Leveraging Label Specific Discriminant Mapping Features for Multi-Label Learning

YUMENG GUO, Tongji University; The Hong Kong Polytechnic University
FULAI CHUNG, The Hong Kong Polytechnic University
GUOZHENG LI, Tongji University; China Academy of Chinese Medical Sciences
JIANCONG WANG, University of Pennsylvania
JAMES C. GEE, University of Pennsylvania

As an important machine learning task, multi-label learning deals with the problem where each sample instance (feature vector) is associated with multiple labels simultaneously. Most existing approaches focus on manipulating the label space, such as exploiting correlations between labels and reducing label space dimension, with identical feature space in the process of classification. One potential drawback of this traditional strategy is that each label might have its own specific characteristics and using identical features for all label cannot lead to optimized performance. In this paper, we propose an effective algorithm named LSDM, i.e. *leveraging Label Specific Discriminant Mapping features for multi-label learning*, to overcome the drawback. LSDM sets diverse ratio parameter values to conduct cluster analysis on the positive and negative instances of identical label. It reconstructs label specific feature space which includes distance information and spatial topology information. Our experimental results show that combining these two parts of information in the new feature representation can better exploit the clustering results in the learning process. Due to the problem of diverse combinations for identical label, we employ simplified linear discriminant analysis to efficiently excavate optimal one for each label and perform classification by querying the corresponding results. Comparison with the state-of-the-art algorithms on a total of 20 benchmark datasets clearly manifests the competitiveness of LSDM.

CCS Concepts: • **Computing methodologies** *Learning latent representations*;

Additional Key Words and Phrases: Machine Learning, Multi-label Learning, Label Specific Features

ACM Reference Format:
Yumeng Guo, Fulai Chung, Guozheng Li, Jiancong Wang, and James C. Gee. 2019. Leveraging Label Specific Discriminant Mapping Features for Multi-Label Learning. *ACM Trans. Web* 1, 1 (February 2019), 21 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

Traditional single-label learning deals with the problem where each instance is associated with only one class label. However, in various real-world applications, multi-label examples exist as shown in Fig. 1, where each instance is associated with multiple class labels simultaneously [35], and

Authors' addresses: Yumeng Guo, Tongji University; The Hong Kong Polytechnic University, Department of Control Science and Engineering; Department of Computing, yumeng_guo@foxmail.com; Fulai Chung, The Hong Kong Polytechnic University, Department of Computing, cskchung@comp.polyu.edu.hk; Guozheng Li, Tongji University; China Academy of Chinese Medical Sciences, Department of Control Science and Engineering; Data Center of Traditional Chinese Medicine, gzli@ndctcm.cn; Jiancong Wang, University of Pennsylvania, Penn Image Computing and Science Laboratory, Department of Radiology, jiancong.wang@pennmedicine.upenn.edu; James C. Gee, University of Pennsylvania, Penn Image Computing and Science Laboratory, Department of Radiology, gee@upenn.edu.

1559-1131/2019/2-ART $15.00
https://doi.org/0000001.0000001

appear also in areas like text categorization where each document may belong to several topics [31], [30], [7], bioinformatics where each gene may be associated with a set of functional classes [1], [5], [37], scene recognition where each image may demonstrate several semantic classes [3], [4], [36]. Multi-label learning aims to build classification models for multi-label objects.

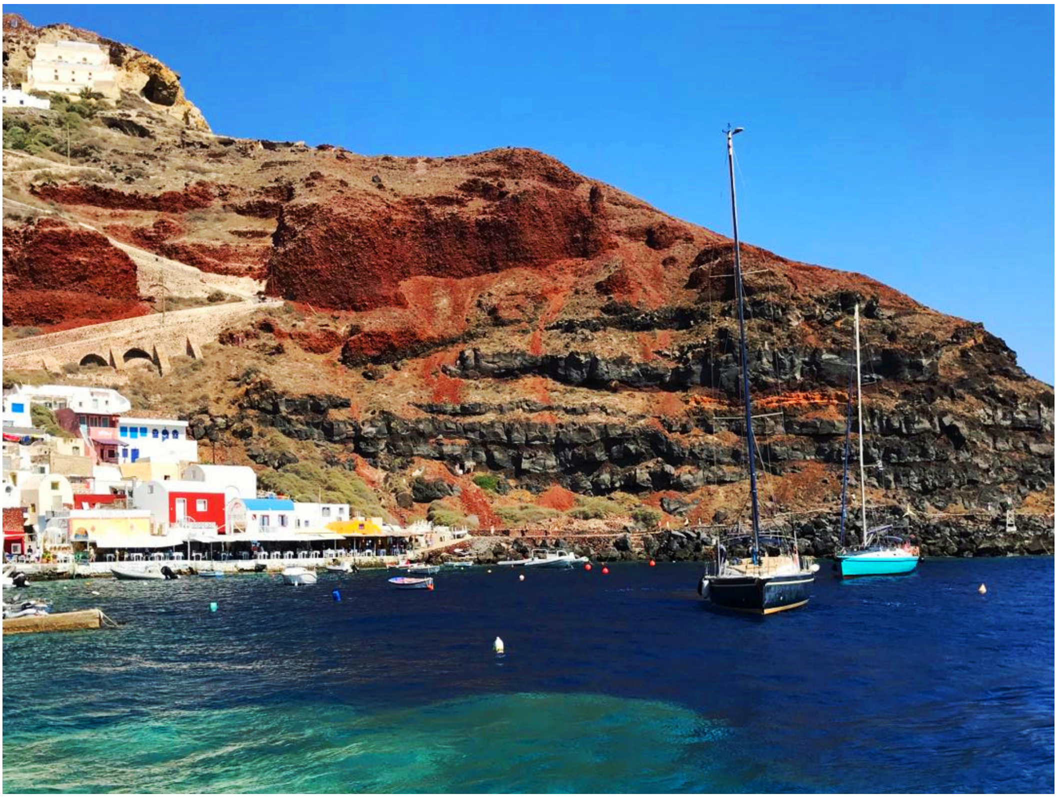


Fig. 1. An exemplar natural scene image which has been annotated with multiple labels: *sky, mountain, house, boat, water* (authors' own photograph).

For the multi-label learning problem, let $\mathcal{X} = \mathbb{R}^d$ denote the d-dimensional feature space and $\mathcal{Y} = \mathbb{R}^q$ $(\{l_k \in \{0,1\}|1 \leq k \leq q\})$ denote the q-dimensional label space. Each instance $\boldsymbol{x} \in \mathcal{X}$ matches a subset of labels $\boldsymbol{y} \subseteq \mathcal{Y}$, which can be equivalently written in the form of a binary vector $\boldsymbol{y} \in \{0,1\}^q$, with each bit l_k indicating the relevant or irrelevant label. Then, the goal of multi-label learning is to build a classier $h : \mathcal{X} \rightarrow 2^{\mathcal{Y}}$ which maps an instance to a subset of labels. One common strategy adopted by existing approaches is manipulating label space $\mathcal{Y}$, such as exploiting correlations between labels and reducing label space dimension, with identical feature representation of the instance, i.e. $\boldsymbol{x}$, to finish the classification task. Although many algorithms [45], [22], [15] have been designed for this strategy, it might only explore partial essence of multi-label learning. Videlicet, it might be suboptimal as the specific characteristics of each label cannot be distinguished from each other. For example, in automatic image annotation, suppose grassland and elephant are two possible classes appearing in the label space, intuitively, color-based features would be preferred in discriminating grassland and non-grassland images, while texture-based features would be preferred in discriminating elephant and non-elephant images.

Encouragingly, a multi-label learning algorithm named LIFT [41], [42] has been recently proposed to explore the label-specific features [16], i.e. the most pertinent and discriminative features for each class label. For each class label $l_k \in \mathcal{Y}$, LIFT performs cluster analysis on the positive and negative training instances, and then reconstructs training and testing feature spaces specific to each label l_k by querying the clustering results. The label specific features of LIFT are represented by distances between the original instances and the cluster centers. However, it still has two drawbacks. First, it utilizes identical ratio parameter to control the number of clusters for each label, i.e. different cluster centers corresponding to different reconstructed feature spaces, which ignores the differences between labels. Second, it only uses the distance information which does not exploit the clustering results comprehensively. Inspired by this work, we ulteriorly explore the reconstructed feature space based on label specific information to overcome the drawbacks of LIFT.

In this paper, we propose to *leverage Label Specific Discriminant Mapping features for multi-label learning* and derive a corresponding algorithm called LSDM. It sets more different values of ratio parameter to generate several groups of cluster centers for each label. It also reconstructs several feature spaces specific to each label by conducting cluster analysis on its positive and negative instances. The label specific features of LSDM includes distance information as in LIFT and new additions represented by the weights of using the cluster centers to linearly represent the original instances. The distance mapping features mainly contain far and near information between instances and the cluster centers, while linear representation features can describe the spatial topological information between them. Through empirical test, we find the combination of the two types of features can better exploit the clustering results.

For the ratio parameter setting, i.e. using several values for each label simultaneously, we attempt to excavate the optimally reconstructed feature space corresponding to ratio value in an efficient manner. An intuitive method to handle this problem is employing classifiers. While the number of ratio value and size of dataset are large, using classifiers should bring high computational complexity leading to time-consuming process. Here, we make an assumption that if a reconstructed feature space for label l_k is good, the distances within positive or negative instances are small and between positive and negative instances are large. Due to this assumption, we utilize the method of simplified linear discriminant analysis (sLDA), which only computes two mean values and scatters of the positive and negative training instances based on the reconstructed feature space for each label without mapping to 1-dimension space (dimension reduction), in order to deal with it efficiently as a compromise.

Briefly, LSDM learns from multi-label data with five intuitive and simplified steps. Firstly, for each class label $l_k \in \mathcal{Y}$, cluster analysis is performed on its positive and negative training instances; Secondly and thirdly, several reconstructed feature spaces based on distance mapping and linear representation with respect to l_k are generated by querying different clustering results. Fourthly, sLDA is employed to excavate optimal one from different reconstructed feature spaces of identical class label efficiently. At last, a family of q classifiers are learned from the excavated results.

The rest of this paper is organized as follows. Section 2, reviews some existing multi-label learning approaches. Section 3, presents the proposed LSDM algorithm. Section 4, presents the design of the experiment. Section 5, reports comparative experimental results over a wide range of multi-label datasets. Finally, Section 6, concludes and discusses several issues for future work.

