IEEE TRANSACTIONS ON
KNOWLEDGE AND
DATA ENGINEERING

Label Enhancement for Label Distribution Learning

Ning Xu, Yun-Peng Liu, and Xin Geng*, *Member, IEEE*

Abstract—Label distribution is more general than both single-label annotation and multi-label annotation. It covers a certain number of labels, representing the degree to which each label describes the instance. The learning process on the instances labeled by label distributions is called *label distribution learning* (LDL). Unfortunately, many training sets only contain simple logical labels rather than label distributions due to the difficulty of obtaining the label distributions directly. To solve this problem, one way is to recover the label distributions from the logical labels in the training set via leveraging the topological information of the feature space and the correlation among the labels. Such process of recovering label distributions from logical labels is defined as *label enhancement* (LE), which reinforces the supervision information in the training sets. This paper proposes a novel LE algorithm called *Graph Laplacian Label Enhancement* (GLLE). Experimental results on one artificial dataset and fourteen real-world LDL datasets show clear advantages of GLLE over several existing LE algorithms. Furthermore, experimental results on eleven multi-label learning datasets validate the advantage of GLLE over the state-of-the-art multi-label learning approaches.

Index Terms—Label enhancement, label distribution learning, multi-label learning, learning with ambiguity.

1 Introduction

Learning with ambiguity is a hot topic in recent machine learning and data mining research. A learning process is essentially building a mapping from the instances to the labels. This paper mainly focuses on the ambiguity at the label side of the mapping, i.e., one instance is not necessarily mapped to one label. Multi-label learning (MLL) [1] studies the problem where each example is represented by a single instance while associated with a set of labels simultaneously, and the task is to learn a multi-label predictor which maps an instance to a relevant label set [2], [3]. During the past decade, multi-label learning techniques have been widely employed to learn from data with rich semantics, such as text [4], image [5], audio [6], video [7], etc.

In most of the supervised data, an instance $\boldsymbol{x}$ is assigned with $l_{\boldsymbol{x}}^{y} \in \{0,1\}$ to each possible label y, representing whether y describes $\boldsymbol{x}$. In this paper, $l_{\boldsymbol{x}}^{y}$ is called *logical label* as $l_{\boldsymbol{x}}^{y}$ reflects the logical relationship between the label and the instance. Logical label answers the essential question "which label can describe the instance", but not involves the explicit relative importance of each label. To solve this problem, a more natural way to label an instance $\boldsymbol{x}$ is to assign a real number $d_{\boldsymbol{x}}^{y}$ to each possible label y, representing the degree to which y describes $\boldsymbol{x}$. Without loss of generality, assume that $d_{\boldsymbol{x}}^{y} \in [0,1]$. Further suppose that the label set is complete, i.e., using all the labels in the set can always fully describe the instance. Then, $\sum_y d_{\boldsymbol{x}}^{y} = 1$. Such $d_{\boldsymbol{x}}^{y}$ is called the *description degree* of y to $\boldsymbol{x}$. For a particular instance, the description degrees of all the labels constitute a real-valued vector called *label distribution*, which describes the instance more comprehensively than logical labels. The learning process on the instances labeled by label distributions is therefore called *label distribution learning* (LDL) [8]. Label distribution is more general than logical labels in most supervised learning problems because the relevance or irrelevance of a label to an instance is essentially relative in mainly three aspects:

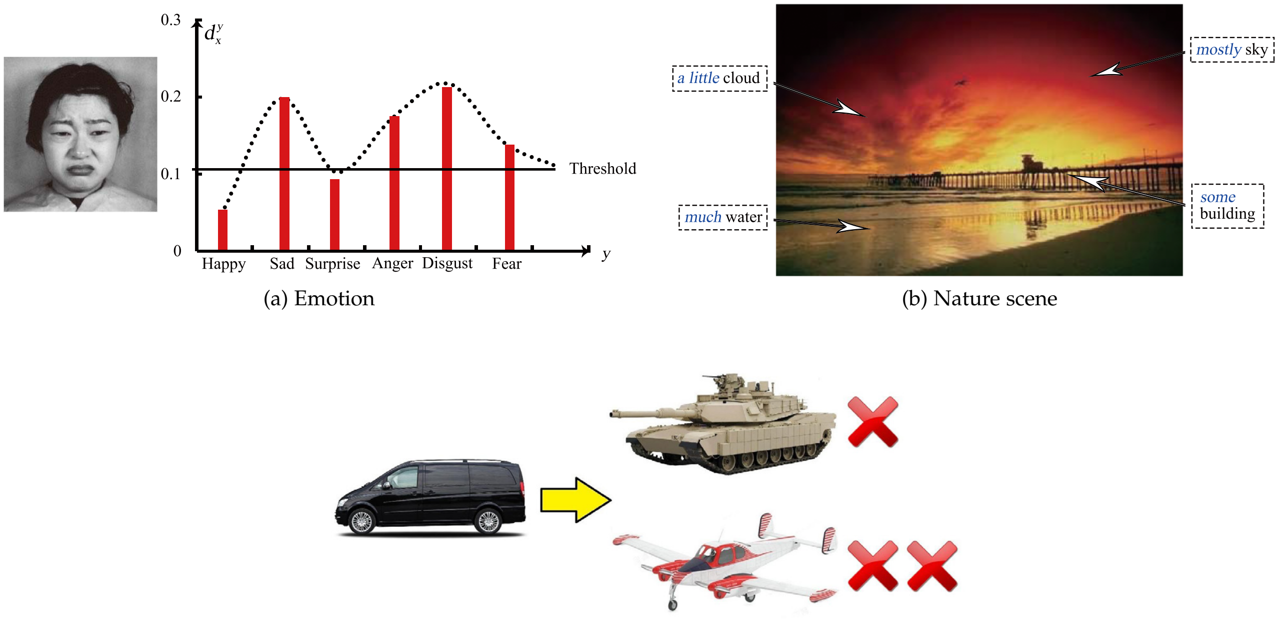
- The differentiation between the relevant and irrelevant labels is relative. A bipartite partition of the label set into relevant and irrelevant labels with respect to an instance is actually a simplification of the real problem. In many cases, the boundary between relevant and irrelevant labels is not clear. For example, in emotion analysis from facial expressions, a facial expression often conveys a complex mixture of basic emotions (e.g., happy, sad, surprise, anger, disgust and fear) [9]. As shown in Fig. 1(a), for an expression, different basic emotions exhibit different intensities. The partition between the relevant and irrelevant emotions depends on the choice of the threshold. But there is no absolutely subjective criterion to determine the threshold.
- When multiple labels are associated with an instance, the relative importance among them is more likely to be different rather than exactly equal. For example, in Fig. 1(b), a natural scene image may be annotated with the labels sky, water, building and cloud simultaneously, but the relative importance of each label to this image is different.
- The "irrelevance" of each irrelevant label may be very different. For example, in Fig. 1(c), for a car, the label airplane is more irrelevant than the label tank.

However, in most training sets, label distribution is not explicitly available. It is difficult to obtain the label distributions directly because the process of quantifying the description degrees is costly. Therefore, we need a way

- *Ning Xu, Yun-Peng Liu and Xin Geng are with the School of Computer Science and Engineering, and the Key Lab of Computer Network and Information Integration (Ministry of Education), Southeast University, Nanjing 211189, China.*

**Corresponding author. E-mail: xgeng@seu.edu.cn*

Digital Object Identifier 10.1109/TKDE.2019.2947040



(a) Emotion

(b) Nature scene

(c) Target

Fig. 1: Three examples about the relevance or irrelevance of each label.

to recover the label distributions from the logical labels in the training set by exploiting the relative importance of each label. This process is called *label enhancement* (LE) in this paper. LE reinforces the supervision information in the training sets via leveraging the topological information in the feature space and the correlation among the labels. After the label distributions are recovered, more effective supervised learning can be achieved by leveraging the label distributions [10], [11].

Note that although there is no explicit concept of LE defined in existing work, some methods with similar function to LE have been proposed. For example, logical labels are transferred to a discretized bivariate Gaussian label distribution centered at the coarse ground-truth label by using priori knowledge in head pose estimation [12] and facial age estimation [13]. Some work [14], [15] builds the membership degrees to the labels, which can constitute a label distribution. Some work [10], [11] establishes the relationship between instances and labels by graph and transfer logical labels into label distributions.

Preliminary results of this paper have been reported in a shorter conference version [16]. While only the topological information of the feature space is used in [16], here we consider the label correlations and explore them in the experiments. Moreover, an alternating solution is used to solve the optimization problem. Besides, the predictive experiment on MLL datasets is conducted to show that GLLE performs favorably against state-of-the-art multi-label learning approaches.

The rest of this paper is organized as follows. Firstly, some related work is briefly reviewed and discussed in Section 2. Secondly, the formulation of LE and the details of the existing LE algorithms are proposed in Section 3. Then, a new LE algorithm is proposed in Section 4. After that, the results of the comparative experiments are reported in Section 5. Finally, conclusions are drawn in Section 6.

2 Related Work

LDL is a novel learning paradigm, which labels an instance with a label distribution and learns a map-ping from instance to label distribution straightly. There are several algorithms [8] designed for LDL. In general, they can be grouped into three categories. PT-Bayes and PT-SVM aim to change the training examples into weighted single-label instances by sampling, which can transform LDL problem as SLL or MLL learning solved by SVM and Naive Bayes. AA-Bayes and AA-BP adapt the hard-threshold labels to the soft ones by some specific mechanisms, which extend traditional MLL algorithms to deal with label distributions. Besides, there are some specialized algorithms to directly match the LDL problems, and the representative one is SA-BFGS. This algorithm is proposed by applying maximum entropy model with Kullback-Leibler divergence as loss function to learn the label distribution.

LDL has been successfully applied to many real applications, such as facial landmark detection [17], age estimation [18], [19], head pose estimation [12], multi-label ranking for natural scene images [20], zero-shot Learning [21] and emotion analysis from texts [22]. According to the theoretical analysis [23], LDL is approximate to the optimal classifier via learning on the instances labeled by the ground-truth label distributions. However, in most training sets, the label distribution is not explicitly available. There are few work to deal with this situation. One recent paper [10] adopts the propagation technique to generate the label distributions without considering the correlations between the labels.

Label distribution explicitly models label ambiguity with the description degree, which is not the probability that y

correctly labels $\boldsymbol{x}$, but the proportion that y accounts for in a full class description of $\boldsymbol{x}$. Therefore, label distribution can be distinguished from the previous studies on probabilistic labels [24], [25], [26], where the basic assumption is that only one correct label is assigned to each instance. Probabilistic labels are mainly used when the real label of the instance cannot be obtained with certainty. In practice, it is usually difficult to determine the probability (or confidence) of a label. In most cases, it relies on the prior knowledge of the human experts, which is a highly subjective and variable process. As a result, the problem of learning from probabilistic labels has not been extensively studied to date.

From the conceptual point of view, it is worthwhile to distinguish description degree from the concept membership used in fuzzy classification. Membership is designed to handle the status of partial truth, which is a truth value which ranges between completely true and completely false. On the other hand, description degree reflects the ambiguity of the label description of the instance, i.e., one label may only partially describe the instance, but it is completely true that the label describes the instance. Fortunately, although the concept of membership is fundamentally different from description degree, some methods [14], [15] which focus on generating membership can be applied to generate label distributions.

3 Label Enhancement

3.1 Formulation of Label Enhancement

First of all, the main notations used in this paper are listed as follows. The instance variable is denoted by $\boldsymbol{x}$, the particular i-th instance is denoted by $\boldsymbol{x}_i$, the label variable is denoted by y, the particular j-th label value is denoted by y_j, the logical label vector of $\boldsymbol{x}_i$ is denoted by $\boldsymbol{l}_i = (l^{y_1}_{\boldsymbol{x}_i}, l^{y_2}_{\boldsymbol{x}_i}, ..., l^{y_c}_{\boldsymbol{x}_i})^\top$, where c is the number of possible labels. The description degree of y to $\boldsymbol{x}$ is denoted by $d^y_{\boldsymbol{x}}$, and the label distribution of $\boldsymbol{x}_i$ is denoted by $\boldsymbol{d}_i = (d^{y_1}_{\boldsymbol{x}_i}, d^{y_2}_{\boldsymbol{x}_i}, ..., d^{y_c}_{\boldsymbol{x}_i})^\top$. Let $\mathcal{X} = \mathbb{R}^q$ denote the q-dimensional feature space. Then, the process of LE can be defined as follows.

