Multi-Label Learning with Local Similarity of Samples

Wenfang Zhu[1], Weiwei Li[2] and Xiuyi Jia[1]

[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210014, China,
[2]College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
{zwf.ang, jiaxy}@njust.edu.cn, liweiwei@nuaa.edu.cn

***Abstract*—Multi-label learning has been successfully applied to solve instance multi-semantics problems. Moreover, the topology information of samples is often adopted in existing works to improve the prediction performance, in which the similarity of samples is usually calculated in the entire feature space. However, in real-world applications, each label is often determined by a subset of the original features, so when we focus on different labels, the similarity of two instances may be different. In this paper, we propose a multi-label learning method by exploiting the local similarity of samples. Specifically, the smoothness assumption is applied to assume that if the feature subset is similar between samples, the corresponding label should be similar. In addition, L1 regularization is also adopted to sparse the weight coefficients when constraining the output space of the instance. The experimental results on several data sets validate the effectiveness of the proposed method.**

***Index Terms*—multi-label, smoothness assumption, local similarity**

I. Introduction

Multi-label learning deals with instances having a set of class labels simultaneously, which widely exist in real-world applications. The goal of multi-label learning is to learn a model that assigns an appropriate set of labels to an unseen example from the training data. In recent years, this technique has been increasingly studied and widely applied to various fields including text annotation [1], [2], [3], automatic image annotation [4], [5], music emotion categorization [6], [7], [8] and so on [9], [10].

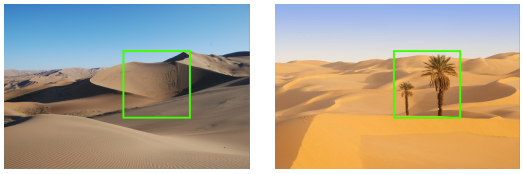
Based on the smoothness assumption, it is thought that the similar samples in the feature space should also have similar properties in the label space, so the neighbors' information of samples is often used as a kind of additional information to improve the performance of the model. For example, in ML-KNN [11], the *maximum a posteriori* (MAP) principle is used to determine the label set of unseen examples, which is based on the k-nearest neighbor information of samples. In LSF-CI [12], if two instances are neighbors to each other, their similarity is 1, so that the label space of the constrained instances are similar. Zhao and Guo [13] selected the k-nearest neighbors of the samples and applied the Laplacian manifold regularization to solve the incomplete label problem.

*Corresponding author: Xiuyi Jia (jiaxy@njust.edu.cn)

Many studies have tried to exploit the sample similarity in multi-label learning, and most of them calculated the similarity between samples based on the whole features, and all the labels share the similarity matrix of samples derived from the feature space. However, in many real-world applications, when we focus on different labels, the similarity of two samples may be different, we call this case that the samples are locally similar. For example, in Figure 1, image (a) has the label *desert* and *sky*, image (b) has the label *desert*, *sky* and *trees*. When we focus on the two labels of *desert* and *sky*, image (a) and image (b) are similar, but when we only focus on the label *trees*, the two images are not similar. Since a label is determined by a subset of features, if we use the whole feature space to calculate the similarity of samples on specific labels, the unrelated features may affect the final calculation result, resulting in inaccurate similarity. In Figure 1, we can easily observe that both the *desert* and the *sky* occupy most of the two images, so it can be said that the two images have a high similarity. If the similarity of the samples is defined globally to determine the relevance of the labels, it is easy to erroneously deduce that the two images have the same value on the label *trees*. However, if we look at the local part, as shown by the green box in the two images, they are significantly different, so that the two images have different values on the label *trees*. Therefore, calculating the corresponding sample similarity for each label separately can make more accurate use of the structural information of the sample and its neighbors, that is, similar samples have similar outputs.

In this paper, we propose a novel and effective multi-label learning method, which uses the local similarity of samples, named ML-LSS (Multi-Label learning with Local Similarity of Samples). In our method, for different labels, we only use the feature subset of the label when calculating its corresponding similarity matrix, and its feature subset is obtained by using the dimensionality reduction method that maximizes the dependency between the feature and the label. In addition, when we constrain the label space of the sample, L1-regularization is applied to sparse the weight parameter vector in which non-zero items represent the selected label-specific features, and the irrelevant features have a coefficient of 0 [12], [14], which is in line with our idea that we only use partial features to solve the local similarity of the samples.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE



(a) *sky*, *desert* (b) *sky*, *desert*, *trees*

Fig. 1. Illustration of the local similarity of samples, in the local area within the green box, the values of the two pictures are not similar in the label *trees*.

Finally, based on the smoothness assumption, we assume that if the feature subset is similar between samples, the outputs of the corresponing label should be similar. We use the Proximal Gradient Descent (PGD) to optimize the model. Comparison experiments on eleven data sets show that the proposed method is effective and can improve the performance of multi-label learning by using the local similarity of samples.

The main contributions of this study can be summarized as follows: 1) We propose a novel and effective multi-label learning method, which uses the local similarity of samples, named ML-LSS; 2) We extract label-specific features by using a dimensionality reduction method that maximizes the dependence between features and labels, and use corresponding feature subsets to calculate a local similarity matrix of samples for each label. Then, we assume that if the feature subset is similar between samples, the outputs of the corresponding label should be similar. Based on the idea that only partial features are used to construct the local similarity of samples, we use L1-regularization to sparse the weight parameter vector in which non-zero items represent the selected label-specific features.

The rest of the paper is organized as follows: Section II briefly reviews some related works. Section III introduces the proposed algorithm. Section IV reports the experiments on real-world data sets. Finally, Section V concludes this paper.

II. Related Work

To solve the problem of multi-label learning, a large number of algorithms have been proposed for multi-label learning. The existing multi-label learning algorithms can be divided into two categories, namely problem transformation and algorithm adaption [15]. On one hand, algorithm adaptation methods work by fitting algorithms to data, i.e., Binary Relevance (BR) [16], Label Power set (LP) [6]. On the other hand, problem transformation methods work by fitting data to algorithms, such as deep neural network [17], decision tree [18]. In addition, many researchers further use the neighbors' information of samples. For example, Zhang and Zhou [11] considered label prior probabilities gained from the k-nearest neighbors of the instance and utilized *maximum a posteriori* (MAP) principle to determine proper labels in their ML-KNN method. The WELL [19] method solved the weak label problem, and it constructed a similarity matrix of samples and embedded the correlation between labels in the model. In the optimization process, it not only solved the weight coefficient, but also solved the sample "appropriate similarity" matrix. LSF-CI [12] used the k-nearest neighbor graph model to create an instance similarity matrix. Specifically, the similarity between samples with mutual neighbors is 1, and the rest are 0.