2 RELATED WORK

Recently, multi-label learning has received rapidly increasing attention from machine learning and pattern recognition communities, due to its widely existing applications in real world. There is a rich body of work on the research of multi-label learning. Generally, according to the popular taxonomy presented in [33] the existing approaches can be categorized into two classes, namely, problem

Table 1. Summary of Major Mathematical Notations

Notations	Mathematical Meanings
$\mathcal{X}$	d-dimensional feature space $\mathbb{R}^d$
$\mathcal{Y}$	q-dimensional label space $\mathbb{R}^q$ $(\{l_k \in \{0,1\} \mid 1 \le k \le q\})$
$\boldsymbol{x}$	d-dimensional feature vector $(x_1, x_2, \ldots, x_d)^{\mathrm{T}}$ $(\boldsymbol{x} \in \mathcal{X})$
$\boldsymbol{y}$	label set associated with $\boldsymbol{x}$ $(\boldsymbol{y} \subseteq \mathcal{Y})$
$\mathcal{D}$	multi-label training set $\{(\boldsymbol{x}_i, \boldsymbol{y}_i) \mid 1 \le i \le m\}$
$\mathcal{P}_k(\mathcal{N}_k)$	training set's positive (negative) instances set according to label l_k $\{\boldsymbol{x}_i \mid (\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}, l_k \in \boldsymbol{y}_i\}(\{\boldsymbol{x}_i \mid (\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}, l_k \notin \boldsymbol{y}_i\})$
$\boldsymbol{p}_k(\boldsymbol{n}_k)$	partition center for $\mathcal{P}_k(\mathcal{N}_k)$
$\lvert \cdot \rvert$	$\lvert\mathcal{A}\rvert$ returns the cardinality of set $\mathcal{A}$
$\lceil \cdot \rceil$	$\lceil a \rceil$ returns the retained integer of number a
$\mathfrak{L}$	binary learner for classifier $\mathfrak{c}$ induction
$\phi_k(\cdot)$	$\phi_k(\boldsymbol{x})$ returns the reconstructed label specific features for label l_k and instance $\boldsymbol{x}$ which is the concatenation of distance mapping features $\phi_k^{'}$ and linear representation features $\phi_k^{''}$
β	a ratio parameter controlling the number of retained clusters $\beta \in [0,1]$
$\phi_k^{\beta}(\cdot)$	a proper feature mapping according to label l_k and one β value
$\mathcal{P}_k^{\beta}(\mathcal{N}_k^{\beta})$	the reconstructed training set's positive (negative) instances set according to label l_k, one β value and $\phi_k^{\beta}(\cdot)$ $\{\phi_k^{\beta}(\boldsymbol{x}_i) \mid (\phi_k^{\beta}(\boldsymbol{x}_i), \boldsymbol{y}_i) \in \mathcal{D}, l_k \in \boldsymbol{y}_i\}(\{\phi_k^{\beta}(\boldsymbol{x}_i) \mid (\phi_k^{\beta}(\boldsymbol{x}_i), \boldsymbol{y}_i) \in \mathcal{D}, l_k \notin \boldsymbol{y}_i\})$
$\boldsymbol{m}_{pk}^{\beta}(\boldsymbol{m}_{nk}^{\beta})$	the feature vectors' mean value of $\mathcal{P}_k^{\beta}(\mathcal{N}_k^{\beta})$
$s_{pk}^{\beta}(s_{nk}^{\beta})$	the scatters of $\mathcal{P}_k^{\beta}(\mathcal{N}_k^{\beta})$
$\mathcal{B}_k$	a new binary training set reconstructed from the original multi-label training set $\mathcal{D}$ for label l_k
$rank_f(\cdot,\cdot)$	$rank_f(\boldsymbol{x}, l)$ returns the rank of l in $\mathcal{Y}$ based on the descending order induced from $f(\boldsymbol{x}, \cdot)$

transformation approaches and algorithm adaptation approaches. On the other hand, according to the order of label correlations, the could be roughly categorized into three major classes [43], namely, first-order, second-order and high-order approaches.

Problem transformation approaches tackle multi-label learning problems by turning them into one or more single-label learning problems that are solved with a single-label learning algorithm. Thus, many conventional single-label algorithms can be employed in this class, such as support vector machines, k-nearest neighbor and decision trees [9], etc. For first-order, the binary relevance (BR) approach [2], which decomposes a multi-label learning problem into q independent binary classification problems, is simple and effective, but it ignores label correlation. For second-order or high-order, the label powerset (LP) approach [33] considers each unique set of labels as a new label, then treats them as a multi-class problem. Compared with BR, LP can utilize the label correlations in the training data. It is possible to generate a larger number of new labels with limited training examples for them. Also, it cannot predict unseen label sets. Some approaches have been proposed to overcome these problems, such as random k labelsets (RAkEL) [34] and multi-label classification using ensembles of pruned sets (EPS) [27]. There is also an approach which is called calibrated label ranking (CLR) [12]. CLR introduces an artificial calibration label that, in each example, to separate the relevant from the irrelevant labels. The major merit of problem transformation approaches lies in their operational flexibility which combines existing single-label algorithms and is conceptually simple to boost the algorithm design, while the effectiveness of these approaches might be suboptimal.

Algorithm adaptation approaches tackle multi-label learning directly by adapting single-label algorithms to multi-label cases. The process of training classifiers and predicting an unseen instance in this kind of algorithms is similar to traditional single-label algorithms. The major merit of algorithm adaptation approaches is that they can utilize the characteristics of a multi-label learning problem in a more concise and elegant way. Especially, these approaches exploit second-order (pairwise) or high-order label correlations. For second-order approaches, they can utilize the ranking criterion, such as the multi-label pairwise perceptron (MLPP) [23], or the co-occurrence patterns, such as the two stage voting architecture (TSVA) [24]. For high order approaches, they can impose all other class labels' influences on each label or part of class labels (label subsets), e.g., utilizing hypothesis of linear combination: instance-based learning and logistic regression (IBLR-ML and IBLR-ML+) [8], nonlinear mapping: dependent binary relevance (DBR) [25], shared subspace [20], randomly selecting the label subsets [21], and utilizing graph structure to determine the specific label subsets: conditional dependency networks (CDN) [13]. Obviously, algorithm adaptation approaches could take the advantage of strong label correlations to a certain extent, while suffering high computational complexities.

A common property of existing approaches is that they handle multi-label learning problem mainly by focusing on the perspective of output space. However, new type of approaches have been proposed which exploit label-specific features are exploited to benefit the discrimination of different class labels. The first and famous one is LIFT [41], [42]. Others are derived from or inspired by LIFT, e.g., utilizing discriminative features for each Label (ML-DFL) [40], fuzzy rough set (FRS-LIFT) [39], meta-label-specific features (MLSF) [32], label-specific features with class-dependent labels in a sparse stacking way (LLSF-DL) [17], selected label-dependent features (SLEF) [26], label-specific features and local pairwise label correlation (LF-LPLC) [38] and performing joint feature selection and classification (JFSC) [18]. In the next section, we will present the LSDM algorithm which handles multi-label data by reconstructing feature space via label specific discriminant mapping features.

3 THE LSDM ALGORITHM

Given a training set $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) | 1 \leq i \leq m\}$ with m multi-label training examples, where $\boldsymbol{x}_i \in \mathcal{X}$ is a d-dimensional feature vector and $\boldsymbol{y}_i \subseteq \mathcal{Y}$ is the set of relevant labels associated with $\boldsymbol{x}_i$. Then, LSDM learns from $\mathcal{D}$ by taking five elementary detailed steps, i.e. label specific information extraction, distance mapping feature construction, linear representation feature construction, simplified linear discriminant analysis of reconstructed feature space and classification model generation.

3.1 Label Specific Information Extraction

The information that could effectively capture the specific characteristics of each label is firstly extracted, so as to facilitate its discrimination process. It refers to the information from inherent properties of the training set with respect to each class label. More specifically, for one class label $l_k \in \mathcal{Y}$, we divide the training set into two parts: positive instances set $\mathcal{P}_k$ as well as negative instances set $\mathcal{N}_k$, which correspond to:

$$\begin{aligned} \mathcal{P}_k &= \{\boldsymbol{x}_i | (\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}, l_k \in \boldsymbol{y}_i\} \\ \mathcal{N}_k &= \{\boldsymbol{x}_i | (\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}, l_k \notin \boldsymbol{y}_i\} \end{aligned} \tag{1}$$

Intuitively, $\mathcal{P}_k$ and $\mathcal{N}_k$, defined as label specificity for each label, consist of training instances with and without label l_k respectively.

To extract label specific information from $\mathcal{P}_k$ and $\mathcal{N}_k$, LSDM chooses to employ partitions of $\mathcal{P}_k$ and $\mathcal{N}_k$, respectively, as the foundation of reconstructed feature space. Therefore, suppose $\mathcal{P}_k$ is partitioned into m_k^+ disjoint partitions whose centers are denoted as $\mathcal{C}_k^p = \{\boldsymbol{p}_k^1, \boldsymbol{p}_k^2, \ldots, \boldsymbol{p}_k^{m_k^+}\}$ ($\mathcal{C}_k^p \in \mathbb{R}^{d \times m_k^+}, \boldsymbol{p}_k \in \mathbb{R}^d$). Similarly, $\mathcal{N}_k$ is also partitioned into m_k^- disjoint partitions whose centers are denoted as $\mathcal{C}_k^n = \{\boldsymbol{n}_k^1, \boldsymbol{n}_k^2, \ldots, \boldsymbol{n}_k^{m_k^-}\}$ ($\mathcal{C}_k^n \in \mathbb{R}^{d \times m_k^-}, \boldsymbol{n}_k \in \mathbb{R}^d$). To determine the appropriate partitions, we consider to optimize the reconstruction error, respectively, defined as:

$$\begin{aligned} \text{minimize} \quad & \sum_{ip=1}^{m_k^p} \|\mathcal{C}_k^p \boldsymbol{s}_{ip}^p - \boldsymbol{x}_{ip}^p\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{s}_{ip}^p\|_{0,1} = 1, \forall ip = 1, \ldots, m_k^p \end{aligned}$$

and

$$\begin{aligned} \text{minimize} \quad & \sum_{in=1}^{m_k^n} \|\mathcal{C}_k^n \boldsymbol{s}_{in}^n - \boldsymbol{x}_{in}^n\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{s}_{in}^n\|_{0,1} = 1, \forall in = 1, \ldots, m_k^n \end{aligned}$$

where $\boldsymbol{s}_{ip}^p \in \mathbb{R}^{m_k^+}$, $\boldsymbol{s}_{in}^n \in \mathbb{R}^{m_k^-}$, $\boldsymbol{x}_{ip}^p \in \mathcal{P}_k$, $\boldsymbol{x}_{in}^n \in \mathcal{N}_k$, $m_k^p = |\mathcal{P}_k|$ and $m_k^n = |\mathcal{N}_k|$ are the number of positive and negative instances for each class label respectively. $|\cdot|$ returns the set cardinality.

However, to determine the centers of partitions, it is hard to be optimized due to the condition $\|\boldsymbol{s}\|_{0,1} = 1$ $(0 - norm, 1 - norm)$. As a compromise, the popular k-means clustering algorithm is employed to handle this [19]. Although it might be suboptimal due to the centers of initialization and the number of iterations, it is effective and simple. Multi-label learning tasks usually suffer from the issue of class-imbalance [45], where the number of positive instances for each class label is much smaller than the number of negative ones, i.e., $m_k^p \ll m_k^n$. To alleviate the potential risks brought by the class-imbalance problem, LSDM sets equivalent number of clusters for $\mathcal{P}_k$ and $\mathcal{N}_k$, i.e., $m_k^+ = m_k^- = m_k$. In this way, clustering information gained from positive instances as well as negative instances are treated with equal importance.

Specifically, the number of clusters retained for $\mathcal{P}_k$ and $\mathcal{N}_k$ is set as follow:

$$m_k = \lceil \beta \cdot min(m_k^p, m_k^n) \rceil \tag{2}$$

where $\lceil \cdot \rceil$ denotes the retained integer and $\beta \in [0, 1]$ is a ratio parameter controlling the number of retained clusters. In this paper, we set several values for β simultaneously, which is different from LIFT.

3.2 Distance Mapping

To exploit the label specific information represented by the cluster centers with regard to each label, distance mapping features as in LIFT are constructed by using distances between instances and the cluster centers as feature representation. Intuitively, the cluster centers generated by the k-means algorithm characterize the underlying structure of the original feature space with regard to l_k, which can be served as appropriate building blocks (prototypes) for the construction of label specific features. Here, a mapping $\phi_k^{'} : \mathcal{X} \rightarrow \mathcal{Z}_k^{'}$ from the original d-dimensional input space $\mathcal{X}$ to the $2m_k$-dimensional distance mapping feature space is created as follow:

$$\phi_k^{'\mathrm{T}}(\boldsymbol{x}) = [d_{p_k}^1, \ldots, d_{p_k}^{m_k}, d_{n_k}^1, \ldots, d_{n_k}^{m_k}] \tag{3}$$

where

$$d_{p_k}^{id} = \|\boldsymbol{x} - \boldsymbol{p}_k^{id}\|_2 \text{ and } d_{n_k}^{id} = \|\boldsymbol{x} - \boldsymbol{n}_k^{id}\|_2 \quad (1 \leq id \leq m_k)$$

Here, we employ the Euclidean distance $(2 - norm)$ as the metric in this paper.

