# Multilabel Distribution Learning Based on Multi-Output Regression and Manifold Learning

Chao Tan, Sheng Chen, *Fellow, IEEE*, Genlin Ji, and Xin Geng, *Member, IEEE*

*Abstract*—**Real-world multilabel data are high dimensional, and directly using them for label distribution learning will incur extensive computational costs. We propose a multilabel distribution learning algorithm based on multi-output regression through manifold learning, referred to as MDLRML. By exploiting smooth, similar spaces' information provided by the samples' manifold learning and the label distribution learning, we link the two spaces' manifolds. This facilitates using the topological relationship of the manifolds in the feature space to guide the manifold construction of the label space. The smoothest regression function is used to fit the manifold data, and a locally constrained multi-output regression is designed to improve the data's local fitting. Based on the regression results, we enhance the logical labels into the label distributions, thereby mining and revealing the label's hidden information regarding importance or significance. Extensive experimental results using real-world multilabel data sets show that the proposed MDLRML algorithm significantly improves the multilabel distribution learning accuracy and efficiency over several existing state-of-the-art schemes.**

*Index Terms*—**Multilabel learning, label distribution learning, multi-output regression, manifold learning**

## I. Introduction

In traditional machine learning, the primary goal is to resolve the problems related to single-label learning. Here, single-label learning means that a sample belongs to only one predefined class. But in real life, multiple class labels are often applicable to a sample, and the process of determining which labels are applicable to a sample is referred to as multilabel learning (MLL). In recent years, MLL has become popular in a variety of fields, including text classification, gene function analysis, image recognition, and video detection [1, 2].

A hot research topic related to MLL is label distribution learning (LDL). Researchers have employed LDL in a variety of ways, with different degrees of success. Geng *et al.* [3] were the first to propose the concept of label distribution, in an effort to solve the problem of insufficient training samples in the age-estimation problem. Geng *et al.* [4] proposed an algorithm for LDL based on improved iterative scaling, called IIS-LDL, to estimate facial age, which is the first proposal of LDL algorithm. Two further algorithms for LDL based on a quasi-Newton method (named BFGS-LDL) and the neural network-based method (referred to as CPNN), respectively, were also proposed later [5]. Geng [3] further proposed more algorithms for LDL, including AA-BP and AA-KNN. A label distribution support vector regressor (LDSVR) algorithm [6] considered LDL as a regression problem. In addition, LDL has been applied to practical applications. For example, Zhou *et al.* [7] used emotion distribution to identify facial expressions. LDL also has been studied in the fields of natural scene annotation [8], crowd counting [9], and video analysis [10].

Real-life MLL applications are usually high-dimensional. For high-dimensional multilabel datasets, many original machine learning methods perform poorly without effective feature extraction for data, leading to prohibitive computational costs. Conversely, reducing dimensionality as a preprocessing step, for example, by extracting the optimal representative features or transforming the original data into a low-dimensional space [11, 12], often improves greatly learning algorithms.

Manifold learning is a powerful tool for dimensionality reduction. Three famous local methods in manifold learning are locally linear embedding (LLE) [13], Laplacian eigenmaps (LEM) [14] and local tangent space alignment algorithm (LTSA) [15]. LLE [13] projects data points onto a low-dimensional space that preserves local geometric properties, using local neighborhood information of each point, while LEM [14] uses the weighted distance between two points as the loss function to get dimension reduction results. By contrast, LTSA [15] constructs a local tangent space for each point and obtains globally low-dimensional embedding results through affine transformation of the local tangent spaces, to increase the class separability. For multilabel problems, it is possible to deal with each single label individually using manifold learning based dimensionality reduction. The work [16] presented one of the first attempts to use the LLE method in the label space for MLL. Later Xu *et al.* [17] also used the LEM method in a label enhancement algorithm. However, a strong correlation always exists between class labels, and processing each label separately may degrade class separability. To mitigate the degradation in class separability, the data manifold retains the original data's intrinsic geometry to the greatest extent after performing dimensionality reduction. To effectively avoid degrading class separability, both the sample attributes and label distribution should be jointly considered.

Currently a few multilabel dimensionality-reduction methods exist, which partially exploit label information. Yu *et al.* [18] proposed a multilabel latent semantic index (MLSI) to preserve input information and capture correlation between multiple outputs. Zhang and Zhou [19] performed multilabel dimensionality reduction via dependency maximization (MDDM), which essentially maximizes feature attributes and class label dependence. Park and Lee [20] used generalized

C. Tan (Corresponding author) and G. Ji are with the School of Computer Science and Technology, Nanjing Normal University, Nanjing 210023, China (E-mails: tutu_tanchao@163.com, glji@njnu.edu.cn).

S. Chen is with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K., and also with King Abdulaziz University, Jeddah 21589, Saudi Arabia (E-mail: sqc@ecs.soton.ac.uk).

X. Geng is with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China (E-mail: xgeng@seu.edu.cn).

This work was supported by National Natural Science Foundation of China under Grant 61702270 and the China Postdoctoral Science Foundation under grant 2017M621592.

linear discriminant analysis (LDA) to contend with multilabel problems, by using least-squares regression to process high-dimensional multilabel text data. Li *et al.* [12] proposed a method called multilabel dimensionality reduction via semisupervised discriminant analysis (MSDA), which only uses partial label information and exploits the graph weight matrix of sample attributes and the similarity correlation matrix of partial sample labels to preserve the global and local intrinsic geometry of the original data. Xu [21] built a weighted multilabel LDA framework to consolidate two existing multilabel LDA-type methods with binary and correlation-based weight forms, and further collected two additional weight forms with entropy and fuzzy principles. Mikalsen *et al.* [22] presented a noisy multilabel semi-supervised dimensionality reduction (NMLSDR) method to denoise the noisy multilabel data and label the unlabeled data simultaneously. The NMLSDR learns a projection matrix for reducing the dimensionality by maximizing the dependence between the enlarged and denoised multilabel space and the features in the projected space. Ji *et al.* [23] considered a framework for extracting shared structures in multilabel classification. A common subspace is assumed to be shared among multiple labels, and the optimal solution to the proposed formulation is obtained by solving a generalized eigenvalue problem. For high-dimensional problems, direct computation of the solution is expensive, and an efficient algorithm was developed in [23].

For many real-world learning problems, data contain only logical labels rather than label distributions. A solution is to transform logical label into label distribution by mining the information on label importance contained in the training samples, which improves LDL's precision. Geng *et al.* [24] proposed a label-enhancement algorithm for LDL, which relies on mining label-related information hidden in the training samples to convert the training samples' original logical labels into label distributions. Although label distribution is not given explicitly, often it is included implicitly in the training samples. If this implicit label distribution can be recovered using some appropriate method, LDL can be harnessed to mine more semantic information. Xu *et al.* [17] proposed an algorithm called graph Laplacian label enhancement (GLLE) to recover label distributions from logical labels. This label enhancement reinforces the supervision information in training sets. In order to preserve the local structure of the label distribution in the embedding space, Peng *et al.* [25] proposed a multi-scale locality preserving (MSLP) algorithm to implement LDL label embedding. For real-world data, the labels may encounter problems, such as redundancy and noise. This is because the noise is inevitably generated during collection and preparation of some data points. MSLP is designed to be insensitive to these data points in order to alleviate this problem.

It can be seen that label embedding (LE) has become a hot research topic in MLL and LDL. In particular, the effective exploitation of label correlations is crucial for successful application of LDL. In order to realize LE in LDL, the following issues must be tackled properly: 1) How to exploit the information of label distributions efficiently, 2) How to recover the label vector to satisfy the constraints of label distribution, and 3) Learning must be a multi-output regression. These considerations motivate our work. In this paper, we propose a new multilabel distribution learning algorithm based on multi-output regression and manifold learning, called MDLRML. Our contributions are summarized as follows.

- We apply manifold learning to multilabel data to map raw sample data onto a low-dimensional subspace.
- We exploit the feature space manifold's topological relationship to guide the reconstruction of the label space manifold by constructing a smooth regression function.
- We also design a constrained multi-output regression to improve the data's local fitting.
- Finally, based on the regression results, we enhance the logical label to a label distribution for the samples' multilabel distribution learning and predicting.

Extensive experiments using real-world multilabel datasets with ground-true label distributions demonstrate the effectiveness and superior performance of the proposed MDLRML algorithm over a range of the existing methods.

Before we proceed, we define a few technical terminologies.

1) **Label manifold**: The label manifold is reconstructed with the transferred local topological structure from the feature manifold and logical labels. With the label manifold, a mapping from the feature manifold to the label manifold can be effectively found with a regression process. The label manifold contains more semantic information, which is beneficial to the learning process.
2) **Label embedding**: The feature manifold and label manifold are two different spaces but they share the local topological structure according to the smoothness assumption. The labeling manifold is not explicitly provided in the training examples. To reconstruct the label manifold, the key is this local topology. In order to study the label manifold, the label space should be extended to Euclidean space. This process is called label embedding.
3) **Label distribution**: This is a new machine learning paradigm, where each instance is annotated by label distribution. The label distribution represents the degree to which each label describes an instance.
4) **Label extension**: According to the smoothness assumption, the local topological structure can be transferred from the feature space to the label space. This is called the extension from the logical label space to the Euclidean label space.

## II. Related Work

### A. Multilabel learning based on manifold learning

Among the previous research efforts in MLL, many have focused on converting logical label space into Euclidean label space. Hou *et al.* [16] explored manifolds in the label space by treating labels as numbers. In this way, the label set contains more semantic information, which is beneficial to the learning process. The multilabel manifold learning algorithm, referred to as ML$^2$, proposed by Hou *et al.* [16] naturally induces a local topology according to the smoothing hypothesis [26] by extending from the logical label space to the Euclidean label space. However, ML$^2$ extends the original logical label space to the Euclidean space without dimensionality reduction.

According to Hou *et al.* [16], this method is the first attempt to explore the label space's manifold structure in MLL.

Xiang *et al.* [27] proposed an algorithm for nonlinear dimensionality reduction (NLDR). Tangent space projection is estimated at each data point on the manifold, through which the data point itself and its neighbors are represented in tangent space with local coordinates. An optimization framework is developed based on reconstruction error analysis, which is capable of yielding a global optimum. Motivated by the observation that the true cluster assignment matrix for high-dimensional data can be embedded in a linear space spanned by the data, Nie *et al.* [28] proposed a spectral embedded clustering (SEC) framework, in which a linearity regularization is added into the objective function to naturally deal with out-of-sample data. They also presented a new Laplacian matrix constructed from a local regression of each pattern and incorporated it into their framework to capture both local and global discriminative information for clustering. Xiang *et al.* [29] reformulated the LLE and LTSA algorithms within a unified regression framework, in terms of locally linear transformations The authors also presented an improved LLE algorithm which learns the manifold in the way similar to LLE but with significant performance improvement.

For many real-world data, performing dimensionality reduction as a preprocessing will greatly improve the learning algorithm's performance. Tai and Lin [30] attempted to reduce computational costs by seeking a major correlation between labels, especially for datasets with a large number of labels, where the Euclidean space's cardinality is a combination of the logical label vectors. Sun *et al.* [31] projected the feature and label spaces onto new spaces, by maximizing the correlation between the two spaces' projections. Both these approaches reduce the label space's dimensionality.

The well-established methods for MLL include the MLL algorithm based on neural network model (BP-MLL) [32], the multilabel naive Bayes classification algorithm (MLNB) [33], the multilabel lazy learning approach (ML-kNN) [34], and the ML$^2$ [16].

### *B. Multilabel distribution learning*

MLL studies the problem that each example or instance is associated with a set of labels. The learning process essentially constructs a mapping from instances onto labels, that is, the task is to learn the multilabel predictor that maps an instance onto the relevant label set [35, 36]. Geng [3] defines the learning process of an instance's labeled distribution as LDL. LDL, multi-output regression [37], and multi-ordinal regression [38] have some similarities, because they all use the numerical labels to annotate instances.

To label an instance $x$, a natural way is to assign a real number $d_x^y$ to each possible label $y$, and $d_x^y$ represents the degree to which $y$ describes $x$. Assume that $d_x^y \in [0,\ 1]$, and further assume that the label set is complete, that is, all the labels in the set fully describe an instance so that $\sum_y d_x^y = 1$. Then the description of the above condition is called the description degree $d_x^y$ of $y$ to $x$. For a particular instance, the description degrees of all the labels constitute a probability-distribution data form and, therefore, it is called the label distribution. The learning process on a dataset labeled by a label distribution is hence referred to as LDL [3].

The emergence of LDL makes it possible to learn richer semantics than just multilabels from data, such as more accurately characterizing the relative importance of different multilabels associated with the same instance. Geng [3] argued that both single-label learning and MLL may be regarded as special cases of LDL. In other words, LDL may be considered as a more general machine learning framework. However, the application of LDL is based on the assumption that each instance is labeled by a label distribution set that covers the importance of all labels, which often is not met in practice. The data in practical applications are mostly labeled by single label or multilabels with uniform label distribution, namely, single logical label or multiple logical labels, which lacks complete label distribution information. Despite this, the supervision information in these data is essentially following a certain unknown label distribution. Although this label distribution is not given explicitly, often it is contained implicitly in the training samples. Therefore, if this information can be recovered through some appropriate method, the label distribution learnt can help us discern more semantic information.

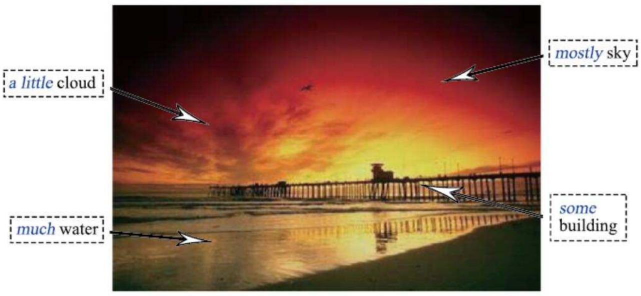


Fig. 1. An example about the relevance or irrelevance of each label.

