Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Label enhancement via manifold approximation and projection with graph convolutional network

Chao Tan [a,*], Sheng Chen [b], Xin Geng [c], Yunyao Zhou [a], Genlin Ji [a]

[a] *School of Computer and Electronic Information/School of Artificial Intelligence, Nanjing Normal University, Nanjing 210023, China*
[b] *School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK*
[c] *School of Computer Science and Engineering, Southeast University, Nanjing 210096, China*

ARTICLE INFO

Keywords:
Multi-label classification
Label distribution learning
Manifold learning
Robust linear regression
Graph convolutional network

ABSTRACT

Label enhancement (LE) aims to enrich logical labels into their corresponding label distributions. But existing LE algorithms fail to fully leverage the structural information in the feature space to improve LE learning. To address this key issue, we first apply manifold learning to map the relatedness between low-dimensional feature samples to the label space. Based on the smoothness assumption of manifolds, the implicit correlation between low-dimensional feature and label spaces effectively promotes the LE process, enabling the learning model to accurately capture the mapping relationship between feature and label manifolds. This leads to an LE based on feature representation (LEFR) algorithm. We also propose an LE algorithm based on graph convolutional network (GCN), called LE-GCN. Inspired by the relationship between threshold connections and label connections, we extend GCN to the LE field for the first time to fully exploit the hidden relationships between nodes and labels. By enhancing node information with threshold connections and label connections, the label learning accuracy reaches a new level. Experiments on real-world datasets show that our LEFR and LE-GCN outperform several state-of-the-art LE algorithms.

1. Introduction

In the past few years, learning with ambiguity has become a research hotspot in the field of machine learning and data mining. A large number of studies have shown that multi-label learning (MLL) is an effective learning method [1], but there still exist many challenges that remain to be tackled for this important learning paradigm. Although the importance of each label is typically considered to be equal in diverse MLL applications, that is, the contribution degree of each label to the example is assumed to be equal, the importance of different labels to the example is often different for many real-world problems. For example, expressions usually contain several different emotional components. In some cases, we not only need to know the several emotional components that the expression contains, but also need to understand the strength of these emotional components in the expression.

To address this critical issue, Geng [2] proposed a novel label distribution learning (LDL) paradigm. The key idea underpinning LDL is as follows. The description degree of all the labels related to an instance constitutes a real-valued vector called label distribution, which describes the instance more comprehensively than the logical labels. LDL methods have been successfully applied to many problems, such as age estimation [3], emotion classification [4], soft video parsing [5], person re-identification [6], etc. In order to apply the LDL method to fully learn label information, we firstly need to obtain the label distributions of the dataset. However, instances in real life are often annotated with logical labels instead of label distributions, and manually annotating instances with label distributions is time-consuming and costly. Getting the label distributions of the dataset is often very difficult in most training sets [7].

Although most real-life data are labeled with logical labels [8], the supervision information in the data follows certain label distribution. The method of recovering this hidden supervision information is called label enhancement (LE) [7]. LE is the process of transforming the original logical labels in the training samples to label distributions. It uses the label correlation implicit in the data to effectively strengthen the supervision information of the examples, and thus enables the LDL to obtain better prediction results. This process is illustrated in Fig. 1. This landscape image contains complete information about the sample. But the logical labeling only assigns value of 1 to some important information, such as 'mountain' and 'building', and it may ignore/miss

* Corresponding author.
E-mail addresses: 73022@njnu.edu.cn (C. Tan), sqc@ecs.soton.ac.uk (S. Chen), xgeng@seu.edu.cn (X. Geng), 21200526@njnu.edu.cn (Y. Zhou), glji@njnu.edu.cn (G. Ji).

https://doi.org/10.1016/j.patcog.2024.110447
Received 30 April 2023; Received in revised form 8 March 2024; Accepted 20 March 2024
Available online 24 March 2024