3.3 Linear Representation

We consider that the distance mapping features are not good enough to capture all the label specific information, so we employ linear representation features which are using the weights derived from the instances linearly represented by the cluster centers with regard to l_k to further exploit such information, . The first subset of features mainly contain far and near distance information between instances and the cluster centers, while the second subset of features model the spatial topological information between them. The combination of these two subsets of features can well utilize the label specific information. Specifically, each instance can be represented as the linear weighted sum of all the positive and negative cluster centers. Here, a mapping $\phi_k^{''} : \mathcal{X} \rightarrow \mathcal{Z}_k^{''}$ from the original d-dimensional input space $\mathcal{X}$ to the $2m_k$-dimensional linear representation feature space is created as follows:

$$\phi_k^{''\mathrm{T}}(\boldsymbol{x}) = [w_k^1, w_k^2, \ldots, w_k^{2m_k}] \tag{4}$$

where w_k^j $(1 \leq j \leq 2m_k)$ is the reconstructed weight for each of the positive and negative cluster centers of each class label and $\boldsymbol{w}_k^{\mathrm{T}} = \{w_k^1, w_k^2, \ldots, w_k^{2m_k}\}$ $(\boldsymbol{w}_k \in \mathbb{R}^{2m_k})$. Further for representation consistency, we constrain $\sum_{j=1}^{2m_k} w_k^j = 1$.

Accordingly, the problem above can be defined as a solution problem as follows:

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^{m} \|\boldsymbol{x}_i - \sum_{j=1}^{2m_k} w_{k_i}^j \mathcal{C}_k^j\|_2^2 \\ &\text{subject to} && \sum_{j}^{2m_k} w_{k_i}^j = 1, \forall j = 1, \ldots, 2m_k \end{aligned} \tag{5}$$

where $\mathcal{C}_k = [\mathcal{C}_k^p, \mathcal{C}_k^n] = \{\boldsymbol{p}_k^1, \ldots, \boldsymbol{p}_k^{m_k}, \boldsymbol{n}_k^1, \ldots, \boldsymbol{n}_k^{m_k}\}$ $(\mathcal{C}_k \in \mathbb{R}^{d \times 2m_k})$, and $\mathcal{C}_k^j$ is the j-th column of $\mathcal{C}_k$ for l_k.

To compute $\boldsymbol{w}_{ki}$, whose computational process is similar to the computation of local linear embedding [29], we can solve a linear equation as follow:

$$\mathcal{CM}_{ki} \cdot \boldsymbol{w}^{'}{}_{ki} = \mathbf{1}$$

where $\boldsymbol{w}^{'}{}_{ki} \in \mathbb{R}^{2m_k}$ and $\mathbf{1}^{\mathrm{T}} = \{1, \ldots, 1\}$ $(\mathbf{1} \in \mathbb{R}^d)$. $\mathcal{CM}_{ki}$ is the covariance matrix about $\boldsymbol{x}_i$ and $\mathcal{C}_k$ for each class label l_k

$$\mathcal{CM}_{ki} = \begin{pmatrix} (\boldsymbol{x}_i - \mathcal{C}_k^1)^{\mathrm{T}}(\boldsymbol{x}_i - \mathcal{C}_k^1) & \cdots & (\boldsymbol{x}_i - \mathcal{C}_k^1)^{\mathrm{T}}(\boldsymbol{x}_i - \mathcal{C}_k^{2m_k}) \\ \vdots & \ddots & \vdots \\ (\boldsymbol{x}_i - \mathcal{C}_k^{2m_k})^{\mathrm{T}}(\boldsymbol{x}_i - \mathcal{C}_k^1) & \cdots & (\boldsymbol{x}_i - \mathcal{C}_k^{2m_k})^{\mathrm{T}}(\boldsymbol{x}_i - \mathcal{C}_k^{2m_k}) \end{pmatrix}$$

To compute the $\boldsymbol{w}^{'}{}_{ki}$ and avoid $\mathcal{CM}_{ki}$ is not invertible, we should use the regularization form $\mathcal{CM}^{'}{}_{ki}$, then

$$\boldsymbol{w}^{'}{}_{ki} = \mathcal{CM}^{'}{}_{ki}^{-1} \cdot \mathbf{1}$$

At last, we normalize the form $\boldsymbol{w}^{'}{}_{ki}$ to get the reconstructed weights $\boldsymbol{w}_{ki}$ according to Eq. (6).

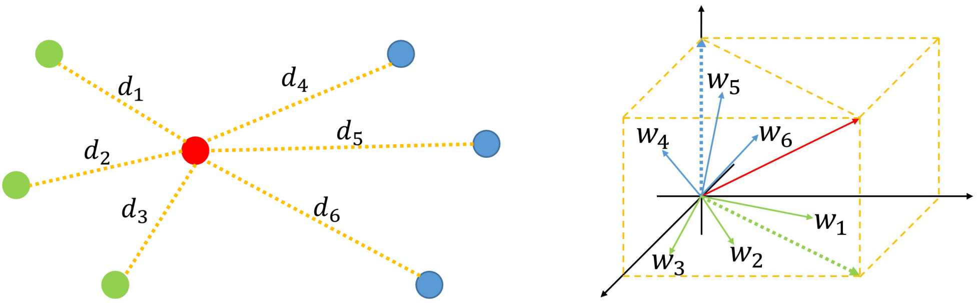
$$\boldsymbol{w}_{ki} = \frac{\boldsymbol{w}^{'}{}_{ki}}{\sum_{j=1}^{2m_k} w^{'}{}_{ki}^{j}} \tag{6}$$

To better understand the concepts of the second and the third steps, Fig. 2 illustrates the schematics of distance mapping and linear representation.

3.4 Simplified Linear Discriminant Analysis

For l_k and β, a mapping $\phi_k : \mathcal{X} \rightarrow \mathcal{Z}_k$ from the original d-dimensional input space $\mathcal{X}$ to the $4m_k$-dimensional reconstructed feature space is created as follow:

$$\phi_k^{\mathrm{T}}(\boldsymbol{x}) = [\phi_k^{'\mathrm{T}}(\boldsymbol{x}), \phi_k^{''\mathrm{T}}(\boldsymbol{x})] \tag{7}$$



(a) *Distance Mapping* (b) *Linear Representation*

Fig. 2. Schematics of (a) distance mapping and (b) linear representation. Red point (arrow) represents original instance, green and blue points (arrows) represent the cluster centers from positive and negative instances respectively.

where $\phi_k(\boldsymbol{x})$ is the label specific features for label l_k and instance $\boldsymbol{x}$ which is the coalition of distance mapping and linear representation features.

Specifically, about parameter sensitivity, for each identical class label l_k, each β value corresponds to proper feature mapping ϕ_k^β. Hence, for one class label $l_k \in \mathcal{Y}$ and β, we reconstruct the training set into two parts: positive instances set $\mathcal{P}_k^\beta$ as well as negative instances set $\mathcal{N}_k^\beta$, which correspond to:

$$\mathcal{P}_k^\beta = \{\phi_k^\beta(\boldsymbol{x}_i) | (\phi_k^\beta(\boldsymbol{x}_i), \boldsymbol{y}_i) \in \mathcal{D}, l_k \in \boldsymbol{y}_i\}$$
$$\mathcal{N}_k^\beta = \{\phi_k^\beta(\boldsymbol{x}_i) | (\phi_k^\beta(\boldsymbol{x}_i), \boldsymbol{y}_i) \in \mathcal{D}, l_k \notin \boldsymbol{y}_i\}$$

To excavate an appropriate ϕ_k well predicting unseen instances, one intuitive strategy is to exploit diverse β values to generate multiple mappings, and then train classifiers with regard to each ϕ_k^β and test them to discriminate optimal ϕ_k. However, when the number of ratio value and size of dataset are large, this strategy faces the dilemma that training too many classifiers increases computational complexity and time also.

We assume that if a reconstructed feature space for label l_k is good, the distances for instances within class are small and those between classes are large. Here, class means positive or negative instances for label l_k. Due to this assumption, we propose a simplified linear discriminant analysis (sLDA), which only computes two mean values and scatters of positive and negative training instances based on reconstructed feature space for each label without mapping to 1-dimension space (dimension reduction), in order to deal with it efficiently as a compromise. Simplified linear discriminant analysis is employed to achieve highly efficient excavation of optimal feature space from different reconstructed feature spaces of identical class label. Intuitively, a mapping which can separate the instances belonging to l_k or not as well as possible is good. Specifically, $\boldsymbol{m}_{pk}^\beta$ and $\boldsymbol{m}_{nk}^\beta$ are the feature vectors' mean value of $\mathcal{P}_k^\beta$ and $\mathcal{N}_k^\beta$ respectively, denoted as follows:

$$\begin{aligned} \boldsymbol{m}_{pk}^\beta &= \frac{\sum_{i=1}^m \phi_k^\beta(\boldsymbol{x}_i)\delta_i^\beta}{\sum_{i=1}^m \delta_i^\beta} \\ \boldsymbol{m}_{nk}^\beta &= \frac{\sum_{i=1}^m \phi_k^\beta(\boldsymbol{x}_i)(1-\delta_i^\beta)}{\sum_{i=1}^m (1-\delta_i^\beta)} \end{aligned} \tag{8}$$

Algorithm 1 The LSDM Algorithm

```
Inputs:
    $\mathcal{D}$: multi-label training set $\{(\boldsymbol{x}_i, \boldsymbol{y}_i) | 1 \leq i \leq m\}$
      $(\boldsymbol{x}_i \in \mathcal{X}, \boldsymbol{y}_i \subseteq \mathcal{Y}, \mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{l_1, l_2, \ldots, l_q\})$
    $\beta$: ratio parameter as used in Eq. (2)
    $\mathfrak{L}$: binary learner for classifier induction
    $\boldsymbol{u}$: unseen instance $(\boldsymbol{u} \in \mathcal{X})$
Outputs:
    $\boldsymbol{y}$: predicted label set for $\boldsymbol{u}$ $(\boldsymbol{y} \subseteq \mathcal{Y})$

for $k = 1$ to $q$ do
    Form $\mathcal{P}_k$ and $\mathcal{N}_k$ based on $\mathcal{D}$ according to Eq. (1)
    for diverse $\beta$ do
        Perform $k$-means clustering on $\mathcal{P}_k$ and $\mathcal{N}_k$, each with $m_k$ clusters as defined in Eq. (2)
        Create the mapping $\phi_k^{'}$ for $l_k$ according to Eq. (3)
        Create the mapping $\phi_k^{''}$ for $l_k$ according to Eq. (4)
        Generate the mapping $\phi_k$ for $l_k$ according to Eq. (7)
        Compute $\boldsymbol{m}_{pk}^{\beta}$ and $\boldsymbol{m}_{nk}^{\beta}$ according to Eq. (8)
        Compute $s_{pk}^{\beta}$ and $s_{nk}^{\beta}$ according to Eq. (9)
        Compute $J(\phi_k^{\beta})$ according to Eq. (11)
    end for
    Excavate the optimal $\phi_k$ due to the largest $J(\phi_k^{\beta})$
end for
for $k = 1$ to $q$ do
    Form $\mathcal{B}_k$ according to Eq. (12)
    Induce $\mathfrak{c}_k$ by invoking $\mathfrak{L}$ on $\mathcal{B}_k$, i.e. $\mathfrak{c}_k \leftarrow \mathfrak{L}(\mathcal{B}_k)$
end for
Return $\boldsymbol{y}$ according to Eq. (13)
```

where $\delta_i^{\beta} = 1$ with regard to $\phi_k^{\beta}(\boldsymbol{x}_i) \in \mathcal{P}_k^{\beta}$ and $\delta_i^{\beta} = 0$ with regard to $\phi_k^{\beta}(\boldsymbol{x}_i) \in \mathcal{N}_k^{\beta}$. And the scatters, indicated as s_{pk}^{β} and s_{nk}^{β}, of $\mathcal{P}_k^{\beta}$ and $\mathcal{N}_k^{\beta}$ are defined as follows:

$$\begin{aligned} s_{pk}^{\beta} &= \sum_{i=1}^{m} \|\phi_k^{\beta}(\boldsymbol{x}_i) - \boldsymbol{m}_{pk}^{\beta}\|_2^2 \delta_i^{\beta} \\ s_{nk}^{\beta} &= \sum_{i=1}^{m} \|\phi_k^{\beta}(\boldsymbol{x}_i) - \boldsymbol{m}_{nk}^{\beta}\|_2^2 (1 - \delta_i^{\beta}) \end{aligned} \tag{9}$$