In most supervised learning problems, label distribution is more general than logical label because the boundaries between relevance or irrelevance of labels to instance are unclear. When multiple labels are associated with an instance, the relative importance of them are more likely to be different than equal, that is, we do not have uniform label distribution in general. In Fig. 1, a natural scene image is annotated with labels like sky, water, buildings, and clouds. But the relative importance of each label to the image is different.

Unlike MLL, LDL represents greater flexibility in processing label ambiguity because it uses the label distribution instead of a binary label vector to annotate each instance.

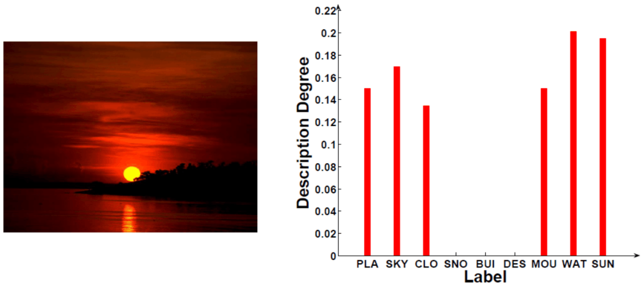


Fig. 2. Label distribution of a natural scene image.

Fig. 2 shows an example of LDL in the field of natural scene annotation, where the description degree represents the relevant degree of each label with the image.

However, it is difficult to obtain the label distribution directly because the process of quantifying the description degree is expensive. We need an effective way of recovering the label distributions from the logical labels of the training set by exploiting the correlation between the topological information in the feature space and the labels. This process is referred to as LE [25]. LE enhances supervised information in the training set by exploiting the relative importance of each label. Some works establish the relationship between the instances and the labels via graphics, and convert the logical label into the label distribution [16, 39]. After the label distribution is recovered, more effective supervised learning methods can be used by utilizing the label distribution.

To convert the logical labels into the label distributions, the existing LDL algorithms include the LDSVR [6], the CPNN [5], and the AA-KNN [3]. Differing from these well-known methods, our MDLRML is based on multi-output regression via manifold learning. The algorithm migrates the feature space's topological structure into the label space. By using the supervised information of the original label space to guide the learning of the feature manifold space, the main features of the original samples are extracted. With the reduced-dimensional main features and their associated label estimates to form the multi-output regression, the unknown label distributions are estimated with enhanced accuracy.

## III. Proposed Method

The original data are generally distributed on certain manifolds in both the feature and label spaces, and the manifolds of these two spaces are linked according to the smooth hypothesis [26]. Therefore, the feature space manifold's topological relationship can act as a guide in constructing the label space manifold. Hence, we can reconstruct the label manifold by transferring the local topology structures from the feature manifold and the existing logical labels. In other words, the feature and label manifolds are in two different spaces, and yet they share some local topology. This property of shared local topology can be utilized by manifold learning methods to guide the label manifold's reconstruction and to preserve the structural information between two spaces.

For high-dimensional data, the ultimate goal of dimensionality reduction is for linear transformation of data in low-dimensional subspaces to be quite close to the intrinsic dimension, that is, the minimum dimension required to represent high-dimensional data in the low-dimensional manifold subspaces, thereby preserving the global and local geometry structure. Our proposed algorithm nonlinearly reduces dimensionality on multilabel data and associates the samples' feature space to the manifold of the data's label space, by combining their label distributions to indicate the description degree of each label per instance. Then we can reconstruct the label manifold automatically from the multilabel data.

When performing feature extraction on big data streams, we can approximate the feature manifold by incrementally aligning the overlapping local linear neighborhood patches. Then we find the weight of each neighborhood patch using least-squares procedure, and reconstruct the label manifold using a local topological structure transformed from a feature manifold and a logical label. We design a multi-output regressor to learn and predict the samples' multilabel distribution and use the quadratic programming method for reconstruction. Finally, we establish the correlated relationships between the samples and labels based on the regression results, thereby enhancing the logical labels to the label distributions.

Based on the aforementioned considerations, our MDLRML algorithm primarily solves the following critical problems.

### A. *Reconstructing label manifold via manifold learning*

The feature space's topology structure for a multilabel training set $\mathcal{S}$ can be represented by a graph $\mathcal{G}=(\mathcal{X},\mathcal{E},\mathcal{W})$, where $\mathcal{X}\triangleq\{\boldsymbol{X}=[\boldsymbol{x}_1\cdots\boldsymbol{x}_N]|\boldsymbol{x}_i\in\mathbb{R}^m, 1\le i\le N\}$ is a set of vertices composed of $N$ examples $\boldsymbol{x}_i$, and $\mathcal{E}$ is a set of edges, while $\mathcal{W}\triangleq\{\boldsymbol{W}\in\mathbb{R}^{m\times d}\}$ is an edge weight matrix set of the graph, which reflects the mapping relationship between samples $\boldsymbol{x}_i$ in the feature manifold space $\mathbb{R}^m$ and the label space $\mathbb{R}^d$. In practice, $m\gg d$. First, in the feature space, we assume that the manifold of the examples' distribution satisfies local linearity, that is, any example $\boldsymbol{x}_i$ can be reconstructed by linearly combining its $k$-nearest neighbors, and we obtain the reconstruction weight matrix $\widetilde{\boldsymbol{W}}\in\mathbb{R}^{m\times\widehat{d}}$ by minimizing the following Frobenius-norm based cost function:

$$\min\|\boldsymbol{X}-\widetilde{\boldsymbol{W}}\boldsymbol{T}\|_F, \tag{1}$$

where $\boldsymbol{T}=\big[\boldsymbol{t}_1\cdots\boldsymbol{t}_N\big]\in\mathbb{R}^{\widehat{d}\times N}$ whose columns represent the eigenvectors of $\boldsymbol{X}$ in a low-dimension ($\widehat{d}$-dimension) space. Specifically, we seek $\boldsymbol{t}_i$ for $1\le i\le N$ to preserve as much of the local geometry in the $d$-dimensional feature space.

When new sample $\boldsymbol{x}_{N+1}$ arrives, its $k$ neighbours on the original data set $\mathcal{X}$ are selected. Assume that the $k$ neighbours of $\boldsymbol{x}_{N+1}$ contains sample $\boldsymbol{x}_i$, and we denote these $k$ samples by $\boldsymbol{X}_i=\big[\boldsymbol{x}_{i_1}\cdots\boldsymbol{x}_{i_k}\big]$. We can construct the matrix $\boldsymbol{B}_i=\boldsymbol{V}_i\boldsymbol{V}_i^{\mathrm{T}}$ and extract its features, where $\boldsymbol{V}_i\in\mathbb{R}^{k\times\widehat{d}}$ represents the feature vectors of $\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{X}_i$ corresponding to its $\widehat{d}$ largest eigenvalues [40]. Assume that $\boldsymbol{B}_i$ is approximated by its first $\widehat{d}$ largest singular values and their corresponding singular vectors. We can solve the feature-extraction problem using the singular value decomposition (SVD) on $\boldsymbol{B}_i$. Specifically, $\boldsymbol{B}_i=\boldsymbol{V}_i\boldsymbol{V}_i^{\mathrm{T}}\approx\boldsymbol{Q}_i\boldsymbol{\Sigma}\boldsymbol{P}_i^{\mathrm{T}}$, where $\boldsymbol{\Sigma}\triangleq\mathrm{diag}\big\{\sigma_1,\cdots,\sigma_{\widehat{d}}\big\}$ is a diagonal matrix containing the first $\widehat{d}$ largest singular values of $\boldsymbol{B}_i$ arranged in descending order, while $\boldsymbol{Q}_i=\big[\boldsymbol{q}_{i_1}\cdots\boldsymbol{q}_{i_{\widehat{d}}}\big]\in\mathbb{R}^{k\times\widehat{d}}$ and $\boldsymbol{P}_i\in\mathbb{R}^{k\times\widehat{d}}$ are the two matrices composed of the corresponding $\widehat{d}$ left and right singular vectors, respectively.

Inspired by the LTSA [15], we construct the optimal low-dimensional ($\widehat{d}$-dimension) coordinates $\boldsymbol{t}_i$, $1\le i\le N$, in (1), based on the coordinates of $\boldsymbol{Q}_i=\big[\boldsymbol{q}_{i_1}\cdots\boldsymbol{q}_{i_{\widehat{d}}}\big]$ in the feature space spanned by $\boldsymbol{B}$'s principal components. Specifically, let $\boldsymbol{T}_i=\big[\boldsymbol{t}_{i_1}\cdots\boldsymbol{t}_{i_k}\big]\in\mathbb{R}^{\widehat{d}\times k}$ contain $\boldsymbol{t}_i$. Then $\boldsymbol{T}_i$ is given by

$$\boldsymbol{T}_i=\frac{1}{k}\boldsymbol{T}_i\boldsymbol{1}_k\boldsymbol{1}_k^{\mathrm{T}}+\boldsymbol{Y}_i\boldsymbol{Q}_i^{\mathrm{T}}+\boldsymbol{E}_i, \tag{2}$$

where $\boldsymbol{1}_k$ is the $k$-dimensional column vector whose element are all ones, $\boldsymbol{Y}_i\in\mathbb{R}^{\widehat{d}\times\widehat{d}}$ is the local affine transformation, and $\boldsymbol{E}_i \in \mathbb{R}^{\widehat{d}\times k}$ represents the reconstruction error. To save as much of the low-dimensional geometry information in the feature space as possible, we need to find $\boldsymbol{T}_i$, i.e., $\boldsymbol{t}_i$, and $\boldsymbol{Y}_i$ to minimize the reconstruction error $\boldsymbol{E}_i$ as follows:

$$\min_{\boldsymbol{t}_i,1\le i\le N}\sum_{i=1}^{N}\left(\|\boldsymbol{E}_i\|_F^2=\left\|\boldsymbol{T}_i\left(\boldsymbol{I}_k-\frac{1}{k}\boldsymbol{1}_k\boldsymbol{1}_k^{\mathrm{T}}\right)-\boldsymbol{Y}_i\boldsymbol{Q}_i^{\mathrm{T}}\right\|_F^2\right),\quad (3)$$

where $\boldsymbol{I}_k$ is the $k$-dimensional identity matrix. By using $\boldsymbol{Y}_i = \boldsymbol{T}_i\big(\boldsymbol{I}_k - \frac{1}{k}\boldsymbol{1}_k\boldsymbol{1}_k^{\mathrm{T}}\big)\boldsymbol{Q}_i$, we have $\boldsymbol{E}_i = \boldsymbol{T}_i\big(\boldsymbol{I}_k - \frac{1}{k}\boldsymbol{1}_k\boldsymbol{1}_k^{\mathrm{T}}\big)\big(\boldsymbol{I}_k - \boldsymbol{Q}_i\boldsymbol{Q}_i^{\dagger}\big)$, where $\boldsymbol{Q}_i^{\dagger}$ is the Moore-Penrose generalized inverse of $\boldsymbol{Q}_i$.

In order to obtain a unique solution, the centralization and normalization constraints are added to the coordinates $\boldsymbol{T}$, and the reconstruction error can be expressed as:

$$\sum_{i=1}^{N}\left\|\boldsymbol{T}_i\left(\boldsymbol{I}_k-\frac{1}{k}\boldsymbol{1}_k\boldsymbol{1}_k^{\mathrm{T}}\right)\left(\boldsymbol{I}_k-\boldsymbol{Q}_i\boldsymbol{Q}_i^{\dagger}\right)\right\|_F^2=\mathrm{trace}\big(\boldsymbol{T}\boldsymbol{\Phi}\boldsymbol{T}^{\mathrm{T}}\big),\quad (4)$$

where $\boldsymbol{\Phi} = \sum_{i=1}^{N} \boldsymbol{S}_i\boldsymbol{R}_i\boldsymbol{R}_i^{\mathrm{T}}\boldsymbol{S}_i^{\mathrm{T}}$ in which $\boldsymbol{S}_i \in \mathbb{R}^{N\times k}$ is the 0-1 selection matrix such that $\boldsymbol{T}_i = \boldsymbol{T}\boldsymbol{S}_i$ and $\boldsymbol{R}_i$ is given by

$$\boldsymbol{R}_i = \left(\boldsymbol{I}_k - \frac{1}{k}\boldsymbol{1}_k\boldsymbol{1}_k^{\mathrm{T}}\right)\left(\boldsymbol{I}_k - \boldsymbol{Q}_i\boldsymbol{Q}_i^{\dagger}\right).\quad (5)$$

The eigenvectors corresponding to the $\widehat{d}$ largest eigenvalues of the matrix $\boldsymbol{\Phi}$ are the low-dimensional embeddings that minimize the reconstruction error.

### *B. Multi-output regression function for fitting manifold data*

The LE algorithm based on manifold [24] assumes that the data are distributed on certain manifolds in both the feature space and label space, and it uses the smooth hypothesis to link the manifolds of the two spaces so that the topological relationship of the feature space manifold can be utilized to guide the construction of the label space manifold. Assuming that the manifold of the example distribution satisfies the local linearity, that is, any example $\boldsymbol{x}_i$ can be reconstructed by a linear combination of its $k$-nearest neighbors, the reconstruction weight matrix $\widetilde{\boldsymbol{W}}$ can be obtained by minimizing the reconstruction error as presented in Subsection III-A.