Given a training set $\mathcal{S} = \{(\boldsymbol{x}_i, \boldsymbol{l}_i) | 1 \leq i \leq n\}$, where $\boldsymbol{x}_i \in \mathcal{X}$ and $\boldsymbol{l}_i \in \{0,1\}^c$, LE recovers the label distribution $\boldsymbol{d}_i$ of $\boldsymbol{x}_i$ from the logical label vector $\boldsymbol{l}_i$, and thus transforms $\mathcal{S}$ into a LDL training set $\mathcal{E} = \{(\boldsymbol{x}_i, \boldsymbol{d}_i) | 1 \leq i \leq n\}$.

3.2 Existing Label Enhancement Algorithms

3.2.1 *Fuzzy Label Enhancement*

The LE algorithm based on fuzzy clustering (FCM) [14] employs fuzzy C-means clustering [27] which attempts to cluster feature vectors by iteratively minimizing an objective function. Supposing that fuzzy C-means clustering divides the training set $\mathcal{S}$ into p clusters and $\boldsymbol{\mu}_k$ denotes the k-th cluster prototype. Then, the membership degree of $\boldsymbol{x}_i$ to the k-th cluster is calculated by

$$m^k_{\boldsymbol{x}_i} = \frac{1}{\sum_{j=1}^{p} \left(\frac{Dist(\boldsymbol{x}_i, \boldsymbol{\mu}_k)}{Dist(\boldsymbol{x}_i, \boldsymbol{\mu}_j)} \right)^{\frac{1}{\beta - 1}}}, \tag{1}$$

where $\beta > 1$, and $Dist(,)$ is the Euclidean distance. Then, the matrix $\boldsymbol{A}$ providing soft connections between classes and clusters is constructed by initializing a $c \times p$ zero matrix $\boldsymbol{A}$ and updating each row $\boldsymbol{A}_j$ through

$$\boldsymbol{A}_{(j)} = \boldsymbol{A}_{(j)} + \boldsymbol{m}_{\boldsymbol{x}_i}, \text{if } l^{y_j}_{\boldsymbol{x}_i} = 1, \tag{2}$$

where $\boldsymbol{m}_{\boldsymbol{x}_i} = [m^1_{\boldsymbol{x}_i}, m^2_{\boldsymbol{x}_i}, ..., m^p_{\boldsymbol{x}_i}]$. Then, the membership degree vector of $\boldsymbol{x}_i$ to the labels is calculated by using fuzzy composition $\tilde{\boldsymbol{d}}_i = \boldsymbol{A} \circ \boldsymbol{m}^\top_{\boldsymbol{x}_i}$. Finally, the label distribution corresponding to each instance is generated via the softmax normalization $d^y_{\boldsymbol{x}_i} = \frac{e^{\tilde{d}^y_{\boldsymbol{x}_i}}}{\sum_y e^{\tilde{d}^y_{\boldsymbol{x}_i}}}$.

For each label y_j, the LE algorithm based on kernel method (KM) [15] divides $\mathcal{S}$ into two sets, i.e., $C^{y_j}_+$ and $C^{y_j}_-$. $C^{y_j}_+$ contains such sample point $\boldsymbol{x}_i$ with $l^{y_j}_{\boldsymbol{x}_i} = 1$ and $C^{y_j}_-$ contains such sample point $\boldsymbol{x}_i$ with $l^{y_j}_{\boldsymbol{x}_i} = 0$. Then, the center of $C^{y_j}_+$ in the feature space is defined by $\boldsymbol{\psi}^{y_j} = \frac{1}{n_+} \sum\limits_{\boldsymbol{x}_i \in C^{y_j}_+} \varphi(\boldsymbol{x}_i)$, where n_+ is the number of the samples in $C^{y_j}_+$ and $\varphi(\boldsymbol{x}_i)$ is a nonlinear transformation of $\boldsymbol{x}$ to a higher dimensional feature space. Then, the radius of $C^{y_j}_+$ is calculated by

$$r = \max \| \boldsymbol{\psi}^{y_j} - \varphi(\boldsymbol{x}_i) \|. \tag{3}$$

The distance between a sample $\boldsymbol{x}_i \in C^{y_j}_+$ and the center of $C^{y_j}_+$ is calculated by

$$s_i = \| \varphi(\boldsymbol{x}_i) - \boldsymbol{\psi}^{y_j} \|. \tag{4}$$

The calculations involving $\varphi(\boldsymbol{x}_i)$ can be obtained indirectly though the kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i) \cdot \varphi(\boldsymbol{x}_j)$. Then, the membership degree of $\boldsymbol{x}_i$ to each label can be calculated by

$$\tilde{d}^{y_j}_{\boldsymbol{x}_i} = \left\{ \begin{array}{cc} 1 - \sqrt{\frac{s_i^2}{r^2 + \delta}} & \text{if } l^{y_j}_{\boldsymbol{x}_i} = 1 \\ 0 & \text{if } l^{y_j}_{\boldsymbol{x}_i} = 0 \end{array} \right., \tag{5}$$

where $\delta > 0$. Finally, the membership degrees are transferred to the label distribution via the softmax normalization.

3.2.2 *Graph-based Label Enhancement*

The LE algorithm based on label propagation (LP) [10] recovers the label distributions from logical labels by using iterative label propagation technique [28]. Let $\mathcal{G}$ denotes the fully-connected graph constructed over $\mathcal{S}$, and then the $n \times n$ symmetric similarity matrix $\boldsymbol{A}$ is specified for $\mathcal{G}$ as

$$a_{ij} = \left\{ \begin{array}{cc} \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2} \right) & \text{if } i \neq j \\ 0 & \text{if } i = j \end{array} \right. . \tag{6}$$

Correspondingly, the label propagation matrix $\boldsymbol{P}$ is constructed from the similarity matrix through $\boldsymbol{P} = \hat{\boldsymbol{A}}^{-\frac{1}{2}} \boldsymbol{A} \hat{\boldsymbol{A}}^{-\frac{1}{2}}$. Here $\hat{\boldsymbol{A}}$ is a diagonal matrix with the elements $\hat{a}_{ii} = \sum\limits_{j=1}^{n} a_{ij}$. At the t-th iteration, the label distribution matrix $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_n]$ is updated by propagating labeling-importance information with the label propagation matrix $\boldsymbol{P}$ as

$$\boldsymbol{D}^{(t)} = \alpha \boldsymbol{P} \boldsymbol{D}^{(t-1)} + (1 - \alpha) \boldsymbol{L}, \tag{7}$$

where $\boldsymbol{L} = [\boldsymbol{l}_1, \boldsymbol{l}_2, ..., \boldsymbol{l}_n]$ is the logical label matrix in training set and the initial matrix $\boldsymbol{D}^{(0)} = \boldsymbol{L}$. Specifically, $\alpha \in (0,1)$ is the balancing parameter which controls the fraction of the information inherited from the label propagation and the logical label matrix. Finally, $\boldsymbol{D}^{(t)}$ will converge to $\boldsymbol{D}^{*}$, and the label distributions are normalized via the softmax normalization.

The LE algorithm based on manifold learning (ML) [11] considers that the topological structure of the feature space can be represented by a graph $\mathcal{G}$. $\boldsymbol{W}$ is the weight matrix whose element w_{ij} represents the weight of the relationship between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. This method assumes that each data point can be optimally reconstructed by using a linear combination of its neighbors [29], [30]. Then, the approximation of the feature manifold is to induce the minimization of

$$\Theta\left(\boldsymbol{W}\right) = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \sum_{j \neq i} w_{ij} \boldsymbol{x}_j\|^2, \tag{8}$$

where $w_{ij} = 0$ unless $\boldsymbol{x}_j$ is one of $\boldsymbol{x}_i$'s K-nearest neighbors and $\sum_{j=1}^{n} w_{ij} = 1$. According to the smoothness assumption [31], the topological structure of the feature space can be transferred to the label space local by local. Then, the reconstruction of the label manifold can infer to the minimization of

$$\begin{aligned} \Psi\left(\boldsymbol{d}\right) &= \sum_{i=1}^{n} \|\boldsymbol{d}_i - \sum_{j \neq i} w_{ij} \boldsymbol{d}_j\|^2 \\ \text{s.t.} \quad & d_{\boldsymbol{x}_i}^{y_l} l_{\boldsymbol{x}_i}^{y_l} > \lambda, \forall 1 \leq i \leq n, 1 \leq j \leq c, \end{aligned} \tag{9}$$

where $\lambda > 0$. The label distributions are generated with the optimization by using a constrained quadratic programming process. Finally, $\boldsymbol{d}_i$ can be normalize via the softmax normalization.

4 The Proposed Approach

In this section, a new LE algorithm named Graph Laplacian Label Enhancement (GLLE) is proposed, which recovers the label distributions from the logical labels in the training set via leveraging the topological information of the feature space and the correlation among the labels.

4.1 Optimization Framework

Given a training set $\mathcal{S}$, the feature matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n]$ and the logical label matrix $\boldsymbol{L} = [\boldsymbol{l}_1, \boldsymbol{l}_2, ..., \boldsymbol{l}_n]$ are constructed. Our aim is to recover the label distribution matrix $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_n]$ from the logical label matrix $\boldsymbol{L}$. To solve this problem, GLLE assumes the parametric model

$$\boldsymbol{d}_i = \boldsymbol{W}^{\top} \varphi(\boldsymbol{x}_i) + \boldsymbol{b} = \hat{\boldsymbol{W}} \boldsymbol{\phi}_i, \tag{10}$$

where $\boldsymbol{W} = [\boldsymbol{w}^1, ..., \boldsymbol{w}^c]$ is a weight matrix and $\boldsymbol{b} \in \mathbb{R}^c$ is a bias vector. $\varphi(\boldsymbol{x})$ is a nonlinear transformation of $\boldsymbol{x}$ to a higher dimensional feature space. For convenient describing, $\hat{\boldsymbol{W}} = [\boldsymbol{W}^{\top}, \boldsymbol{b}]$ and $\boldsymbol{\phi}_i = [\varphi(\boldsymbol{x}_i); 1]$ are set. Accordingly, the goal of our method is to determine the best parameter $\hat{\boldsymbol{W}}^*$ that can generate a reasonable label distribution $\boldsymbol{d}_i$ given the instance $\boldsymbol{x}_i$. Thus our goal becomes to find the optimal model $\hat{\boldsymbol{W}}^*$ which minimizes

$$\hat{\boldsymbol{W}}^* = \underset{\hat{\boldsymbol{W}}}{\arg\min} \; L(\hat{\boldsymbol{W}}) + \lambda_1 \Omega(\hat{\boldsymbol{W}}) + \lambda_2 Z(\hat{\boldsymbol{W}}), \tag{11}$$

where L is a loss function, Ω is the functions to leverage the topological information of the feature space, and Z is the function to leverage the correlation among the labels. Note that LE is essentially a pre-processing applied to the training set, which is different from standard supervised learning. Therefore, our optimization does not need to consider the overfitting problem.

Since the information in the label distributions is inherited from the initial logical labels, $L(\hat{\boldsymbol{W}})$ in Eq. (11) is defined as the least squares (LS) loss function

$$\begin{aligned} L(\hat{\boldsymbol{W}}) &= \sum_{i=1}^{n} \|\hat{\boldsymbol{W}} \boldsymbol{\phi}_i - \boldsymbol{l}_i\|^2 \\ &= \operatorname{tr}[(\hat{\boldsymbol{W}} \boldsymbol{\Phi} - \boldsymbol{L})^{\top} (\hat{\boldsymbol{W}} \boldsymbol{\Phi} - \boldsymbol{L})], \end{aligned} \tag{12}$$

where $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_n]$.