To alleviate the potential risks brought by the class-imbalance problem, we average the scatters as follows:

$$\begin{aligned} \overline{s_{pk}^{\beta}} &= \frac{\sum_{i=1}^{m} \|\phi_k^{\beta}(\boldsymbol{x}_i) - \boldsymbol{m}_{pk}^{\beta}\|_2^2 \delta_i^{\beta}}{\sum_{i=1}^{m} \delta_i^{\beta}} \\ \overline{s_{nk}^{\beta}} &= \frac{\sum_{i=1}^{m} \|\phi_k^{\beta}(\boldsymbol{x}_i) - \boldsymbol{m}_{nk}^{\beta}\|_2^2 (1 - \delta_i^{\beta})}{\sum_{i=1}^{m} (1 - \delta_i^{\beta})} \end{aligned} \tag{10}$$

Table 2. Characteristics of Data Sets

Data set	$\lvert\mathcal{S}\rvert$	$dim(\mathcal{S})$	$L(\mathcal{S})$	$F(\mathcal{S})$	$LCard(\mathcal{S})$	$LDen(\mathcal{S})$	$DL(\mathcal{S})$	$PDL(\mathcal{S})$	Domain	URL*
CAL500	502	68	174	numeric	26.044	0.150	502	1.000	music	URL 1
emotions	593	72	6	numeric	1.869	0.311	27	0.046	music	URL 1
genbase	662	1185	27	nominal	1.252	0.046	32	0.048	biology	URL 1
medical	978	1449	45	nominal	1.245	0.028	94	0.096	text	URL 2
language log	1460	1004	75	nominal	1.180	0.016	286	0.196	text	URL 2
enron	1702	1001	53	nominal	3.378	0.064	753	0.442	text	URL 2
image	2000	294	5	numeric	1.236	0.247	20	0.010	images	URL 3
scene	2407	294	6	numeric	1.074	0.179	15	0.006	images	URL 1
yeast	2417	103	14	numeric	4.237	0.303	198	0.082	biology	URL 3
slashdot	3782	1079	22	nominal	1.181	0.054	156	0.041	text	URL 2
corel5k	5000	499	374	nominal	3.522	0.009	3175	0.635	images	URL 1
rcv1(subset1)	6000	944	101	numeric	2.880	0.029	1028	0.171	text	URL 1
rcv1(subset2)	6000	944	101	numeric	2.634	0.026	954	0.159	text	URL 1
bibtex	7395	1836	159	nominal	2.402	0.015	2856	0.386	text	URL 1
corel16k(sample1)	13766	500	153	nominal	2.859	0.019	4803	0.349	images	URL 1
corel16k(sample2)	13761	500	164	nominal	2.882	0.018	4868	0.354	images	URL 1
eurlex(suject matter)	19348	5000	201	numeric	2.213	0.011	2504	0.129	text	URL 1
eurlex(directory code)	19348	5000	412	numeric	1.292	0.003	1615	0.084	text	URL 1
tmc2007	28596	981	22	nominal	2.158	0.098	1341	0.047	text	URL 1
mediamill	43907	120	101	numeric	4.376	0.043	6555	0.149	video	URL 1

* URL 1: *http://mulan.sourceforge.net/datasets.html*
URL 2: *http://meka.sourceforge.net/#datasets*
URL 3: *http://cse.seu.edu.cn/people/zhangml/Resources.htm#data*

After mapping, for well separating the instances belonging to l_k or not with regard to each β, we hope the distance between $\boldsymbol{m}_{pk}^{\beta}$ and $\boldsymbol{m}_{nk}^{\beta}$ is as far as possible and the scatters of $\mathcal{P}_k^{\beta}$ and $\mathcal{N}_k^{\beta}$ are as small as possible. Hence, we employ sLDA to denote this as follows:

$$J(\phi_k^{\beta}) = \frac{\|\boldsymbol{m}_{pk}^{\beta} - \boldsymbol{m}_{nk}^{\beta}\|_2^2}{\overline{s_{pk}^{\beta}} + \overline{s_{nk}^{\beta}}} \tag{11}$$

where $J(\phi_k^{\beta})$ denotes the performance of each β with regard to l_k. We excavate the optimal ϕ_k due to the largest $J(\phi_k^{\beta})$.

3.5 Classification Model Generation

A family of q classifiers $\{\mathfrak{c}_1, \mathfrak{c}_2, \ldots, \mathfrak{c}_q\}$ are induced with the generated label specific discriminant mapping features. Here, for each class label $l_k \in \mathcal{Y}$, a new binary training set $\mathcal{B}_k$ with m examples is reconstructed from the original multi-label training set $\mathcal{D}$ and the optimal mapping ϕ_k with regard to largest $J(\phi_k^{\beta})$ as follows:

$$\mathcal{B}_k = \{(\phi_k(\boldsymbol{x}_i), \boldsymbol{y}_i(k) | (\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D})\} \tag{12}$$

Here, $\boldsymbol{y}_i(k) = +1$ if $l_k \in \boldsymbol{y}_i$; otherwise, $\boldsymbol{y}_i(k) = -1$. Based on $\mathcal{B}_k$ any binary learner $\mathfrak{L}$ can be applied to induce a classifier $\mathfrak{c}_k : \mathcal{Z}_k \to \mathbb{R}$ for l_k. The detailed discussion about the binary learner is described in Experiments part.

Give an unseen instance $\boldsymbol{u} \in \mathcal{X}$, its associated label set is predicted as

$$\boldsymbol{y} = \{l_k | \mathfrak{c}_k(\phi_k(\boldsymbol{u})) > 0, 1 \leq k \leq q\} \tag{13}$$

In other words, classification model f_k corresponding to each label l_k can be viewed as the composition of $\mathfrak{c}_k$ and ϕ_k, i.e. $f_k(u) = [\mathfrak{c}_k \circ \phi_k](\boldsymbol{u}) = \mathfrak{c}_k(\phi_k(\boldsymbol{u}))$. Algorithm 1 illustrates the procedure of LSDM in detail.

4 EXPERIMENTS

4.1 Datasets

For each dataset, $|S|$, $dim(S)$, $L(S)$ and $F(S)$ denote the number of examples, number of features, number of possible class labels, and feature type for S respectively. In addition, several other multi-label properties [35], [28] are further used, including label cardinality $LCard(S)$ which measures the average number of labels per example, label density $LDen(S)$ which normalizes $LCard(S)$ by the number of possible labels, distinct label sets $DL(S)$ which counts the number of distinct label combinations in S, proportion of distinct label sets $PDL(S)$ which normalizes $DL(S)$ by the number of example. They are denoted as:

$$\begin{aligned} LCard(S) &= \frac{1}{p}\sum_{i=1}^{p}|\boldsymbol{y}_i| \\ LDen(S) &= \frac{LCard(S)}{L(S)} \\ DL(S) &= |\{Y|(x,Y)\in S\}| \\ PDL(S) &= \frac{DL(S)}{|S|} \end{aligned}$$

In summary, detailed characteristics of all multi-label datasets used in the experiments are demonstrated in Table 2. Roughly ordered by $|S|$, ten regular-scale datasets (first part, $|S| < 5000$) as well as ten large-scale datasets (second part, $|S| \geq 5000$) are included. Furthermore, dimensionality reduction is performed on three text datasets with huge number of features ($dim(S) > 47000$), including *rcv1(subset 1)*, *rcv1(subset 2)*, and *tmc2007*. Specifically, the top 2% features with the highest document frequency are retained. These datasets cover a broad range of cases with diversified multi-label properties. Therefore, experimental studies reported in this paper are quite comprehensive to provide a solid basis for thorough evaluation of LSDM's effectiveness.

4.2 Evaluation Measures

In the multi-label learning community, it is well known that the performance evaluation of multi-label learning differs from that of classical single-label learning because each example could have multiple labels simultaneously. Following the notations in Table 1, six standard evaluation measures are introduced for evaluating the performance of our proposed method from multiple aspects, namely, *Average precision, Macro-averaging AUC, Hamming loss, Coverage, One-error, Ranking loss* [35], [45]. They are defined as follows:

- Average precision:

$$avgprec = \frac{1}{t}\sum_{i=1}^{t}\frac{1}{|Y_i|}\sum_{l_k\in Y_i}\frac{|\mathcal{R}(\boldsymbol{x}_i,l_k)|}{rank_f(\boldsymbol{x}_i,l_k)}$$

where

$$\mathcal{R}(\boldsymbol{x}_i,l_k) = \{l_j|rank_f(\boldsymbol{x}_i,l_j), l_j\in Y_i\}$$

Here, $rank_f(\boldsymbol{x}_i,l_k) = \sum_{j=1}^{q}\|f_j(\boldsymbol{x}_i)\geq f_k(\boldsymbol{x}_i)\|$ returns the rank of l_k when all class labels in $\mathcal{Y}$ are sorted in descending order according to $\{f_1(\boldsymbol{x}_i), f_2(\boldsymbol{x}_i), \ldots, f_q(\boldsymbol{x}_i)\}$. Average precision evaluates the average fraction of relevant labels ranked higher than a particular label $l_k \in Y_i$.

Table 3. Predictive Performance of Benchmarking Algorithms (mean ± std. Deviation) on the Ten Regular-Scale Datasets