We propose a multi-output regression for fitting manifold data. Given the training set $\boldsymbol{X} = [\boldsymbol{x}_1 \cdots \boldsymbol{x}_N]$ and the corresponding logical label matrix $\boldsymbol{L} = [\boldsymbol{l}_1 \cdots \boldsymbol{l}_N]$, where $\boldsymbol{l}_i \in \{0,1\}^d$, our aim is to recover the label distribution matrix $\boldsymbol{D} = [\boldsymbol{d}_1 \cdots \boldsymbol{d}_N]$, where $\boldsymbol{d}_i \in [0,\ 1]^d$, from the logical label matrix $\boldsymbol{L}$. Traditional label space is spanned by the label vectors $\boldsymbol{l}_i$. In order to study the label manifold, the label space should be extended to a Euclidean space. Each dimension of the space still corresponds to a label, but the value is extended from logical $\{0,1\}$ to real $[0,\ 1]$. Such numerical label can be used to represent the label's relevance to the example, and it carries more semantic information to describe the instance more comprehensively than the logical label [16]. To solve this label extension problem, we consider the model:

$$\boldsymbol{d}_i = \boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\varphi}_i + \boldsymbol{b},\quad (6)$$

where $\boldsymbol{\Theta} = \big[\boldsymbol{\theta}_1\ \boldsymbol{\theta}_2 \cdots \boldsymbol{\theta}_d\big] \in \mathbb{R}^{d\times d}$ is the weight matrix in the label space and $\boldsymbol{b} = [b_1\ b_2 \cdots b_d]^{\mathrm{T}} \in \mathbb{R}^d$, while $\boldsymbol{\varphi}_i \in \mathbb{R}^d$ is the nonlinear transformation to a low dimensional feature space, i.e., the low-dimensional embedding coordinates corresponding to the $\widehat{d}$ largest eigenvalues of the matrix $\boldsymbol{\Phi}$ that minimize the reconstruction error. Then the multi-output ($d$-output) support vector regression (SVR) can be generalized by minimizing the following objective function

$$F\big(\boldsymbol{\Theta},\boldsymbol{b}\big) = \frac{1}{2}\sum_{j=1}^{d}\big\|\boldsymbol{\theta}_j\big\|^2 + \sum_{i=1}^{N} L(u_i),\quad (7)$$

where $L(\cdot)$ is the chosen SVR loss function, $u_i = \|\boldsymbol{e}_i\|$, and $\boldsymbol{e}_i = \boldsymbol{l}_i - \boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\varphi}_i - \boldsymbol{b}$.

By solving the quadratic programming problem associated with the cost function (7) to determine the numerical label $\boldsymbol{d}_i$, the proposed algorithm based on manifold reconstructs the feature and label spaces' manifolds. The algorithm uses the smoothness assumption to migrate the feature space's topological relationship into the label space, thereby enhancing the logical label to the label distribution. Specifically, the $L_2$ norm is used to define the loss function in (7) [6]

$$L(u_i) = \begin{cases} 0, & u_i < \varepsilon, \\ (u_i - \varepsilon)^2, & u_i \ge \varepsilon. \end{cases}\quad (8)$$

To minimize (7), the iterative quasi-Newton method known as iterative reweighed least squares (IRWLS) [41] is used. Firstly, $L(u_i)$ is approximated by its first order Taylor expansion at the solution of the current $\iota$th iteration, denoted by $\boldsymbol{e}_i^{(\iota)}$ and $u_i^{(\iota)}$, namely,

$$L^{'}(u_i) = L\big(u_i^{(\iota)}\big) + \frac{\partial L(u)}{\partial u}\Big|_{u=u_i^{(\iota)}}\frac{\big(\boldsymbol{e}_i^{(\iota)}\big)^{\mathrm{T}}}{u_i^{(\iota)}}\big(\boldsymbol{e}_i - \boldsymbol{e}_i^{(\iota)}\big).\quad (9)$$

Then we further construct a quadratic approximation

$$L^{''}(u_i) = L\big(u_i^{(\iota)}\big) + \frac{\partial L(u)}{\partial u}\Big|_{u=u_i^{(\iota)}}\frac{u_i^2 - \big(u_i^{(\iota)}\big)^2}{2u_i^{(\iota)}} = \frac{1}{2}a_i^{(\iota)}u_i^2 + \tau,\quad (10)$$

where $\tau$ is a constant term and

$$a_i^{(\iota)} = \frac{1}{u_i^{(\iota)}}\frac{\partial L(u)}{\partial u}\Big|_{u=u_i^{(\iota)}} = \begin{cases} 0, & u_i^{(\iota)} < \varepsilon, \\ \frac{2\left(u_i^{(\iota)} - \varepsilon\right)}{u_i^{(\iota)}}, & u_i^{(\iota)} \ge \varepsilon. \end{cases}\quad (11)$$

Substituting (10) into (7) yields the weighted least-squares (WLS) problem given by

$$F^{''}\big(\boldsymbol{\Theta},\boldsymbol{b}\big) = \frac{1}{2}\sum_{j=1}^{d}\big\|\boldsymbol{\theta}_j\big\|^2 + \frac{1}{2}\sum_{i=1}^{N} a_i^{(\iota)}u_i^2 + \tau.\quad (12)$$

In order to solve the WLS problem (12), each component $\big(\boldsymbol{\theta}_j, b_j\big)$ can be independently solved by equating its gradient to zero [42] for $j = 1, \cdots, d$:

$$\frac{\partial F^{''}}{\partial \boldsymbol{\theta}_j} = \boldsymbol{\theta}_j - \boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{D}_{\boldsymbol{a}}\big(\boldsymbol{\omega}_j - \boldsymbol{\Psi}\boldsymbol{\theta}_j - \boldsymbol{1}_N b_j\big) = \boldsymbol{0}_d,\quad (13)$$

$$\frac{\partial F^{''}}{\partial b_j} = \big(\boldsymbol{a}^{(\iota)}\big)^{\mathrm{T}}\big(\boldsymbol{\omega}_j - \boldsymbol{\Psi}\boldsymbol{\theta}_j - \boldsymbol{1}_N b_j\big) = 0.\quad (14)$$

Here $\boldsymbol{\Psi} = \big[\boldsymbol{\varphi}_1\ \boldsymbol{\varphi}_2 \cdots \boldsymbol{\varphi}_N\big]^{\mathrm{T}} \in \mathbb{R}^{N\times d}$ is the nonlinear transformation of the samples $\boldsymbol{x}_i$ to the $d$-dimensional space, that is, the optimal solution of (4), and $\boldsymbol{D}_{\boldsymbol{a}} \in \mathbb{R}^{N\times N}$ is the diagonal matrix given by $\boldsymbol{D}_{\boldsymbol{a}} = \text{diag}\big\{a_1^{(\iota)}, a_2^{(\iota)} \cdots a_N^{(\iota)}\big\}$, while $\boldsymbol{\omega}_j = \big[[\boldsymbol{l}_1]_j\ [\boldsymbol{l}_2]_j \cdots [\boldsymbol{l}_N]_j\big]^{\mathrm{T}}$, $[\boldsymbol{l}]_j$ denotes the $j$th element of $\boldsymbol{l}$, and $\mathbf{0}_d$ is the $d$-dimensional zero vector. Furthermore, $\boldsymbol{a}^{(\iota)} = \big[a_1^{(\iota)}\ a_2^{(\iota)} \cdots a_N^{(\iota)}\big]^{\mathrm{T}}$. Then the optimal solution of (12) is found by solving the following linear equations for $1 \le j \le d$

$$\begin{bmatrix} \boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{D}_{\boldsymbol{a}}\boldsymbol{\Psi} + \boldsymbol{I}_d & \boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{a}^{(\iota)} \\ \big(\boldsymbol{a}^{(\iota)}\big)^{\mathrm{T}}\boldsymbol{\Psi} & \mathbf{1}_N^{\mathrm{T}}\boldsymbol{a}^{(\iota)} \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_j \\ b_j \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{D}_{\boldsymbol{a}}\boldsymbol{\omega}_j \\ \big(\boldsymbol{a}^{(\iota)}\big)^{\mathrm{T}}\boldsymbol{\omega}_j \end{bmatrix}. \quad (15)$$

The multi-output SVR cannot be solved as the standard SVR but we can use an iterative method, similar to the one proposed in [42]. This procedure first obtains $\boldsymbol{\theta}_j$ and $b_j$. Then, the direction of the optimal solution of (15) is used as the descending direction for optimizing $F\big(\boldsymbol{\Theta}, \boldsymbol{b}\big)$. The algorithm iterates until the SVR solution is reached, and the solution for the next $(\iota+1)$th iteration, denoted as $\boldsymbol{\Theta}^{(\iota+1)}$ and $\boldsymbol{b}^{(\iota+1)}$, is obtained via a line search algorithm along this direction.

### C. Enhancing to label distribution with sigmoid function

Sigmoid function can be used to constrain each component of a distribution to within the range of $[0,\ 1]$. With the sigmoid function acting directly as the regression target, we have the regression problem that minimizes the objective function:

$$\Gamma(\boldsymbol{\Theta}, \boldsymbol{b}) = \frac{1}{2}\sum_{j=1}^{d} \big\|\boldsymbol{\theta}_j\big\|^2 + \alpha\sum_{i=1}^{N} L(u_i) + \beta\sum_{i=1}^{N} L(r_i), \quad (16)$$

where $\alpha + \beta = 1$, $L(\cdot)$ is the loss function defined in (8), $r_i = \|\boldsymbol{c}_i\|$, and $\boldsymbol{c}_i = \boldsymbol{l}_i - \boldsymbol{f}(\boldsymbol{x}_i) \in \mathbb{R}^d$, while the SVR label distribution $\boldsymbol{f}(\boldsymbol{x}_i)$ is defined by an element-wise sigmoid vector [6]

$$\big[\boldsymbol{f}(\boldsymbol{x}_i)\big]_j = \frac{1}{1 + \exp\Big(-\big[\boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\varphi}_i\big]_j - b_j\Big)}, 1 \le j \le d. \quad (17)$$

For notational convenience, we will denote (17) concisely as

$$\boldsymbol{f}(\boldsymbol{x}_i) = \frac{1}{1 + \exp\big(-\boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\varphi}_i - \boldsymbol{b}\big)}. \quad (18)$$

However, substituting (17) into (16) does not lead to a convex quadratic form and it is difficult to find the optimization results. To address this problem, an alternative regression is adopted to reform the minimization of (16) to a convex quadratic programming process [6]. Specifically, the 'error' vector $\boldsymbol{c}_i$ is defined by the alternative regression as

$$\big[\boldsymbol{c}_i\big]_j = -\log\left(\frac{1}{\big[\boldsymbol{l}_i\big]_j} - 1\right) - \big[\boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\varphi}_i\big]_j - b_j, 1 \le j \le d, \quad (19)$$

or 'concisely' as

$$\boldsymbol{c}_i = -\log\left(\frac{1}{\boldsymbol{l}_i} - 1\right) - \boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\varphi}_i - \boldsymbol{b}. \quad (20)$$

Similarly, we adopt the IRWLS. The loss function $L(r_i)$ can be approximated by the solution of the current $\iota$th iteration, denoted by $\boldsymbol{c}_i^{(\iota)}$ and $r_i^{(\iota)}$, as

$$L^{''}(r_i) = \frac{1}{2}q_i^{(\iota)} r_i^2 + \tau, \quad (21)$$

where

$$q_i^{(\iota)} = \frac{1}{r_i^{(\iota)}}\frac{\partial L(r)}{\partial r}\Big|_{r=r_i^{(\iota)}} = \begin{cases} 0, & r_i^{(\iota)} < \varepsilon, \\ \frac{2\left(r_i^{(\iota)} - \varepsilon\right)}{r_i^{(\iota)}}, & r_i^{(\iota)} \ge \varepsilon, \end{cases} \quad (22)$$

and $\tau$ is a constant term. Hence we arrive at

$$\Gamma^{''}\big(\boldsymbol{\Theta}, \boldsymbol{b}\big) = \frac{1}{2}\sum_{j=1}^{d}\big\|\boldsymbol{\theta}_j\big\|^2 + \frac{1}{2}\sum_{i=1}^{N}\Big(\alpha a_i^{(\iota)} u_i^2 + \beta q_i^{(\iota)} r_i^2\Big) + \tau. \quad (23)$$

The solution of the WLS problem (23) can readily obtained by solving the following linear equations for $1 \le j \le d$

$$\begin{bmatrix} \alpha\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{D}_{\boldsymbol{a}}\boldsymbol{\Psi} + \boldsymbol{I}_d & \alpha\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{a}^{(\iota)} \\ \alpha\big(\boldsymbol{a}^{(\iota)}\big)^{\mathrm{T}}\boldsymbol{\Psi} & \alpha\mathbf{1}_N^{\mathrm{T}}\boldsymbol{a}^{(\iota)} \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_j \\ b_j \end{bmatrix} = \begin{bmatrix} \alpha\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{D}_{\boldsymbol{a}}\boldsymbol{\omega}_j - \beta\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{D}_{\boldsymbol{q}}\boldsymbol{\eta}_j \\ \alpha\big(\boldsymbol{a}^{(\iota)}\big)^{\mathrm{T}}\boldsymbol{\omega}_j - \beta\big(\boldsymbol{q}^{(\iota)}\big)^{\mathrm{T}}\boldsymbol{\eta}_j \end{bmatrix}, \quad (24)$$

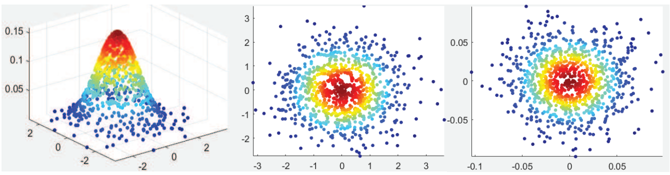
where $\boldsymbol{D}_{\boldsymbol{q}} = \text{diag}\big\{q_1^{(\iota)}, \cdots, q_N^{(\iota)}\big\}$, and $\boldsymbol{q}^{(\iota)} = \big[q_1^{(\iota)} \cdots q_N^{(\iota)}\big]^{\mathrm{T}}$, while

$$\boldsymbol{\eta}_j = \left[\log\left(\frac{1}{[\boldsymbol{l}_1]_j} - 1\right) \cdots \log\left(\frac{1}{[\boldsymbol{l}_N]_j} - 1\right)\right]^{\mathrm{T}} \in \mathbb{R}^N. \quad (25)$$

The direction of the optimal solution of (24) is used as the descending direction for optimizing $\Gamma(\boldsymbol{\Theta}, \boldsymbol{b})$, and the solution for the next $(\iota+1)$ iteration, denoted as $\boldsymbol{\Theta}^{(\iota+1)}, \boldsymbol{b}^{(\iota+1)}$ is obtained via a line search algorithm along this direction. By calculating the optimal solution of (24), it is convenient to find the direction of the optimal solution for the WLS problem (23), and the solution for the optimization (16).