For multi-label learning, feature selection approaches fall into two types: transformation-based approaches and direct approaches [20]. Problem transformation approaches transform a multi-label instance into multiple single-label instances, then exploit a classical single-label feature selection approach like the filter method, wrapper method or embedded method directly. For the direct approaches, Zhang and Zhou [21] proposed to use the HSIC method to find a lower-dimensional feature space in which the dependence between the features and labels are maximized. Lee and Kim [22] achieved feature selection on multi-label data by maximizing the mutual information between selected features and labels. Yan and Li [23] proposed a graph-margin based feature selection for multi-label data.

Label-specific feature selection in multi-label learning has also attracted a lot of attention in recent years. LIFT [24] was an algorithm to exploit label-specific features for multi-label learning. For each label, LIFT constructed features by conducting clustering analysis on its positive and negative instances. Yan et al. [25] employed the information theory to implement label-specific feature selection and assigned different weights to the different class instances according to the imbalance rate. Jia et al. [26] discussed the efficiency of jointly combining label-specific features and correlation information for multi-label learning. In LLSF [14], L1-regularization was applied to get the weights of features for each label, and the feature with zero weight value does not affect the final discrimination. Meanwhile, LLSF required that strongly correlated labels should have a large similarity between their weight vectors. Furthermore, LSF-CI [12] improved the LLSF algorithm by adding instance correlation.

III. The Proposed Approach

A. Multi-Label Learning

Assume $\mathcal{X} \in \mathcal{R}^{d \times n}$ be the input space and $\mathcal{Y} = \{-1, +1\}^q$ be the label space with q possible labels. We denote by $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ the training data that consists of n instances. $x_i \in \mathcal{X}$ is a d-dimensional feature vector $x_i = [x_{1i}, x_{2i}, \cdots, x_{di}]^T$ and $y_i = [y_{i1}, y_{i2}, \cdots, y_{iq}]$ is the label vector of x_i, each element $y_{ik} = 1$ if the label y_k is related to x_i, otherwise $y_{ik} = -1$. The goal of multi-label learning is to predict a label set for an unseen instance, so we need to learn a classifier: $f : \mathcal{X} \to \mathcal{Y}$. We assume f consists of q sub-functions, one for each label, i.e., $f = [f_1, f_2, \cdots, f_q]$. In this study, we apply the linear model for each f_k :

$$f_k(x_i) = x_i^T W_k, \tag{1}$$

Here, we add an additional dimension with a constant value of 1 for each data x_i $(1 \le i \le n)$, so $x_i = [x_{1i}, x_{2i}, \cdots, x_{di}, 1]^T$. The offset term b_j has been expanded into W_k, and $W_k = [W_{1k}, W_{2k}, \cdots, W_{dk}, b_k]^T$ represents the weight parameters of the linear model corresponding to the k-th label.

B. Combining Local Similarity of Samples

In reality, a label is often determined by a subset of features, rather than all features, and different labels may correspond to different subsets. With the previous discussion, we know that the samples are locally similar. In this paper, the learning procedure of ML-LSS consists of two steps. The first is to select the corresponding feature subset for each label, and the second is to use the feature subset to calculate the local similarity matrix of samples.

We attempt to find a lower-dimensional feature space for each label in which the dependence between the features and the label are maximized. Many criteria can be used to measure such dependence and here we adopt the Hilbert-Schmidt independence criterion (HSIC) [27], due to its simplicity and HSIC well measures the independence of two variables [21], [28]. HSIC is a kernel-based dependence metric for random variables, which measures the dependence between the feature and the label by computing the Hilbert-Schmidt-norm of the cross-covariance operator over the domain $\mathcal{Q} \times \mathcal{Y}$ reproducing kernel Hilbert spaces (RKHSs). We consider a linear projection P, assume that the instance x is projected into the new space $\mathcal{Q}$ by $\phi(x) = P^T x$. Then, we try to maximize the dependence between the feature description $\phi(x) \in \mathcal{Q}$ and the label $y_k \in \mathcal{Y}$ $(1 \leq k \leq q)$. Therefore, for each label y_k, we use HSIC to measure the correlation of the original feature space and label y_k. An empirical estimate of HSIC is:

$$\mathrm{HSIC}(\mathcal{Q}, \mathcal{Y}, P_{xy}) = (N-1)^{-2} tr(HKHL), \tag{2}$$

where P_{xy} is the joint distribution and $tr(\cdot)$ is the trace of matrix. $K = [K_{ij}]_{N\times N}$ and $L = [l_{ij}]_{N\times N}$ are the matrices of the inner product of instances in $\mathcal{Q}$ and $\mathcal{Y}$, and these could also be considered as the kernel matrices of $\mathcal{X}$ and y_k with kernel functions. $K = \langle \phi(x), \phi(x)' \rangle$, $L(y_k, y_k')$, $H = [H_{ij}]_{N\times N}$, $H_{ij} = \frac{\delta_{ij}-1}{N}$, and the δ_{ij} takes 1 when $i = j$ and 0 otherwise. In Eq. (2), the normalization term does not affect the result, so we only need to consider $tr(HKHL)$ in the solution process. Denote $X = [x_1, x_2, \cdots, x_n]$, x_i represents the i-th sample, $Y_k = [y_{1k}, y_{2k}, \cdots, y_{nk}]^T$, y_{ik} represents the k-th label belongs to the i-th sample. We adopt linear kernel matrix, thus $\phi(x) = P_k^T X$ and $K = \langle \phi(x), \phi(x)' \rangle = X^T P_k P_k^T X$, $L = Y_k Y_k^T$. P_k is the projection matrix we need to solve, and we can rewrite the optimization as follows:

$$P_k^* = \arg\max_{P_k} tr(HX^T P_k P_k^T HL). \tag{3}$$

The matrix P_k reduces the feature dimension to d-dimension and denote $P_k = [p_1, p_2, \cdots, p_d] (d \ll D)$, the column vectors of the matrix P_k forms a basis spanning of the new space. By constraining the basis to be orthonormal, we have

$$\max_{P_k} tr(HX^T P_k P_k^T XHL) \quad s.t. \quad (P_k^i)^T P_k^j = \delta_{ij}, \tag{4}$$

here, P_k^i $(1 \leq i \leq d)$ represents the i-th column of the P_k matrix. To solve this problem, we have:

$$\begin{aligned} tr(HX^T P_k P_k^T XHL) &= tr(\sum_{i=1}^{d} HX^T P_k^i (P_k^i)^T XHL) \\ &= \sum_{i=1}^{d} tr(HX^T P_k^i (P_k^i)^T XHL) = \sum_{i=1}^{d} (P_k^i)^T (XHLHX^T) P_k^i. \end{aligned} \tag{5}$$

It is easy to obtain the optimal $(P_k^i)^*$ by using the Lagrangian model. We can get the eigenvalues of $XHLHX^T$, assuming they are sorted as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0 \geq \cdots \geq \lambda_D$, the optimal $(P_k^i)^*$ are the normalized eigenvectors corresponding to the largest d eigenvalues.