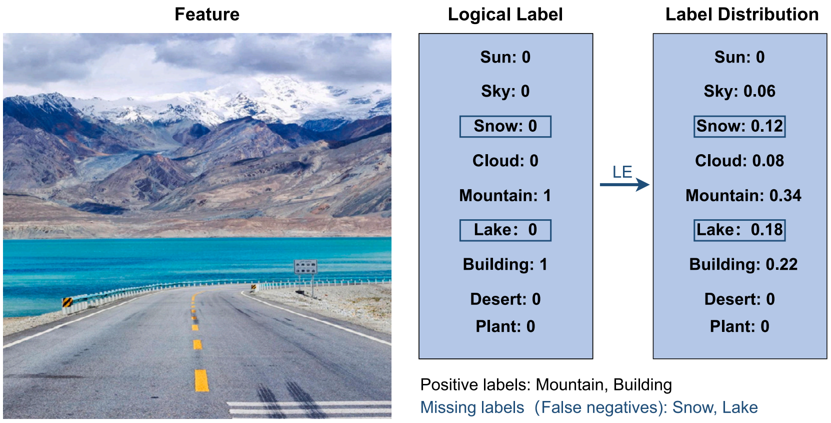

Fig. 1. An example of label enhancement. Some labels, such as 'lake' and 'snow', are missing (false negative). LE methods can pay special attention to recover these missing labels.

relatively important information like 'lake' and 'snow'. LE can enhance the important information within samples by assigning a descriptive measure to each label based on its significance.

Effective LE methods should be capable of mining the low-dimensional topological structure information of the feature space, transferring the sample correlation information to the label space, and enhancing the logical labels into label distributions. In this context, this paper proposes an LE method based on feature representation, referred to as LEFR. First, we use the manifold learning method, known as local tangent space alignment (LTSA) [9], to obtain the low-dimensional structure of data points, and construct a new similarity matrix to map the correlation of samples in the low-dimensional feature space onto the label space. The initial label distributions can be obtained through this manifold learning process. Then, the label distribution prediction model is constructed based on the manifold structure. According to the smoothness property of manifold, the implicit correlation in the low-dimensional feature space and label space effectively facilitates the LE process, and the learning model can accurately capture the mapping relationship between the feature manifold space and label space. The experimental results show that the proposed LEFR model outperforms several state-of-the-art techniques, in terms of label distribution prediction performance.

The traditional graph similarity matrix is based on the co-occurrence probability matrix between labels or established according to the index relationship between nodes but nodes and labels are not combined to construct a graph [10,11]. In the work [12], for example, each node and the label are fully connected, and there is no information to be exploited. By extending the graph convolutional network (GCN) [13] to the LE field for the first time, this paper also proposes an LE algorithm based on GCN, called LE-GCN. Our inspiration is the relationship between the threshold connection and the label connection. Specifically, the feature nodes of the original samples are connected through a distance threshold, the hidden relationship between the feature nodes is extracted, and the feature nodes and their corresponding marked nodes are connected to fully mine the hidden relationship between the nodes and the markers. This realizes the information reshaping of feature space and label space. Then the information is injected into the GCN. The GCN uses this reshaping information to aggregate and update the training sample node information, and obtain a new feature node. We design an objective function to optimize our GCN output. The distance between the predicted label distributions obtained by the GCN using the reshaping information and label propagation (LP) constitutes this objective function, thereby obtaining the prediction score corresponding to each instance label.

In summary, our main contributions are as follows.

1. The first proposed LE method, called the LEFR, builds a framework based on feature representations and applies manifold learning to extract underlying low-dimensional feature information. Intermediate label distributions are obtained via LP. A novel sample similarity matrix mines relationships between sample feature and label spaces. By combining low-dimensional feature space, label information and estimated distributions, an enhanced label distribution prediction model is established and trained via gradient descent optimization to achieve accurate predictions.
2. The second proposed LE algorithm, referred to as LE-GCN, employs the GCN to determine feature node connection thresholds based on similarity attributes and enable information reshaping between the feature and label spaces, which allows LE-GCN to generate accurate label distributions. LE-GCN includes virtual label nodes and weighted edges between samples and their corresponding single labels. Effective information propagation is enabled by GCN within the feature space, where feature representations are transformed and passed between connected nodes. Subsequently, under the operation of LP, the high-quality propagated feature information is accurately converted into label distribution representations.
3. The experimental results validate that both LEFR and LE-GCN outperform the existing state-of-the-art LE methods, in terms of label recovery and label prediction performance. In particular, the proposed LE-GCN achieves the best label recovery accuracy, owing to its GCN propagation mechanism which iteratively converts the feature space information propagation driven by the virtual labels into label distribution representations. The proposed LEFR attains the best label prediction accuracy through its natural integration of manifold space, label space and similarity weights as well as optimization with an effective quasi-Newton method.