**Algorithm 1:** Procedure of MDLRML

```
Require: Multilabel sample set $\boldsymbol{X}$ of $N$ examples, logical
    label matrix $\boldsymbol{L}$, and edge weight matrix $\boldsymbol{W}$ reflecting
    mapping relationship of sample $\boldsymbol{x}$ between feature
    manifold space $\mathbb{R}^m$ and label space $\mathbb{R}^d$.
    New sample $\boldsymbol{x}_{N+1}$ and matrix $\boldsymbol{X}_i = [\boldsymbol{x}_{i_1} \cdots \boldsymbol{x}_{i_k}]$
    containing its $k$ neighbors which includes sample $\boldsymbol{x}_i$.
Ensure: Label distribution $\boldsymbol{D}$ for multilabel sample set $\boldsymbol{X}$.
  Part I. Reconstructing label manifold via manifold
      learning:
  I.1 Construct matrix $\boldsymbol{B}_i = \boldsymbol{V}_i\boldsymbol{V}_i^{\mathrm{T}}$ and extract its features,
      where $\boldsymbol{V}_i$ contains feature vectors of $\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{X}_i$
      corresponding to $d$ largest eigenvalues, for $1 \le i \le N$.
  I.2 Perform SVD on $\boldsymbol{B}_i$ to obtain matrix $\boldsymbol{Q}_i$ consisting
      of singular vectors corresponding to $d$ largest singular
      values, for $1 \le i \le N$.
  I.3 Optimize objective function (3), select eigenvectors
      of $\boldsymbol{\Phi}$ corresponding to its first $d$ largest eigenvalues as
      low-dimensional embeddings that minimize
      reconstruction error.
  Part II. Multilabel distribution learning with sigmoid
      function enhanced multi-output regression:
  II.1 Form the sigmoid function enhanced multi-output
      SVR model with the cost function (16).
  II.2 Minimize (16) by using IRWLS to construct WLS
      problem (23) whose optimal solution is solution (24).
```



(a) (b) (c)

Fig. 3. (a) The synthetic data points in the feature space; (b) the true numerical label points; and (c) the reconstructed numerical label points.

Let the cost function value of the WLS problem (23) at the $\iota$th iteration be $L_{\text{cost}}(\iota)$. Then the iterative procedure is terminated when

$$\frac{L_{\text{cost}}(\iota-1)-L_{\text{cost}}(\iota)}{L_{\text{cost}}(\iota-1)}<\xi \tag{26}$$

where $\xi$ is small positive number, such as $\xi=10^{-5}$.

### D. Proposed MDLRML algorithm

By combining the results of Subsections III-A to III-C, we arrive at our proposed MDLRML algorithm, which is summarized in Algorithm 1.

We use a three-dimensional synthetic dataset with 1000 points to illustrate the effectiveness of our method in recovering the label distributions. The third dimension of the data is calculated as the Gaussian distribution of the first two dimensions. The Gaussian distribution has zero mean, and the variance of each dimension is 1. Fig. 3 (a) depicts the data points in the feature space. The label space is two-dimensional, and Fig. 3 (b) shows the true label points. Comparing Fig. 3 (a) with Fig. 3 (b) reveals the rationality of the smoothness assumption, i.e., the points close to each other in the feature space are also close in the label space. Fig. 3 (c) shows the reconstructed label points, where the reconstruction weight matrix $\widetilde{\boldsymbol{W}}$ is obtained by minimizing the cost function (1). Observe that the reconstructed label points closely resemble the true label points. Note that in Part I of reconstructing the label manifold, dimensionality reduction has occurred on the original dataset, and this results in a change in the scale of the estimated label space.

*1) Convergence analysis:* In Algorithm 1, Part II involves using the IRWLS algorithm to solve the corresponding multi-output SVR problem [42]. It is well known that the convergence of the IRWLS to a stationary point is guaranteed [43]. Therefore, the convergence of the MDLRML to a solution is guaranteed. The rate of convergence has also been proved for the IRWLS [43]. Hence our MDLRML is guaranteed to converge to a solution in finite number of iterations.

*2) Remark:* In the proposed MDLRML, without the sigmoid function enhancement, i.e., without the third term in the cost function (16), the algorithm still works. We will refer to the algorithm consisting of only Subsections III-A and III-B as the MDLRML-3. Similarly, the algorithm without the second term in the cost function (16), denoted as the MDLRML-2, will also work. In the experimental evaluation section, we will test all these three algorithms, to evaluate their achievable performance. It is expected that the MDLRML will outperform both the MDLRML-2 and MDLRML-3.

*3) Complexity analysis:* The complexity of feature decomposition in Part I.1 is on the order of $k^3$, and the complexity of SVD in I.2 is on the order of $d\times k^3$, while the complexity of I.3 can be worked out to be $d\times N\times k^3$. Therefore, the computational complexity of Part I, reconstructing label manifold via manifold learning, is on the order of $d\times N\times k^3$, denoted as $\mathsf{O}\big(d\times N\times k^3\big)$.

Let the number of iterations for the IRWLS in Part II be upper bounded by $I_{\text{up}}$. The complexity per iteration of the IRWLS follows the complexity of SVR, which is on the order of $\mathsf{O}\big(N^3\big)$. Therefore, the complexity of Part II, multilable distribution learning, is on the order of $\mathsf{O}\big(I_{\text{up}}\times N^3\big)$.

Since the number of samples $N$ is much larger than $d$ and $k$, the computational complexity of the MDLRML is on the order of $\mathsf{O}\big(I_{\text{up}}\times N^3\big)$.

Interestingly, the complexity of the MDLRML may be lower than those of the MDLRML-3 and MDLRML-2. This is because the data form for the MDLRML-3 has one Kronecker delta matrix $\boldsymbol{D_a}$ which is sparse, and the data form for the MDLRML-2 has one Kronecker delta matrix $\boldsymbol{D_q}$ which is sparse, while the MDLRML has two sparse Kronecker delta matrices $\boldsymbol{D_a}$ and $\boldsymbol{D_q}$. Therefore, the data for for the MDLRML are sparser than those for the MDLRML-2 and MDLRML-3. As a result, the MDLRML may take less number of iterations to converge than the other two algorithms. This analysis will be further confirmed in the next section.

## IV. Experimental evaluation

### A. Experimental setup

*1) Comparison algorithms:* In the experimental evaluation of our proposed MDLRML, MDLRML-3 and MDLRML-2 algorithms, we choose four well-established MLL algorithms for comparison, which are the BP-MLL [32], the MLNB [33],

TABLE I
Characteristics of multilabel datasets [44] used in experimental evaluation with MLL metrics

| Dataset | $S$ | $T$ | $dim(S)$ | $L(S)$ | $LCard(S)$ | $LDen(S)$ | $DL(S)$ | $F(S)$ |
|---|---|---|---|---|---|---|---|---|
| Emotions | 415 | 178 | 72 | 6 | 1.869 | 0.311 | 27 | numeric |
| Medical | 645 | 333 | 1449 | 45 | 1.245 | 0.028 | 94 | nominal |
| Cal500 | 250 | 252 | 68 | 174 | 26.044 | 0.150 | 502 | numeric |
| Birds | 320 | 325 | 260 | 19 | 1.014 | 0.053 | 133 | numeric |
| Enron | 1123 | 579 | 1001 | 53 | 3.378 | 0.064 | 753 | nominal |
| Yeast | 1200 | 1217 | 103 | 14 | 4.237 | 0.303 | 198 | numeric |
| Image | 1000 | 1000 | 294 | 5 | 1.236 | 0.247 | 20 | numeric |
| Scene | 1211 | 1196 | 294 | 6 | 1.074 | 0.179 | 15 | numeric |
| Corel5k | 2500 | 2500 | 499 | 374 | 3.522 | 0.009 | 3175 | nominal |
| Bibtex | 3700 | 3695 | 1836 | 159 | 2.402 | 0.015 | 2856 | nominal |

the ML-kNN [34], and the ML$^2$ [16]. We also select three well-known LDL algorithms, the LDSVR [6], the CPNN [5], and the AA-KNN [3], as the benchmarks for comparison with our MDLRML, MDLRML-3 and MDLRML-2.

*2) Datasets:* To compare our algorithms with the aforementioned MLL and LDL algorithms, we select 10 real-world multilabel datasets from Mulan website [44] for performance evaluation. Table I summarizes the features of these datasets. Half of these datasets are regular-sized and half of them are large-scale. These datasets therefore cover a wide range of multilabel attributes. In Table I, $S$ is the number of examples, $T$ is the number of testing samples, $dim(S)$ denotes the feature dimensions, $L(S)$ is the number of class labels, $LCard(S)$ is the label cardinality, $LDen(S)$ is the label density, $DL(S)$ denotes the distinct label sets, and $F(S)$ is the feature type.

To compare the accuracy of the estimated label distributions obtained by our algorithms with those obtained by the aforementioned LDL algorithms, we use 15 real-world datasets of [3] with the known ground-true label distributions. Table II summarizes these datasets.

*3) Evaluation metrics:* We choose five widely used evaluation metrics for MLL, and they are: Hamming loss, one error, coverage, ranking loss, and average precision. For average precision, the larger the value, the better the performance. Hence we use ↑ after this evaluation metric, i.e., average precision ↑. For the other four metrics, the smaller the values, the better the performance. Therefore, we use ↓ after the evaluation index.

For the 15 real-world datasets of Table II with the known ground-true label distributions used in LDL, we can compare the estimated label distributions with the ground-true label distributions. We employ six representative label distribution evaluation indicators [3] for this task, which are: Chebyshev distance, Clark distance, Canberra distance, Kullback-Leibler divergence, cosine coefficient, and intersectional similarity.

TABLE II
MULTILABEL DATASETS WITH GROUND-TRUE LABEL DISTRIBUTIONS FROM [3] USED IN EXPERIMENTAL EVALUATION WITH LDL METRICS

| Dataset | Examples ($N$) | Features ($m$) | Labels ($d$) |
|---|---|---|---|
| Yeast-alpha | 2465 | 24 | 18 |
| Yeast-cdc | 2465 | 24 | 15 |
| Yeast-elu | 2465 | 24 | 14 |
| Yeast-diau | 2465 | 24 | 7 |
| Yeast-heat | 2465 | 24 | 6 |
| Yeast-spo | 2465 | 24 | 6 |
| Yeast-cold | 2465 | 24 | 4 |
| Yeast-dtt | 2465 | 24 | 4 |
| Yeast-spo5 | 2465 | 24 | 3 |
| Yeast-spoem | 2465 | 24 | 2 |
| Human Gene | 30542 | 36 | 68 |
| Natural Scene | 2000 | 294 | 9 |
| Movie | 7755 | 1869 | 5 |
| SJAFFE | 213 | 243 | 6 |
| SBU_3DFE | 2500 | 243 | 6 |

### *B. Experimental result comparison based on MLL metrics*

*1) Comparison of achievable MLL metrics:* In the experimental comparison of our algorithms with the seven chosen benchmark algorithms using the 10 real-world datasets, half the examples in each dataset are selected randomly as a training set, and the remaining half are used to form a test set. No preprocessing is performed on data. We set the algorithmic parameters of the MDLRML as follows. The number of $k$ nearest neighbours is set to $k = 10$. This value is chosen simply to be consistent with the value of $k$ used in the benchmark algorithms [16]. We choose $\widehat{d} = 8$. In fact, we have tested the values of $\widehat{d}$ from 1 to 9, and the results obtained are all

TABLE III
PERFORMANCE COMPARISON OF 10 ALGORITHMS ON 10 DATASETS [44] USING HAMMING LOSS ↓

| Algorithms | BP-MLL | MLNB | ML-kNN | ML$^2$ | LDSVR | CPNN | AA-KNN | MDLRML | MDLRML-2 | MDLRML-3 |
|---|---|---|---|---|---|---|---|---|---|---|
| Yeast | 0.4500(9) | 0.2061(3) | 0.1980(2) | 0.2267(4) | 0.3037(6) | 0.6964(10) | 0.2297(5) | **0.1910**(1) | 0.3058(7.5) | 0.3058(7.5) |
| Emotions | 0.2987(7) | 0.2414(4) | 0.2584(5) | 0.2865(6) | 0.2996(8) | 0.7097(10) | 0.3006(9) | **0.2247**(1) | 0.2406(2.5) | 0.2406(2.5) |
| Medical | 0.0290(4) | 0.0362(5) | 0.0178(2) | 0.3405(8) | 0.9721(9) | 0.9732(10) | 0.0184(3) | **0.0116**(1) | 0.1739(6.5) | 0.1739(6.5) |
| Cal500 | 0.1472(3) | 0.2062(6) | 0.1416(2) | 0.3701(9) | 0.1488(4) | 0.8522(10) | 0.1814(5) | **0.1412**(1) | 0.2630(7.5) | 0.2630(7.5) |
| Birds | 0.0683(4) | 0.0704(5) | 0.0546(3) | 0.1330(9) | 0.0517(2) | 0.9491(10) | 0.0748(6) | **0.0514**(1) | 0.1273(7.5) | 0.1273(7.5) |
| Image | 0.3056(8) | 0.2108(5) | 0.1888(2) | 0.2450(7) | 0.7516(9) | 0.7522(10) | 0.2158(6) | **0.1644**(1) | 0.2054(3.5) | 0.2054(3.5) |
| Scene | 0.2904(9) | 0.1225(4) | 0.0962(2) | 0.1803(7) | 0.1810(8) | 0.8194(10) | 0.1134(3) | **0.0872**(1) | 0.1559(5.5) | 0.1559(5.5) |
| Enron | 0.0682(4) | 0.1162(6) | 0.0623(2) | 0.2049(7) | 0.0677(3) | 0.9339(10) | 0.0705(5) | **0.0570**(1) | 0.2924(8.5) | 0.2924(8.5) |
| Corel5k | 0.0094(2) | 0.0138(4) | **0.0093**(1) | 0.3922(8) | 0.9907(9.5) | 0.9907(9.5) | 0.0114(3) | 0.0156(5) | 0.2251(6.5) | 0.2251(6.5) |
| Bibtex | 0.0160(4) | 0.0846(7) | 0.0135(2) | 0.0747(6) | 0.0149(3) | 0.9853(10) | 0.0165(5) | **0.0094**(1) | 0.2785(8.5) | 0.2785(8.5) |
| Average rank | 5.4(5) | 4.9(3) | 2.3(2) | 7.1(9) | 6.15(6) | 9.95(10) | 5.0(4) | **1.4**(1) | 6.4(7.5) | 6.4(7.5) |