4.2 Leveraging Topological Information

In order to mine the hidden label importance from the training examples via leveraging the topological information of the feature space, the local similarity matrix $\boldsymbol{A}$ is specified as follows.

- Step 1. An edge is put between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ if $\boldsymbol{x}_i$ is among K-nearest neighbors of $\boldsymbol{x}_j$ or $\boldsymbol{x}_j$ is among K-nearest neighbors of $\boldsymbol{x}_i$.
- Step 2. If $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are connected, then
$$a_{ij} = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right), \tag{13}$$
where $\sigma > 0$ is the width parameter for similarity calculation which is fixed to be 1 in this paper.

According to the smoothness assumption [31], the points close to each other are more likely to share a label. Intuitively, if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ have a high degree of similarity, as measured by a_{ij}, then $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$ should be near to one another. This intuition leads to the following function $\Omega(\hat{\boldsymbol{W}})$ in Eq. (11):

$$\begin{aligned} \Omega(\hat{\boldsymbol{W}}) &= \sum_{i,j} a_{ij} \|\boldsymbol{d}_i - \boldsymbol{d}_j\|^2 \\ &= \operatorname{tr}(\boldsymbol{D}\boldsymbol{G}\boldsymbol{D}^{\top}) \\ &= \operatorname{tr}(\hat{\boldsymbol{W}} \boldsymbol{\Phi} \boldsymbol{G} \boldsymbol{\Phi}^{\top} \hat{\boldsymbol{W}}^{\top}), \end{aligned} \tag{14}$$

where $\boldsymbol{G} = \hat{\boldsymbol{A}} - \boldsymbol{A}$ is the graph Laplacian and $\hat{\boldsymbol{A}}$ is the diagonal matrix whose elements are $\hat{a}_{ii} = \sum_{j=1}^{n} a_{ij}$.

4.3 Handling Label Correlations

The label correlations [32] are considered to provide helpful extra information to recover the label distributions from logical labels in the training sets. Specifically, the more correlative two labels are, the closer the corresponding description degrees should be. In other words, $\boldsymbol{d}^i$ should more be more similar to $\boldsymbol{d}^j$ if the i-th and j-th labels are more correlated. Here $\boldsymbol{d}^i$ is the vector constituted by all the description degrees of the i-th label, i.e., $\boldsymbol{d}^i = (d_{\boldsymbol{x}_1}^{y_i}, d_{\boldsymbol{x}_2}^{y_i}, ..., d_{\boldsymbol{x}_n}^{y_i})$. Assuming that the label correlations are measured by the label correlation matrix $\boldsymbol{R}$ whose elements are r_{ij}, $Z(\hat{\boldsymbol{W}})$ in Eq. (11) is defined as:

$$\begin{aligned} Z(\hat{\boldsymbol{W}}) &= \sum_{i,j} r_{ij} \|\boldsymbol{d}^i - \boldsymbol{d}^j\|^2 \\ &= \mathrm{tr}(\boldsymbol{D}^\top \boldsymbol{C} \boldsymbol{D}) \\ &= \mathrm{tr}(\boldsymbol{\Phi}^\top \hat{\boldsymbol{W}}^\top \boldsymbol{C} \hat{\boldsymbol{W}} \boldsymbol{\Phi}), \end{aligned} \tag{15}$$

where $\boldsymbol{C} = \hat{\boldsymbol{R}} - \boldsymbol{R}$ is the Laplacian matrix, and $\hat{\boldsymbol{R}}$ is the diagonal matrix whose elements are $\hat{r}_{ii} = \sum\limits_{j=1}^{n} r_{ij}$.

In real-world tasks, however, label correlations are naturally local, where a label correlation may be shared by only a subset of instances rather than all the instances [33]. Assume that the training data can be separated into m groups $\{G_1, G_2, ..., G_m\}$, where instances in the same group share the same subset of label correlations. Inspired by [34], the groups can be discovered via clustering. Therefore, the following function is proposed instead of Eq. (15):

$$\begin{aligned} Z(\hat{\boldsymbol{W}}) &= \sum_{i=1}^{m} \mathrm{tr}(\boldsymbol{D}_i^\top \boldsymbol{C}_i \boldsymbol{D}_i) \\ &= \sum_{i=1}^{m} \mathrm{tr}(\boldsymbol{\Phi}_i^\top \hat{\boldsymbol{W}}^\top \boldsymbol{C}_i \hat{\boldsymbol{W}} \boldsymbol{\Phi}_i), \end{aligned} \tag{16}$$

where $\boldsymbol{D}_i$ is the matrix consisting of the label distributions corresponding to all the instance in G_i, $\boldsymbol{C}_i$ is the Laplacian matrix representing the local correlation in G_i, $\boldsymbol{\Phi}_i$ is the feature matrix representing the higher dimensional features to the instance in G_i.

Formulating the LE problem into an optimization framework over Eq. (12), Eq. (14) and Eq. (16), the following optimization problem is obtained:

$$\begin{aligned} \min_{\hat{\boldsymbol{W}}} \quad & \mathrm{tr}[(\hat{\boldsymbol{W}}\boldsymbol{\Phi} - \boldsymbol{L})^\top (\hat{\boldsymbol{W}}\boldsymbol{\Phi} - \boldsymbol{L})] + \lambda_1 \mathrm{tr}(\hat{\boldsymbol{W}}\boldsymbol{\Phi}\boldsymbol{G}\boldsymbol{\Phi}^\top \hat{\boldsymbol{W}}^\top) \\ & + \lambda_2 \sum_{i=1}^{m} \mathrm{tr}(\boldsymbol{\Phi}_i^\top \hat{\boldsymbol{W}}^\top \boldsymbol{C}_i \hat{\boldsymbol{W}} \boldsymbol{\Phi}_i). \end{aligned} \tag{17}$$

In this paper, instead of specifying any label correlation matrix, each Laplacian matrix $\boldsymbol{C}_i$ is learned directly. Note that optimization w.r.t. $\boldsymbol{C}_i$ can not guarantee that $\boldsymbol{C}_i$ is the normalized Laplacian matrix, and may lead to the trivial solution $\boldsymbol{C}_i = \boldsymbol{0}$. To avoid the problems, $\boldsymbol{C}_i$ is decomposed as $\boldsymbol{E}_i \boldsymbol{E}_i^\top$ and the constrain $\mathrm{diag}(\boldsymbol{E}_i \boldsymbol{E}_i^\top) = \boldsymbol{1}$ is added. Then the following formulation is obtained:

$$\begin{aligned} \min_{\hat{\boldsymbol{W}}, \boldsymbol{E}} \quad & \mathrm{tr}[(\hat{\boldsymbol{W}}\boldsymbol{\Phi} - \boldsymbol{L})^\top (\hat{\boldsymbol{W}}\boldsymbol{\Phi} - \boldsymbol{L})] + \lambda_1 \mathrm{tr}(\hat{\boldsymbol{W}}\boldsymbol{\Phi}\boldsymbol{G}\boldsymbol{\Phi}^\top \hat{\boldsymbol{W}}^\top) \\ & + \lambda_2 \sum_{i=1}^{m} \mathrm{tr}(\boldsymbol{\Phi}_i^\top \hat{\boldsymbol{W}}^\top \boldsymbol{E}_i \boldsymbol{E}_i^\top \hat{\boldsymbol{W}} \boldsymbol{\Phi}_i) \\ \text{s.t.} \quad & \mathrm{diag}(\boldsymbol{E}_i \boldsymbol{E}_i^\top) = \boldsymbol{1}, i = 1, 2, ..., m. \end{aligned} \tag{18}$$

If the best parameter $\hat{\boldsymbol{W}}^*$ is determined, the label distribution $\boldsymbol{d}_i$ can be generated through Eq. (10). Finally, $\boldsymbol{d}_i$ is normalized via the softmax normalization.

According to the representor's theorem [35], under fairly general conditions, a learning problem can be expressed as a linear combination of the training examples in the feature space, i.e. $\boldsymbol{w}^j = \sum_i \boldsymbol{\theta}^j \varphi(\boldsymbol{x}_i)$. If this expression is replaced into Eq. (18), it will generate the inner product$< \varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j) >$, and then the kernel trick can be applied.

4.4 The Alternating Solution

We solve the optimization problem in Eq. (18) in an alternating way, i.e., optimizing one of the two variables with the other fixed. When $\hat{\boldsymbol{W}}$ is fixed to solve $\boldsymbol{E}$, Eq. (18) can be reduced to:

$$\begin{aligned} \min_{\boldsymbol{E}} \quad & \sum_{i=1}^{m} \mathrm{tr}(\boldsymbol{\Phi}_i^\top \hat{\boldsymbol{W}}^\top \boldsymbol{E}_i \boldsymbol{E}_i^\top \hat{\boldsymbol{W}} \boldsymbol{\Phi}_i) \\ \text{s.t.} \quad & \mathrm{diag}(\boldsymbol{E}_i \boldsymbol{E}_i^\top) = \boldsymbol{1}, i = 1, 2, ..., m. \end{aligned} \tag{19}$$

Note that Eq. (19) can be further decomposed into m optimization problems, where the i-th one is:

$$\begin{aligned} \min_{\boldsymbol{E}_i} \quad & \mathrm{tr}(\boldsymbol{\Phi}_i^\top \hat{\boldsymbol{W}}^\top \boldsymbol{E}_i \boldsymbol{E}_i^\top \hat{\boldsymbol{W}} \boldsymbol{\Phi}_i) \\ \text{s.t.} \quad & \mathrm{diag}(\boldsymbol{E}_i \boldsymbol{E}_i^\top) = \boldsymbol{1}. \end{aligned} \tag{20}$$

The optimization of Eq. (20) uses projected gradient descent. The gradient of the objective w.r.t. $\boldsymbol{E}_i$ is

$$\nabla_{\boldsymbol{E}_i} = 2\hat{\boldsymbol{W}} \boldsymbol{\Phi}_i \boldsymbol{\Phi}_i^\top \hat{\boldsymbol{W}}^\top \boldsymbol{E}_i. \tag{21}$$

To satisfy the constraint $\mathrm{diag}(\boldsymbol{E}_i \boldsymbol{E}_i^\top) = \boldsymbol{1}$, each row of $\boldsymbol{E}_i$ is projected onto the unit norm ball after each update

$$\boldsymbol{e}_{i,(j)} \leftarrow \frac{\boldsymbol{e}_{i,(j)}}{\|\boldsymbol{e}_{i,(j)}\|}, \tag{22}$$

where $\boldsymbol{e}_{i,(j)}$ is the j-th row of $\boldsymbol{E}_i$.