The rest of this paper is organized as follows. The related work is presented in Section 2. Our proposed two algorithms are detailed in Section 3. The results of comparative experiments under different tasks are reported in Section 4. The paper is concluded in Section 5.

2. Survey of existing LE algorithms

The existing LE algorithms can be divided into the three categories. The first category is the LE based on fuzzy theory, which utilizes the fuzzy mathematics to construct the fuzzy membership degree of each label class through fuzzy operation or fuzzy clustering. Typical fuzzy-based LE methods include the fuzzy clustering-based LE algorithm [14] and kernel-based LE algorithm [15]. It is worth noting that the purpose of this type of methods is generally to introduce ambiguity into the originally logical labels. But it is not clear how this can enhance the logical label into a label distribution. Another category is the graph-based LE, which uses graph models to represent topological structures of instances, and enhances logical labels into label distributions by establishing the relationship between the instance correlations and the label correlations. Typical graph-based LE methods include the LE based on label propagation (LP) [1], the LE based on manifold learning (ML) [16], the graph Laplacian-based LE (GLLE) [7], the LE with sample correlations (LESC) [17], and the privileged LE with multi-label learning (PLEML) [18]. The third category includes the generative LE methods. LEVI [19] and GLEMR [20] consider the label distribution as a latent variable and infer the approximate posterior density of the label distribution based on the variational lower bound to improve recovery performance. ConLE [21] integrates features and logical labels into a unified projection space, and employs an adversarial learning strategy to bring the features and logical labels of the same instance closer in the projection space.

We first introduce the following variables. Let $\boldsymbol{X} = [\boldsymbol{x}_1\ \boldsymbol{x}_2 \cdots \boldsymbol{x}_n] \in \mathbb{R}^{m\times n}$ be the feature matrix of instances, where n is the number of instances and m is the feature dimension. Further let the logical label vector of instance $\boldsymbol{x}_i$ be given by $\boldsymbol{y}_i = \left[y_{i,1}\ y_{i,2} \cdots y_{i,c}\right]^{\mathrm{T}} \in \{0,\ 1\}^c$, where c is the number of labels, $y_{i,k}=1$ if the label $y_{i,k}$ is relevant to $\boldsymbol{x}_i$, and otherwise $y_{i,k}=0$. The logic label matrix is defined by $\boldsymbol{Y}=\left[\boldsymbol{y}_1\ \boldsymbol{y}_2 \cdots \boldsymbol{y}_n\right]$. Moreover, let $\boldsymbol{d}_i = \boldsymbol{d}_{\boldsymbol{x}_i}^{\boldsymbol{y}_i} = \left[d_{\boldsymbol{x}_i}^{y_{i,1}}\ d_{\boldsymbol{x}_i}^{y_{i,2}} \cdots d_{\boldsymbol{x}_i}^{y_{i,c}}\right]^{\mathrm{T}}$ represent the unknown ground-truth label distribution of $\boldsymbol{x}_i$.

2.1. LE based on label propagation (LP)

Zhang et al. [1] applied the LP method in semi-supervised learning to LE. A fully associative graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$ is first constructed using all the training instances, where each vertex of the graph is an instance, i.e., $\mathcal{V} = \{\boldsymbol{x}_i\}_{i=1}^n$, and $\mathcal{E}$ denotes the set of edges. According to $\mathcal{G}$, the feature similarity matrix $\boldsymbol{W}=\left[w_{i,j}\right]_{n\times n}$ between the instances is defined as

$$w_{i,j} = \begin{cases} \exp\left(-\dfrac{\left\|x_i - x_j\right\|^2}{2\tau^2}\right), & i \neq j, \\ 0, & i = j, \end{cases} \quad 1 \leq i, j \leq n, \tag{1}$$