We use the HSIC method to get a projection matrix for each label, so we can get a new space for each label that is maximized by its dependence. For each label, if the subset of features that determine the label is different, the similarity between the two samples may be different. So here we use the new feature space to calculate the local similarity matrix of samples for label y_k by the cosine similarity method, as follows:

$$\mathcal{R}_{ijk} = \frac{(P_k^T x_i)^T (P_k^T x_j)}{\|P_k^T x_i\| \|P_k^T x_j\|}, \tag{6}$$

R_{ijk} represents the similarity of the samples x_i and x_j on label y_k. And $R_{ijk} > 0$ indicates that the sample i and the sample j are positively correlated, $R_{ijk} < 0$ indicates that they are negatively correlated. Furthermore, they are irrelevant if $R_{ijk} = 0$. Based on the manifold assumption, we assume that if x_i and x_j are similar on label y_k, $f_k(x_i)$ and $f_k(x_j)$ should be similar, and vice versa. This idea makes us need to minimize the function below:

$$\Omega(f) = \sum_{k=1}^{q} \sum_{j=1}^{n} \sum_{i=1}^{n} \mathcal{R}_{ijk} \|f_k(x_i) - f_k(x_j)\|_2^2. \tag{7}$$

We adopt the squared Euclidean distance to measure the similarity. Considering that the sample local correlation is calculated based on the features that maximizes the label dependence, $f(x)$ should also be determined by the specific features. According to Eq. (1), if $W_{jk} = 0$, the j-th feature has no effect on the discrimination of the k-th label y_k. Only the features corresponding to the non-zero items in W_k are used to discriminate the k-th label, so the weight coefficient matrix should be a sparse matrix. Therefore, we apply L1-regularization on W_k. The objective function is defined as:

$$\begin{aligned} \min_W \frac{1}{2} \|X^T W - Y\|_F^2 + \lambda_1 \sum_{k=1}^{q} \|W_k\|_1 \\ + \frac{\lambda_2}{2} \sum_{k=1}^{q} \sum_{j=1}^{n} \sum_{i=1}^{n} R_{ijk} \|X_i^T W_k - X_j^T W_k\|_2^2, \end{aligned} \tag{8}$$

λ_1 and λ_2 are the balance factors. The first term is the loss function to measure the distance between the predicted value and the true value of the labels. The second term is an L1-regularization term to sparse the weight parameter vector. The third item is the local similarity constraint of samples, which

represents that the feature subset of label y_k is similar between samples, the distance between the two samples on label y_k should be small.

C. Optimization

The objective function in Eq. (8) can be represented as:

$$G(W) = \min_W \frac{1}{2}\|X^T W - Y\|_F^2 + \lambda_1 \sum_{k=1}^{q} \|W_k\|_1 + \frac{\lambda_2}{2} \sum_{k=1}^{q} tr(W_k^T X L_k X^T W_k), \tag{9}$$

where $L_k = A_k - R_k$, and $A_k = Diag(d_1, d_2, \cdots, d_n)$ is a dialog matrix, $d_i = \sum_{j=1}^{n} R_{ijk}$.

In Eq. (9), the L1 norm is a non-smooth function, which causes the objective function to be non-convex and non-smooth, making it difficult to employ the traditional gradient-descent optimization method. The accelerated proximal gradient method is usually applied to solve the L1 norm optimization problem. Here, we will solve for the weight coefficient $W_k (1 \le k \le q)$ corresponding to each label, the accelerated proximal gradient method divides the optimization target into two parts, which can be expressed as follows:

$$\min_{W_k \in \mathcal{H}} G(W_k) = s(W_k) + g(W_k), \tag{10}$$

where $\mathcal{H}$ represents Hilbert space, $s(W_k)$ is smoothing while $g(W_k)$ is non-smoothing. $s(W_k)$ is further $Lipschitz$ continuous, that is, $s(W_k)$ satisfies the following condition:

$$\|\nabla s(W_k') - \nabla s(W_k)\|_2^2 \le Lip \|\Delta W_k\|_2^2 \quad (\forall W_k', W_k), \tag{11}$$

where $\Delta W_k = W_k' - W_k$, Lip is the $Lipschitz$ constant. Considering the second order Taylor series of $s(W_k)$ at the current estimate of the parameter vector $W_k^{(t)}$:

$$\begin{aligned} s(W_k) &\cong s(W_k^{(t)}) + < \nabla s(W_k^{(t)}), W_k - W_k^{(t)} > \\ &+ \frac{Lip}{2}\|W_k - W_k^{(t)}\|_2^2 \\ &= \frac{Lip}{2}\|W_k - (W_k^t - \frac{1}{Lip}\nabla s(W_k^{(t)}))\|_2^2 + const, \end{aligned} \tag{12}$$

where $const$ is a constant unrelated to W_k, and $< ., . >$ represents the inner product. The minimum value of Eq. (12) can be obtained on $W_k^{(t+1)}$:

$$W_k^{(t+1)} = \arg\min_{W_k} \frac{Lip}{2}\|W_k - W_k^{(t)} - \frac{1}{Lip}\nabla s(W_k^{(t)})\|_2^2 + g(W_k). \tag{13}$$

For the Eq. (9) and Eq. (10), $s(W_k)$and $g(W_k)$ can be represented as:

$$\begin{aligned} s(W_k) &= \frac{1}{2}\|X^T W_k - Y_k\|_2^2 + \lambda_2 tr(W_k^T X L_k X^T W_k), \\ g(W_k) &= \lambda_1 \|W_k\|_1. \end{aligned} \tag{14}$$