TABLE IV
PERFORMANCE COMPARISON 10 ALGORITHMS ON 10 DATASETS [44] USING RANKING LOSS ↓

| Algorithms | BP-MLL | MLNB | ML-kNN | ML$^2$ | LDSVR | CPNN | AA-KNN | MDLRML | MDLRML-2 | MDLRML-3 |
|---|---|---|---|---|---|---|---|---|---|---|
| Yeast | 0.4450(7) | 0.2323(2) | **0.1716**(1) | 0.3325(6) | 0.4974(8) | 0.9708(10) | 0.5054(9) | 0.2937(3) | 0.2990(4.5) | 0.2990(4.5) |
| Emotions | 0.4803(8) | 0.2285(4) | 0.2827(5) | 0.3579(6) | 0.5899(9) | 0.8511(10) | 0.4283(7) | **0.1812**(2) | **0.1812**(2) | **0.1812**(2) |
| Medical | 0.2445(6) | 0.0623(2) | **0.0555**(1) | 0.4984(7) | 0.5000(8) | 0.8982(10) | 0.5039(9) | 0.1440(5) | 0.1059(3.5) | 0.1059(3.5) |
| Cal500 | 0.1996(4) | 0.1882(3) | 0.1880(2) | 0.4981(7) | 0.5005(8) | 0.8621(10) | 0.7750(9) | **0.1769**(1) | 0.4616(5.5) | 0.4616(5.5) |
| Birds | 0.3964(7) | **0.2115**(1) | 0.3035(3) | 0.4051(8) | 0.4374(9) | 0.3132(4) | 0.7335(10) | 0.2916(2) | 0.3159(5.5) | 0.3159(5.5) |
| Image | 0.7956(9) | 0.2231(5) | 0.2008(4) | 0.2470(6) | 0.5000(8) | 0.8892(10) | 0.3139(7) | **0.1352**(1) | 0.1402(2.5) | 0.1402(2.5) |
| Scene | 0.5992(8) | 0.1070(5) | 0.1059(4) | 0.1915(7) | 0.6556(9) | 0.8609(10) | 0.1838(6) | **0.0577**(1) | 0.0612(2.5) | 0.0612(2.5) |
| Enron | 0.3738(6) | 0.1776(2) | **0.1201**(1) | 0.4493(7) | 0.4741(8) | 0.9621(10) | 0.8563(9) | 0.3076(3) | 0.3126(4.5) | 0.3126(4.5) |
| Corel5k | 0.2695(2) | 0.4145(3) | **0.2672**(1) | 0.5101(9) | 0.5000(8) | 0.4990(7) | 0.9444(10) | 0.4303(4) | 0.4436(5.5) | 0.4436(5.5) |
| Bibtex | 0.4764(7) | 0.2037(4) | 0.2427(6) | 0.2199(5) | 0.5012(8) | 0.6954(9) | 0.7416(10) | **0.1017**(2) | **0.1017**(2) | **0.1017**(2) |
| Average rank | 6.4(6) | 3.1(3) | 2.8(2) | 6.8(7) | 8.3(8) | 9.0(10) | 8.6(9) | **2.4**(1) | 3.8(4.5) | 3.8(4.5) |

TABLE V
PERFORMANCE COMPARISON OF 10 ALGORITHMS ON 10 DATASETS [44] USING ONE ERROR ↓

| Algorithms | BP-MLL | MLNB | ML-kNN | ML$^2$ | LDSVR | CPNN | AA-KNN | MDLRML | MDLRML-2 | MDLRML-3 |
|---|---|---|---|---|---|---|---|---|---|---|
| Yeast | 0.7034(10) | 0.2475(6) | 0.2454(5) | 0.4286(7.5) | 0.4286(7.5) | **0.0714**(2.5) | 0.4999(9) | **0.0714**(2.5) | **0.0714**(2.5) | **0.0714**(2.5) |
| Emotions | 0.7022(10) | 0.4100(5) | 0.4213(6) | 0.6667(8.5) | 0.6667(8.5) | **0.3333**(2.5) | 0.4899(7) | **0.3333**(2.5) | **0.3333**(2.5) | **0.3333**(2.5) |
| Medical | 0.4024(6) | 0.4324(8) | 0.2583(2) | 0.4737(9) | 0.5000(10) | 0.4290(7) | **0.1579**(1) | 0.3684(4) | 0.3684(4) | 0.3684(4) |
| Cal500 | 0.1071(5.5) | 0.1111(7) | 0.1071(5.5) | 0.0827(4) | 0.8563(10) | 0.3333(8) | 0.5862(9) | **0.0747**(1) | 0.0753(2.5) | 0.0752(2.5) |
| Birds | 0.7989(8) | 0.5287(6) | 0.7126(7) | 0.9474(10) | 0.4990(5) | 0.8421(9) | 0.4737(4) | **0.3396**(1) | 0.3421(2.5) | 0.3421(2.5) |
| Image | 0.6710(10) | 0.4030(6) | 0.3630(5) | 0.2000(3.5) | 0.5000(8) | 0.5470(9) | 0.4990(7) | 0.2000(3.5) | **0**(1.5) | **0**(1.5) |
| Scene | 0.8269(10) | 0.3002(5) | 0.2575(3) | 0.6667(9) | 0.4999(7) | 0.3333(6) | 0.5000(8) | 0.3000(4) | **0**(1.5) | **0**(1.5) |
| Enron | 0.2642(4) | 0.5009(7) | 0.4076(5) | 0.8846(9) | 0.9615(10) | 0.5050(8) | 0.4808(6) | **0.0639**(1) | 0.0692(2.5) | 0.0692(2.5) |
| Corel5k | 0.9716(10) | 0.8868(6) | 0.7856(5) | 0.9564(9) | 0.4890(3) | **0.4400**(1) | 0.4419(2) | 0.7680(4) | 0.9390(7.5) | 0.9390(7.5) |
| Bibtex | 0.4547(4) | 0.5681(5) | 0.6363(6) | 0.6792(7) | 0.9497(9) | 0.9874(10) | 0.7688(8) | **0.3322**(1) | 0.3396(2.5) | 0.3396(2.5) |
| Average rank | 7.75(9) | 6.1(5.5) | 4.95(4) | 7.65(8) | 7.8(10) | 6.3(7) | 6.1(5.5) | **2.45**(1) | 2.95(2.5) | 2.95(2.5) |

TABLE VI
PERFORMANCE COMPARISON OF 10 ALGORITHMS ON 10 DATASETS [44] USING COVERAGE ↓

| Algorithms | BP-MLL | MLNB | ML-kNN | ML$^2$ | LDSVR | CPNN | AA-KNN | MDLRML | MDLRML-2 | MDLRML-3 |
|---|---|---|---|---|---|---|---|---|---|---|
| Yeast | 0.8990(10) | 0.6629(2) | **0.6385**(1) | 0.8950(8) | 0.8982(9) | 0.8845(7) | 0.8794(6) | 0.8620(3) | 0.8725(4.5) | 0.8725(4.5) |
| Emotions | 0.3089(10) | 0.1960(8) | 0.2320(9) | 0.1723(7) | 0.1568(4) | 0.1703(6) | 0.1661(5) | **0.1480**(1) | 0.1523(2.5) | 0.1523(2.5) |
| Medical | 0.2955(8) | 0.1934(5) | 0.3564(9) | 0.7684(10) | 0.2087(7) | 0.2081(6) | 0.1374(4) | 0.0590(3) | **0.0526**(1.5) | **0.0526**(1.5) |
| Cal500 | 1.3386(10) | **0.1346**(1) | 1.3045(9) | 0.2313(6) | 0.2284(3) | 0.2316(8) | 0.2315(7) | 0.2270(2) | 0.2287(5) | 0.2286(4) |
| Birds | 0.4415(10) | 0.2695(3) | 0.3606(9) | 0.3031(8) | 0.3014(7) | **0.2309**(1) | 0.2899(6) | 0.2660(2) | 0.2809(4.5) | 0.2809(4.5) |
| Image | 2.1460(10) | 1.1700(9) | 1.0760(8) | 0.9962(7) | 0.9608(2) | 0.9648(4) | 0.9644(3) | **0.9350**(1) | 0.9686(5.5) | 0.9686(5.5) |
| Scene | 2.0761(10) | 0.6296(4) | 0.6405(5) | 1.0617(7) | 1.0843(9) | 1.0773(8) | 1.0505(6) | **0.1870**(1) | 0.1990(2.5) | 0.1990(2.5) |
| Enron | 0.2369(2) | **0.2313**(1) | 1.6046(10) | 0.5029(9) | 0.4936(6) | 0.5028(8) | 0.4956(7) | 0.4390(3) | 0.4405(4.5) | 0.4405(4.5) |
| Corel5k | 0.1980(5) | 0.2062(7) | 0.1983(6) | 0.1912(4) | 1.5023(8.5) | 1.5023(8.5) | 1.5126(10) | **0.1830**(1) | 0.1836(2.5) | 0.1836(2.5) |
| Bibtex | 0.7356(10) | 0.3788(8) | 0.6146(9) | 0.3513(5) | 0.3382(4) | 0.3598(7) | 0.3585(6) | **0.1980**(1) | 0.2632(2.5) | 0.2632(2.5) |
| Average rank | 8.5(10) | 4.8(4) | 7.5(9) | 7.1(8) | 5.95(5) | 6.35(7) | 6.0(6) | **1.8**(1) | 3.55(3) | 3.45(2) |

TABLE VII
PERFORMANCE COMPARISON OF 10 ALGORITHMS ON 10 DATASETS [44] USING AVERAGE PRECISION ↑

| Algorithms | BP-MLL | MLNB | ML-kNN | ML$^2$ | LDSVR | CPNN | AA-KNN | MDLRML | MDLRML-2 | MDLRML-3 |
|---|---|---|---|---|---|---|---|---|---|---|
| Yeast | 0.4297(8) | 0.7481(3) | 0.7566(2) | 0.4366(7) | 0.3965(9) | 0.3064(10) | 0.4779(6) | **0.8316**(1) | 0.5047(4.5) | 0.5047(4.5) |
| Emotions | 0.5161(6) | 0.7324(2) | 0.6897(3) | 0.4442(9) | 0.4900(8) | 0.3123(10) | 0.4926(7) | **0.7831**(1) | 0.6498(4.5) | 0.6498(4.5) |
| Medical | 0.2081(8) | 0.6080(2) | **0.7898**(1) | 0.4683(6) | 0.0480(9) | 0.0467(10) | 0.3692(7) | 0.5748(5) | 0.6011(3.5) | 0.6011(3.5) |
| Cal500 | 0.4783(5) | 0.4372(6) | 0.4882(4) | 0.1644(9) | 0.1676(8) | 0.1598(10) | 0.1705(7) | **0.7883**(1) | 0.7862(2.5) | 0.7862(2.5) |
| Birds | 0.2460(4) | 0.5423(2) | 0.3875(3) | 0.1407(7) | 0.0759(10) | 0.1013(9) | 0.1131(8) | **0.6841**(1) | 0.1542(5.5) | 0.1542(5.5) |
| Image | 0.5111(7) | 0.7386(2) | **0.7649**(1) | 0.4905(8) | 0.2729(9) | 0.2645(10) | 0.5954(6) | 0.7258(3) | 0.7219(4.5) | 0.7219(4.5) |
| Scene | 0.4200(8) | 0.8191(5) | 0.8378(2) | 0.3078(9) | 0.7859(6) | 0.2954(10) | 0.7649(7) | **0.8394**(1) | 0.8354(3.5) | 0.8354(3.5) |
| Enron | 0.2057(4) | 0.2135(2) | **0.5509**(1) | 0.1234(7) | 0.0747(10) | 0.0828(9) | 0.1201(8) | 0.2130(3) | 0.2019(5.5) | 0.2019(5.5) |
| Corel5k | 0.2012(6) | 0.2200(5) | 0.1929(7) | 0.2930(4) | 0.0141(9.5) | 0.0141(9.5) | 0.0252(8) | **0.3470**(1) | 0.3340(2.5) | 0.3340(2.5) |
| Bibtex | 0.0659(8) | **0.3874**(1) | 0.3057(6) | 0.3872(2) | 0.0226(9) | 0.0182(10) | 0.1111(7) | 0.3637(5) | 0.3862(3.5) | 0.3862(3.5) |
| Average rank | 6.4(6) | 3.0(2.5) | 3.0(2.5) | 6.8(7) | 8.75(9) | 9.75(10) | 7.1(8) | **2.2**(1) | 4.0(4.5) | 4.0(4.5) |

similar. Similarly to [16], we set $\lambda = 1$ and $\epsilon = 0.01$. We set $\alpha = \beta = 0.5$ to maintain the balance of the two loss terms in (16). The termination threshold is set to $\xi = 10^{-5}$.

We used 10-fold cross-validation on each dataset, and we record each algorithm's average performance on the five MLL evaluation metrics in Tables III to VII, respectively, where bold-font value indicates the best performance among the algorithms. The entries of each row in a table record the metric values achieved by the corresponding algorithms, where the bracketed numbers indicate the ranks achieved for this dataset. For example, for the row corresponding to the dataset Yeast in Table III, the Hamming loss metric entry for the MLNB is 0.2061(3). This indicates that the MLNB attains the Hamming loss value of 0.2061 and ranks the second best for Yeast. From the experimental results, we see that on the regular-sized and large-scale datasets, our MDLRML ranks first in more than half of the evaluation metrics. The effectiveness of our MDLRML is especially evident on large-scale datasets. The last row of each table compares the average ranks of the 10 algorithms over all the 10 datasets for the corresponding evaluation metric. It can be seen that our MDLRML consistently comes out top.

As expected, the MDLRML outperforms the MDLRML-2 and MDLRML-3 on every metric over all the datasets. It can also be seen that the MDLRML-2 and MDLRML-3 achieve the same performance. This demonstrates the effectiveness of our proposed approach of combining the two loss terms.