Comparing	*Average precision*↑									
algorithm	*CAL500*	*emotions*	*genbase*	*medical*	*language log*	*enron*	*image*	*scene*	*yeast*	*slashdot*
LSDM	**0.5044±0.0055**	**0.8259±0.0311**	**0.9985±0.0026**	**0.8836±0.0315**	**0.4205±0.0136**	**0.6994±0.0231**	**0.8336±0.0185**	**0.8905±0.0154**	**0.7711±0.0116**	**0.6972±0.0156**
LIFT	0.5032±0.0060	0.8237±0.0285	**0.9985±0.0027**	0.8781±0.0329	0.4163±0.0130	0.6969±0.0197	0.8248±0.0164	0.8869±0.0171	0.7693±0.0112	0.6957±0.0146
BR	0.4990±0.0045	0.8182±0.0306	0.9983±0.0030	0.8718±0.0372	0.3917±0.0122	0.6670±0.0255	0.7983±0.0169	0.8463±0.0180	0.7596±0.0127	0.6832±0.0162
MLkNN	0.4821±0.0049	0.8009±0.0274	0.9910±0.0055	0.7703±0.0405	0.3045±0.0193	0.6334±0.0202	0.7902±0.0131	0.8669±0.0152	0.7632±0.0171	0.5004±0.0166
CC	0.4181±0.0092	0.7952±0.0262	0.9968±0.0042	0.8597±0.0476	0.3913±0.0135	0.6086±0.0245	0.7036±0.0292	0.8170±0.0193	0.7209±0.0111	0.5275±0.1054
ECC	0.4810±0.0052	0.8213±0.0300	0.9979±0.0041	0.8759±0.0389	0.3971±0.0125	0.6482±0.0222	0.7922±0.0198	0.8564±0.0115	0.7525±0.0122	0.6686±0.0199
Comparing	*Macro-averaging AUC*↑									
algorithm	*CAL500*	*emotions*	*genbase*	*medical*	*language log*	*enron*	*image*	*scene*	*yeast*	*slashdot*
LSDM	0.5188±0.0066	**0.8549±0.0319**	0.8684±0.1123	0.8897±0.0435	**0.4652±0.0331**	**0.5995±0.0347**	**0.8643±0.0156**	**0.9489±0.0090**	**0.6922±0.0107**	0.7524±0.0391
LIFT	0.5159±0.0066	0.8535±0.0339	0.8684±0.1122	0.8880±0.0381	0.4642±0.0340	0.5985±0.0380	0.8597±0.0195	0.9488±0.0094	0.6913±0.0112	**0.7558±0.0397**
BR	0.5136±0.0081	0.8421±0.0296	**0.8692±0.1125**	**0.8955±0.0398**	0.4359±0.0307	0.5672±0.0303	0.8316±0.0195	0.9157±0.0110	0.6437±0.0114	0.7433±0.0434
MLkNN	0.5148±0.0045	0.8443±0.0261	0.8647±0.1099	0.8504±0.0482	0.3557±0.0271	0.5223±0.0299	0.8309±0.0177	0.9337±0.0087	0.6845±0.0152	0.5306±0.0222
CC	0.5088±0.0043	0.8286±0.0279	0.8596±0.1133	0.8453±0.0470	0.4403±0.0307	0.5564±0.0326	0.8065±0.0228	0.9105±0.0137	0.6378±0.0130	0.7371±0.0417
ECC	**0.5236±0.0066**	0.8361±0.0251	0.8656±0.1136	0.8584±0.0422	0.4547±0.0357	0.5693±0.0298	0.8318±0.0181	0.9337±0.0089	0.6700±0.0119	0.7436±0.0436
Comparing	*Hamming loss*↓									
algorithm	*CAL500*	*emotions*	*genbase*	*medical*	*language log*	*enron*	*image*	*scene*	*yeast*	*slashdot*
LSDM	**0.1370±0.0014**	**0.1824±0.0162**	0.0024±0.0015	0.0116±0.0004	**0.0150±0.0015**	**0.0457±0.0023**	**0.1522±0.0105**	**0.0776±0.0065**	**0.1906±0.0060**	**0.0386±0.0012**
LIFT	**0.1370±0.0019**	0.1849±0.0154	0.0024±0.0015	0.0114±0.0009	**0.0150±0.0014**	**0.0457±0.0018**	0.1550±0.0095	0.0782±0.0055	0.1909±0.0060	0.0387±0.0010
BR	0.1378±0.0016	0.1922±0.0153	**0.0005±0.0004**	0.0111±0.0004	0.0174±0.0011	0.0498±0.0030	0.1768±0.0095	0.1038±0.0078	0.1990±0.0050	0.0399±0.0007
MLkNN	0.1402±0.0022	0.1920±0.0241	0.0051±0.0023	0.0178±0.0005	0.0257±0.0011	0.0520±0.0020	0.1706±0.0070	0.0850±0.0073	0.1931±0.0079	0.0519±0.0005
CC	0.1532±0.0081	0.2060±0.0298	**0.0005±0.0008**	0.0124±0.0007	0.0255±0.0011	0.0562±0.0026	0.2520±0.0158	0.1125±0.0086	0.2087±0.0047	0.0781±0.0171
ECC	0.1385±0.0019	0.1874±0.0226	**0.0005±0.0004**	**0.0110±0.0009**	0.0254±0.0011	0.0515±0.0029	0.1783±0.0174	0.0942±0.0064	0.2002±0.0068	0.0413±0.0025
Comparing	*Coverage*↓									
algorithm	*CAL500*	*emotions*	*genbase*	*medical*	*language log*	*enron*	*image*	*scene*	*yeast*	*slashdot*
LSDM	**0.7562±0.0093**	**0.2788±0.0465**	0.0134±0.0007	**0.0418±0.0019**	**0.1608±0.0021**	**0.2223±0.0042**	**0.1637±0.0327**	**0.0637±0.0106**	0.4486±0.0320	0.1057±0.0048
LIFT	0.7654±0.0094	0.2805±0.0467	0.0135±0.0007	0.0434±0.0020	0.1650±0.0022	0.2228±0.0042	0.1684±0.0337	0.0647±0.0108	0.4538±0.0324	**0.1048±0.0048**
BR	0.8512±0.0090	0.2849±0.0475	**0.0129±0.0006**	0.0462±0.0010	0.4419±0.0019	0.2298±0.0043	0.1877±0.0375	0.0888±0.0148	0.4588±0.0328	0.1094±0.0050
MLkNN	0.7607±0.0088	0.2965±0.0494	0.0162±0.0008	0.0704±0.0016	0.3729±0.0023	0.2475±0.0047	0.1952±0.0390	0.0785±0.0131	**0.4456±0.0318**	0.1873±0.0085
CC	0.8514±0.0162	0.2931±0.0489	0.0131±0.0007	0.0524±0.0022	0.4427±0.0019	0.2587±0.0049	0.2509±0.0502	0.1064±0.0177	0.5300±0.0379	0.1453±0.0066
ECC	0.7757±0.0098	0.2789±0.0466	0.0132±0.0007	0.0716±0.0019	0.3395±0.0019	0.2371±0.0045	0.1940±0.0388	0.0816±0.0136	0.4568±0.0326	0.1244±0.0057
Comparing	*One-error*↓									
algorithm	*CAL500*	*emotions*	*genbase*	*medical*	*language log*	*enron*	*image*	*scene*	*yeast*	*slashdot*
LSDM	**0.1175±0.0190**	**0.2310±0.0443**	**0.0000±0.0000**	0.1493±0.0425	**0.6688±0.0157**	**0.2473±0.0288**	**0.2530±0.0347**	**0.1869±0.0266**	0.2226±0.0131	**0.3979±0.0186**
LIFT	0.1195±0.0161	**0.2310±0.0489**	**0.0000±0.0000**	0.1575±0.0425	0.6724±0.0166	0.2479±0.0280	0.2680±0.0323	0.1940±0.0277	0.2226±0.0125	0.4016±0.0159
BR	0.3171±0.0206	0.2377±0.0552	0.0015±0.0047	0.1501±0.0553	0.7009±0.0152	0.2714±0.0445	0.3085±0.0293	0.2551±0.0289	0.2226±0.0122	0.4170±0.0212
MLkNN	0.1323±0.0154	0.2766±0.0470	0.0121±0.0119	0.2978±0.0414	0.8060±0.0146	0.3073±0.0274	0.3205±0.0215	0.2239±0.0302	0.2400±0.0178	0.6386±0.0202
CC	0.3351±0.0302	0.2950±0.0465	0.0045±0.0072	0.1718±0.0410	0.7056±0.0193	0.3702±0.0601	0.4695±0.0523	0.2995±0.0295	0.2644±0.0189	0.6660±0.1878
ECC	0.1371±0.0184	0.2478±0.0535	0.0015±0.0047	**0.1485±0.0328**	0.6983±0.0151	0.2820±0.0431	0.3175±0.0337	0.2426±0.0235	**0.2191±0.0102**	0.4268±0.0257
Comparing	*Ranking loss*↓									
algorithm	*CAL500*	*emotions*	*genbase*	*medical*	*language log*	*enron*	*image*	*scene*	*yeast*	*slashdot*
LSDM	**0.1802±0.0025**	**0.1390±0.0332**	0.0010±0.0021	**0.0260±0.0019**	**0.1440±0.0086**	**0.0756±0.0065**	**0.1365±0.0154**	**0.0596±0.0095**	**0.1628±0.0090**	0.0901±0.0069
LIFT	0.1821±0.0021	0.1412±0.0289	0.0011±0.0022	0.0270±0.0023	0.1485±0.0081	0.0757±0.0053	0.1424±0.0144	0.0611±0.0107	0.1649±0.0093	**0.0897±0.0070**
BR	0.2021±0.0020	0.1453±0.0281	**0.0008±0.0020**	0.0299±0.0041	0.4292±0.0079	0.0816±0.0079	0.1660±0.0157	0.0897±0.0105	0.1715±0.0082	0.0932±0.0067
MLkNN	0.1891±0.0021	0.1599±0.0294	0.0028±0.0039	0.0508±0.0026	0.4638±0.0073	0.0924±0.0055	0.1774±0.0162	0.0769±0.0078	0.1654±0.0096	0.1727±0.0097
CC	0.2305±0.0090	0.1646±0.0263	0.0009±0.0020	0.0436±0.0056	0.4299±0.0068	0.0968±0.0146	0.2458±0.0178	0.1114±0.0154	0.2090±0.0055	0.1300±0.0213
ECC	0.1955±0.0031	0.1415±0.0319	0.0010±0.0022	0.0355±0.0036	0.3266±0.0076	0.0847±0.0072	0.1735±0.0196	0.0807±0.0056	0.1758±0.0080	0.1072±0.0098

- Macro-averaging AUC:

$$AUC_{macro} = \frac{1}{q}\sum_{k=1}^{q} AUC_k$$

$$= \frac{1}{q}\sum_{k=1}^{q} \frac{|\{(\boldsymbol{x}', \boldsymbol{x}'')|f_k(\boldsymbol{x}) \geq f_k(\boldsymbol{x}), (\boldsymbol{x}', \boldsymbol{x}'') \in \mathcal{P}_k \times \mathcal{N}_k\}|}{m_k^p m_k^n}$$

  Here, the AUC value on each class label (i.e. AUC_k) is calculated based on the relationship between AUC and the Wilcoxon-Mann-Whitney statistic [14].

- Hamming loss:

$$hloss = \frac{1}{t}\sum_{i=1}^{t} |h(\boldsymbol{x}_i)\Delta Y_i|$$

  Here, $h(\boldsymbol{x}_i) = \{l_k | f_k(\boldsymbol{x}_i) > 0, 1 \leq k \leq q\}$ corresponds to the predicted set of relevant labels for $\boldsymbol{x}_i$, and Δ stands for the symmetric difference between two sets. Hamming loss evaluates the fraction of instance-label pairs which have been misclassified, i.e. a relevant label is missed or an irrelevant label is predicted.

Table 4. Predictive Performance of Benchmarking Algorithms (mean ± std. Deviation) on the Ten Large-Scale Datasets