When $\boldsymbol{E}$ is fixed to solve $\hat{\boldsymbol{W}}$, the task becomes:

$$\begin{aligned} \min_{\hat{\boldsymbol{W}}} \quad & \mathrm{tr}[(\hat{\boldsymbol{W}}\boldsymbol{\Phi} - \boldsymbol{L})^\top (\hat{\boldsymbol{W}}\boldsymbol{\Phi} - \boldsymbol{L})] + \lambda_1 \mathrm{tr}(\hat{\boldsymbol{W}}\boldsymbol{\Phi}\boldsymbol{G}\boldsymbol{\Phi}^\top \hat{\boldsymbol{W}}^\top) \\ & + \lambda_2 \sum_{i=1}^{m} \mathrm{tr}(\boldsymbol{\Phi}_i^\top \hat{\boldsymbol{W}}^\top \boldsymbol{E}_i \boldsymbol{E}_i^\top \hat{\boldsymbol{W}} \boldsymbol{\Phi}_i). \end{aligned} \tag{23}$$

The optimization of Eq. (23) uses an effective quasi-Newton method BFGS [36]. As to the optimization of the target function $T(\hat{\boldsymbol{W}})$, the computation of BFGS is mainly related to the first-order gradient, which can be obtained through

$$\begin{aligned} \nabla_{\hat{\boldsymbol{W}}} =& 2\hat{\boldsymbol{W}}\boldsymbol{\Phi}\boldsymbol{\Phi}^\top - 2\boldsymbol{L}\boldsymbol{\Phi}^\top + \lambda_1 \hat{\boldsymbol{W}}\boldsymbol{\Phi}\boldsymbol{G}^\top \boldsymbol{\Phi}^\top + \lambda_1 \hat{\boldsymbol{W}}\boldsymbol{\Phi}\boldsymbol{G}\boldsymbol{\Phi}^\top \\ & + 2\lambda_2 \sum_{i=1}^{m} (\boldsymbol{E}_i \boldsymbol{E}_i^\top \hat{\boldsymbol{W}} \boldsymbol{\Phi}_i \boldsymbol{\Phi}_i^\top). \end{aligned} \tag{24}$$

5 Experiments

5.1 Recovery Experiment

As shown in Fig. 2, we recover the label distributions from the logical labels via the LE algorithms, and then compare the recovered label distributions with the ground-truth label distributions.

TABLE 1: Statistics of the 15 datasets used in the recovery experiment and LDL predictive experiment

No.	Dataset	#Examples	#Features	#Labels
1	Artificial (Ar)	2601	3	3
2	SJAFFE (SJ)	213	243	6
3	NaturalScene (NS)	2,000	294	9
4	Yeast-spoem (spoem)	2,465	24	2
5	Yeast-spo5 (spo5)	2,465	24	3
6	Yeast-dtt (dtt)	2,465	24	4
7	Yeast-cold (cold)	2,465	24	4
8	Yeast-heat (heat)	2,465	24	6
9	Yeast-spo (spo)	2,465	24	6
10	Yeast-diau (diau)	2,465	24	7
11	Yeast-elu (elu)	2,465	24	14
12	Yeast-cdc (cdc)	2,465	24	15
13	Yeast-alpha (alpha)	2,465	24	18
14	SBU_3DFE (3DFE)	2,500	243	6
15	Movie (Mov)	7,755	1,869	5

5.1.1 Datasets

There are in total 15 datasets used in the experiments including an artificial toy dataset, 14 LDL real-world datasets[1]. Some basic statistics about these 15 datasets are given in Table 1.

The first dataset is an artificial toy dataset which is generated to show in a direct and visual way whether the LE algorithms can recover the label distributions from the logical labels. In this dataset, the instance $\boldsymbol{x}$ is of three-dimensional and there are three labels. The label distribution $\boldsymbol{d} = [d_{\boldsymbol{x}}^{y_1}, d_{\boldsymbol{x}}^{y_2}, d_{\boldsymbol{x}}^{y_3}]$ of $\boldsymbol{x} = [x_1, x_2, x_3]^\top$ is created in the following way.

$$t_i = ax_i + bx_i^2 + cx_i^3 + d, i = 1, ..., 3, \tag{25}$$

$$\psi_1 = (\boldsymbol{h}_1^\top \boldsymbol{t})^2, \psi_2 = (\boldsymbol{h}_2^\top \boldsymbol{t} + \beta_1 \psi_1)^2, \psi_3 = (\boldsymbol{h}_3^\top \boldsymbol{t} + \beta_2 \psi_2)^2, \tag{26}$$

$$d_{\boldsymbol{x}}^i = \frac{\psi_i}{\psi_1 + \psi_2 + \psi_3}, i = 1, ..., 3, \tag{27}$$

where $\boldsymbol{t} = [t_1, t_2, t_3]^\top$, $x_i \in [-1, 1]$, $a = 1$, $b = 0.5$, $c = 0.2$, $d = 1$, $\boldsymbol{h}_1 = [4, 2, 1]^\top$, $\boldsymbol{h}_2 = [1, 2, 4]^\top$, $\boldsymbol{h}_3 = [1, 4, 2]^\top$, and $\beta_1 = \beta_2 = 0.01$. In order to show the results of LE algorithms in a direct and visual way, the examples of the toy dataset are selected from a certain manifold in the feature space. The first two components of the instance $\boldsymbol{x}$, x_1 and x_2, are located at a grid of the interval 0.04 within the range $[-1, 1]$, and there are in total $51 \times 51 = 2601$ instances. The third component x_3 is calculated by

$$x_3 = \sin((x_1 + x_2) \times \pi). \tag{28}$$

Then, the label distribution $\boldsymbol{d}$ corresponding to each is calculated via Eq. (25)-(27).

The second to the fourteen datasets are real-world LDL datasets [8] collected from biological experiments on the yeast genes, facial expression images, natural scene images and movies, respectively.

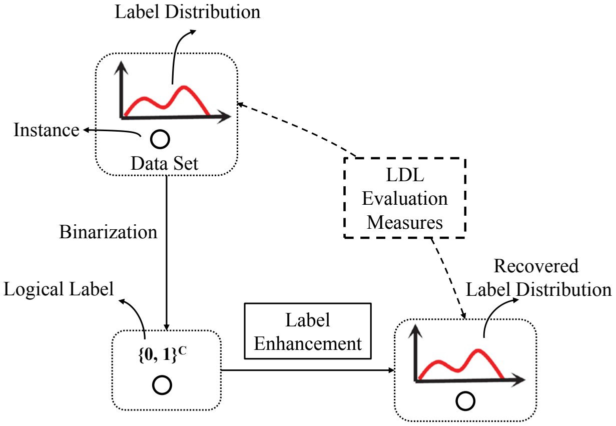


Fig. 2: The schematic diagram of the recovery experiment.

TABLE 2: The distribution distance/similarity measures

Measure	Formula
Chebyshev ↓	$Dis_1(\boldsymbol{d}, \hat{\boldsymbol{d}}) = \max_j \lvert d_j - \hat{d}_j \rvert$
Clark ↓	$Dis_2(\boldsymbol{d}, \hat{\boldsymbol{d}}) = \sqrt{\sum_{j=1}^{c} \frac{(d_j - \hat{d}_j)^2}{(d_j + \hat{d}_j)^2}}$
Canberra ↓	$Dis_3(\boldsymbol{d}, \hat{\boldsymbol{d}}) = \sum_{j=1}^{c} \frac{\lvert d_j - \hat{d}_j \rvert}{d_j + \hat{d}_j}$
Kullback-Leibler ↓	$Dis_4(\boldsymbol{d}, \hat{\boldsymbol{d}}) = \sum_{j=1}^{c} d_j \ln \frac{d_j}{\hat{d}_j}$
cosine ↑	$Sim_1(\boldsymbol{d}, \hat{\boldsymbol{d}}) = \frac{\sum_{j=1}^{c} d_j \hat{d}_j}{\sqrt{\sum_{j=1}^{c} d_j^2} \sqrt{\sum_{j=1}^{c} \hat{d}_j^2}}$
intersection ↑	$Sim_2(\boldsymbol{d}, \hat{\boldsymbol{d}}) = \sum_{j=1}^{c} \min(d_j, \hat{d}_j)$

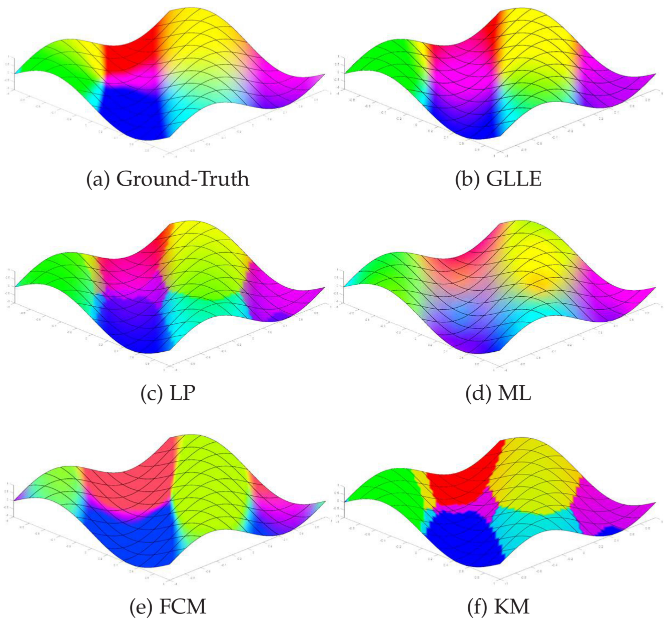
5.1.2 Evaluation Measures

When the ground-truth label distributions are available in the test datasets, a natural choice of the evaluation measure is the distance or similarity measure. There are many distance/similarity measures proposed in clustering or classification work. For example, a fuzzy extension of the rand index [37] is proposed for classifiers and clustering. Zhang [38], [39] proposed generalized adjusted rand indices and generalized pair-counting similarity measures for clustering and cluster ensembles. Cha [40] performed a semantic similarity analysis on 41 measures for the distance/similarity between distributions from 8 syntactic families.

The output of LE algorithm is label distribution rather than logical output of clustering or classification, which makes some commonly used measures inapplicable. As suggested in [8], we select six measures , i.e., Chebyshev distance (Cheb), Clark distance (Clark), Canberra metric (Canber), Kullback-Leibler divergence (KL), cosine coefficient (Cosine) and intersection similarity (Intersec), which belong to the Minkowski family, the χ^2 family, the L_1 family, the Shannons entropy family, the inner product family, and the intersection family, respectively. The first four are distance measures and the last two are similarity measures.

Suppose the real label distribution is $\boldsymbol{d} = [d_1, d_2, ..., d_c]$,

[1] http://cse.seu.edu.cn/PersonalPage/xgeng/LDL/index.htm



(a) Ground-Truth (b) GLLE

(c) LP (d) ML

(e) FCM (f) KM

Fig. 3: Comparison between the ground-truth and recovered label distributions (regarded as RGB colors) on the artificial manifold.

the predicted label distribution is $\hat{\boldsymbol{d}} = [\hat{d}_1, \hat{d}_2, ..., \hat{d}_c]$, then the formulae of the six measures are summarized in Table 2, where the "↓" after the distance measures indicates "the smaller the better", and the "↑" after the similarity measures indicates "the larger the better". Considering that the selected measures all come from different families, the selected measures are significantly different in both syntax and semantics.

5.1.3 Methodology

The four algorithms described in Section 3.2, i.e., FCM [14], KM [15], LP [10], ML [11], and our GLLE are all applied to the 15 datasets shown in Table 1. For each compared algorithm, we adopt the suggested configuration in their literature, i.e., the parameter α in LP is set to 0.5, the number of neighbors K for ML is set to $c{+}1$, the parameter β in FCM is set to 2, and the kernel function in KM is Gaussian kernel. For GLLE, the parameter λ_1 and λ_2 are chosen among $\{10^{-2}, 10^{-1}, ..., 100\}$, and the number of neighbors K is set to $c+1$. The kernel function in GLLE is Gaussian kernel.

We consider the following LDL learning setting. With each instance, a label distribution is associated. The training set, however, contains for each instance not the actual distribution, but a set of labels. The set includes the labels with the highest weights in the distribution, and is the smallest set such that the sum of these weights exceeds a given threshold. This setting can model, for instance, the way in which users label images or add keywords to texts: it assumes that users add labels starting with the most relevant ones, until they feel the labeling is sufficiently complete. Therefore, the logical labels in the datasets can be binarized from the real label distributions as follows. For each instance $\boldsymbol{x}$, the greatest description degree $d_{\boldsymbol{x}}^{y_j}$ is found, and the label y_j is set to relevant label, i.e., $l_{\boldsymbol{x}}^{y_j} = 1$. Then, we calculate the sum of the description degrees of all the current relevant labels $H = \sum_{y_j \in \mathcal{Y}^+} d_{\boldsymbol{x}}^{y_j}$, where $\mathcal{Y}^+$ is the set of the current relevant labels. If H is less than a predefined threshold T, we continue finding the greatest description degree among other labels excluded from $\mathcal{Y}^+$ and select the label corresponding to the greatest description degree into $\mathcal{Y}^+$. This process continues until $H > T$. Finally, the logical labels to the labels in $\mathcal{Y}^+$ are set to 1, and other logical labels are set to 0. In our experiments, $T = 0.5$.