Therefore, the $\nabla s(W_k)$ is calculated by:

$$\nabla s(W_k) = X(X^T W_k - Y_k) + 2\lambda_2 (X L_k X^T W_k). \tag{15}$$

Then, we have:

$$\begin{aligned} &\|\nabla s(W_k + \Delta W_k) - \nabla s(W_k)\|_2^2 \\ =&\|XX^T \Delta W + 2\lambda_2 X L_k X^T \Delta W\|_2^2 \\ \le&2\|XX^T \Delta W_k\|_2^2 + 2\|2\lambda_2 X L_k X^T \Delta W_k\|_2^2 \\ =&2(\|XX^T\|_2^2 + \|2\lambda_2 X L_k X\|_2^2)\|\Delta W_k\|_2^2. \end{aligned} \tag{16}$$

Thus, the $Lipschitz$ constant can be calculated by:

$$Lip = \sqrt{2(\|XX^T\|_2^2 + \|2\lambda_2 X L_k X^T\|_2)}. \tag{17}$$

Then, combined with Eq. (12), we can get the optimal solution of W_k iteratively by:

$$W_k^{(t+1)} = \arg\min_{W_k} \frac{Lip}{2}\|W_k - Z\|_2^2 + g(W_k), \tag{18}$$

$$Z = W_k^{(t)} - \frac{1}{Lip}(X(X^T W_k - Y) + \lambda_2 X L_k X^T W_k). \tag{19}$$

The previous work [29] has shown that setting $W_k^{(t)} = W_k^{(t)} + \frac{b^{(t-1)}-1}{b^{(t)}}(W_k^{(t)} - W_k^{(t-1)})$ can improve the convergence rate to $O(t^{-2})$, where $b^{(t)}$ satisfying $(b^{(t)})^2 - b^{(t)} \le (b^{(t-1)})^2$, and $W_k^{(t)}$ is the result of W_k at the t-th iteration. The closed solution of Eq. (18) can be calculated by a soft threshold method which is defined as:

$$(W_k^i)^{(t+1)} = \begin{cases} Z_i^{(t)} - \lambda_2/Lip, & \lambda_1/Lip < Z_i^{(t)}; \\ 0, & |Z_i^{(t)}| \le \lambda_1/Lip; \\ Z_i^{(t)} + \lambda_2/Lip, & Z_i^{(t)} < -\lambda_1/Lip. \end{cases} \tag{20}$$

Here, W_k^i and Z_i represent the i-th row of the W_k and Z matrices respectively. The detailed algorithm of the proposed method is shown in Algorithm 1.

IV. Experiments

A. Experiments Setup

In this study, we conduct extensive experiments on various multi-label real-world datasets. For each data set S, the number of instances is denoted as $|S|$, the number of features is denoted as $dim(S)$, and the number of labels is denoted as $L(S)$. In addition, $LCard(S)$ denotes cardinality representing the average value of labels belonging to instances, and 'Domain' denotes the types of the datasets. Table I summarizes the detailed characteristics of the eleven real-world multi-label datasets. We utilize five commonly used multi-label evaluation measures to evaluate the performance of the trained classifiers on the multi-label datasets: Hamming loss, Ranking loss, One error, Coverage and Average precision [15].

The performance of ML-LSS is compared against three well-established and two state-of-the-art multi-label learning algorithms, including LIFT [24], ML-KNN [11], MLFE [30], BR [16] and LSF-CI [12].

Algorithm 1: The ML-LSS algorithm

Input: The training set $D = \{x_i, y_i\}_{i=1}^{n}$, parameters λ_1, λ_2 , γ and the convergence criterion ξ.
Output: The regression parameters matrix W.

for *k=1 to q* **do**
  solve P_k by Eq. (5);
  comput the local simility martrix R_k by Eq. (6);
  initialize the parameters;
  $b_0, b_1 \leftarrow 1$, $W_k^{(0)}, W_k^{(1)} \leftarrow (XX^T + \gamma I)^{-1} XY_k$;
  $t \leftarrow 1$;
  while *stopping criterion is not satisfied* **do**
    $W_k^{(t)} = W_k^{(t)} + \frac{b^{(t-1)}-1}{b^{(t)}}(W_k^{(t)} - W_k^{(t-1)})$;
    update Z by Eq. (19);
    update $(W_k^i)^{(t+1)}$ by Eq. (20);
    $b^{(t+1)} \leftarrow \frac{1+\sqrt{4((b^{(t)})^2+1)}}{2}$;
    $t \leftarrow t + 1$;
  end
  return W_k;
end
return W.

TABLE I
STATISTICS OF THE ELEVEN DATASETS.

Datasets	$\|S\|$	$dim(S)$	$L(S)$	$LCard(S)$	Domain
flags	194	19	7	3.39	images
birds	645	260	19	1.01	audio
cal500	502	68	174	26.04	audio
genbase	662	1186	27	1.25	biology
medical	978	1449	45	1.25	text
llog	1460	1004	75	1.18	text
yeast	2417	103	14	4.24	biology
shashdot	3782	1079	22	1.18	text
arts	5000	462	26	1.64	text
corel5k	5000	499	374	3.52	images
education	5000	550	33	1.46	text

With the previous discussion, LIFT [24] proposed the idea of label-specific features for multi-label learning. ML-KNN [11] was an algorithm that works similarly as ML-LSS by using the neighbors' information of samples. BR [16] was a representative algorithm of problem transformation methods, which decomposed a multi-label learning problem into q independent binary (one-vs-rest) classification problems. MLFE [30] applied the multi-output regression techniques to train the prediction model under the MLFE framework which enriches the label information by utilizing the structural information of the feature space. LSF-CI [12] attempted to learn label specific features for each label with consideration of label correlation in label space and instance correlation in feature space simultaneously.

For the comparing algorithms, parameter configurations suggested in corresponding literatures are used, i.e. MLFE: parameters β_1, β_2 and β_3 are chosen among $\{1, 2, ..., 10\}$, $\{1, 10, 15\}$ and $\{1, 10\}$ respectively; LSF-CI: parameters λ_1 and λ_2 are selected from $\{2^{-10}, 2^{-9}, \cdots, 2^{10}\}$, λ_3 is selected from $\{2^{-12}, 2^{-11}, \cdots, 2^{12}\}$. For ML-LSS, the values of the parameters λ_1 and λ_2 are selected among $\{2^{-5}, 2^{-4}, \cdots, 2^{6}\}$ with cross-validation on the training set. For performance evaluation, we perform a 10×5-fold cross-validation on each dataset, where the mean metric value, as well as standard deviation, are recorded for each comparing algorithm.