Comparing	*Average precision↑*									
algorithm	*corel5k*	*rcv1-s1*	*rcv1-s2*	*bibtex*	*corel16k-s1*	*corel16k-s2*	*eurlex-sm*	*eurlex-dc*	*tmc2007*	*mediamill*
LSDM	**0.2928±0.0025**	**0.5978±0.0056**	**0.6223±0.0060**	**0.5672±0.0088**	**0.3136±0.0007**	**0.3113±0.0025**	**0.5900±0.0189**	**0.5932±0.0105**	**0.7433±0.0035**	**0.7047±0.0016**
LIFT	0.2880±0.0048	0.5918±0.0049	0.6180±0.0047	0.5602±0.0090	0.3083±0.0024	0.3076±0.0020	0.5830±0.0131	0.5842±0.0103	0.7432±0.0010	0.7000±0.0021
BR	0.2789±0.0038	0.5511±0.0035	0.5857±0.0024	0.5391±0.0085	0.2827±0.0052	0.2766±0.0022	0.4158±0.0146	0.3915±0.0135	0.6772±0.0031	0.5089±0.0020
MLkNN	0.2437±0.0038	0.4502±0.0143	0.4772±0.0083	0.3403±0.0079	0.2803±0.0023	0.2727±0.0040	0.4292±0.0118	0.3757±0.0072	0.6147±0.0056	0.6757±0.0018
CC	0.1717±0.0869	0.4553±0.0111	0.4963±0.0470	0.5228±0.0099	0.2492±0.0133	0.2553±0.0072	0.3804±0.0792	0.3918±0.0132	0.6651±0.0089	0.5091±0.0209
ECC	0.2528±0.0048	0.5601±0.0052	0.5965±0.0038	0.5388±0.0079	0.2925±0.0033	0.2883±0.0026	0.3913±0.0120	0.4621±0.0128	0.7071±0.0050	0.6155±0.0177
Comparing	*Macro-averaging AUC↑*									
algorithm	*corel5k*	*rcv1-s1*	*rcv1-s2*	*bibtex*	*corel16k-s1*	*corel16k-s2*	*eurlex-sm*	*eurlex-dc*	*tmc2007*	*mediamill*
LSDM	**0.6063±0.0160**	**0.9065±0.0105**	**0.8962±0.0078**	**0.9093±0.0044**	**0.6981±0.0042**	**0.7209±0.0036**	**0.8068±0.0085**	**0.8164±0.0095**	**0.8978±0.0018**	**0.6687±0.0002**
LIFT	0.6058±0.0168	0.9018±0.0109	0.8937±0.0130	0.9085±0.0036	0.6966±0.0048	0.7116±0.0033	0.7973±0.0090	0.8149±0.0083	0.8974±0.0018	0.6395±0.0002
BR	0.5333±0.0180	0.8732±0.0143	0.8803±0.0065	0.8734±0.0027	0.6527±0.0034	0.6669±0.0083	0.5861±0.0033	0.5643±0.0045	0.8446±0.0027	0.5085±0.0001
MLkNN	0.4629±0.0069	0.6713±0.0079	0.6779±0.0189	0.6687±0.0071	0.5637±0.0027	0.5711±0.0053	0.6450±0.0083	0.5601±0.0062	0.7423±0.0029	0.5097±0.0001
CC	0.5315±0.0114	0.8146±0.0237	0.7921±0.0235	0.8596±0.0016	0.6327±0.0040	0.6413±0.0067	0.5837±0.0046	0.5649±0.0049	0.8317±0.0096	0.5196±0.0002
ECC	0.5517±0.0153	0.8607±0.0145	0.8705±0.0097	0.8690±0.0032	0.6548±0.0041	0.6648±0.0039	0.6292±0.0067	0.5804±0.0051	0.8807±0.0029	0.5237±0.0002
Comparing	*Hamming loss↓*									
algorithm	*corel5k*	*rcv1-s1*	*rcv1-s2*	*bibtex*	*corel16k-s1*	*corel16k-s2*	*eurlex-sm*	*eurlex-dc*	*tmc2007*	*mediamill*
LSDM	**0.0094±0.0000**	**0.0260±0.0002**	**0.0222±0.0009**	**0.0124±0.0001**	0.0189±0.0000	0.0177±0.0000	0.0314±0.0021	**0.0023±0.0009**	0.0680±0.0010	**0.0305±0.0002**
LIFT	0.0095±0.0001	0.0261±0.0002	0.0228±0.0002	0.0125±0.0001	0.0188±0.0000	0.0176±0.0000	0.0318±0.0022	0.0033±0.0009	**0.0675±0.0006**	0.0308±0.0003
BR	0.0123±0.0001	0.0266±0.0002	0.0233±0.0002	0.0125±0.0001	**0.0187±0.0000**	**0.0175±0.0000**	0.0342±0.0013	0.0030±0.0009	0.0771±0.0003	0.0311±0.0003
MLkNN	0.0096±0.0000	0.0276±0.0005	0.0244±0.0002	0.0137±0.0002	0.0188±0.0000	0.0176±0.0000	**0.0299±0.0012**	0.0029±0.0010	0.0807±0.0006	0.0332±0.0003
CC	0.0196±0.0006	0.0314±0.0013	0.0287±0.0031	0.0127±0.0002	0.0203±0.0003	0.0189±0.0001	0.0344±0.0013	0.0030±0.0009	0.0789±0.0004	0.0412±0.0067
ECC	0.0145±0.0001	0.0269±0.0002	0.0240±0.0002	0.0127±0.0001	0.0188±0.0000	0.0177±0.0001	0.0349±0.0010	0.0029±0.0009	0.0763±0.0005	0.0383±0.0011
Comparing	*Coverage↓*									
algorithm	*corel5k*	*rcv1-s1*	*rcv1-s2*	*bibtex*	*corel16k-s1*	*corel16k-s2*	*eurlex-sm*	*eurlex-dc*	*tmc2007*	*mediamill*
LSDM	0.2953±0.0008	**0.1264±0.0088**	**0.1234±0.0033**	0.1402±0.0052	**0.3119±0.0037**	**0.3066±0.0016**	0.2116±0.0126	**0.0805±0.0022**	0.1354±0.0009	0.1890±0.0020
LIFT	0.2955±0.0008	0.1285±0.0086	0.1250±0.0022	**0.1388±0.0059**	0.3280±0.0021	0.3169±0.0037	**0.2105±0.0152**	0.0857±0.0028	**0.1345±0.0011**	0.1953±0.0017
BR	**0.2908±0.0008**	0.1473±0.0135	0.1376±0.0035	0.1585±0.0090	0.3190±0.0021	0.3106±0.0019	0.2269±0.0114	0.1520±0.0025	0.1704±0.0023	0.5696±0.0037
MLkNN	0.3068±0.0008	0.2342±0.0091	0.2270±0.0044	0.3498±0.0082	0.3412±0.0022	0.3342±0.0020	0.2244±0.0080	0.1512±0.0019	0.2314±0.0024	**0.1810±0.0018**
CC	0.4759±0.0013	0.2887±0.0237	0.2410±0.0198	0.1784±0.0111	0.3726±0.0024	0.3625±0.0022	0.2255±0.0111	0.1520±0.0025	0.1824±0.0097	0.4484±0.0044
ECC	0.2969±0.0008	0.1486±0.0153	0.1395±0.0058	0.1622±0.0110	0.3264±0.0021	0.3180±0.0019	0.2287±0.0088	0.1462±0.0024	0.1475±0.0029	0.2394±0.0024
Comparing	*One-error↓*									
algorithm	*corel5k*	*rcv1-s1*	*rcv1-s2*	*bibtex*	*corel16k-s1*	*corel16k-s2*	*eurlex-sm*	*eurlex-dc*	*tmc2007*	*mediamill*
LSDM	**0.6820±0.0077**	0.4185±0.0073	**0.3983±0.0054**	**0.3824±0.0082**	**0.6925±0.0040**	**0.6796±0.0052**	**0.4034±0.0212**	**0.4910±0.0084**	**0.3191±0.0058**	**0.1470±0.0040**
LIFT	0.6874±0.0192	**0.4149±0.0079**	0.4107±0.0026	0.3889±0.0127	0.6973±0.0076	0.6857±0.0081	0.4137±0.0219	0.5020±0.0087	0.3211±0.0029	0.1483±0.0036
BR	0.7700±0.0074	0.4519±0.0080	0.4393±0.0034	0.4045±0.0100	0.7229±0.0129	0.7215±0.0068	0.5556±0.0167	0.6615±0.0068	0.3977±0.0031	0.2426±0.0057
MLkNN	0.7442±0.0059	0.5730±0.0184	0.5452±0.0098	0.6010±0.0153	0.7384±0.0075	0.7473±0.0047	0.5648±0.0146	0.6777±0.0080	0.4456±0.0066	0.1672±0.0038
CC	0.8356±0.1101	0.5471±0.0190	0.5134±0.0789	0.4047±0.0059	0.7829±0.0375	0.7538±0.0119	0.6618±0.2260	0.6611±0.0059	0.4089±0.0025	0.2829±0.1337
ECC	0.7136±0.0092	0.4543±0.0096	0.4269±0.0084	0.3973±0.0082	0.7030±0.0113	0.6970±0.0108	0.5596±0.0162	0.6217±0.0042	0.3722±0.0082	0.2023±0.0387
Comparing	*Ranking loss↓*									
algorithm	*corel5k*	*rcv1-s1*	*rcv1-s2*	*bibtex*	*corel16k-s1*	*corel16k-s2*	*eurlex-sm*	*eurlex-dc*	*tmc2007*	*mediamill*
LSDM	**0.1230±0.0031**	**0.0500±0.0047**	**0.0509±0.0014**	0.0754±0.0020	**0.1570±0.0027**	**0.1538±0.0014**	0.1487±0.0065	**0.1169±0.0012**	0.0690±0.0007	0.0554±0.0006
LIFT	0.1232±0.0039	0.0513±0.0047	0.0518±0.0010	**0.0747±0.0018**	0.1656±0.0037	0.1595±0.0016	**0.1479±0.0119**	0.1242±0.0010	**0.0685±0.0008**	0.0576±0.0005
BR	0.1241±0.0047	0.0633±0.0086	0.0617±0.0016	0.0860±0.0012	0.1632±0.0020	0.1581±0.0015	0.1620±0.0059	0.2200±0.0019	0.0954±0.0011	0.1499±0.0011
MLkNN	0.1346±0.0047	0.1136±0.0052	0.1139±0.0021	0.2149±0.0025	0.1761±0.0018	0.1705±0.0022	0.1577±0.0070	0.2293±0.0016	0.1467±0.0011	**0.0544±0.0008**
CC	0.2435±0.1814	0.1223±0.0094	0.1037±0.0094	0.0931±0.0009	0.1879±0.0037	0.1812±0.0054	0.1602±0.0051	0.2200±0.0019	0.1021±0.0057	0.1509±0.0327
ECC	0.1260±0.0038	0.0606±0.0064	0.0571±0.0021	0.0857±0.0018	0.1645±0.0012	0.1587±0.0016	0.1636±0.0058	0.2048±0.0015	0.0777±0.0020	0.0762±0.0033

- Coverage:

$$coverage = \frac{1}{q}\left(\frac{1}{t}\sum_{i=1}^{t}\max rank_f(\boldsymbol{x}_i, l_k) - 1\right)$$

  Coverage evaluates how many steps are needed, on average, to move down the ranked label list of an example so as to cover all its relevant labels. Furthermore, the coverage measure is normalized by the number of possible class labels (i.e. q) in this paper.

- One-error

$$one - error = \frac{1}{t}\sum_{t=1}^{t}\|[arg \max_{l_k \in \mathcal{Y}} f_k(\boldsymbol{x}_i)] \notin Y_i\|$$

  Here, for any predicate π, $\|\pi\|$ returns 1 if π holds and 0 otherwise. One-error evaluates the fraction of examples whose top-ranked predicted label is not in the ground-truth relevant label set.

- Ranking loss

$$rloss = \frac{1}{t}\sum_{i=1}^{t}\frac{|\{(l_k, l_j)|f_k(\boldsymbol{x}_i) \leq f_j(\boldsymbol{x}_i), (l_k, l_j) \in Y_i \times \overline{Y}_i\}|}{|Y_i||\overline{Y}_i|}$$

Here,$\overline{Y}_i$ is the complementary set of Y_i in $\mathcal{Y}$. Ranking loss evaluates the average fraction of misordered label pairs, i.e. an irrelevant label of an example is ranked higher than its relevant one.

Note that for all the six multi-label evaluation measures, their values vary between $[0, 1]$. Furthermore, for *Average precision* and *Macro-averaging AUC*, the larger the values the better the performance. While for the other four measures, the smaller the values the better the performance. These measures serve as good indicators for comprehensive comparative studies as they evaluate the performance of the learned models from various aspects.

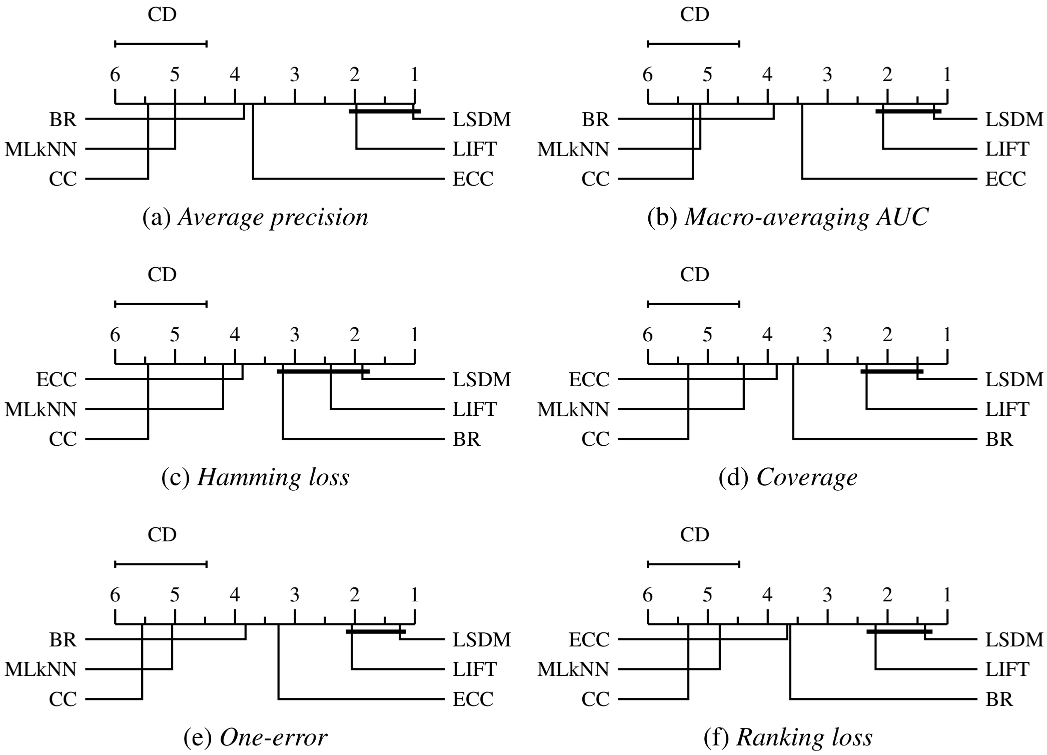
4.3 Multi-label Classifiers

We compare our proposed algorithm with the following five methods: 1) *Label specific features* (LIFT) [41], [42]; 2) *Binary relevance* (BR) [2]; 3) *Multi-label k Nearest Neighbors* (MLkNN) [44]; 4) *Classifier chain* (CC) and 5) *Ensemble of classifier chains* (ECC) [28].