*2) Computational time comparison:* Table VIII compares the run times measured in seconds of all the 10 algorithms. The experiments are carried out on Matlab 2016a, running on a PC with i5-6200 2.30 GHz processor of 4 cores and 8 GB of RAM. It can be seen that our MDLRML comes out on the top, and it consistently outperforms the other 7 MLL and LDL benchmark algorithms. Observe that the MDLRML attains faster run times than the MDLRML-2 on most of the datasets. Also the MDLRML-3 seems converging slower than the MDLRML-2 for all these 10 datasets, as it imposes higher run times than the latter. The results of run time comparison

TABLE VIII
RUN TIME (S) ↓ COMPARISON OF 10 ALGORITHMS ON 10 DATASETS [44]

| Algorithms | BP-MLL | MLNB | ML-kNN | $\text{ML}^2$ | LDSVR | CPNN | AA-KNN | MDLRML | MDLRML-2 | MDLRML-3 |
|---|---|---|---|---|---|---|---|---|---|---|
| Yeast | 50.0773(10) | 1.3516(7) | 0.4063(4) | 0.4836(5) | 7.4243(9) | 4.2781(8) | 1.1855(6) | **0.1250**(1) | 0.1563(2) | 0.3937(3) |
| Emotions | 1.1157(10) | 0.3014(7) | 0.1094(5) | 0.0005(2) | 0.1430(6) | 0.4149(9) | 0.06612(4) | **0.0001**(1) | 0.0025(3) | 0.3342(8) |
| Medical | 62.7626(10) | 0.3320(5) | 0.2500(4) | 0.1716(3) | 0.6256(7) | 21.7916(9) | 1.6063(8) | **0.0781**(1) | 0.0938(2) | 0.3534(6) |
| Cal500 | 30.9375(10) | 0.3175(8) | 0.2969(6) | 0.1404(5) | 0.0528(4) | 12.9231(9) | 0.0453(2) | **0.0238**(1) | 0.0469(3) | 0.3336(7) |
| Birds | 22.1667(10) | 0.7372(8) | 0.0938(5) | 0.0469(3) | 0.0566(4) | 2.0752(9) | 0.1276(6) | **0.0001**(1.5) | **0.0001**(1.5) | 0.3547(7) |
| Image | 34.1562(10) | 0.8424(7) | 0.3438(5) | 0.2500(3) | 0.4310(6) | 2.3032(9) | 1.2883(8) | 0.1092(2) | **0.0938**(1) | 0.3332(4) |
| Scene | 36.7543(10) | 0.3126(3) | 0.4063(6) | 0.3438(5) | 1.1168(7) | 2.5011(9) | 2.2282(8) | **0.1094**(1) | 0.1875(2) | 0.3366(4) |
| Enron | 85.6709(10) | 0.3190(4) | 0.4219(6) | 0.1875(3) | 1.4750(7) | 44.0872(9) | 2.7003(8) | 0.1719(2) | **0.1250**(1) | 0.3464(5) |
| Corel5k | 264.4987(9) | 7.0152(6) | 3.8750(4) | 1.5444(3) | 4.5927(5) | 285.8024(10) | 13.5505(7) | 0.8125(2) | **0.7656**(1) | 179.0355(8) |
| Bibtex | 326.5432(8) | 8.2442(5) | 7.2656(4) | 4.5625(3) | 37.9287(6) | 416.3579(9) | 255.3396(7) | **2.7500**(1) | 2.9063(2) | 468.1460(10) |
| Average rank | 9.7(10) | 6.0(5) | 4.9(4) | 3.5(3) | 6.1(6) | 9.0(9) | 6.4(8) | **1.35**(1) | 1.85(2) | 6.2(7) |

therefore indicates that the MDLRML typically converges faster than both the MDLRML-2 and MDLRML-3.

TABLE IX
FRIEDMAN STATISTICS $F_F$, IN TERMS OF EACH EVALUATION METRIC AND THE CRITICAL VALUE AT A SIGNIFICANCE LEVEL OF 0.05 (COMPARING ALGORITHMS 8, DATASETS 10)

| Evaluation metric | $F_F$ | Critical value |
|---|---|---|
| Hamming loss | 20.4255 | 2.690 |
| Ranking loss | 30.6226 | |
| One error | 4.5484 | |
| Coverage | 5.0281 | |
| Average precision | 30.5232 | |
| Run time (s) | 30.4490 | |

*3) Friedman test and critical difference diagram:* Friedman test statistically compares relative performance among multiple algorithms over a number of datasets [46]. We use Friedman test to statistically compare the performance of our MDLRML and the other 7 benchmark MLL and LDL algorithms on the 10 datasets. Table IX shows the Friedman statistic $F_F$ and critical value on each evaluation metric at a significance level of 0.05, among the eight comparison algorithms and 10 datasets.

As confirmed in Table IX, the $F_F$ values on all the evaluation metrics are greater than the critical value. Therefore, Bonferroni-Dunn test [46] can be employed as a post hoc test to show the algorithms' relative performances. Specifically, based on Table IX, we use Nemenyi test [46] to check the average ordering comparison between two algorithms. Figs. 4 to 9 represent these results with a critical difference (CD) graph for each evaluation metric, respectively. When the significance level is 0.05, the number of comparison algorithms is 8, and the number of datasets is 10, the CD value is CD = 2.9467 for Nemenyi test. In the CD diagram, the average ordering of each algorithm is marked on the same coordinate axis. If the difference between the average order of the two algorithms is less than the CD value, then there is no significant difference between the two algorithms and they will be connected by a line segment in the CD graph. Algorithms not connected with the MDLRML in the CD diagrams are considered to have significant performance difference from the control algorithm, given the CD value of 2.9467 at a significance level of 0.05.

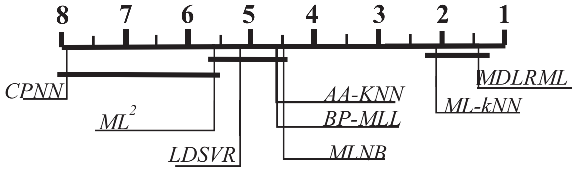


Fig. 4. CD diagrams given CD = 2.9467 of Nemenyi tests on the eight algorithms for Hamming loss evaluation metric

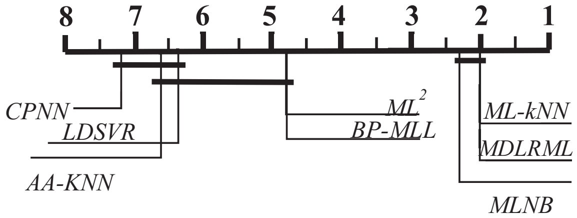


Fig. 5. CD diagrams given CD = 2.9467 of Nemenyi tests on the eight algorithms for ranking loss evaluation metric

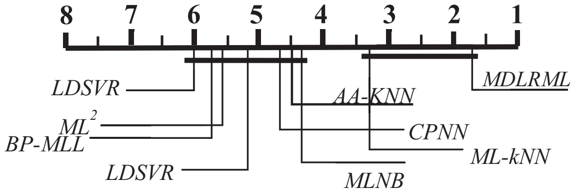


Fig. 6. CD diagrams given CD = 2.9467 of Nemenyi tests on the eight algorithms for one error evaluation metric

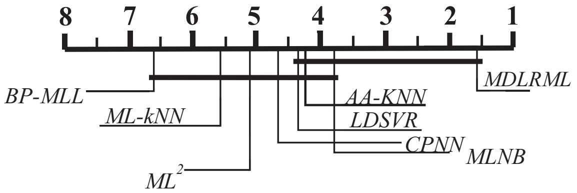


Fig. 7. CD diagrams given CD = 2.9467 of Nemenyi tests on the eight algorithms for coverage evaluation metric

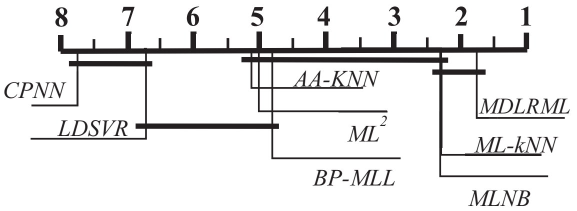


Fig. 8. CD diagrams given CD = 2.9467 of Nemenyi tests on the eight algorithms for average precision evaluation metric

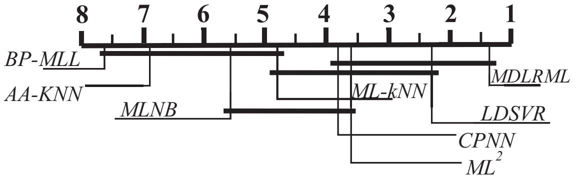


Fig. 9. CD diagrams given CD = 2.9467 of Nemenyi tests on the eight algorithms for run time (s) evaluation metric

### C. Experimental result comparison based on LDL metrics

Experimental results of the six LDL algorithms, i.e., the bechmarks LDSVR, CPNN and AA-KNN as well as our MDLRML, MDLRML-2 and MDLRML-3, applied to the 15 real-world datasets of [3] are compared in Tables X to XV for the six evaluation metrics, respectively, where we calculate the corresponding algorithms' average ranks over the 15 datasets in the last row of each table. The results show that our MDLRML consistently performs best among the 6 LDL algorithms for all the 6 measures. From the average ranks given in Tables X to XV, we can infer that the algorithms' rankings over the six evaluation indicators are as follows

$$\begin{aligned}\text{MDLRML} &\succ \text{MDLRML-2} \approx \text{MDLRML-3} \\ &\succ \text{LDSVR} \approx \text{CPNN} \succ \text{AA-KNN}.\end{aligned}$$

The results of the run time comparison for all the six algorithms over the 15 real-world datasets of [3] are presented in Table XVI. In addition to achieve the best LDL performance, our MDLRML also has the fastest run time. Observe that the MDLRML is significantly faster than both the MDLRML-2

TABLE X
EXPERIMENTAL RESULTS OF 6 LDL ALGORITHMS ON 15 DATASETS [3] MEASURED BY CHEBYSHEV DISTANCE ↓

| Algorithms | LDSVR | CPNN | AA-KNN | MDLRML | MDLRML-2 | MDLRML-3 |
|---|---|---|---|---|---|---|
| Yeast-alpha | 0.0080(5) | **0.0073**(2.5) | 0.0090(6) | **0.0073**(2.5) | **0.0073**(2.5) | **0.0073**(2.5) |
| Yeast-cdc | 0.0141(5) | 0.0138(4) | 0.0229(6) | **0.0073**(1) | 0.0083(2.5) | 0.0083(2.5) |
| Yeast-elu | 0.0267(6) | 0.0262(5) | 0.0189(4) | **0.0076**(2) | **0.0076**(2) | **0.0076**(2) |
| Yeast-diau | 0.0414(4) | 0.0505(5) | 0.0572(6) | **0.0205**(2) | **0.0205**(2) | **0.0205**(2) |
| Yeast-heat | 0.0481(4) | 0.0504(5) | 0.0669(6) | **0.0111**(2) | **0.0111**(2) | **0.0111**(2) |
| Yeast-spo | 0.1774(4) | 0.1969(6) | 0.1830(5) | **0.0277**(2) | **0.0277**(2) | **0.0277**(2) |
| Yeast-cold | 0.0773(5) | 0.0727(4) | 0.0833(6) | **0.0174**(2) | **0.0174**(2) | **0.0174**(2) |
| Yeast-dtt | 0.0417(4) | 0.0436(5) | 0.0543(6) | **0.0080**(1) | 0.0084(2.5) | 0.0084(2.5) |
| Yeast-spo5 | 0.1722(4) | 0.1751(5) | 0.2006(6) | **0.0531**(1) | 0.0539(2.5) | 0.0539(2.5) |
| Yeast-spoem | 0.3369(6) | 0.2869(5) | 0.1512(4) | **0.0096**(2) | **0.0096**(2) | **0.0096**(2) |
| Human Gene | 0.0179(5) | 0.0202(6) | 0.0140(4) | **0.0130**(2) | **0.0130**(2) | **0.0130**(2) |
| Natural Scene | 0.4045(6) | 0.1877(4) | 0.2473(5) | **0.1317**(1.5) | **0.1317**(1.5) | 0.1334(3) |
| Movie | 0.1697(6) | 0.1420(4) | 0.1544(5) | **0.1418**(2) | **0.1418**(2) | **0.1418**(2) |
| SJAFFE | 0.1300(5) | 0.1416(6) | 0.1183(4) | **0.0566**(1.5) | **0.0566**(1.5) | 0.0728(3) |
| SBU_3DFE | 0.1790(5) | 0.1745(4) | 0.2229(6) | **0.0794**(2) | **0.0794**(2) | **0.0794**(2) |
| Average rank | 4.93(5) | 4.7(4) | 5.27(6) | **1.77**(1) | 2.07(2) | 2.27(3) |