5.1.4 Recovery Performance

In order to visually show the results of the LE algorithms on the artificial dataset, the description degrees of the three labels are regarded as the three color channels of the RGB color space, respectively. In this way, the color of a point in the feature space will visually represent its label distribution. Thus, the label distribution recovered by the LE algorithms can be compared with the ground-truth label distribution through observing the color patterns on the manifold. For easier comparison, the images are visually enhanced by applying a decorrelation stretch process. The results are shown in Fig. 3. It can be seen that GLLE recovers almost identical color patterns with the ground-truth. LP, ML and FCM can also recover similar color patterns with the ground-truth. However, KM fails to obtain a reasonable result.

For quantitative analysis, Table 3 tabulates the results of the five LE algorithms on all the datasets evaluated by Cheb and Cosine, and the best performance on each dataset is highlighted by boldface. For each evaluation metric, ↓ indicates the smaller the better while ↑ indicates the larger the better. Note that since each LE algorithm only runs once, there is no record of standard deviation. GLLE ranks *1st* in 94.4% cases across six evaluation measures. Thus, GLLE generally performs better than other LE algorithms.

5.2 LDL Predictive Experiment

In order to further test the effectiveness of LDL after the LE pre-process on the logical-labeled datasets, we first recover the label distributions from the logical labels via the LE algorithms, and then use the recovered label distributions for LDL training. Finally, the trained LDL models are tested on the new test dataset, and the label distribution predictions are compared with those predictions made by the LDL model directly trained on the ground-truth label distributions. The LDL training algorithm used in this paper is SA-BFGS [8]. The process is shown in Fig. 4. The datasets and evaluation measures in section 5.1 are used in this experiments.

5.2.1 Methodology

In this experiment, the compared LE algorithms and parameters configuration are identical to Section 5.1. 'FCM', 'KM', 'LP', 'ML' and 'GLLE' represent the predictions made by the LDL model trained on the label distributions recovered by each LE algorithm, respectively. In addition, the prediction performance upper bound is made by the LDL model directly trained on the ground-truth label distributions. All the algorithms are tested via ten-fold cross validation.

TABLE 3: Recovery Results (value(rank)) Evaluated by Six LDL Measures

Comparing								Cheb ↓								Avg.
algorithm	Ar	SJ	NS	spoem	spo5	dtt	cold	heat	spo	diau	elu	cdc	alpha	3DFE	Mov	Rank
FCM	0.188(3)	0.132(3)	0.368(5)	0.233(3)	0.162(3)	0.097(2)	0.141(3)	0.169(4)	0.130(3)	0.124(3)	0.052(3)	0.051(3)	0.044(3)	0.135(3)	0.230(4)	3.2
KM	0.260(5)	0.214(5)	0.306(4)	0.408(5)	0.277(5)	0.257(5)	0.252(5)	0.175(5)	0.175(5)	0.152(5)	0.078(5)	0.076(5)	0.063(5)	0.238(5)	0.234(5)	4.93
LP	0.130(2)	0.107(2)	**0.275(1)**	0.163(2)	0.114(2)	0.128(3)	0.137(2)	0.086(2)	0.090(2)	0.099(2)	0.044(2)	0.042(2)	0.040(2)	**0.123(1)**	0.161(2)	1.93
ML	0.227(4)	0.186(4)	0.295(2)	0.403(4)	0.273(4)	0.244(4)	0.242(4)	0.165(3)	0.171(4)	0.148(4)	0.072(4)	0.071(4)	0.057(4)	0.233(4)	0.164(3)	3.73
GLLE	**0.108(1)**	**0.087(1)**	0.296(3)	**0.088(1)**	**0.099(1)**	**0.052(1)**	**0.066(1)**	**0.049(1)**	**0.062(1)**	**0.053(1)**	**0.023(1)**	**0.022(1)**	**0.020(1)**	0.126(2)	**0.122(1)**	1.2
Comparing								Clark ↓								Avg.
algorithm	Ar	SJ	NS	spoem	spo5	dtt	cold	heat	spo	diau	elu	cdc	alpha	3DFE	Mov	Rank
FCM	0.561(3)	0.522(3)	2.486(5)	0.401(3)	0.395(3)	0.329(2)	0.433(2)	0.580(3)	0.520(2)	0.838(3)	0.579(2)	0.739(2)	0.821(2)	0.482(2)	0.859(2)	2.6
KM	1.251(5)	1.874(5)	2.448(3)	1.028(5)	1.059(5)	1.477(5)	1.472(5)	1.802(5)	1.811(5)	1.886(5)	2.768(5)	2.885(5)	3.153(5)	1.907(5)	1.766(5)	4.87
LP	0.487(2)	0.502(2)	2.482(4)	0.272(2)	0.274(2)	0.499(3)	0.503(3)	0.568(2)	0.558(3)	0.788(2)	0.973(3)	1.014(3)	1.185(3)	0.580(3)	0.913(3)	2.67
ML	1.041(4)	1.519(4)	2.388(2)	1.004(4)	1.036(4)	1.446(4)	1.440(4)	1.764(4)	1.768(4)	1.844(4)	2.711(4)	2.825(4)	3.088(4)	1.848(4)	1.140(4)	3.87
GLLE	**0.452(1)**	**0.377(1)**	**2.343(1)**	**0.132(1)**	**0.197(1)**	**0.143(1)**	**0.176(1)**	**0.213(1)**	**0.266(1)**	**0.296(1)**	**0.295(1)**	**0.306(1)**	**0.337(1)**	**0.391(1)**	**0.569(1)**	1
Comparing								Canber ↓								Avg.
algorithm	Ar	SJ	NS	spoem	spo5	dtt	cold	heat	spo	diau	elu	cdc	alpha	3DFE	Mov	Rank
FCM	0.797(3)	1.081(3)	6.974(5)	0.534(3)	0.563(3)	0.501(2)	0.734(2)	1.157(2)	0.998(2)	1.895(3)	1.689(2)	2.415(2)	2.883(2)	1.020(2)	1.664(2)	2.53
KM	1.779(5)	4.010(5)	6.795(4)	1.253(5)	1.382(5)	2.594(5)	2.566(5)	3.849(5)	3.854(5)	4.261(5)	9.110(5)	9.875(5)	11.809(5)	4.121(5)	3.444(5)	4.933
LP	0.668(2)	1.064(2)	6.79(3)	0.365(2)	0.401(2)	0.941(3)	0.924(3)	1.293(3)	1.231(3)	1.748(2)	3.381(3)	3.644(3)	4.544(3)	1.245(3)	1.720(3)	2.67
ML	1.413(4)	3.138(4)	6.477(2)	1.226(4)	1.355(4)	2.549(4)	2.519(4)	3.779(4)	3.772(4)	4.180(4)	8.949(4)	9.695(4)	11.603(4)	4.001(4)	1.934(4)	3.87
GLLE	**0.617(1)**	**0.781(1)**	**6.299(1)**	**0.183(1)**	**0.305(1)**	**0.248(1)**	**0.305(1)**	**0.430(1)**	**0.548(1)**	**0.671(1)**	**0.902(1)**	**0.959(1)**	**1.134(1)**	**0.828(1)**	**1.045(1)**	1
Comparing								KL ↓								Avg.
algorithm	Ar	SJ	NS	spoem	spo5	dtt	cold	heat	spo	diau	elu	cdc	alpha	3DFE	Mov	Rank
FCM	0.267(3)	0.107(3)	3.565(5)	0.208(3)	0.123(3)	0.065(2)	0.113(3)	0.147(3)	0.110(3)	0.159(3)	0.059(2)	0.091(2)	0.100(2)	0.094(2)	0.381(4)	2.87
KM	0.309(5)	0.558(5)	3.014(4)	0.531(5)	0.334(5)	0.617(5)	0.586(5)	0.586 (5)	0.562(5)	0.538(5)	0.617(5)	0.630(5)	0.630(5)	0.603(5)	0.452(5)	4.93
LP	0.160(2)	0.077(2)	**1.595(1)**	0.067(2)	0.042(2)	0.103(3)	0.103(2)	0.089(2)	0.084(2)	0.127(2)	0.109(3)	0.111(3)	0.121(3)	0.105(3)	0.177(2)	2.27
ML	0.274(4)	0.391(4)	2.275(2)	0.503(4)	0.317(4)	0.586(4)	0.556(4)	0.556(4)	0.532(4)	0.509(4)	0.589(4)	0.601(4)	0.602(4)	0.565(4)	0.218(3)	3.8
GLLE	**0.131(1)**	**0.050(1)**	2.663(3)	**0.027(1)**	**0.034(1)**	**0.013(1)**	**0.019(1)**	**0.017(1)**	**0.029(1)**	**0.027(1)**	**0.013(1)**	**0.014(1)**	**0.013(1)**	**0.069(1)**	**0.123(1)**	1.13
Comparing								Cosine ↑								Avg.
algorithm	Ar	SJ	NS	spoem	spo5	dtt	cold	heat	spo	diau	elu	cdc	alpha	3DFE	Mov	Rank
FCM	0.933(3)	0.906(3)	0.593(5)	0.878(3)	0.922(3)	0.959(2)	0.922(3)	0.883(3)	0.909(3)	0.882(3)	0.950(2)	0.929(2)	0.922(2)	0.912(3)	0.773(5)	3
KM	0.918(5)	0.827(5)	0.748(3)	0.812(5)	0.882(5)	0.759(5)	0.779(5)	0.779(5)	0.800(5)	0.799(5)	0.758(5)	0.754(5)	0.751(5)	0.812(5)	0.880(4)	4.8
LP	0.974(2)	0.941(2)	**0.860(1)**	0.950(2)	0.969(2)	0.921(3)	0.925(2)	0.932(2)	0.939(2)	0.915(2)	0.918(3)	0.916(3)	0.911(3)	0.922(2)	0.929(2)	2.2
ML	0.925(4)	0.857(4)	0.818(2)	0.815(4)	0.884(4)	0.763(4)	0.784(4)	0.783(4)	0.803(4)	0.803(4)	0.763(4)	0.759 (4)	0.756(4)	0.815(4)	0.919(3)	3.8
GLLE	**0.980(1)**	**0.958(1)**	0.733(4)	**0.978(1)**	**0.971(1)**	**0.988(1)**	**0.982(1)**	**0.984(1)**	**0.974(1)**	**0.975(1)**	**0.987(1)**	**0.987(1)**	**0.987(1)**	**0.927(1)**	**0.936(1)**	1.2
Comparing								Intersec ↑								Avg.
algorithm	Ar	SJ	NS	spoem	spo5	dtt	cold	heat	spo	diau	elu	cdc	alpha	3DFE	Mov	Rank
FCM	0.812(3)	0.821(3)	0.379(5)	0.767(3)	0.838(3)	0.894(2)	0.833(2)	0.807(2)	0.836(2)	0.760(3)	0.883(2)	0.847(2)	0.844(2)	0.827(2)	0.677(4)	2.67
KM	0.740(5)	0.593(5)	0.453(4)	0.592(5)	0.724(5)	0.541(5)	0.559(5)	0.559(5)	0.575(5)	0.588(5)	0.539(5)	0.533(5)	0.532(5)	0.579(5)	0.649(5)	4.93
LP	0.870(2)	0.837(2)	**0.626(1)**	0.837(2)	0.886(2)	0.786(3)	0.794(3)	0.805(3)	0.819(3)	0.788(2)	0.782(3)	0.779(3)	0.774(3)	0.810(3)	0.778(3)	2.53
ML	0.773(4)	0.661(4)	0.567(2)	0.597(4)	0.727(4)	0.546(4)	0.565(4)	0.564(4)	0.580(4)	0.593(4)	0.544(4)	0.538(4)	0.537(4)	0.587(4)	0.779(2)	3.73
GLLE	**0.892(1)**	**0.872(1)**	0.518(3)	**0.912(1)**	**0.901(1)**	**0.939(1)**	**0.924(1)**	**0.929(1)**	**0.909(1)**	**0.906(1)**	**0.936(1)**	**0.937(1)**	**0.938(1)**	**0.850(1)**	**0.831(1)**	1.13