- *Label specific features* (LIFT) [3]: The basic idea of this algorithm is that it firstly utilizes cluster analysis on the positive and negative instances to construct features specific to each label, and then performs training and testing as q independent binary classification problem by querying the clustering results. It could be viewed as a degenerated version of LSDM where the label specific discriminant mapping features $\phi_k(\boldsymbol{x})$ is only kept to the distance mapping features $\phi_k^{'}(\boldsymbol{x})$ and sLDA is not employed to excavate the optimally reconstructed feature space.
- *Binary relevance* (BR): The basic idea of this algorithm is to decompose the multi-label learning problem into q independent binary classification problems, where each binary classification problem corresponds to a possible label in the label space. It could be viewed as a plain version of LSDM where the feature space is kept to the original features $\boldsymbol{x}$.
- *Multi-label k Nearest Neighbors* (MLkNN). The basic idea of this algorithm is adapting k nearest neighbors techniques to deal with multi-label data, where maximum a posteriori (MAP) rule is utilized to make prediction by reasoning with the labeling information embodied in the neighbors.
- *Classifier chain* (CC): The basic idea of this algorithm is to transform the multi-label learning problem into a chain of binary classification problems, where subsequent binary classifiers in the chain are successively built upon the predictions of preceding ones.
- *Ensemble of classifier chains* (ECC): The basic idea of this algorithm is to employ ensemble learning to address chain order randomness as an ensemble of classifier chains. Here, the ensemble size is set to be 50 to accommodate sufficient number of classifier chains.

4.4 Experimental Setup

The parameter of LSDM, i.e. ratio β as used in Eq. (2) is set to be from 0.01 to 0.2 (0.01, 0.05, 0.1, 0.2) in this paper. As we mentioned above our proposed algorithm LSDM can utilize all these β values at the same time and employ sLDA to efficiently excavate the optimal ϕ_k. This is different from LIFT, because it can only utilize one β value at a time. The parameter settings of LIFT, MLkNN and ECC are as suggested in the corresponding literatures. The ratio of LIFT is set to be 0.1. The number of nearest neighbors of MLkNN is set to be 10. The ensemble size and sampling ratio of ECC are set to be 50 and 50% respectively. For fair comparison, LSDM, LIFT, BR, CC and ECC employ LIBSVM (with linear kernel) [6] as the binary learner. For the regular-scale datasets, we apply ten-fold cross validation. However, for the large-scale datasets, on each dataset, 50% examples are randomly sampled to form the training set, and the rest are used to form the test set. And then, the sampling process is repeated for ten times.



(a) *Average precision*

(b) *Macro-averaging AUC*

(c) *Hamming loss*

(d) *Coverage*

(e) *One-error*

(f) *Ranking loss*

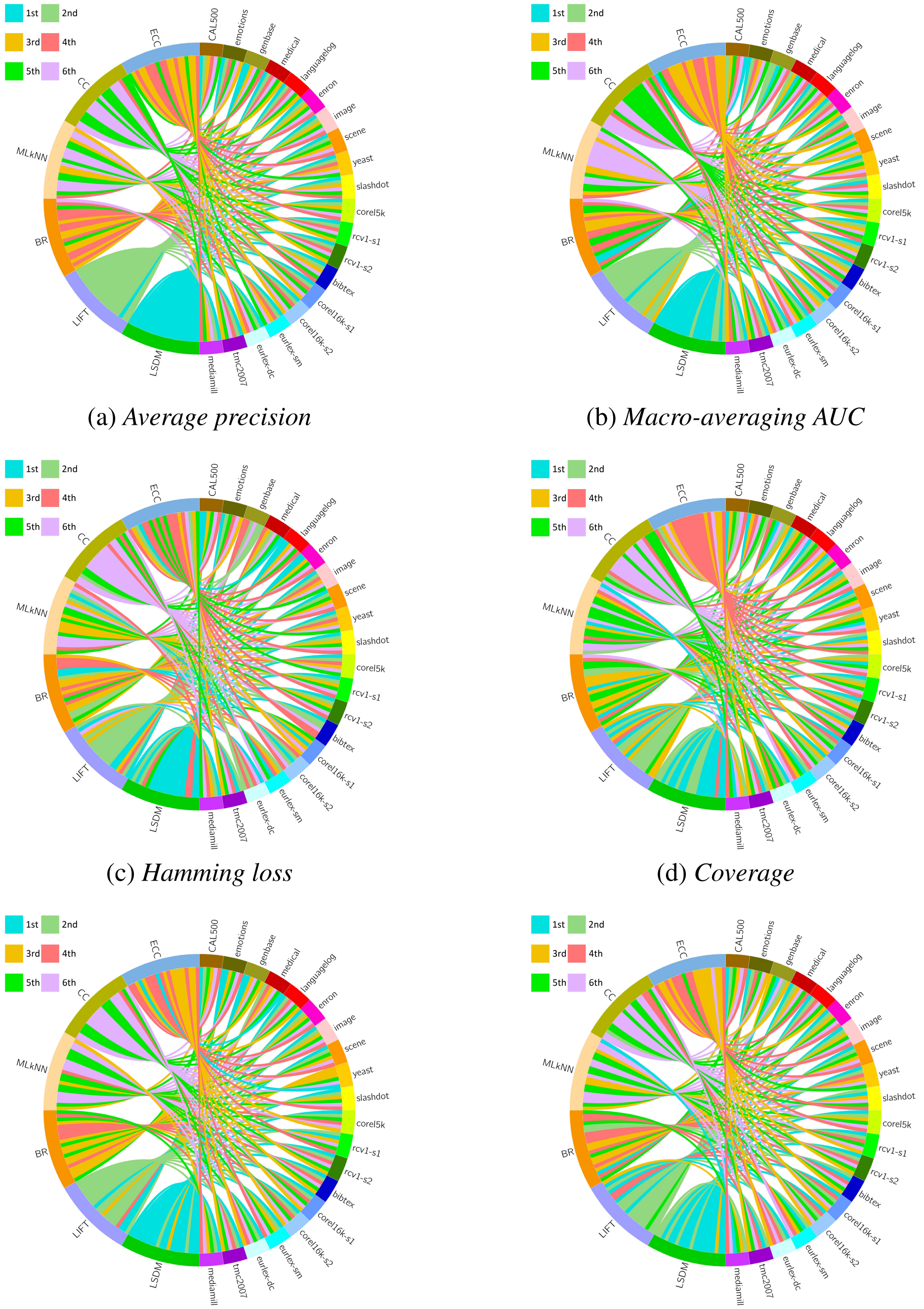
Fig. 3. Comparison of LSDM (control algorithm) against other comparing algorithms with *Bonferroni-Dunn test* under each evaluation measure. Algorithms not connected with LSDM in the CD diagram are considered to have significantly different performance from the control algorithm (significance level $\alpha = 0.05$).

5 RESULTS

Table 3 and Table 4 report the detailed experimental results of all comparing algorithms with six evaluation measures on the regular-scale and large-scale datasets respectively. For each evaluation measure, "↑" indicates "the larger the better" while "↓" indicates "the smaller the better". Furthermore, the boldfaced values represent the best performance among the six comparing algorithms.

To analyze the performance among the comparing algorithms systematically, we employ *Friedman test* [10] which is regarded as the favorable statistical test for comparisons among multiple algorithms over a number of datasets. Given k comparing algorithms and N data sets, let r_i^j denote the rank of the j-th algorithm on the i-th data set (mean ranks are shared in case of ties). Let $R_j = \frac{1}{N}\sum_{i=1}^{N} r_i^j$ denote the average rank for the j-th algorithm, under the null hypothesis (i.e. all algorithms have "equal" performance), the following Friedman statistic F_F will be distributed according to the F-distribution with $k-1$ numerator degrees of freedom and $(k-1)(N-1)$ denominator degrees of freedom:

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}$$



(a) *Average precision*

(b) *Macro-averaging AUC*

(c) *Hamming loss*

(d) *Coverage*

(e) *One-error*

(f) *Ranking loss*

Fig. 4. Comparison of LSDM against other comparing algorithms under each evaluation measure. Each dataset connects all the algorithms with different color curves simultaneously and the size of identical color curve area on the left half circle denotes the rank performance of each algorithm corresponding to all datasets.

Where

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_{j=1}^{k} R_j^2 - \frac{k(k+1)^2}{4}\right]$$

Table 5 summarizes the Friedman statistics F_F and the corresponding critical values on each evaluation measure. At significance level $\alpha = 0.05$, the null hypothesis of "equal" performance among the comparing algorithms is clearly rejected on each evaluation measure. Consequently, *Bonferroni-Dunn test* [11] is employed as the *post-hoc test* [10] to demonstrate the relative performance among the comparing algorithms, where LSDM is regarded as the control algorithm. Here, the average rank difference between LSDM and one comparing algorithm is compared with the *critical difference* (CD):

$$\text{CD} = q_\alpha \sqrt{\frac{k(k+1)}{6N}}$$

For *Bonferroni-Dunn test*, we have $q_\alpha = 2.576$ at significance level $\alpha = 0.05$ and thus $\text{CD} = 1.524$ $(k = 6, N = 20)$. Accordingly, the performance between LSDM and one comparing algorithm is deemed to be significantly different if their average ranks differ by at least one CD. Fig. 3 illustrates the CD diagrams [11] on each evaluation measure, where the average rank of each comparing algorithm is marked along the axis (lower ranks to the right). In each subfigure, any comparing algorithm whose average rank is within one CD to that of LSDM is interconnected to each other with a thick line. Otherwise, it is considered to have significant different performance against LSDM. Furthermore, Fig. 4 illustrates the rank performance of each algorithm corresponding to all datasets on each evaluation measure. In each subfigure, each dataset connects each algorithm with different color curves (performance ranking) simultaneously and the area of different color curves combination for each algorithm on the left half circle denotes its final performance ranking corresponding to all datasets. For example, for subfigure (a) *Average precision*, we can see that LSDM with area of 20 blue curves (1st ranking) combination on the left half circle, which means it ranks first for all datasets. For LIFT, it is with area of 19 celadon curves (2nd ranking) and 1 blue curve (1st ranking). For BR, it is with area of 7 orange curves (3rd ranking), 10 pink curves (4th ranking), 2 green curves (5th ranking) and 1 purple curve (6th ranking). The other three are described in the same way.

Table 5. Summary of the Friedman Statistics F_F in Terms of Each Evaluation Measure and the Critical Value
(# Comparing Algorithms $k = 6$; # Datasets $N = 20$)

Evaluation measure	F_F	critical value($\alpha = 0.05$)
Average precision	98.3357	2.3102
Macro-averaging AUC	56.1412	
Hamming loss	17.4334	
Converage	23.0420	
One-error	74.0395	
Ranking loss	34.3922	

Based on the above experimental results, the following observations can be apparently made:

1) As shown in Fig. 3, the performance in terms of each evaluation measure between LSDM and LIFT is not significantly different due to their average ranks within least one CD, but we can also consider that LSDM achieves comparable performance against LIFT. Because the *Bonferroni-Dunn test* measures the average rank and LIFT is excellent to beat other comparing algorithms, as shown in Fig. 4 LSDM exceeding LIFT at most rank circumstances is superior. As LIFT can be viewed as a degenerated version of LSDM where the label specific discriminant

mapping features $\phi_k(\boldsymbol{x})$ is only kept to the distance mapping features $\phi_k^{'}(\boldsymbol{x})$. The superior performance of LSDM against LIFT clearly verifies the effectiveness of linear representation features and employing sLDA to excavate the optimally reconstructed feature space.