B. Experiments Result

In Table II, we summarize the detailed experimental results of five comparing algorithms on each data set, where the best performance among the five comparing algorithms is highlighted in boldface.

Across all the 55 configurations (i.e. 5 criteria $\times$ 11 datasets as shown in Table II), ML-LSS ranks in 1^{st} place at 35 cases, in 2^{nd} place at 12 cases, in 3^{rd} and 4^{th} places at only 8 cases, and never ranks in 5^{th} and 6^{th} places.

To further analyze the relative performance between the comparing algorithms, *Friedman test* [31] is used as the statistical test in this paper. Table III summarizes the Friedman statistic F_F and the corresponding critical value on each evaluation metric. For each evaluation metric, the null hypothesis of indistinguishable performance among the comparing algorithms is clearly rejected at the 0.05 significance level. Therefore, *Bonferroni-Dunn test* [31] at 0.05 significance level is employed to test whether our proposed method ML-LSS achieves competitive performance against the comparing algorithms. Here, the ML-LSS is regarded as a control algorithm whose average level difference with the comparison algorithm is calibrated with *critical difference* (CD). Accordingly, ML-LSS is deemed to have a significantly different performance to one comparing algorithm if their average ranks differ by at least one CD (CD=2.055 in this paper: # comparing algorithms k = 6, # data sets N = 11).

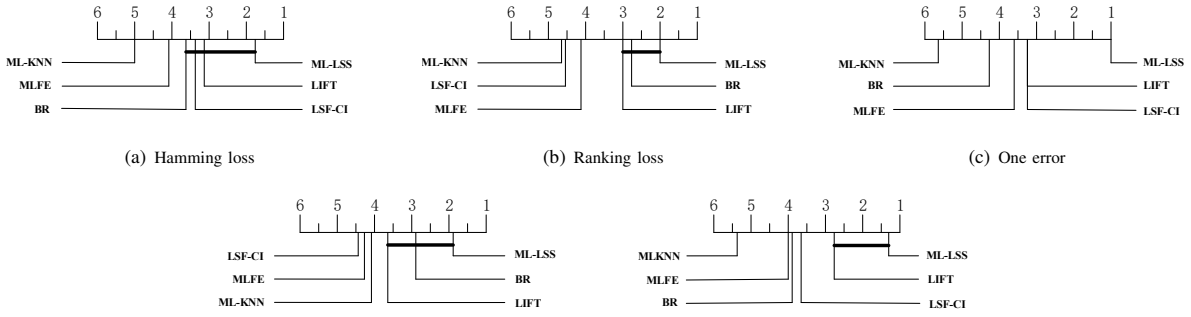
Figure 2 illustrates the CD diagrams [31] on each evaluation metric, where the average rank of each comparing algorithm is marked along the axis (lower ranks to the right). In each sub-figure, any comparing algorithm whose average rank is within one CD to that of ML-LSS is connected with a thick line. Otherwise, any algorithm not connected with ML-LSS is considered to have significant different performance between them. From Figure 2, we can get the following observations:

- ML-LSS achieves an optimal average rank in terms of all evaluation metrics.
- ML-LSS significantly outperforms ML-KNN and MLFE in terms of all evaluation metrics.
- ML-LSS is significantly outperforming other comparing algorithm in terms of one error, comparable to LIFT in terms of hamming loss, ranking loss, coverage and average precision, comparable to BR in terms of hamming loss, ranking loss and coverage, comparable to LSF-CI in terms of hamming loss, average precision, and significantly outperforms LIFT and BR LSF-CI on all the other cases.

TABLE II

COMPARISON RESULTS (MEAN±STD.) OF MULTI-LABEL METHODS ON REAL-WORLD DATASETS. THE BEST PERFORMANCE ON EACH MEASURE IS MARKED IN BOLD.