TABLE XI
EXPERIMENTAL RESULTS OF 6 LDL ALGORITHMS ON 15 DATASETS [3] MEASURED BY CLARK DISTANCE ↓

| Algorithms | LDSVR | CPNN | AA-KNN | MDLRML | MDLRML-2 | MDLRML-3 |
|---|---|---|---|---|---|---|
| Yeast-alpha | 0.1713(6) | 0.1141(4) | 0.1262(5) | **0.1027**(2) | **0.1027**(2) | **0.1027**(2) |
| Yeast-cdc | 0.2178(5) | 0.2005(4) | 0.2857(6) | **0.1041**(1) | 0.1179(2) | 0.1205(3) |
| Yeast-elu | 0.2494(6) | 0.2351(4) | 0.2386(5) | **0.1013**(2) | **0.1013**(2) | **0.1013**(2) |
| Yeast-diau | 0.2890(4) | 0.3287(5) | 0.3393(6) | **0.0913**(2) | **0.0913**(2) | **0.0913**(2) |
| Yeast-heat | 0.2534(5) | 0.2492(4) | 0.3438(6) | **0.0535**(2) | **0.0535**(2) | **0.0535**(2) |
| Yeast-spo | 0.4690(4) | 0.5093(6) | 0.4928(5) | **0.1221**(2) | **0.1221**(2) | **0.1221**(2) |
| Yeast-cold | 0.1979(4) | 0.2077(5) | 0.2392(6) | **0.0424**(2) | **0.0424**(2) | **0.0424**(2) |
| Yeast-dtt | 0.1244(4) | 0.1366(6) | 0.1363(5) | **0.0191**(1) | 0.0200(2.5) | 0.0200(2.5) |
| Yeast-spo5 | 0.4098(4) | 0.4233(5) | 0.4501(6) | **0.1023**(1) | 0.1038(2.5) | 0.1038(2.5) |
| Yeast-spoem | 0.5671(6) | 0.5117(5) | 0.3180(4) | **0.0136**(2) | **0.0136**(2) | **0.0136**(2) |
| Human Gene | 1.0551(4) | 1.0739(5) | 1.3913(6) | **0.9659**(2) | **0.9659**(2) | **0.9659**(2) |
| Natural Scene | 2.3735(6) | 2.1469(5) | 2.1008(4) | **2.0868**(1) | 2.0936(2.5) | 2.0936(2.5) |
| Movie | 0.7311(6) | 0.6558(5) | 0.5541(4) | **0.4354**(1) | 0.4355(2.5) | 0.4355(2.5) |
| SJAFFE | 0.3860(5) | 0.5163(6) | 0.3324(4) | **0.2365**(1.5) | **0.2365**(1.5) | 0.2464(3) |
| SBU_3DFE | 0.4379(4) | 0.4457(5) | 0.5710(6) | **0.2560**(2) | **0.2560**(2) | **0.2560**(2) |
| Average rank | 4.87(4) | 4.93(5) | 5.20(6) | **1.63**(1) | 2.1(2) | 2.27(3) |

TABLE XII
EXPERIMENTAL RESULTS OF 6 LDL ALGORITHMS ON 15 DATASETS [3] MEASURED BY CANBERRA DISTANCE ↓

| Algorithms | LDSVR | CPNN | AA-KNN | MDLRML | MDLRML-2 | MDLRML-3 |
|---|---|---|---|---|---|---|
| Yeast-alpha | 0.5686(6) | 0.3792(4) | 0.4057(5) | **0.3328**(2) | **0.3328**(2) | **0.3328**(2) |
| Yeast-cdc | 0.7528(5) | 0.6780(4) | 0.9368(6) | **0.3179**(1) | 0.3227(2) | 0.4268(3) |
| Yeast-elu | 0.7263(6) | 0.7003(5) | 0.6932(4) | **0.3028**(2) | **0.3028**(2) | **0.3028**(2) |
| Yeast-diau | 0.6489(4) | 0.7181(5) | 0.7262(6) | **0.1637**(2) | **0.1637**(2) | **0.1637**(2) |
| Yeast-heat | 0.5499(5) | 0.5282(4) | 0.7659(6) | **0.1124**(2) | **0.1124**(2) | **0.1124**(2) |
| Yeast-spo | 0.9634(4) | 1.0667(6) | 0.9917(5) | **0.2344**(2) | **0.2344**(2) | **0.2344**(2) |
| Yeast-cold | 0.3574(4) | 0.3990(5) | 0.4401(6) | **0.0707**(2) | **0.0707**(2) | **0.0707**(2) |
| Yeast-dtt | 0.2436(5) | 0.2709(6) | 0.2184(4) | **0.0319**(1) | 0.0334(2.5) | 0.0334(2.5) |
| Yeast-spo5 | 0.5826(4) | 0.5843(5) | 0.6554(6) | **0.1623**(1) | 0.1647(2.5) | 0.1647(2.5) |
| Yeast-spoem | 0.7600(6) | 0.6749(5) | 0.3972(4) | **0.0192**(2) | **0.0192**(2) | **0.0192**(2) |
| Human Gene | 6.2419(3) | **6.1261**(1) | 9.6774(6) | 6.1410(2) | 6.5104(4.5) | 6.5104(4.5) |
| Natural Scene | 6.8702(6) | 5.5629(5) | **5.2013**(1) | 5.3370(2) | 5.3707(3.5) | 5.3707(3.5) |
| Movie | 1.2497(5) | 1.2530(6) | 0.9741(4) | **0.7995**(1) | 0.7996(2.5) | 0.7996(2.5) |
| SJAFFE | 0.7755(5) | 1.0884(6) | 0.6243(4) | **0.4796**(1.5) | 0.5541(3) | **0.4796**(1.5) |
| SBU_3DFE | 0.9382(4) | 0.9765(5) | 1.1770(6) | **0.5488**(2) | **0.5488**(2) | **0.5488**(2) |
| Average rank | 4.8(4.5) | 4.8(4.5) | 4.87(6) | **1.7**(1) | 2.43(3) | 2.4(2) |

TABLE XIII
Experimental results of 6 LDL algorithms on 15 datasets [3] measured by Kullback-Leibler divergence ↓

| Algorithms | LDSVR | CPNN | AA-KNN | MDLRML | MDLRML-2 | MDLRML-3 |
|---|---|---|---|---|---|---|
| Yeast-alpha | 0.0030(6) | 0.0015(4) | 0.0018(5) | **0.0012**(2) | **0.0012**(2) | **0.0012**(2) |
| Yeast-cdc | 0.0064(5) | 0.0054(4) | 0.0111(6) | **0.0014**(1) | 0.0018(2) | 0.0020(3) |
| Yeast-elu | 0.0094(6) | 0.0085(5) | 0.0080(4) | **0.0015**(2) | **0.0015**(2) | **0.0015**(2) |
| Yeast-diau | 0.0198(4) | 0.0258(5) | 0.0273(6) | **0.0022**(2) | **0.0022**(2) | **0.0022**(2) |
| Yeast-heat | 0.0193(5) | 0.0187(4) | 0.0367(6) | **0.0009**(2) | **0.0009**(2) | **0.0009**(2) |
| Yeast-spo | 0.0952(4) | 0.1182(6) | 0.1047(5) | **0.0050**(2) | **0.0050**(2) | **0.0050**(2) |
| Yeast-cold | 0.0201(4) | 0.0219(5) | 0.0290(6) | **0.0008**(2) | **0.0008**(2) | **0.0008**(2) |
| Yeast-dtt | 0.0077(4) | 0.0092(5) | 0.0096(6) | **0.0001**(1) | 0.0002(2.5) | 0.0002(2.5) |
| Yeast-spo5 | 0.0836(4) | 0.0934(5) | 0.1144(6) | **0.0066**(1) | 0.0068(2.5) | 0.0068(2.5) |
| Yeast-spoem | 0.2484(6) | 0.1858(5) | 0.0538(4) | **0.0001**(2) | **0.0001**(2) | **0.0001**(2) |
| Human Gene | 0.0355(4.5) | 0.0355(4.5) | 0.0594(6) | **0.0284**(2) | **0.0284**(2) | **0.0284**(2) |
| Natural Scene | 1.4482(6) | 0.6021(4) | 0.6874(5) | **0.5689**(1) | 0.5884(2.5) | 0.5884(2.5) |
| Movie | 0.0955(4) | 0.1190(6) | 0.0994(5) | **0.0756**(2) | **0.0756**(2) | **0.0756**(2) |
| SJAFFE | 0.0619(5) | 0.1073(6) | 0.0473(4) | **0.0200**(1.5) | **0.0200**(1.5) | 0.0247(3) |
| SBU_3DFE | 0.0925(5) | 0.0809(4) | 0.1625(6) | **0.0250**(2) | **0.0250**(2) | **0.0250**(2) |
| Average rank | 4.83(4.5) | 4.83(4.5) | 5.33(6) | **1.70**(1) | 2.07(2) | 2.23(3) |

TABLE XIV
Experimental results of 6 LDL algorithms on 15 datasets [3] measured by cosine coefficient ↑

| Algorithms | LDSVR | CPNN | AA-KNN | MDLRML | MDLRML-2 | MDLRML-3 |
|---|---|---|---|---|---|---|
| Yeast-alpha | 0.9971(6) | 0.9985(4) | 0.9981(5) | **0.9988**(2) | **0.9988**(2) | **0.9988**(2) |
| Yeast-cdc | 0.9935(5) | 0.9946(4) | 0.9893(6) | **0.9982**(1.5) | **0.9982**(1.5) | 0.9980(3) |
| Yeast-elu | 0.9904(6) | 0.9913(5) | 0.9921(4) | **0.9985**(2) | **0.9985**(2) | **0.9985**(2) |
| Yeast-diau | 0.9832(4) | 0.9778(5) | 0.9766(6) | **0.9978**(2) | **0.9978**(2) | **0.9978**(2) |
| Yeast-heat | 0.9821(5) | 0.9826(4) | 0.9648(6) | **0.9990**(2) | **0.9990**(2) | **0.9990**(2) |
| Yeast-spo | 0.9032(4) | 0.8815(6) | 0.8944(5) | **0.9950**(2) | **0.9950**(2) | **0.9950**(2) |
| Yeast-cold | 0.9803(4) | 0.9785(5) | 0.9710(6) | **0.9991**(2) | **0.9991**(2) | **0.9991**(2) |
| Yeast-dtt | 0.9925(4) | 0.9909(5) | 0.9903(6) | **0.9998**(2) | **0.9998**(2) | **0.9998**(2) |
| Yeast-spo5 | 0.9310(4) | 0.9268(5) | 0.9075(6) | **0.9935**(1) | 0.9933(2.5) | 0.9933(2.5) |
| Yeast-spoem | 0.8293(6) | 0.8808(5) | 0.9556(4) | **0.9998**(2) | **0.9998**(2) | **0.9998**(2) |
| Human Gene | 0.9647(4) | 0.9639(5) | 0.9420(6) | **0.9717**(2) | **0.9717**(2) | **0.9717**(2) |
| Natural Scene | 0.4386(6) | 0.7315(4) | 0.6838(5) | **0.7555**(1) | 0.7408(2.5) | 0.7408(2.5) |
| Movie | **0.9371**(1) | 0.9225(5) | 0.9205(6) | 0.9301(2) | 0.9300(3.5) | 0.9300(3.5) |
| SJAFFE | 0.9360(5) | 0.8879(6) | 0.9482(4) | **0.9797**(1.5) | **0.9797**(1.5) | 0.9735(3) |
| SBU_3DFE | 0.9018(5) | 0.9147(4) | 0.8432(6) | **0.9740**(2) | **0.9740**(2) | **0.9740**(2) |
| Average rank | 4.6(4) | 4.8(5) | 5.4(6) | **1.8**(1) | 2.1(2) | 2.3(3) |

TABLE XV
Experimental results of 6 LDL algorithms on 15 datasets [3] measured by intersectional similarity ↑

| Algorithms | LDSVR | CPNN | AA-KNN | MDLRML | MDLRML-2 | MDLRML-3 |
|---|---|---|---|---|---|---|
| Yeast-alpha | 0.9694(6) | 0.9789(4) | 0.9771(5) | **0.9815**(2) | **0.9815**(2) | **0.9815**(2) |
| Yeast-cdc | 0.9495(5) | 0.9546(4) | 0.9378(6) | **0.9786**(1.5) | **0.9786**(1.5) | 0.9715(3) |
| Yeast-elu | 0.9480(6) | 0.9488(5) | 0.9502(4) | **0.9783**(2) | **0.9783**(2) | **0.9783**(2) |
| Yeast-diau | 0.9136(4) | 0.9037(5) | 0.9024(6) | **0.9771**(2) | **0.9771**(2) | **0.9771**(2) |
| Yeast-heat | 0.9119(5) | 0.9154(4) | 0.8753(6) | **0.9812**(2) | **0.9812**(2) | **0.9812**(2) |
| Yeast-spo | 0.8226(4) | 0.8031(6) | 0.8170(5) | **0.9610**(2) | **0.9610**(2) | **0.9610**(2) |
| Yeast-cold | 0.9108(4) | 0.9001(5) | 0.8887(6) | **0.9826**(2) | **0.9826**(2) | **0.9826**(2) |
| Yeast-dtt | 0.9394(5) | 0.9325(6) | 0.9444(4) | **0.9920**(1) | 0.9916(2.5) | 0.9916(2.5) |
| Yeast-spo5 | 0.8278(4) | 0.8249(5) | 0.7994(6) | **0.9469**(1) | 0.9461(2.5) | 0.9461(2.5) |
| Yeast-spoem | 0.6631(6) | 0.7131(5) | 0.8488(4) | **0.9904**(2) | **0.9904**(2) | **0.9904**(2) |
| Human Gene | 0.9071(3) | **0.9097**(1) | 0.8567(6) | 0.9095(2) | 0.9038(4.5) | 0.9038(4.5) |
| Natural Scene | 0.3522(6) | 0.5683(3) | 0.5008(5) | **0.5751**(1) | 0.5601(4) | 0.5684(2) |
| Movie | 0.8203(4) | 0.7973(6) | 0.8183(5) | **0.8406**(2) | **0.8406**(2) | **0.8406**(2) |
| SJAFFE | 0.8603(5) | 0.7990(6) | 0.8817(4) | **0.9141**(1.5) | 0.9053(3) | **0.9141**(1.5) |
| SBU_3DFE | 0.8210(5) | 0.8255(4) | 0.7771(6) | **0.9047**(2) | **0.9047**(2) | **0.9047**(2) |
| Average rank | 4.8(5) | 4.6(4) | 5.2(6) | **1.73**(1) | 2.4(3) | 2.27(2) |

and MDLRML-3 for these 15 datasets.

## V. Conclusions

A novel multilabel distribution learning algorithm has been proposed based on multi-output regression via manifold learning. In ourd MDLRML, manifold learning has been applied to multilabel data to map feature data onto a low-dimensional subspace. We have used the feature space manifold's topological structure to guide the label space manifold learning and to construct a smooth multi-output regression function. We have further enhanced the multi-output regression with sigmoid function to improve LDL. Experimental results have shown that this approach significantly improves the effectiveness of multilabel distribution learning. Specifically, employing a large number of real-world multilabel datasets and using several existing MLL and LDL algorithms as the benchmarks, we have demonstrated that our MDLRML not only achieves the best performance, in terms of a wide range of MLL and LDL metrics, but also has the fastest run time.