where τ is a hyperparameter. The LP matrix $\boldsymbol{P}$ is constructed from $\boldsymbol{W}$ as $\boldsymbol{P}=\bar{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{W}\bar{\boldsymbol{D}}^{-\frac{1}{2}}$, where the diagonal matrix $\bar{\boldsymbol{D}}=\mathrm{diag}\left\{\bar{d}_1, \ldots, \bar{d}_n\right\}$ and the diagonal element $\bar{d}_i=\sum_{j=1}^n w_{i,j}$ is the sum of all the elements in the ith row of $\boldsymbol{W}$. Let $\boldsymbol{F}=\left[f_{i,j}\right]_{n\times c}$ be the label importance matrix, whose initial state is defined as $\boldsymbol{F}^{(0)}=\boldsymbol{Y}^{\mathrm{T}}$. At the tth iteration, $\boldsymbol{F}$ is updated according to $\boldsymbol{P}$ as

$$\boldsymbol{F}^{(t)} = \alpha\boldsymbol{P}\boldsymbol{F}^{(t-1)} + (1-\alpha)\boldsymbol{Y}^{\mathrm{T}}, \tag{2}$$

where $\alpha \in (0,\ 1)$ is the balancing parameter which controls the fraction of the information inherited from the LP matrix and the logical label matrix. Clearly,

$$\boldsymbol{F}^{(t)} = (\alpha\boldsymbol{P})^t\boldsymbol{Y}^{\mathrm{T}} + (1-\alpha)\sum_{i=0}^{t-1}(\alpha\boldsymbol{P})^i\boldsymbol{Y}^{\mathrm{T}}, \tag{3}$$

and $\boldsymbol{F}^{(t)}$ converges to

$$\boldsymbol{F}^{\star} = (1-\alpha)(\boldsymbol{I}_n - \alpha\boldsymbol{P})^{-1}\boldsymbol{Y}^{\mathrm{T}}, \tag{4}$$

where $\boldsymbol{I}_n$ denotes the $(n\times n)$ identity matrix. Each row of $\boldsymbol{F}^{\star}$ is then normalized to obtain the estimated label distribution

$$\hat{d}_{\boldsymbol{x}_i}^{y_{i,j}} = \frac{f_{i,j}^{\star}}{\sum_{k=1}^c f_{i,k}^{\star}},\ 1 \leq i \leq n,\ 1 \leq j \leq c. \tag{5}$$

The LE algorithm based on LP represents the topological structure between instances by using graph model. It first constructs a LP matrix based on the correlation between examples, and then uses the different path weights in the propagation process to make the description degree of different labels different, to reflect the inter-label relationship embedding in the training data. However, this LP algorithm imposes high complexity, as it calculates the paired distances in the whole feature space. The more serious point is that this LP algorithm is essentially the propagation of logical label and uses the normalization to force the logical label to be the label distribution, which cannot reflect the essence of LE: namely, ability to predict the label distribution of unknown instances through the relationship between known instances.

2.2. LE algorithm based on manifold learning (ML)

Hou et al. [16] proposed an LE method based on the manifold learning (ML). Similar to the LE based on LP, a fully connected graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$ is constructed using the training examples. Again let $\boldsymbol{W} = \left[w_{i,j}\right]_{n\times n}$ be the weight matrix, with $w_{i,j}$ representing the weight of the edge connecting $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. To explore the local topological structure in the training set, the local topological structure between the examples, namely, $\boldsymbol{W}$, is obtained by solving a quadratic programming problem according to the locally linear embedding [22]. The matrix $\boldsymbol{W}$ is used to describe the local reconstruction relationship between the examples. This local reconstruction relationship between examples is migrated to the label space, and the label distributions can be reconstructed/estimated by solving anther quadratic programming problem.

The ML establishes the relationship between instances' correlation and labels' correlation based on the smoothness hypothesis. It needs to reconstruct the structural information in the label space from the feature space, and then through the quadratic programming to solve the label distribution estimation. These two steps need to construct the separate objective functions, and they are solved separately, which inevitably reduces the effectiveness and the prediction accuracy of the algorithm. Like the LP algorithm [1], the ML itself has no direct ability to predict the label distribution of new sample unseen in training.