data	algorithm	Hamming loss↓	Ranking loss↓	One error↓	Coverage↓	Average precision↑
flags	LIFT	0.2656±0.0069(2)	**0.2161±0.0053(1)**	0.2379±0.0217(3)	**0.5385±0.0044(1)**	**0.8084±0.0053(1)**
	ML-KNN	0.3239±0.0046(6)	0.2483±0.0070(6)	0.2544±0.0221(6)	0.5737±0.0070(6)	0.7868±0.0075(6)
	MLFE	0.2737±0.0070(4)	0.2346±0.0083(4)	0.2367±0.0189(2)	0.5537±0.0050(4)	0.7981±0.0077(4)
	BR	**0.2645±0.0063(1)**	0.2240±0.0067(2)	0.2396±0.0104(4)	0.5429±0.0054(2)	0.8034±0.0043(3)
	LSF-CI	0.2861±0.0059(5)	0.2426±0.0061(5)	0.2397±0.0099(5)	0.5601±0.0055(5)	0.7913±0.0045(5)
	ML-LSS	0.2681±0.0058(3)	0.2306±0.0049(3)	**0.2150±0.0171(1)**	0.5520±0.0029(3)	0.8072 ±0.0055(2)
cal500	LIFT	0.1386±0.0004(4)	0.1843±0.0007(3)	0.1307±0.0061(3)	0.7667±0.0030(5)	0.4989±0.0015(4)
	ML-KNN	0.1390±0.0003(5)	0.1865±0.0006(4)	0.1515±0.0072(6)	0.7665±0.0052(4)	0.4886±0.0012(6)
	MLFE	0.1402±0.0017(6)	0.1865±0.0013(5)	0.1321±0.0025(4)	0.7653±0.0005(3)	0.4984±0.0000(5)
	BR	**0.1371±0.0003(1)**	0.1799±0.0003(2)	0.1181±0.0023(2)	0.7475±0.0015(2)	0.5030±0.0008(2)
	LSF-CI	0.1379±0.0005(3)	0.1866±0.0010(6)	0.1451±0.0057(5)	0.7821±0.0025(6)	0.5025±0.0015(3)
	ML-LSS	0.1378±0.0000(2)	**0.1794±0.0005(1)**	**0.1165±0.0010(1)**	**0.7457±0.0014(1)**	**0.5031±0.0009(1)**
genbase	LIFT	0.0083±0.0006(5)	0.0148±0.0020(6)	0.0044±0.0051(4)	0.0448±0.0026(6)	0.9778±0.0037(5)
	ML-KNN	0.0048±0.0003(4)	0.0059±0.0011(4)	0.0111±0.0028(5)	0.0195±0.0015(4)	0.9871±0.0022(4)
	MLFE	0.0012±0.0002(2)	0.0030±0.0005(2)	0.0025±0.0007(3)	0.0143±0.0009(2)	0.9939±0.0010(3)
	BR	0.0092±0.0051(6)	0.0124±0.0052(5)	0.2215±0.1395(6)	0.0246±0.0050(5)	0.8883±0.0668(6)
	LSF-CI	0.0019±0.0004(3)	0.0036±0.0007(3)	0.0023±0.0015(2)	0.0152±0.0012(3)	0.9946±0.0009(2)
	ML-LSS	**0.0008±0.0001(1)**	**0.0021±0.0004(1)**	**0.0023±0.0008(1)**	**0.0128±0.0006(1)**	**0.9953±0.0006(1)**
medical	LIFT	0.0124±0.0002(3)	0.0283±0.0018(5)	0.1612±0.0067(4)	0.0433±0.0026(5)	0.8724±0.0043(4)
	ML-KNN	0.0158±0.0003(6)	0.0450±0.0019(6)	0.2507±0.0058(6)	0.0641±0.0023(6)	0.8035±0.0044(6)
	MLFE	**0.0100±0.0001(1)**	**0.0210±0.0012(1)**	0.1346±0.0029(2)	0.0342±0.0010(2)	0.8969±0.0028(2)
	BR	0.0132±0.0039(5)	0.0266±0.0041(4)	0.2419±0.1141(5)	0.0409±0.0041(4)	0.8373±0.0623(5)
	LSF-CI	0.0130±0.0006(4)	0.0246±0.0019(3)	0.1395±0.0067(3)	0.0378±0.0028(3)	0.8948±0.0046(3)
	ML-LSS	0.0101±0.0002(2)	0.0213±0.0009(2)	**0.1339±0.0063(1)**	**0.0332±0.0013(1)**	**0.9010±0.0028(1)**
birds	LIFT	0.0459±0.0007(4)	0.1880±0.0080(4)	0.4433±0.0166(4)	0.1301±0.1301(4)	0.5943±0.0110(4)
	ML-KNN	0.0510±0.0009(5)	0.1988±0.0066(5)	0.5435±0.0152(5)	0.1336±0.0039(5)	0.5443±0.0046(5)
	MLFE	0.0458±0.0006(3)	0.1632±0.0044(2)	0.3560±0.0137(2)	0.1192±0.0029(2)	0.6555±0.0084(2)
	BR	0.0541±0.0003(6)	0.2845±0.0028(6)	0.6903±0.0206(6)	0.1864±0.0019(6)	0.4089±0.0103(6)
	LSF-CI	0.0456±0.0009(2)	0.1633±0.0019(3)	0.3592±0.0103(3)	0.1195±0.0015(3)	0.6541±0.0053(3)
	ML-LSS	**0.0455±0.0010(1)**	**0.1627±0.0065(1)**	**0.3507±0.0130(1)**	**0.1191±0.0033(1)**	**0.6580±0.0058(1)**
llog	LIFT	0.0178±0.0001(6)	0.1520±0.0041(4)	0.6670±0.0070(2)	0.1962±0.0047(6)	0.4222±0.0053(2)
	ML-KNN	0.0158±0.0001(5)	0.1665±0.0022(6)	0.8041±0.0069(6)	0.1750±0.0024(4)	0.3025±0.0040(6)
	MLFE	0.0156±0.0001(4)	0.1474±0.0010(3)	0.7146±0.0062(5)	0.1823±0.0013(5)	0.3612±0.0018(5)
	BR	0.0154±0.0001(3)	**0.1319±0.0014(1)**	0.7066±0.0092(4)	**0.1442±0.0017(1)**	0.3876±0.0047(4)
	LSF-CI	0.0153±0.0001(2)	0.1522±0.0028(5)	0.6704±0.0042(3)	0.1662±0.0023(3)	0.4212±0.0024(3)
	ML-LSS	**0.0153±0.0000(1)**	0.1421±0.0018(2)	**0.6534±0.0052(1)**	0.1569±0.0017(2)	**0.4267±0.0023(1)**
yeast	LIFT	**0.1931±0.0009(1)**	**0.1662±0.0007(1)**	0.2264±0.0042(2)	0.4549±0.0014(3)	**0.7666±0.0015(1)**
	ML-KNN	0.1955±0.0010(2)	0.1694±0.0010(3)	0.2356±0.0035(4)	**0.4505±0.0014(1)**	0.7614±0.0015(3)
	MLFE	0.2073±0.0007(5)	0.1782±0.0012(5)	0.2441±0.0013(5)	0.4691±0.0011(5)	0.7490±0.0011(5)
	BR	0.1994±0.0005(4)	0.1724±0.0003(4)	0.2268±0.0022(3)	0.4602±0.0004(4)	0.7585±0.0005(4)
	LSF-CI	0.4108±0.0015(6)	0.3392±0.0020(6)	0.3554±0.0043(6)	0.6237±0.0017(6)	0.6175±0.0017(6)
	ML-LSS	0.1993±0.0008(3)	0.1692±0.0006(2)	**0.2213±0.0024(1)**	0.4544±0.0007(2)	0.7622±0.0010(2)
slashdot	LIFT	0.0173±0.0002(5)	0.0465±0.0006(3)	0.0891±0.0011(3)	0.0444±0.0005(3)	0.8928±0.0013(3)
	ML-KNN	0.0221±0.0001(6)	0.0500±0.0005(4)	0.0944±0.0004(6)	0.0463±0.0006(4)	0.8829±0.0007(6)
	MLFE	0.0158±0.0002(3)	0.0591±0.0012(6)	0.0897±0.0012(5)	0.0569±0.0012(6)	0.8865±0.0007(5)
	BR	0.0165±0.0002(4)	0.0413±0.0006(2)	0.0895±0.0013(4)	0.0374±0.0007(2)	0.8950±0.0007(2)
	LSF-CI	0.0152±0.0001(2)	0.0547±0.0012(5)	0.0856±0.0013(2)	0.0526±0.0012(5)	0.8927±0.0008(4)
	ML-LSS	**0.0150±0.0001(1)**	**0.0406±0.0005(1)**	**0.0856±0.0009(1)**	**0.0372±0.0006(1)**	**0.8993±0.0007(1)**
arts	LIFT	**0.0531±0.0001(1.5)**	0.1143±0.0005(2)	0.4625±0.0021(4)	0.1737±0.0006(2)	0.6233±0.0012(3)
	ML-KNN	0.0599±0.0000(6)	0.1492±0.0007(6)	0.6150±0.0034(6)	0.2072±0.0008(4)	0.5225±0.0021(6)
	MLFE	0.0581±0.0002(5)	0.1486±0.0005(5)	0.4794±0.0038(5)	0.2198±0.0010(6)	0.5991±0.0015(5)
	BR	0.0535±0.0023(3)	**0.1119±0.0025(1)**	0.4590±0.0341(3)	**0.1708±0.0025(1)**	0.6555±0.0084(2)
	LSF-CI	0.0550±0.0001(4)	0.1430±0.0007(4)	0.4518±0.0030(2)	0.2139±0.0012(5)	0.6202±0.0009(4)
	ML-LSS	**0.0531±0.0001(1.5)**	0.1293±0.0004(3)	**0.4489±0.0026(1)**	0.2006±0.0005(3)	**0.6300±0.0017(1)**
corel5k	LIFT	**0.0094±0.0000(2)**	0.1257±0.0005(3)	0.6588±0.0034(3)	0.2995±0.0015(3)	**0.3028±0.0012(1)**
	ML-KNN	0.0094±0.0009(5)	0.1358±0.0003(4)	0.7394±0.0023(6)	0.3098±0.0007(4)	0.2437±0.0008(6)
	MLFE	0.0098±0.0000(6)	0.2149±0.0035(6)	0.6579±0.0014(4)	0.4735±0.0066(6)	0.2857±0.0008(4)
	BR	0.0094±0.0002(4)	**0.1237±0.0006(1)**	0.6763±0.0207(5)	**0.2892±0.0013(1)**	0.2772±0.0052(5)
	LSF-CI	**0.0094±0.0000(2)**	0.1726±0.0012(5)	0.6477±0.0025(2)	0.4011±0.0016(5)	0.2962±0.0009(2)
	ML-LSS	**0.0094±0.0000(2)**	0.1246±0.0006(2)	**0.6443±0.0047(1)**	0.2941±0.0012(2)	0.2978±0.0012(2)
education	LIFT	**0.0375±0.0001(1.5)**	**0.0708±0.0005(1)**	0.4663±0.0015(4)	0.0992±0.0007(2)	0.6420±0.0015(2)
	ML-KNN	0.0383±0.0001(5)	0.0783±0.0003(3)	0.4999±0.0036(6)	0.1056±0.0003(3)	0.6125±0.0023(6)
	MLFE	0.0384±0.0005(6)	0.0946±0.0044(6)	0.4634±0.0009(2)	0.1367±0.0061(6)	0.6375±0.0014(3)
	BR	0.0375±0.0018(3)	0.0710±0.0019(2)	0.4739±0.0327(5)	**0.0991±0.0018(1)**	0.6366±0.0166(4)
	LSF-CI	0.0378±0.0001(4)	0.0932±0.0009(5)	0.4636±0.0026(3)	0.1352±0.0011(5)	0.6362±0.0011(5)
	ML-LSS	**0.0375±0.0001(1.5)**	0.0796±0.0007(4)	**0.4615±0.0023(1)**	0.1160±0.0008(4)	**0.6421±0.0015(1)**