TABLE 4: The average ranks of five algorithms on six measures

Criterion	FCM	KM	LP	ML	GLLE
Cheb	4.40	4.27	2.20	3.13	1.00
Clark	4.33	4.07	2.40	3.07	1.13
Canber	4.20	4.13	2.40	3.13	1.13
KL	4.37	4.30	2.20	3.13	1.00
Cosine	4.53	4.27	2.13	3.07	1.00
Intersec	4.40	4.27	2.13	3.13	1.07

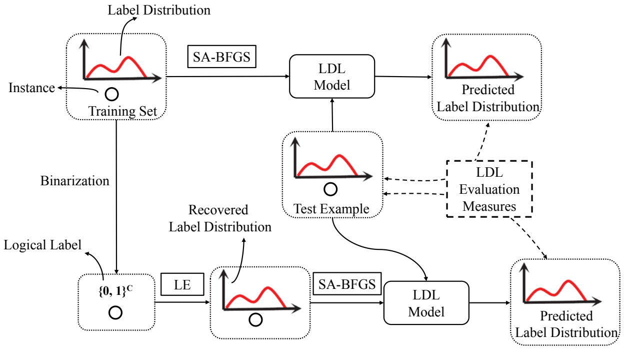
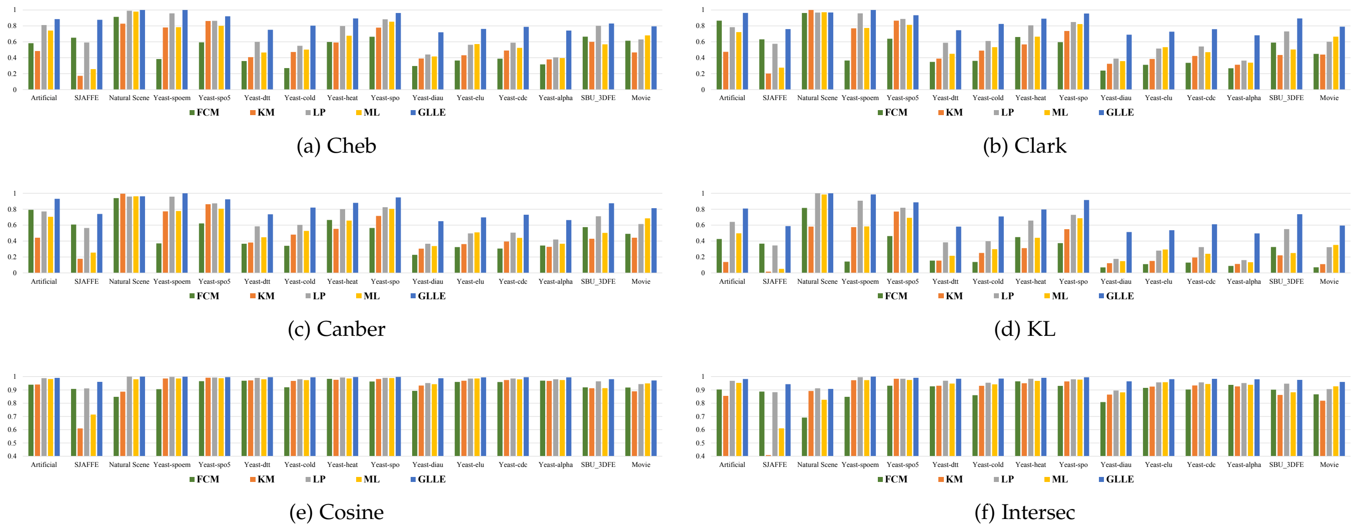


Fig. 4: The schematic diagram of the LDL predictive experiment.

5.2.2 Predictive Performance

The histograms of the ratio of the LE + LDL performance over the performance upper-bound (i.e., predictions made by the LDL model trained on the ground-truth label distributions) are given in Fig. 5. The average rank of each algorithm over all the datasets is shown in Table 4. Based on the experimental results, GLLE ranks *1st* in 96.7% cases across all the evaluation measures. Thus, GLLE achieves superior performance over other LE algorithms.

Note that in most cases, GLLE is very close to the performance upper bound, especially on the Nature Scene and Yeast-spoem datasets. But the difference between them is relatively larger on a few datasets (SJAFFE, Yeast-cold, Yeast-diau and Yeast-alpha). This is because that the description degrees constituting each ground-truth label distribution in these datasets are almost equal. Thus, the binarization process to generate the logical labels might become unstable. It is hard to recover the reasonable label distributions from these logical labels. When the description degrees constituting each ground-truth label distribution in the datasets (e.g., the Nature Scene, Yeast-spoem and Yeast-spo datasets) are much different, the binarization process can easily differentiate the relevant labels and the irrelevant labels, which is helpful to recover the reasonable label distributions. Compared with the second best algorithm, on average, GLLE's distance to the performance upper bound is closer by 65.0% on Cheb, 62.6% on Clark, 60.3% on Canber, 57.0% on KL, 41.9% on Cosine, and 55.6% on Intersec, respectively. The results of the LDL predictive performances prove the effectiveness of LDL after LE pre-process by using GLLE on the logical-labeled training sets.



(a) Cheb

(b) Clark

(c) Canber

(d) KL

(e) Cosine

(f) Intersec

Fig. 5: The ratio of the LE + LDL performance over the performance upper-bound

TABLE 5: Statistics of the 10 datasets used in MLL predictive experiment

No.	Dataset	#Examples	#Features	#Labels
1	cal500	502	68	174
2	emotion	593	72	6
3	medical	978	1,449	45
4	llog	1,460	1,004	75
5	enron	1,702	1,001	53
6	msra	1,868	898	19
7	image	2,000	294	5
8	scene	2,407	294	5
9	slashdot	3,782	1,079	22
10	corel5k	5,000	499	374

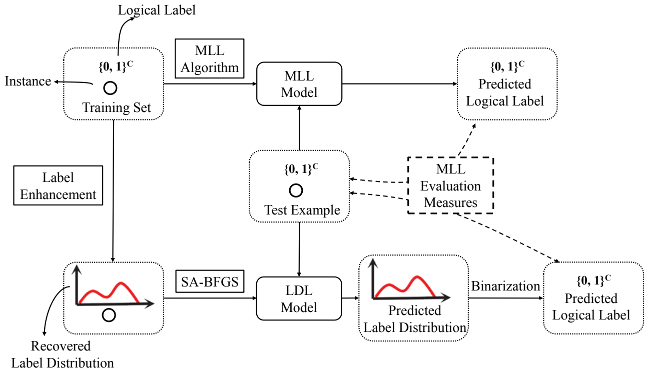


Fig. 6: The schematic diagram of the MLL predictive experiment.

5.3 MLL Predictive Experiment

As mentioned in section 1, more effective supervised learning can be achieved when the label distributions are recovered. In this experiment, the effective performance of MLL prediction based on LE can be validated. Firstly, the label distributions are recover from the logical labels in MLL datasets via the LE algorithms. After that, the LDL model can be trained on the recovered label distributions by SA-BFGS. Finally, the multi-label predictions are binarized from label distribution predictions made by the LDL model, which enable the comparison with the predictive performance of the state-of-the-art MLL algorithms. This process is shown in Fig. 6.

5.3.1 Datasets

There are ten MLL datasets[2] used in the experiments. Some basic statistics about these datasets are given in Table 5. The MLL datasets cover a broad range of cases with diversified multi-label properties and thus serve as a solid basis for thorough comparative studies.

5.3.2 Evaluation Measures

Five widely-used MLL evaluation metrics are selected in this experiment, i.e., *Hamming loss*, *One-error*, *Coverage*, *Ranking loss* and *Average precision* [3]. Note that for all the five multi-label metrics, their values vary between [0,1]. Furthermore, for average precision, the *larger* the values the better the performance; While for the other four metrics, the *smaller* the values the better the performance. These metrics serve as good indicators for comprehensive comparative studies as they evaluate the performance of the learned models from various aspects.

5.3.3 Methodology

In this experiment, GLLE is compared with several state-of-the-art multi-label learning algorithms:

- Binary Relevance (BR): This is a first-order approach which decomposes the multi-label learning problem into q independent binary classification problems [41].
- Calibrated Label Ranking (CLR): This is a second-order approach which transforms the multi-label learning problem into the label ranking problem,

[2] http://mulan.sourceforge.net/datasets.htm

where ranking among labels is carried out by preference learning techniques [42].

- Ensemble of Classifier Chains (ECC): This is a high-order approach which transforms the multi-label learning problem into a chain of binary classification problems with random order, where subsequent classifiers in the chain take predictions of preceding ones as extra input features [43]. The ensemble size is set to 30.
- Random k-labelsets (RAKEL): This is a high-order approach which transforms the multi-label learning problem into an ensemble of multi-class learning problems, where each component multi-class learning problem is derived from a random k-labelset in $\mathcal{Y}$ using label powerset techniques [3], [44]. Here, the ensemble size is set to be $2q$ with $k = 3$ as suggested in the literature [45].

Since the parametric predictor employed by SA-BFGS can be viewed equivalently as multinomial logistic regression models, each of the four comparing algorithms are implemented under the MULAN multi-label learning package [46] by instantiating their base learners with logistic regression models. Note that some work [10], [11] validates the effectiveness of LP and ML in MLL, GLLE is also compared with them.

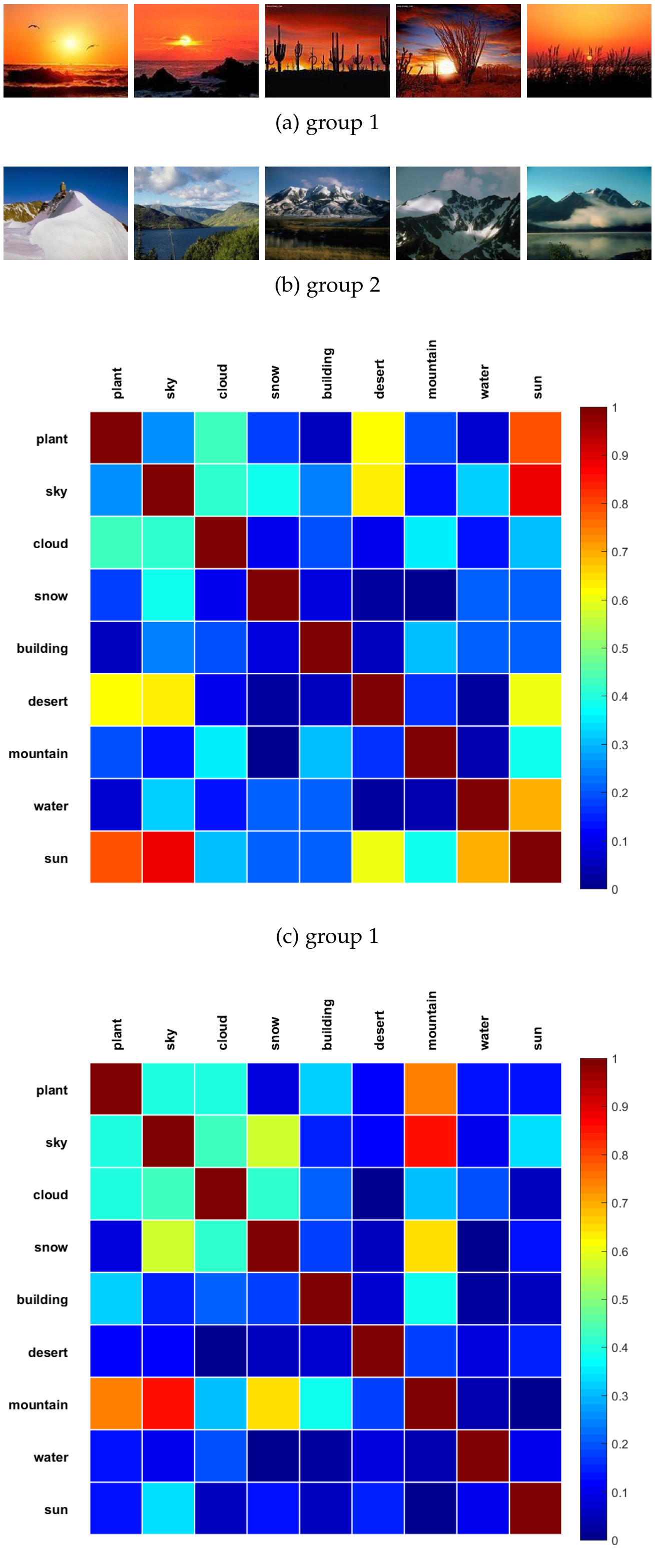
5.3.4 Predictive Performance

Table 6 tabulates the results of the three LE based algorithms (GLLE, LP and ML) and the four state-of-the-art algorithms (BR, CLR, ECC and RAKEL) on the ten MLL datasets evaluated by five evaluation metrics, and the best performance on each dataset is highlighted by boldface. For each evaluation metric, ↓ indicates the smaller the better while ↑ indicates the larger the better. All the algorithms are tested via ten-fold cross validation. The ranks are given in the parentheses right after the performance values. The average rank of each algorithm over all the datasets is also calculated and given in the last row of each table.