TABLE XVI
RUN TIME (S) ↓ COMPARISON OF 6 LDL ALGORITHMS ON 15 DATASETS [3]

| Algorithms | LDSVR | CPNN | AA-KNN | MDLRML | MDLRML2 | MDLRML3 |
|---|---|---|---|---|---|---|
| Yeast-alpha | 1.6316685(3) | 1.8254755(4) | 0.9816280(2) | **0.6749496**(1) | 3.7506492(6) | 3.5801652(5) |
| Yeast-cdc | 1.4815188(4) | 1.4599378(3) | 0.9854936(2) | **0.6171126**(1) | 2.8562506(5) | 2.8783142(6) |
| Yeast-elu | 1.1763668(3) | 1.3851158(4) | 0.9667879(2) | **0.7045028**(1) | 3.3418765(6) | 3.3094362(5) |
| Yeast-diau | 1.1227780(4) | 0.8722209(2) | 0.9620473(3) | **0.7018304**(1) | 2.4062197(6) | 2.3304593(5) |
| Yeast-heat | 1.1543420(4) | 0.7586278(2) | 0.9650402(3) | **0.5900261**(1) | 2.0426195(6) | 2.0198211(5) |
| Yeast-spo | **0.4129708**(1) | 0.7687076(5) | 1.0020879(6) | 0.6912406(4) | 0.6198215(3) | 0.6096032(2) |
| Yeast-cold | **0.3042440**(1) | 0.8562454(5) | 0.9423973(6) | 0.5843705(3) | 0.6082043(4) | 0.5694781(2) |
| Yeast-dtt | 1.0043086(4) | **0.5036830**(1) | 0.9282778(3) | 0.6135728(2) | 2.7701283(5) | 2.8128525(6) |
| Yeast-spo5 | 1.2346091(4) | **0.4690452**(1) | 0.9655296(3) | 0.6076371(2) | 2.6028464(5) | 2.6441552(6) |
| Yeast-spoem | **0.1616213**(1) | 0.3514490(2) | 0.9704378(6) | 0.6063446(3) | 0.6253336(4) | 0.6394916(5) |
| Human Gene | 97.8977045(3) | 61.2968199(2) | **50.1885618**(1) | 179.3913124(4) | 191.6530611(6) | 182.3504735(5) |
| Natural Scene | **0.4168384**(1) | 2.0211242(3) | 1.3799915(2) | 7.2305590(6) | 6.9770856(4) | 7.1842052(5) |
| Movie | 24.8605512(2) | 44.4023725(3) | 274.6507527(6) | **10.5005083**(1) | 270.2719719(5) | 255.3366652(4) |
| SJAFFE | 0.0121351(2) | 0.2062209(6) | 0.0155848(4) | 0.0152976(3) | 0.0221721(5) | **0.0102398**(1) |
| SBU_3DFE | **0.5213146**(1) | 1.9771731(5) | 1.9786696(6) | 0.6452705(3) | 0.7449729(4) | 0.6209796(2) |
| Average rank | 2.53(2) | 3.2(3) | 3.67(4) | **2.4**(1) | 4.93(6) | 4.27(5) |

## REFERENCES

[1] T. N. Rubin, A. Chambers, P. Smyth, and M. Steyvers, "Statistical topic models for multi-label document classification," *Machine Learning*, vol. 88, nos. 1-2, pp. 157-208, 2012.

[2] J. Wang, Y. Zhao, X. Wu, and X.-S. Hua, "A transductive multi-label learning approach for video concept detection," *Pattern Recognition*, vol. 44, nos. 10-11, pp. 2274-2286, 2011.

[3] X. Geng, "Label distribution learning," *IEEE Trans. Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1734-1748, Jul. 2016.

[4] X. Geng, K. Smith-Miles, and Z.-H. Zhou, "Facial age estimation by learning from label distributions," In *Proc. AAAI-10* (Atlanta, GA), Jul. 11-15, 2010, pp. 451-456.

[5] X. Geng, C. Yin, and Z.-H. Zhou, "Facial age estimation by learning from label distributions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 35, no. 10, pp. 2401-2412, Oct. 2013.

[6] X. Geng and P. Hou, "Pre-release prediction of crowd opinion on movies by label distribution learning," In *Proc. IJCAI 2015* (Buenos Aires, Argentina), Jul. 23-31, 2015, pp. 3511-3517.

[7] Y. Zhou, H. Xue, and X. Geng, "Emotion distribution recognition from facial expressions," In *Proc. 23rd Int. Conf. Multimedia* (Brisbane, Australia), Oct. 26-30, 2015, pp. 1247-1250.

[8] X. Geng and L. R. Luo, "Multilabel ranking with inconsistent rankers," In *Proc. CVPR 2014* (Columbus, OH), Jun. 23-282014, pp. 3742-3747.

[9] Z. Zhang, M. Wang, and X. Geng, "Crowd counting in public video surveillance by label distribution learning," *Neurocomputing*, vol. 166, pp. 151-163, 2015.

[10] X. Geng and M. G. Ling, "Soft video parsing by label distribution learning," In *Proc. AAAI-17* (San Francisco, CA), Feb. 4-9, 2017, pp. 1331-1337.

[11] D. Cai, X. F. He, and J. W. Han, "Semi-supervised discriminant analysis," In *Proc. IEEE 11th IEEE Int. Conf. Computer Vision* (Rio de Janeiro, Brazil), Oct. 14-21, 2007, pp. 1-7.

[12] H. Li, P. Li, Y. J. Guo, and M. Wu, "Multi-label dimensionality reduction based on semi-supervised discriminant analysis," *J. Central South Univ. Techno.*, vol. 17, no. 6, pp. 1310-1319, Dec. 2010.

[13] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323-2326, 2000.

[14] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373-1396, 2003.

[15] Z. Y. Zhang and H. Y. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM J. Scientific Computing*, vol. 26, no. 1, pp. 313-338, 2004.

[16] P. Hou, X. Geng, and M. Zhang, "Multi-label manifold learning," In *Proc. AAAI-16* (Phoenix, AZ), Feb. 12-17, 2016, pp. 1680-1686.

[17] N. Xu, A. Tao, and X. Geng, "Label enhancement for label distribution learning," In *Proc. IJCAI'18* (Stockholm, Sweden), Jul. 13-19, 2018, pp. 2926-2932.

[18] K. Yu, S. P. Yu, and V. Tresp, "Multi-label informed latent semantic indexing," In *Proc. 28th Annual Int. ACM SIGIR Conf. Research and Development in Information Retrieval* (Salvador, Brazil), Aug. 15-19, 2005, pp. 258-265.

[19] Y. Zhang and Z.-H. Zhou, "Multi-label dimensionality reduction via dependency maximization," In *Proc. AAAI-08* (Chicago, IL), Jul.13-17, 2008, pp. 1503-1505.

[20] C.-H. Park and M. Lee, "On applying linear discriminant analysis for multi-labeled problems," *Pattern Recognition Lett.*, vol. 29, no. 7, pp. 878-887, May 2008.

[21] J. Xu, "A weighted linear discriminant analysis framework for multi-label feature extraction," *Neurocomputing*, vol. 275, pp. 107-120, Jan. 2018.

[22] K. Φ. Mikalsen, C. Soguero-Ruiz, F. M. Bianchi, and R. Jenssen, "Noisy multi-label semi-supervised dimensionality reduction," *Pattern Recognition*, vol. 90, pp. 257-270, Jun. 2019.

[23] S. W. Ji, L. Tang, S. P. Yu, and J. P. Ye, "A shared-subspace learning framework for multi-label classification," *ACM Trans. Knowledge Discovery from Data*, vol. 4, no. 2, pp. 1-29, May. 2010.

[24] X. Geng, N. Xu, and R. F. Shao, "Label enhancement for label distribution learning," *J. Computer Research and Development*, vol. 54, no. 6, pp. 1171-1184, 2017.

[25] C.-L. Peng, A. Tao, and X. Geng, "Label embedding based on multi-scale locality preservation," In *Proc. IJCAI-18* (Stockholm, Sweden), Jul. 13-19,, 2018, pp. 2623-2629.

[26] X. J. Zhu, *Semi-Supervised Learning with Graphs*. PhD Thesis, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 2005.

[27] S. Xiang, F. Nie, C. Zhang, and C. Zhang, "Nonlinear dimensionality reduction with local spline embedding," *IEEE Trans. Knowledge and Data Eng.*, vol. 21, no. 9, pp. 1285-1298, Sep. 2009.

[28] F. Nie, *et al.*, "Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering," *IEEE Trans. Neural Networks*, vol. 22, no. 11, pp. 1796-1808, Nov. 2011.

[29] S. Xiang, F. Nie, C. Pan, and C. Zhang, "Regression reformulations of LLE and LTSA with locally linear transformation," *IEEE Trans. Systems, Man, and Cybernetics, Part B*, vol. 41, no. 5, pp. 1250-1262, Oct. 2011.

[30] F. Tai and H. T. Lin, "Multilabel classification with principal label space transformation," *Neural Computation*, vol. 24, no. 9, pp. 2508-2542, Sep. 2012.

[31] L. Sun, S. Ji, and J. Ye, "Canonical correlation analysis for

multilabel classification: A least-squares formulation, extensions, and analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 194-200, Jan. 2011.

[32] M.-L. Zhang and Z.-H. Zhou, "Multi-label neural networks with applications to functional genomics and text categorization," *IEEE Trans. Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338-1351, Oct. 2006.

[33] M.-L. Zhang, J. M. Pena, and V. Robles, "Feature selection for multi-label naive Bayes classification," *Information Sciences*, vol. 179, no. 19, pp. 3218-3229, 2009.

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

[35] E. Gibaja and S. Ventura, "A tutorial on multilabel learning," *ACM Computing Surveys*, vol. 47, no. 3, pp. 1-39, Apr. 2015.

[36] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowledge and Data Engineering*, vol. 26, no. 8, pp. 18191837, Aug. 2014.

[37] H. Borchani, G. Varando, C. Bielza, and P. Larranaga, "A survey on multi-output regression," *WIREs Data Mining and Knowledge Discovery*, vol. 5, no. 5, pp. 216233, Sep. 2015.

[38] J. Zeng, *et al.*, "Dimensionality reduction in multiple ordinal regression," *IEEE Trans. Neural Networks and Learning*, vol. 29, no. 9, pp. 4088-4101, Sep. 2018.

[39] Y.-K. Li, M.-L. Zhang, and X. Geng, "Leveraging implicit relative labeling-importance information for effective multi-label learning," In *Proc. 2015 IEEE Int. Conf. Data Mining* (Atlantic City, NJ), Nov. 14-17, 2015, pp. 251260.

[40] C. Tan and J. H. Guan, "A feature space alignment learning algorithm," In *Proc. PRICAI 2012* (Kuching, Malaysia), Sep. 3-7, 2012, pp. 795-800.

[41] F. Párez-Cruz, A. Navia-Vázquez, P. L. Alarcón-Diana, and A. Artés-Rodríguez, "An IRWLS procedure for SVR," In *Proc. 10th European Signal Processing Conf.* (Tampere, Finland), Sep. 4-8, 2000, pp. 1-4.

[42] D. Tuia, *et al.*, "Multioutput support vector regression for remote sensing biophysical parameter estimation," *IEEE Geoscience and Remote Sensing Lett.*, vol. 8, no. 4, pp. 804-808, Jul. 2011.

[43] F. Perez-Cruz, C. Bousono-Calzon and A. Artes-Rodiguez, "Convergence of the IRWLS procedure to the support vector machine solution," *Neural Computation*, vol. 17, no. 1, pp. 1-18, Jan. 2005.

[44] Mulan: A java library for multi-label learning. http://mulan.sourceforge.net/datasets-mlc.html, 2018-03-01.

[45] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819-1837, Aug. 2014.

[46] J. Demšar, "Statistical comparisons of classifiers over multiple datasets," *J. Machine Learning Research*, vol. 7, no. 1, pp. 1-30, 2006.

**Chao Tan** received the B.E. and M.E. degree in Computer Science and Technology from Southeast University in 2005 and 2009, respectively, and received the PhD degree in Computer Science and Technology from Tongji University in 2015. She joined the Nanjing Normal University as a lecturer in 2015 and is an associate professor at present. She is now a postdoctoral researcher in Southeast University. Her research interests generally focus on machine learning, multi-label manifold learning and data mining.

**Sheng Chen** (M'90-SM'97-F'08) received his BEng degree from the East China Petroleum Institute, Dongying, China, in 1982, and his PhD degree from the City University, London, in 1986, both in control engineering. In 2005, he was awarded the higher doctoral degree, Doctor of Sciences (DSc), from the University of Southampton, Southampton, UK. From 1986 to 1999, He held research and academic appointments at the Universities of Sheffield, Edinburgh and Portsmouth, all in UK. Since 1999, he has been with the School of Electronics and Computer Science, the University of Southampton, UK, where he holds the post of Professor in Intelligent Systems and Signal Processing. Dr Chen's research interests include neural network and machine learning, adaptive signal processing, and wireless communications, and nonlinear system modelling. He has published over 700 research papers. Dr. Chen is a Fellow of the United Kingdom Royal Academy of Engineering, a Fellow of IET, a Distinguished Adjunct Professor at King Abdulaziz University, Jeddah, Saudi Arabia, and an original ISI highly cited researcher in engineering (March 2004). Professor Chen has 15,100+ Web of Science citations with h-index 54, and 30,700+ Google Scholar citations with h-index 75.

**Genlin Ji** received the B.E. and M.E. degree in Computer Science and Technology from Nanjing University of Aeronautics and Astronautics in 1986 and 1989, respectively, and received the PhD degree in Computer Science and Technology from Southeast University in 2004. He is now a professor and dean of the school of Computer Science and Technology at Nanjing Normal University. His research interests generally focus on data mining and its application.

**Xin Geng(M'13)** received the BSc and MSc degrees in Computer Science from Nanjing University, China, in 2001 and 2004, respectively, and the PhD degree from Deakin University, Australia in 2008. He is currently a professor and dean of the school of Computer Science and Engineering at Southeast University. His research interests include pattern recognition, machine learning, and computer vision. He has published more than 40 refereed papers and holds four patents in these areas. He is member of the IEEE.