(a) Hamming loss (b) Ranking loss (c) One error

(d) Coverage (e) Average precision

Fig. 2. Comparison of ML-LSS (control algorithm) against five comparing algorithms with the *Bonferroni-Dunn test*. Algorithms not connected with ML-LSS in the CD diagram are considered to have significantly different performance from the control algorithm (CD=2.055 at 0.05 significance level).

TABLE III
FRIEDMAN STATISTICS F_F IN TERMS OF EACH EVALUATION METRICAND THE CRITICAL VALUE AT 0.05 SIGNIFICANCE LEVEL (# COMPARING ALGORITHMS K = 6, # DATA SETS N = 11).

Evaluation metrics	F_F	critical value
Hamming loss	5.0594	
Ranking loss	4.9968	
One error	19.2479	2.4004
Coverage	6.1513	
Average precision	12.7686	

C. Sensitivity Analysis

In order to test the robustness of the proposed algorithm, we also analyzed the influence of the trade-off parameters on the experimental results, including the parameters λ_1 and λ_2. The larger the λ_1, the more important the L1 regularization is, the larger the λ_2 is, the more important the local similarity of samples is. Due to space limitations, we only report the experimental results on the flags dataset using the Ranking loss and One error evaluation measures, the results of other datasets are similar. As shown in Figure 3, we can observe that when the parameter λ_1 and the parameter λ_2 are at an appropriate value, both evaluation measures reach the optimal value. In addition, we compared the running time of the six algorithms on the dataset flags. As shown in Figure 4, our algorithm has the shortest running time, and the MLFE [30] algorithm has the longest running time, which indicates that the time loss of our algorithm is small.

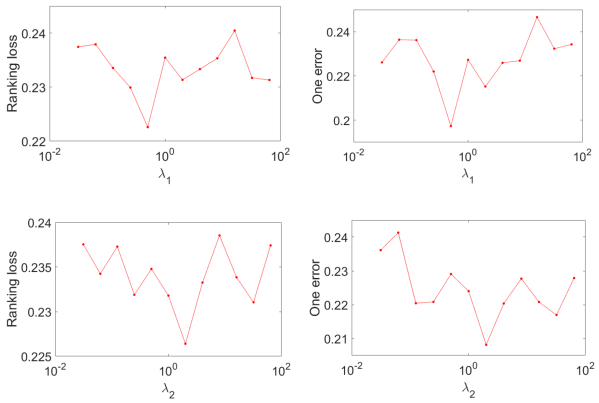


Fig. 3. Influence of λ_1 (λ_2 = 2) and λ_2 (λ_1 = 0.1) with Ranking loss and One error evaluation measures on dataset flags.

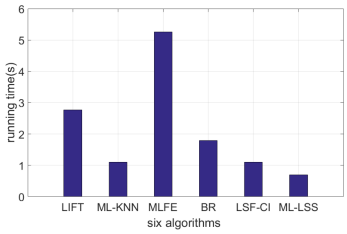


Fig. 4. Comparison of the running time of six algorithms on the dataset flags, assuming the running time is t, the Y-axis represents $\log(t+1)$.