2) As shown in Fig. 3 and 4, LSDM achieves statistically superior or at least comparable performance against BR in terms of each evaluation measure. As BR can be regarded as a plain version of LSDM by keeping the original feature vector unchanged, the superior performance of LSDM against BR clearly verifies the effectiveness of employing label specific discriminant mapping features.
3) Furthermore, LSDM significantly outperforms MLkNN, CC and ECC in terms of each evaluation measure. Note that label transforming to feature learning strategy has been incorporated into CC and ECC to deal with the inherent randomness in their learning procedure, similar strategy may also be utilized by LSDM to account for the randomness in its procedure.
4) As shown in Tables 3, 4 and Fig. 4, across all evaluation measures, LSDM ranks 1st in 71.2% cases on the datasets with sparse features (*genbase, medical, language log, enron, slashdot, rcv1-s1, rcv1-s2, bibtex, eurlex-sm, eurlex-dc and tmc2007*). On the other hand, LSDM ranks 1st in more than 85.2% cases on the datasets with dense features (*CAL500, emotions, image, scene, yeast, corel5k, corel16k-s1, corel16k-s2 and mediamill*). These results indicate that LSDM tends to work better in application domains with dense feature representation than those with sparse feature representation.
5) Furthermore, across all evaluation measures, LSDM ranks 1st in 77.5% cases on all datasets with comparing algorithms. Especially, LSDM ranks 1st in 78.3% cases on regular-scale datasets (Table 3) and ranks 1st in 76.7% cases on large-scale datasets (Table 4). These results indicate that LSDM tends to work stably with arbitrary-scale datasets.

To summarize, LSDM achieves rather competitive performance against other well-established multi-label learning algorithms across 20 benchmark datasets and 6 evaluation measures, which validates the effectiveness of label specific discriminant mapping features. Specifically, the linear representation features and sLDA to deal with the setting of diverse β values show the superiority of LSDM compared with LIFT. The performance advantage is more pronounced on arbitrary-scale datasets with dense features.

6 CONCLUSION

In this paper, we propose a novel algorithm named LSDM to deal with the multi-label learning problem. Previously, numerous multi-label algorithms learn from training examples by manipulating the label space, such as exploiting label correlations and reducing label space dimension. To deal with this problem, LIFT which is the pioneer to learn from training examples by manipulating the feature space, i.e. learning label specific features, conducts cluster analysis on the positive and negative instances with regard to each label and performs training and testing by querying the clustering results which are employed to construct distance mapping features. However, it has two drawbacks: (a) Utilizing identical β value to control the number of clusters for each label ignores the differences between labels; (b) Using the distance information can not exploit the clustering results comprehensively. To overcome these two drawbacks, LSDM sets diverse β values to conduct cluster analysis for identical label and more thoroughly explores the label specific information by using linear representation features which describe the spatial topological information. Due to the problem of diverse reconstructed feature spaces for identical label, it employs sLDA to excavate optimal one with regard to each label efficiently.

The major contribution of our work is to utilize label specific discriminant mapping features, which suggests a promising direction for learning from multi-label data. Experiments across the largest number of benchmark datasets up to date show that: (a) Linear representation and sLDA effectively exploit label specific information; (b) LSDM achieves highly competitive performance against other state-of-the-art multi-label learning algorithms; (c) Multi-label learning algorithms comprising binary classifiers might be improved by utilizing label specific discriminant mapping features.

In the future, it is interesting to design other strategies of generating label specific discriminant mapping features which incorporate these features into other multi-label learning algorithms, and improve LSDM by considering label correlations into the feature construction step.

ACKNOWLEDGMENTS

This work was partially supported by the Natural Science Foundation of China under grant no. 61871141, the GRF, UGC under project Hong Kong PolyU 152039/14E and the Central Research Grant, Hong Kong PolyU under project G-YBVT, the International Exchange Program for Graduate Students, Tongji University (No.2017010007), National Key R&D Program of China (No. 2017YFC1703501), the Fundamental Research Funds for the Central public welfare research institutes (No. ZZ0908032).

REFERENCES

[1] Zafer Barutçuoglu, Robert E. Schapire, and Olga G. Troyanskaya. 2006. Hierarchical multi-label prediction of gene function. *Bioinformatics* 22, 7 (2006), 830–836.

[2] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. 2004. Learning multi-label scene classification. *Pattern Recognition* 37, 9 (2004), 1757–1771.

[3] Ricardo Silveira Cabral, Fernando De la Torre, João Paulo Costeira, and Alexandre Bernardino. 2011. Matrix Completion for Multi-label Image Classification.. In *NIPS*, Vol. 201. 2.

[4] Ricardo Silveira Cabral, Fernando De la Torre, João Paulo Costeira, and Alexandre Bernardino. 2015. Matrix Completion for Weakly-Supervised Multi-Label Image Classification. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 1 (2015), 121–135.

[5] Nicolò Cesa-Bianchi, Matteo Re, and Giorgio Valentini. 2012. Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference. *Machine Learning* 88, 1-2 (2012), 209–241.

[6] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM TIST* 2, 3 (2011), 27.

[7] Li Li1 Baobao Chang, Shi Zhao, and Lei Sha1 Xu Sun1 Houfeng Wang. 2015. Multi-label Text Categorization with Joint Learning Predictions-as-Features Method. (2015), 835–839.

[8] Weiwei Cheng and Eyke Hüllermeier. 2009. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning* 76, 2-3 (2009), 211–225.

[9] Amanda Clare and Ross D King. 2001. Knowledge discovery in multi-label phenotype data. In *Principles of data mining and knowledge discovery*. Springer, 42–53.

[10] Janez Demsar. 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research* 7 (2006), 1–30.

[11] Olive Jean Dunn. 1961. Multiple comparisons among means. *J. Amer. Statist. Assoc.* 56, 293 (1961), 52–64.

[12] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. 2008. Multilabel classification via calibrated label ranking. *Machine learning* 73, 2 (2008), 133–153.

[13] Yuhong Guo and Suicheng Gu. 2011. Multi-Label Classification Using Conditional Dependency Networks. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*. 1300–1305.

[14] James A Hanley and Barbara J McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* 143, 1 (1982), 29–36.

[15] Peng Hou, Xin Geng, and Min-Ling Zhang. 2016. Multi-Label Manifold Learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*. 1680–1686.

[16] Jun Huang, Guorong Li, Qingming Huang, and Xindong Wu. 2015. Learning Label Specific Features for Multi-label Classification. In *Data Mining (ICDM), 2015 IEEE International Conference on*. IEEE, 181–190.

[17] Jun Huang, Guorong Li, Qingming Huang, and Xindong Wu. 2016. Learning label-specific features and class-dependent labels for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering* 28, 12 (2016), 3309–3323.

[18] Jun Huang, Guorong Li, Qingming Huang, and Xindong Wu. 2018. Joint feature selection and classification for multilabel learning. *IEEE transactions on cybernetics* 48, 3 (2018), 876–889.

[19] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. 1999. Data clustering: a review. *ACM computing surveys (CSUR)* 31, 3 (1999), 264–323.

[20] Shuiwang Ji, Lei Tang, Shipeng Yu, and Jieping Ye. 2010. A shared-subspace learning framework for multi-label classification. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 4, 2 (2010), 8.

[21] Abhishek Kumar, Shankar Vembu, Aditya Krishna Menon, and Charles Elkan. 2012. Learning and inference in probabilistic classifier chains with beam search. In *Machine Learning and Knowledge Discovery in Databases*. Springer, 665–680.

[22] Yu-Kun Li, Min-Ling Zhang, and Xin Geng. 2015. Leveraging Implicit Relative Labeling-Importance Information for Effective Multi-label Learning. In *2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14-17, 2015*. 251–260.

[23] Eneldo Loza Mencía and Johannes Fürnkranz. 2008. Pairwise learning of multilabel classifications with perceptrons. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2008, part of the IEEE World Congress on Computational Intelligence, WCCI 2008, Hong Kong, China, June 1-6, 2008*. 2899–2906.

[24] Gjorgji Madjarov, Dejan Gjorgjevikj, and Tomche Delev. 2011. Efficient two stage voting architecture for pairwise multi-label classification. In *AI 2010: Advances in Artificial Intelligence*. Springer, 164–173.

[25] Elena Montañés, Robin Senge, José Barranquero, José Ramón Quevedo, Juan José del Coz, and Eyke Hüllermeier. 2014. Dependent binary relevance models for multi-label classification. *Pattern Recognition* 47, 3 (2014), 1494–1508.

[26] Lishan Qiao, Limei Zhang, Zhonggui Sun, and Xueyan Liu. 2017. Selecting label-dependent features for multi-label classification. *Neurocomputing* 259 (2017), 112–118.

[27] Jesse Read, Bernhard Pfahringer, and Geoff Holmes. 2008. Multi-label classification using ensembles of pruned sets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 995–1000.

[28] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2011. Classifier chains for multi-label classification. *Machine Learning* 85, 3 (2011), 333–359.

[29] Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science* 290, 5500 (2000), 2323–2326.

[30] Timothy N. Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. 2012. Statistical topic models for multi-label document classification. *Machine Learning* 88, 1-2 (2012), 157–208.

[31] Robert E. Schapire and Yoram Singer. 2000. BoosTexter: A Boosting-based System for Text Categorization. *Machine Learning* 39, 2/3 (2000), 135–168.

[32] Lu Sun, Mineichi Kudo, and Keigo Kimura. 2016. Multi-label classification with meta-label-specific features. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*. IEEE, 1612–1617.

[33] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2010. Mining multi-label data. In *Data mining and knowledge discovery handbook*. Springer, 667–685.

[34] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis P. Vlahavas. 2011. Random k-Labelsets for Multilabel Classification. *IEEE Trans. Knowl. Data Eng.* 23, 7 (2011), 1079–1089.

[35] G Tsoumakas, ML Zhang, and ZH Zhou. 2009. Tutorial on learning from multi-label data. In *ECML/PKDD*.

[36] Alexis Vallet and Hiroyasu Sakamoto. 2015. A Multi-Label Convolutional Neural Network for Automatic Image Annotation. *Journal of information processing* 23, 6 (2015), 767–775.

[37] Xiao Wang, Jun Zhang, and Guo-Zheng Li. 2015. Multi-location gram-positive and gram-negative bacterial protein subcellular localization using gene ontology and multi-label classifier ensemble. *BMC bioinformatics* 16, Suppl 12 (2015), S1.

[38] Wei Weng, Yaojin Lin, Shunxiang Wu, Yuwen Li, and Yun Kang. 2018. Multi-label learning based on label-specific features and local pairwise label correlation. *Neurocomputing* 273 (2018), 385–394.

[39] Suping Xu, Xibei Yang, Hualong Yu, Dong-Jun Yu, Jingyu Yang, and Eric CC Tsang. 2016. Multi-label learning with label-specific feature reduction. *Knowledge-Based Systems* 104 (2016), 52–61.

[40] Ju-Jie Zhang, Min Fang, and Xiao Li. 2015. Multi-label learning with discriminative features for each label. *Neurocomputing* 154 (2015), 305–316.

[41] Min-Ling Zhang. 2011. LIFT: Multi-Label Learning with Label-Specific Features. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*. 1609–1614.

[42] Min-Ling Zhang and Lei Wu. 2015. Lift: Multi-Label Learning with Label-Specific Features. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 1 (2015), 107–120.

[43] Min-Ling Zhang and Kun Zhang. 2010. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010*. 999–1008.

[44] Min-Ling Zhang and Zhi-Hua Zhou. 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition* 40, 7 (2007), 2038–2048.

[45] Min-Ling Zhang and Zhi-Hua Zhou. 2014. A Review on Multi-Label Learning Algorithms. *IEEE Trans. Knowl. Data Eng.* 26, 8 (2014), 1819–1837.