheme is grounded on manifold theory on Graph Laplacian. Training data constitute the vertices of this graph. The weight of an edge between two vertices reflects their similarity. In other words, the more similar the two vertices are, the higher weight of their connecting edge is.

We applied the proposed scheme to several standard data sets to compare its performance with MLTSVM, Rnak-SVM, BPMLL and SS-MLLSTSVM based on performance metrics. The evaluation results demonstrate the outstanding performance of LP-MLTSVM compared with other works. As a future work, this scheme might be extended to the structural learning problems.

## REFERENCES

[1] V. Cherkassky and F. M. Mulier, *Learning from Data: Concepts, Theory, and Methods*. Hoboken, NJ, USA: Wiley, 2007.

[2] C. C. Aggarwal, *Data Classification: Algorithms Application*. Boca Raton, FL, USA: CRC Press, 2014. [Online]. Available: http://www.crcnetbase.com/doi/book/10.1201/b17320

[3] G. Melki, A. Cano, V. Kecman, and S. Ventura, "Multi-target support vector regression via correlation regressor chains," *Inf. Sci.*, vols. 415–416, pp. 53–69, Nov. 2017.

[4] Y. Wu, "Statistical learning theory," *Technometrics*, vol. 41, no. 4, pp. 377–378, 1999. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/00401706.1999.10485951

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

[6] K. Gurney, *An Introduction to Neural Networks*. Berlin, Germany: Springer-Verlag, 2018.

[7] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, 1998, doi: 10.1023/A:1009715923555.

[8] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Machine Learning: ECML-98*, C. Nédellec and C. Rouveirol, Eds. Berlin, Germany: Springer, 1998, pp. 137–142.