2.3. Graph Laplacian-based LE (GLLE)

Given the training feature matrix $\boldsymbol{X}$, to recover the label distribution matrix $\boldsymbol{D} = \left[\boldsymbol{d}_1 \cdots \boldsymbol{d}_n\right]$ from the logical label matrix $\boldsymbol{Y}$, GLLE [7] constructs the model

$$\boldsymbol{d}_i = \boldsymbol{\Omega}^{\mathrm{T}}\varphi(\boldsymbol{x}_i) + \boldsymbol{b} = \bar{\boldsymbol{\Omega}}\boldsymbol{\phi}_i, \tag{6}$$

by solving the optimization problem

$$\min_{\bar{\boldsymbol{\Omega}}} L\left(\bar{\boldsymbol{\Omega}}\right) + \lambda\Xi\left(\bar{\boldsymbol{\Omega}}\right). \tag{7}$$

where $\boldsymbol{\Omega} = \left[\boldsymbol{\omega}_1 \cdots \boldsymbol{\omega}_c\right]$ is a weight matrix, $\boldsymbol{b} \in \mathbb{R}^c$ is a bias vector, and $\varphi(\boldsymbol{x})$ is a nonlinear transformation of $\boldsymbol{x}$, while $\bar{\boldsymbol{\Omega}}=\left[\boldsymbol{\Omega}^{\mathrm{T}}\ \boldsymbol{b}\right]$, $\boldsymbol{\phi}_i=\left[\varphi(\boldsymbol{x}_i)\ 1\right]$ and λ is a hyperparameter. The least squares (LS) loss function $L(\cdot)$ is chosen to be

$$L\left(\bar{\boldsymbol{\Omega}}\right)=\sum_{i=1}^n \left\|\bar{\boldsymbol{\Omega}}\boldsymbol{\phi}_i - \boldsymbol{y}_i\right\|^2 = \mathrm{tr}\left(\left(\bar{\boldsymbol{\Omega}}\boldsymbol{\Phi}-\boldsymbol{Y}\right)^{\mathrm{T}}\left(\bar{\boldsymbol{\Omega}}\boldsymbol{\Phi}-\boldsymbol{Y}\right)\right), \tag{8}$$

where $\boldsymbol{\Phi} = \left[\boldsymbol{\phi}_1 \cdots \boldsymbol{\phi}_n\right]$. To mine the hidden label importance from the training examples by exploiting the topological information of the feature space, the authors of [7] specified the local similarity matrix $\boldsymbol{A} = \left[a_{i,j}\right]_{n\times n}$ with

$$a_{i,j} = \begin{cases} \exp\left(-\dfrac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right), & \text{if } \boldsymbol{x}_j \in \boldsymbol{K}(i), \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

where $\boldsymbol{K}(i)$ denotes the set of $\boldsymbol{x}_i$'s K-nearest neighbors, and σ is the width parameter. The cost function $\Xi(\cdot)$ is used to mine hidden label's importance. According to the smoothness assumption [23], the points close to each other are more likely to share a label. This intuition leads to:

$$\Xi\left(\bar{\boldsymbol{\Omega}}\right) = \sum_{i,j} a_{i,j}\left\|\boldsymbol{d}_i - \boldsymbol{d}_j\right\|^2 = \mathrm{tr}\left(\boldsymbol{D}\boldsymbol{G}\boldsymbol{D}^{\mathrm{T}}\right)\mathrm{tr}\left(\bar{\boldsymbol{\Omega}}\boldsymbol{\Phi}\boldsymbol{G}\bar{\boldsymbol{\Omega}}^{\mathrm{T}}\boldsymbol{\Phi}^{\mathrm{T}}\right), \tag{10}$$

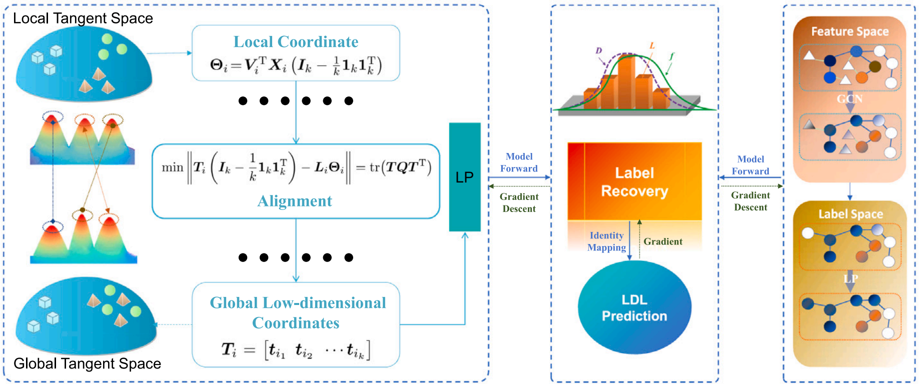
Local Tangent Space
Local Coordinate
$\Theta_i = V_i^{\mathrm{T}} X_i \left(I_k - \frac{1}{k}\mathbf{1}_k\mathbf{1}_k^{\mathrm{T}}\right)$
$\min \left\| T_i \left(I_k - \frac{1}{k}\mathbf{1}_k\mathbf{1}_k^{\mathrm{T}}\right) - L_i\Theta_i \right\| = \mathrm{tr}(TQT^{\mathrm{T}})$
Alignment
Global Low-dimensional Coordinates
$T_i = [t_{i_1}\ t_{i_2}\ \cdots t_{i_k}]$
Global Tangent Space
LP
Model Forward
Gradient Descent
D
L
f
Label Recovery
Identity Mapping
Gradient
LDL Prediction
Model Forward
Gradient Descent
Feature Space
GCN
Label Space
LP