When looking at the average ranks over all the ten real-world datasets, GLLE achieves rather competitive performance over other LE algorithms. Besides, the rankings of each LE based algorithm on five measures are higher than the four state-of-the-art MLL algorithms. . When compared with the four state-of-the-art MLL algorithms, GLLE ranks 1st in 86.0% cases and ranks 2nd in 12.0% cases across, LP ranks 1st in 60.0% cases and ranks 2nd in 22.0% cases, ML ranks 1st in 78.0% cases and ranks 2nd in 18.0% cases, respectively. Thus, each LE based algorithm achieves rather superior performance over the four state-of-the-art multi-label learning algorithms across all the evaluation measures.

5.4 Label Correlations Exploration

As mentioned before, one of our methods advantages is that it could learn label correlations explicitly. In this part, the effectiveness of proposed algorithm in label correlations exploration is examined. The explorations are conducted on the real-world LDL dataset Nature Scene [8]. Nature Scene contains nine labels: plant, sky, cloud, snow, building, desert, mountain, water and sun.



(a) group 1

(b) group 2

(c) group 1

(d) group 2

Fig. 7: Illustration of learned label correlations on Nature Scene.

TABLE 6: Predictive performance of each algorithm (mean±std(rank)) measured by five MLL measures

Comparing	Ranking-loss ↓										Avg.
algorithm	cal500	emotions	medical	llog	enron	image	scene	msra	slashdot	corel5k	Rank
GLLE	**0.180±0.003(1)**	**0.171±0.008(1)**	**0.024±0.004(1)**	0.149±0.010(2)	0.103±0.005(4)	0.183±0.008(2)	0.095±0.004(2)	0.146±0.011(2)	0.115±0.004(2)	0.226±0.002(4)	2.1
LP	0.181±0.003(2)	0.182±0.012(2)	0.034±0.006(4)	**0.125±0.005(1)**	0.091±0.003(2)	**0.181±0.008(1)**	**0.087±0.006(1)**	**0.141±0.014(1)**	0.132±0.005(4)	0.145±0.002(3)	2.1
ML	0.203±0.004(3)	0.195±0.012(3)	**0.024±0.004(1)**	0.164±0.008(4)	0.094±0.002(3)	0.186±0.007(3)	0.096±0.010(3)	0.166±0.014(3)	**0.110±0.003(1)**	0.115±0.002(2)	2.7
BR	0.258±0.003(6)	0.233±0.016(6)	0.091±0.005(5)	0.328±0.007(6)	0.312±0.009(7)	0.314±0.014(7)	0.229±0.010(7)	0.368±0.021(7)	0.240±0.008(6)	0.416±0.003(6)	6.3
CLR	0.239±0.026(5)	0.222±0.014(4)	0.123±0.026(7)	0.190±0.015(5)	**0.089±0.002(1)**	0.294±0.009(5)	0.127±0.003(4)	0.288±0.018(5)	0.260±0.007(7)	**0.114±0.002(1)**	4.4
ECC	0.205±0.004(4)	0.227±0.017(5)	0.032±0.007(3)	0.154±0.009(3)	0.120±0.004(5)	0.276±0.005(6)	0.151±0.005(5)	0.332±0.047(6)	0.123±0.004(3)	0.292±0.003(5)	4.3
RAKEL	0.444±0.005(7)	0.254±0.020(7)	0.095±0.033(6)	0.412±0.010(7)	0.241±0.005(6)	0.311±0.010(4)	0.205±0.008(6)	0.223±0.075(4)	0.190±0.005(5)	0.627±0.004(7)	6.1
Comparing	One-error ↓										Avg.
algorithm	cal500	emotions	medical	llog	enron	image	scene	msra	slashdot	corel5k	Rank
GLLE	**0.115±0.013(1)**	**0.292±0.010(1)**	**0.158±0.010(1)**	**0.677±0.014(1)**	**0.249±0.011(1)**	0.341±0.017(2)	0.272±0.006(2)	**0.065±0.014(1)**	**0.414±0.010(1)**	**0.656±0.006(1)**	1.2
LP	0.120±0.015(3)	0.303±0.027(2)	0.213±0.021(5)	0.748±0.011(3)	0.311±0.013(3)	0.353±0.017(3)	**0.270±0.016(1)**	0.097±0.028(3)	0.558±0.009(5)	0.755±0.005(5)	3.3
ML	0.118±0.014(2)	0.319±0.031(3)	0.158±0.016(2)	0.684±0.017(2)	0.275±0.015(2)	**0.277±0.017(1)**	0.277±0.018(3)	0.093±0.030(2)	0.415±0.010(2)	0.701±0.007(3)	2.2
BR	0.921±0.025(7)	0.375±0.027(6)	0.297±0.036(6)	0.884±0.011(6)	0.648±0.019(7)	0.538±0.019(7)	0.475±0.014(7)	0.464±0.032(7)	0.734±0.017(6)	0.919±0.006(7)	6.6
CLR	0.331±0.111(6)	0.356±0.030(5)	0.688±0.143(7)	0.900±0.019(7)	0.376±0.017(4)	0.514±0.014(5)	0.371±0.008(4)	0.312±0.085(5)	0.979±0.003(7)	0.721±0.007(4)	5.4
ECC	0.191±0.021(4)	0.353±0.040(4)	0.182±0.019(3)	0.785±0.009(4)	0.424±0.013(6)	0.486±0.018(4)	0.373±0.008(5)	0.420±0.105(6)	0.481±0.014(4)	0.699±0.006(2)	4.2
RAKEL	0.286±0.039(5)	0.392±0.035(7)	0.208±0.071(4)	0.838±0.014(5)	0.412±0.016(5)	0.515±0.017(6)	0.444±0.012(6)	0.302±0.103(4)	0.453±0.005(3)	0.819±0.010(6)	5.1
Comparing	Coverage ↓										Avg.
algorithm	cal500	emotions	medical	llog	enron	image	scene	msra	slashdot	corel5k	Rank
GLLE	0.748±0.006(2)	**0.306±0.008(1)**	**0.039±0.006(1)**	**0.157±0.011(1)**	0.279±0.013(4)	0.199±0.006(2)	**0.093±0.003(1)**	0.563±0.016(2)	0.127±0.004(2)	0.492±0.004(4)	2
LP	**0.747±0.007(1)**	0.318±0.031(2)	0.052±0.001(4)	0.159±0.006(2)	**0.242±0.005(1)**	**0.198±0.007(1)**	0.171±0.009(5)	**0.543±0.020(1)**	0.148±0.005(4)	0.328±0.005(3)	2.4
ML	0.803±0.010(5)	0.328±0.011(3)	**0.039±0.006(1)**	0.167±0.009(3)	0.249±0.003(3)	0.204±0.007(3)	0.094±0.008(2)	0.592±0.018(3)	**0.120±0.003(1)**	0.273±0.004(2)	2.6
BR	0.852±0.014(6)	0.363±0.015(6)	0.118±0.007(6)	0.377±0.008(6)	0.601±0.014(7)	0.301±0.012(7)	0.207±0.009(7)	0.759±0.018(7)	0.259±0.009(6)	0.758±0.003(6)	6.4
CLR	0.794±0.010(4)	0.351±0.016(4)	0.143±0.030(7)	0.225±0.016(5)	0.243±0.006(2)	0.286±0.008(5)	0.120±0.007(3)	0.720±0.023(5)	0.272±0.007(7)	**0.267±0.004(1)**	4.3
ECC	0.788±0.008(3)	0.356±0.013(5)	0.048±0.009(3)	0.192±0.010(4)	0.300±0.009(5)	0.272±0.005(4)	0.141±0.004(4)	0.743±0.033(6)	0.139±0.004(3)	0.562±0.007(5)	4.2
RAKEL	0.971±0.001(7)	0.381±0.019(7)	0.117±0.040(5)	0.459±0.011(7)	0.523±0.008(6)	0.298±0.010(6)	0.186±0.006(6)	0.628±0.210(4)	0.212±0.005(5)	0.886±0.004(7)	6
Comparing	Hamming-loss ↓										Avg.
algorithm	cal500	emotions	medical	llog	enron	image	scene	msra	slashdot	corel5k	Rank
GLLE	**0.138±0.002(1)**	0.234±0.009(2)	0.012±0.001(2)	0.021±0.000(5)	**0.056±0.001(1)**	**0.183±0.007(1)**	**0.103±0.002(1)**	**0.205±0.008(1)**	**0.047±0.001(1)**	**0.010±0.000(1)**	1.6
LP	0.167±0.004(6)	**0.223±0.007(1)**	0.017±0.001(5)	**0.016±0.000(1)**	0.063±0.003(4)	0.190±0.005(3)	0.127±0.005(3)	0.279±0.017(4)	0.060±0.002(6)	0.024±0.000(6)	3.9
ML	0.141±0.002(3)	0.251±0.012(3)	0.012±0.001(2)	0.021±0.000(5)	0.057±0.001(2)	0.185±0.007(2)	0.104±0.006(2)	0.266±0.008(3)	**0.047±0.001(1)**	**0.010±0.000(1)**	2.4
BR	0.214±0.004(7)	0.265±0.013(5)	0.022±0.003(6)	0.052±0.003(7)	0.105±0.003(7)	0.287±0.008(6)	0.184±0.005(7)	0.404±0.037(7)	0.130±0.003(7)	0.027±0.000(7)	6.6
CLR	0.165±0.005(5)	0.270±0.011(7)	0.024±0.002(7)	0.019±0.002(4)	0.072±0.002(6)	0.305±0.005(7)	0.181±0.004(6)	0.342±0.033(5)	0.058±0.001(5)	0.011±0.001(3)	5.5
ECC	0.146±0.002(4)	0.254±0.013(4)	0.013±0.001(4)	**0.016±0.000(1)**	0.064±0.001(5)	0.244±0.005(4)	0.133±0.002(4)	0.353±0.037(6)	0.049±0.001(4)	0.015±0.001(5)	4.1
RAKEL	**0.138±0.002(1)**	0.269±0.011(6)	**0.010±0.003(1)**	0.017±0.001(3)	0.058±0.001(3)	0.286±0.007(5)	0.171±0.005(5)	0.237±0.079(2)	0.048±0.001(3)	0.012±0.001(4)	3.3
Comparing	Average-precision ↑										Avg.
algorithm	cal500	emotions	medical	llog	enron	image	scene	msra	slashdot	corel5k	Rank
GLLE	**0.501±0.003(1)**	**0.790±0.006(1)**	0.879±0.011(2)	**0.414±0.010(1)**	**0.678±0.007(1)**	**0.780±0.010(1)**	0.837±0.004(2)	**0.802±0.013(1)**	**0.676±0.006(1)**	0.271±0.003(3)	1.4
LP	0.496±0.005(2)	0.779±0.012(2)	0.837±0.018(4)	0.390±0.009(3)	0.661±0.007(3)	0.775±0.009(2)	**0.842±0.009(1)**	0.800±0.020(2)	0.579±0.009(5)	0.241±0.002(5)	2.9
ML	0.474±0.004(3)	0.768±0.013(3)	**0.881±0.014(1)**	0.396±0.009(2)	0.670±0.005(2)	0.775±0.008(3)	0.834±0.012(3)	0.775±0.015(3)	0.675±0.007(2)	**0.284±0.002(1)**	2.3
BR	0.300±0.005(7)	0.730±0.015(6)	0.762±0.022(5)	0.215±0.009(5)	0.381±0.009(7)	0.649±0.012(7)	0.692±0.010(7)	0.540±0.015(7)	0.427±0.014(6)	0.123±0.003(6)	6.3
CLR	0.395±0.042(5)	0.742±0.016(4)	0.400±0.062(7)	0.194±0.018(7)	0.610±0.008(4)	0.666±0.008(5)	0.778±0.004(4)	0.624±0.022(4)	0.250±0.007(7)	0.274±0.002(2)	4.9
ECC	0.463±0.006(4)	0.740±0.021(5)	0.860±0.015(3)	0.342±0.009(4)	0.559±0.008(5)	0.685±0.008(4)	0.766±0.005(5)	0.567±0.048(6)	0.628±0.009(3)	0.264±0.003(4)	4.3
RAKEL	0.353±0.006(6)	0.717±0.023(7)	0.700±0.234(6)	0.197±0.013(6)	0.539±0.006(6)	0.661±0.010(6)	0.713±0.008(6)	0.601±0.200(5)	0.617±0.004(4)	0.122±0.004(7)	5.9