[9] T. S. Guzella and W. M. Caminhas, "A review of machine learning approaches to spam filtering," *Expert Syst. Appl.*, vol. 36, no. 7, pp. 10206–10222, Sep. 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S095741740900181X
[10] B. Wu, E. Zhong, A. Horner, and Q. Yang, "Music emotion recognition by multi-label multi-layer multi-instance multi-view learning," in *Proc. 22nd ACM Int. Conf. Multimedia*, Nov. 2014, pp. 117–126.
[11] Y. Liu, K. Wen, Q. Gao, X. Gao, and F. Nie, "SVM based multi-label learning with missing labels for image annotation," *Pattern Recognit.*, vol. 78, pp. 307–317, Jun. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320318300372
[12] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.
[13] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 681–687.
[14] M. L. Zhang and Z. H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1338–1351, Oct. 2006.
[15] M.-L. Zhang, "ML-RBF: RBF neural networks for multi-label learning," *Neural Process. Lett.*, vol. 29, no. 2, pp. 61–74, Apr. 2009.
[16] W. Cheng and E. Hüllermeier, "Combining instance-based learning and logistic regression for multilabel classification," *Mach. Learn.*, vol. 76, nos. 2–3, pp. 211–225, Sep. 2009.
[17] Q. Ai, Y. Kang, A. Wang, X. Li, and F. Li, "An effective semi-supervised multi-label least squares twin support vector machine," *IEEE Access*, vol. 8, pp. 213460–213472, 2020.
[18] G. Tsoumakas and I. Katakis, "Multi-label classification," in *Proc. Database Technol.*, 2011, pp. 309–319.
[19] J. Fürnkranz, E. Hüllermeier, E. L. Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *J. Mach. Learn.*, vol. 73, no. 2, pp. 133–153, Nov. 2008.
[20] R. E. Schapire and Y. Singer, "BoosTexter: A boosting-based system for text categorization," *Mach. Learn.*, vol. 39, nos. 2–3, pp. 135–168, May 2000.
[21] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *J. Mach. Learn.*, vol. 85, no. 3, pp. 333–359, Dec. 2011.
[22] M. E. Celebi and K. Aydin, *Unsupervised Learning Algorithms*. Cham, Switzerland: Springer, 2016, pp. 1–558, doi: 10.1007/978-3-319-24211-8.
[23] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Stat. Probab.*, vol. 1, 1967, pp. 281–296.
[24] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3336–3341, 2009.
[25] X. Goldberg, "Introduction to semi-supervised learning," *Synthesis Lectures Artif. Intell. Mach. Learn.*, vol. 6, pp. 1–116, Jun. 2009.
[26] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Mach. Learn.*, vol. 109, no. 2, pp. 373–440, Feb. 2020.
[27] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning. Adaptive Computation and Machine Learning*. Cambridge, MA, USA: MIT Press, 2010.
[28] R. Fujimaki, T. Yairi, and K. Machida, "An approach to spacecraft anomaly detection problem using kernel feature space," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2005, pp. 401–410.
[29] D. Dasgupta and N. S. Majumdar, "Anomaly detection in multidimensional data using negative selection algorithm," in *Proc. Congr. Evol. Comput.*, 2002, pp. 1039–1044.
[30] M. S. Mulekar, "Data mining: Multimedia, soft computing, and bioinformatics," *Technometrics*, vol. 46, no. 3, pp. 368–369, 2004.
[31] Z. Zhang, *Multimedia Data Mining: A Systematic Introduction to Concepts and Theory*. New York, NY, USA: Choice Reviews, 2010.
[32] X. Wang, W. Zhang, Q. Zhang, and G. Z. Li, "MultiP-SChlo: Multi-label protein subchloroplast localization prediction with Chou's pseudo amino acid composition and a novel multi-label classifier," *Bioinformatics*, vol. 31, no. 16, pp. 2639–2645, 2015.
[33] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning* (Springer Texts in Statistics). Jan. 2013, doi: 10.1007/978-1-4614-7138-7.
[34] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. 20th Int. Conf. Mach. Learn.*, 2000, pp. 200–209.
[35] G. Fung and O. L. Mangasarian, "Semi-supervised support vector machines for unlabeled data classification," *Optim. Methods Softw.*, vol. 15, no. 1, pp. 29–44, 2001.
[36] S. Melacci and M. Belkin, "Laplacian support vector machines trained in the primal," *J. Mach. Learn. Res.*, vol. 12, no. 3, pp. 1149–1184, 2011.
[37] Z. Qi, Y. Tian, and Y. Shi, "Laplacian twin support vector machine for semi-supervised classification," *Neural Netw.*, vol. 35, pp. 46–53, Nov. 2012.
[38] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMUCALD02107, 2002.
[39] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, no1, pp. 2399–2434, 2006.
[40] X. Zhu, *Semi-Supervised Learning with Graphs*. Pittsburgh, PA, USA: Carnegie Mellon Univ., 2012.
[41] X. He, "Laplacian regularized D-optimal design for active learning and its application to image retrieval," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 254–263, Jan. 2010.
[42] J. Abernethy, O. Chapelle, and C. Castillo, "Web spam identification through content and hyperlinks," in *Proc. 4th Int. Workshop Adversarial Inf. Retr. Web*, 2008, pp. 41–44.
[43] W.-J. Chen, Y.-H. Shao, C.-N. Li, and N.-Y. Deng, "MLTSVM: A novel twin support vector machine to multi-label learning," *Pattern Recognit.*, vol. 52, pp. 61–74, Apr. 2016.
[44] J. Huang, "A combinatorial view of graph Laplacians," Max Planck Inst. Biol. Cybern., Tübingen, Germany, Tech. Rep. 144, 2005, p. 144. [Online]. Available: https://pure.mpg.de/pubman/faces/ViewItemOverviewPage.jsp?itemId=item_1791364
[45] Z. X. Yang, Y. H. Shao, and X. S. Zhang, "Multiple birth support vector machine for multi-class classification," *Neural Comput. Appl.*, vol. 22, no. 1, pp. 153–161, 2013.
[46] A. Tikhonov, "Regularization of mathematically incorrectly posed problems," *Sov. Math.*, vol. 4, no. 6, pp. 1624–1627, 1963.
[47] W.-J. Chen, Y.-H. Shao, D.-K. Xu, and Y.-F. Fu, "Manifold proximal support vector machine for semi-supervised classification," *Int. J. Speech Technol.*, vol. 40, no. 4, pp. 623–638, Jun. 2014.
[48] O. L. Mangasarian and D. R. Musicant, "Successive overrelaxation for support vector machines," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1032–1037, Sep. 1999.
[49] D. Dua and C. Graff, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA, 2017. [Online]. Available: http://archive.ics.uci.edu/ml

**FARHAD GHAREBAGHI** received the master's degree in computer engineering from Islamic Azad University, Qazvin Branch, Qazvin, Iran, in 2006, where he is currently pursuing the Ph.D. degree in software systems. His research interests include machine learning, classification techniques, and optimization.

**ALI AMIRI** was born in Zanjan, Iran. He received the B.Sc. degree in computer software engineering from Islamic Azad University, Zanjan Branch, Zanjan, and the M.Sc. and Ph.D. degrees in computer engineering, artificial intelligence, and robotic from the Iran University of Science and Technology, Tehran, Iran, in 2006 and 2011, respectively. Currently, he is an Assistant Professor with the Computer Engineering Department, University of Zanjan, Zanjan.