V. CONCLUSION

In the real world, when the labels we focus on are different, the similarity between the samples may be different. We can say that the samples are only locally similar, whereas the previous algorithms assume that all labels share the same sample similarity matrix, which may damage the model performance. In this paper, we propose a method that makes good use of the local similarity of the sample. Specifically, we assume that the different labels are determined by a different subset of the original features, and the feature subset of each label is calculated by the dimensionality reduction method that maximizes the dependence of the features and label. In addition, we apply L1 regularization to sparse weight parameter vectors. Finally, we assume that the subset of features that determine a label is similar between samples, and the value of samples in the label should be similar. We carried out extensive experiments to validate the effectiveness of our algorithm in comparison with other state-of-the-art approaches on various data sets.

ACKNOWLEDGEMENTS

This work is jointly supported by the National Natural Science Foundation of China (Nos. 61906090, 61773208), the Natural Science Foundation of Jiangsu Province (Nos. BK20191287, BK20170809), the Fundamental Research Funds for the Central Universities (No. 30920021131), and the China Postdoctoral Science Foundation (No. 2018M632304).

REFERENCES

[1] Z. He, J. Wu, and P. Lv, "Multi-label text classification based on the label correlation mixture model," *Intelligent Data Analysis*, vol. 21, no. 6, pp. 1371–1392, 2017.

[2] H. Kazawa, T. Izumitani, H. Taira, and E. Maeda, "Maximal margin labeling for multi-topic text categorization," in *Advances in neural information processing systems*, 2005, pp. 649–656.

[3] D. Zha and C. Li, "Multi-label dataless text classification with topic modeling," *Knowledge and Information Systems*, vol. 61, no. 1, pp. 137–160, 2019.

[4] Y. Li, Y. Song, and J. Luo, "Improving pairwise ranking for multi-label image classification," in *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1837–1845.

[5] M. Tan, Q. Shi, A. van den Hengel, C. Shen, J. Gao, F. Hu, and Z. Zhang, "Learning graph structure for multi-label image classification via clique generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4100–4109.

[6] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. P. Vlahavas, "Multi-label classification of music into emotions," in *Proceedings of International Conference on Music Information Retrieval*, 2008, pp. 325–330.

[7] B. Wu, E. Zhong, A. Horner, and Q. Yang, "Music emotion recognition by multi-label multi-layer multi-instance multi-view learning," in *Proceedings of the ACM international conference on Multimedia*, 2014, pp. 117–126.

[8] S. AlZu'bi, O. Badarneh, B. Hawashin, M. Al-Ayyoub, N. Alhindawi, and Y. Jararweh, "Multi-label emotion classification for arabic tweets," in *Proceedings of the International Conference on Social Networks Analysis*, 2019, pp. 499–504.

[9] Z. Zhou and M. Zhang, "Multi-label learning," in *Sammut, C., and Webb, G. I., eds., Encyclopedia of Machine Learning and Data Mining*. Berlin: Springer, 2017, pp. 875–881.

[10] S. Huang, G. Li, W. Huang, and S. Li, "Incremental multi-label learning with active queries," *Journal of Computer Science and Technologys*, vol. 35, no. 2, pp. 234–246, 2020.

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

[12] H. Han, M. Huang, Y. Zhang, X. Yang, and W. Feng, "Multi-label learning with label specific features using correlation information," *IEEE Access*, vol. 7, pp. 11 474–11 484, 2019.

[13] F. Zhao and Y. Guo, "Semi-supervised multi-label learning with incomplete labels," in *Proceedings of International Joint Conference on Artificial Intelligence*, 2015, pp. 4062–4068.

[14] J. Huang, G. Li, Q. Huang, and X. Wu, "Learning label specific features for multi-label classification," in *Proceedings of IEEE International Conference on Data Mining*, 2015, pp. 181–190.

[15] M. Zhang and Z. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[16] 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.

[17] Z. Chen, Z. Chi, H. Fu, and D. Feng, "Multi-instance multi-label image classification: A neural approach," *Neurocomputing*, vol. 99, pp. 298–306, 2013.

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

[19] Y. Y. Sun, Y. Zhang, and Z. H. Zhou, "Multi-label learning with weak label," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2010, pp. 539–598.

[20] R. B. Pereira, A. Plastino, B. Zadrozny, and L. H. C. Merschmann, "Categorizing feature selection methods for multi-label classification," *Artificial Intelligence Review*, vol. 49, no. 1, pp. 1–22, 2016.

[21] Y. Zhang and Z. Zhou, "Multi-label dimensionality reduction via dependence maximization," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2008, pp. 1503–1505.

[22] J. Lee and D. W. Kim, "Feature selection for multi-label classification using multivariate mutual information," *Pattern Recognition Letters*, vol. 34, no. 3, pp. 349–357, 2013.

[23] P. Yan and Y. Li, "Graph-margin based multi-label feature selection," in *Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases*, 2016, pp. 540–555.

[24] M. Zhang, "LIFT: multi-label learning with label-specific features," in *Proceedings of International Joint Conference on Artificial Intelligence*, 2011, pp. 1609–1614.

[25] Y. Yan, S. Li, Z. Yang, X. Zhang, J. Li, A. Wang, and J. Zhang, "Multi-label learning with label-specific feature selection," in *Proceedings of International Conference on Neural Information Processing*, 2017, pp. 305–315.

[26] X. Jia, S. Zhu, and W. Li, "Joint label-specific features and correlation information for multi-label le," *Journal of Computer Science and Technology*, vol. 35, no. 2, pp. 247–258, 2020.

[27] A. Gretton, O. Bousquet, A. J. Smola, and B. Schölkopf, "Measuring statistical dependence with hilbert-schmidt norms," in *Proceedings of International Conference on Algorithmic Learning Theory*, 2005, pp. 63–77.

[28] N. Quadrianto, L. Song, and A. J. Smola, "Kernelized sorting," in *Proceedings of the Annual Conference on Neural Information Processing Systems*, 2008, pp. 1289–1296.

[29] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," *Journal of the Marine Biological Association of the United Kingdom*, vol. 56, no. 3, pp. 707–722, 2009.

[30] Q. Zhang, Y. Zhong, and M. Zhang, "Feature-induced labeling information enrichment for multi-label learning," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2018, pp. 4446–4453.

[31] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.