To show the label correlations learned by GLLE, we use two local groups extracted from Nature Scene. The label correlation learned by GLLE are shown in Fig. 7, and the value in label correlation matrix is scaled into $[0, 1]$. Fig. 7 shows that local label correlation does vary from group to group. For instance, "sun" is highly correlated with "plant", "sky", "desert" and "water" (Fig. 7(c)) in group 1. This can also be seen from the images in Fig. 7(a). Moreover, "desert" co-occurs with "sky" and "sun". In group 2 (Fig. 7(d)), "mountain"is highly correlated with "plant", "sky" and "snow", whereas "desert" occurs less often with plant (Fig. 7(b)). All these correlations are consistent with intuition.

6 Conclusion

This paper shows *label enhancement*, which reinforces the supervision information in the training sets. LE can recover the label distributions from the logical labels in the training sets by utilizing the topological information of the feature space and the correlation among the labels. In order to solve the LE problem, we introduce existing algorithms that can be used for LE and propose a novel method called GLLE. Extensive comparative studies clearly validate the advantage of GLLE against other LE algorithms and the effectiveness of LDL after LE pre-process on the logical-labeled datasets.

LE is motivated by the LDL learning framework on the datasets denoted by logical labels. However, LE might also be used to solve other kinds of problems. Generally speaking, there are at least three scenarios where LE could be helpful:

- There is a dataset that associates the logical labels with the instances. This is the most direct application of LE, as described in this paper.
- There are training examples each associated with a set of candidate labels, among which only one label is valid for the training example. LE could recover the label relevance over each candidate label and the label irrelevance over each non-candidate label.
- There are training examples each associated with logical labels, where some labels are missing. LE could recover the description degrees of all the labels.

Each of the three scenarios actually covers a vast area of applications. A lot of interesting work, both at the theoretical level and at the application level, may be conducted in the future.

7 Acknowledgments

This research was supported by the National Key Research & Development Plan of China (No. 2017YFB1002801), the National Science Foundation of China (61622203), the Jiangsu Natural Science Funds for Distinguished Young Scholar (BK20140022), the Collaborative Innovation Center of Novel Software Technology and Industrialization, and the Collaborative Innovation Center of Wireless Communications Technology.

References

[1] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2006.

[2] E. Gibaja and S. Ventura, "A tutorial on multilabel learning," *ACM Computing Surveys*, vol. 47, no. 3, pp. 1–38, 2015.

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

[4] T. N. Rubin, A. Chambers, P. Smyth, and M. Steyvers, "Statistical topic models for multi-label document classification," *Machine Learning*, vol. 88, no. 1-2, pp. 157–208, 2012.

[5] R. S. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino, "Matrix completion for multi-label image classification," in *Advances in Neural Information Processing Systems*, Granada, Spain, 2011, pp. 190–198.

[6] H.-Y. Lo, J.-C. Wang, H.-M. Wang, and S.-D. Lin, "Cost-sensitive multi-label learning for audio tag annotation and retrieval," *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 518–529, 2011.

[7] J. Wang, Y. Zhao, X. Wu, and X.-S. Hua, "A transductive multi-label learning approach for video concept detection," *Pattern Recognition*, vol. 44, no. 10, pp. 2274–2286, 2011.

[8] X. Geng, "Label distribution learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1734–1748, 2016.

[9] Y. Zhou, H. Xue, and X. Geng, "Emotion distribution recognition from facial expressions," in *Proceedings of the 23rd ACM International Conference on Multimedia*, Brisbane, Australia, 2015, pp. 1247–1250.

[10] Y.-K. Li, M.-L. Zhang, and X. Geng, "Leveraging implicit relative labeling-importance information for effective multi-label learning," in *Proceedings of the 15th IEEE International Conference on Data Mining*, Atlantic City, NJ, 2015, pp. 251–260.

[11] P. Hou, X. Geng, and M.-L. Zhang, "Multi-label manifold learning," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, Phoenix, AZ, 2016, pp. 1680–1686.

[12] X. Geng and Y. Xia, "Head pose estimation based on multivariate label distribution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 1837–1842.

[13] X. Geng, Q. Wang, and Y. Xia, "Facial age estimation by adaptive label distribution learning," in *Proceedings of the 22nd International Conference on Pattern Recognition*, Stockholm, Sweden, 2014, pp. 4465–4470.

[14] N. E. Gayar, F. Schwenker, and G. Palm, "A study of the robustness of knn classifiers trained using soft labels," in *Proceedings of the 2nd International Conference on Artificial Neural Network in Pattern Recognition*, Ulm, Germany, 2006, pp. 67–80.

[15] X. Jiang, Z. Yi, and J. C. Lv, "Fuzzy svm with a new fuzzy membership function," *Neural Computing & Applications*, vol. 15, no. 3-4, pp. 268–276, 2006.

[16] X. Ning, T. An, and G. Xin, "Label enhancement for label distribution learning," in *Proceedings of the International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 2018, pp. 2926–2932.

[17] K. Su and X. Geng, "Soft facial landmark detection by label distribution learning," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, Honolulu, HI, 2019, p. in press.

[18] B.-B. Gao, H.-Y. Zhou, J. Wu, and X. Geng, "Age estimation using expectation of label distribution learning," in *Proceedings of the International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 2018, pp. 712–718.

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

[20] X. Geng and L. Luo, "Multilabel ranking with inconsistent rankers," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 3742–3747.

[21] Z. Huo and X. Geng, "Ordinal zero-shot learning," in *Proceedings of the International Joint Conference on Artificial Intelligence*, Melbourne, Australia, 2017, pp. 1331–1337.

[22] D. Zhou, Y. Zhou, X. Zhang, Q. Zhao, and X. Geng, "Emotion distribution learning from texts," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Austin, TX, 2016, pp. 638–647.

[23] J. Wang and X. Geng, "Theoretical analysis of label distribution learning," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, Honolulu, HI, 2019, p. in press.

[24] B. Quost and T. Denœux, "Learning from data with uncertain labels by boosting credal classifiers," in *Proceedings of the 1st ACM SIGKDD Workshop on Knowledge Discovery from Uncertain Data*, Paris, France, 2009, pp. 38–47.

[25] T. Denœux and L. M. Zouhal, "Handling possibilistic labels in pattern classification using evidential reasoning," *Fuzzy sets and systems*, vol. 122, no. 3, pp. 409–424, 2001.

[26] P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi, "Learning with probabilistic supervision," *Computational learning theory and natural learning systems*, vol. 3, pp. 163–182, 1995.

[27] O. Castillo and P. Melin, *Hybrid intelligent systems for pattern recognition using soft computing: An evolutionary approach for neural networks and fuzzy systems*. Heidelberg: Springer, 2005.

[28] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 3, no. 1, pp. 1–130, 2009.

[29] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[30] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 55–67, 2008.

[31] X. Zhu, J. Lafferty, and R. Rosenfeld, *Semi-supervised learning with graphs*. Carnegie Mellon University, language technologies institute, school of computer science, 2005.

[32] G. Tsoumakas, A. Dimou, E. Spyromitros, and V. Mezaris, "Correlation-based pruning of stacked binary relevance models for multi-label learning," in *Proceedings of the 1st International Workshop on Learning from Multi-Label Data*, Bled, Slovenia, 2009, pp. 101–116.

[33] S.-J. Huang and Z.-H. Zhou, "Multi-label learning by exploiting label correlations locally," in *Twenty-sixth AAAI conference on artificial intelligence*, Toronto, Canada, 2012, pp. 949–955.

[34] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li, "Multi-instance multi-label learning," *Artificial Intelligence*, vol. 176, no. 1, pp. 2291–2320, 2012.

[35] A. J. Smola, *Learning with kernels*. Ph.D. Thesis, GMD, Birlinghoven, German, 1999.

[36] J. Nocedal and S. J. Wright, *Numerical optimization*. New York: Springer, 2006.

[37] R. J. Campello, "A fuzzy extension of the rand index and other related indexes for clustering and classification assessment," *Pattern Recognition Letters*, vol. 28, no. 7, pp. 833–841, 2007.

[38] S. Zhang, H.-S. Wong, and Y. Shen, "Generalized adjusted rand indices for cluster ensembles," *Pattern Recognition*, vol. 45, no. 6, pp. 2214–2226, 2012.

[39] S. Zhang, Z. Yang, X. Xing, Y. Gao, D. Xie, and H.-S. Wong, "Generalized pair-counting similarity measures for clustering and cluster ensembles," *IEEE Access*, vol. 5, pp. 16 904–16 918, 2017.

[40] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *City*, vol. 1, no. 2, p. 1, 2007.

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

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

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

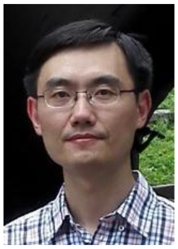
[44] J. Read, B. Pfahringer, and G. Holmes, "Multi-label classification using ensembles of pruned sets," in *IEEE International Conference on Data Mining*, Pisa, Italy, 2008, pp. 995–1000.

[45] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multilabel classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1079–1089, 2011.

[46] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, "Mulan: A java library for multi-label learning," *Journal of Machine learning research*, vol. 12, no. Jul, pp. 2411–2414, 2011.

Ning Xu received the BSc degree in computer science from University of Science and Technology of China, China, in 2010. In the same year, he was admitted to further study at Chinese Academy of Sciences, China. Now, he is currently working toward the PhD degree. His research interests mainly include pattern recognition and machine learning.

Yun-Peng Liu, born in 1994. Master candidate. His main research interests include machine learning and computer vision.



Xin Geng 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 joined the School of Computer Science and Engineering at Southeast University, China, in 2008, and is currently a professor and vice dean (research) of the school. His research interests include pattern recognition, machine learning, and computer vision. He has published more than 50 refereed papers and holds six patents in these areas.