Fig. 2. An overall framework diagram of the two proposed LE algorithms. LEFR is shown in the left and middle parts, and LE-GCN is illustrated in the right and middle parts.

where $G = \bar{A} - A$ is the graph Laplacian, and the diagonal matrix $\bar{A} = \mathrm{diag}\{\bar{a}_1, \ldots, \bar{a}_n\}$ has the diagonal elements $\bar{a}_i = \sum_{j=1}^{n} a_{i,j}$, $1 \le i \le n$. The optimization problem (7) is solved by an effective quasi-Newton method called BFGS [24] to determine the best parameter $\bar{\Omega}^{\star}$. Finally, the label distribution d_i can be generated and normalized by the softmax normalization [7].

Unlike the LP algorithm [1], which imposes high complexity by calculating the distances between all the samples, GLLE calculates the distances between the instance and its K nearest neighbors. GLLE also integrates the topological information of the feature space and the loss function that predicts the label distribution into a single combined objective function, thus, avoiding the need to solve the problem in two steps, as in the case of the ML algorithm [16]. Moreover, since during the LE enhance process, GLLE builds a label predictor model, it has the ability to directly predict the label distribution of new sample unseen in training. However, GLLE has a natural ‘defect’. The first loss function in its objective function constructs a weighted linear model based on the logical label of instances, which directly approximates the predicted label distribution. This does not conform to the ‘physical’ interpretation of label distribution, which indicates the extent to which original label describe instances.

2.4. LE with sample correlations (LESC)

The LE with sample correlations (LESC) via low-rank representation algorithm [17] obtains the label distribution by exploiting the low-rank representation to excavate the global information in the feature space, which is different from GLLE [7] that exploits the local similarity. Similar to GLLE, the LS loss function is adopted as the first term of the optimization objective function. The difference is that LESC adopts the low-rank representation to construct the second term of the optimization formula. The minimized low-rank representation of the feature space is obtained by seeking the low-rank representation among the feature matrix to excavate the global structure of the feature space. Finally, the BFGS algorithm [24] is adopted to solve this optimization and hence to obtain the label distributions.

However, it is difficult to determine the convergence of BFGS applied to the LESC optimization problem. A common practice is to manually set the number of iterations. This is time consuming for large size problems. Like GLLE [7], during the LE process, LESC builds a label predictor model and therefore has the ability to predict the label distribution of new sample unseen in training.

2.5. Privileged LE with multi-label learning (PLEML)

Zhu et al. [18] proposed a privileged LE method with MLL (PLEML). First, it applies an MLL model to generate an auxiliary information for LE. Second, PLEML adopts the learning using privileged information (LUPI) paradigm [25], which is supplied by a teacher about instances at the training stage, to make reasonable use of additional information. Finally, PLEML applies the RSVM+ [25] as the final prediction model, which is a support vector machine discriminative model implementing LUPI.

Although PLEML first utilizes the labels’ correlation in the label space to generate the auxiliary label distribution, the algorithm is divided into two steps, and the label information is lost to a certain extent. Hence, PLEML does not make full use of the correlation between examples in the feature space, and the effect of this is to make the algorithm suboptimal. Since during the LE enhance process, PLEML builds a label predictor model, it has the ability to directly predict the label distribution of new sample unseen in training.

3. Our proposed approaches

First, we define the generic LE problem. Given the training set $\mathcal{S}_{\mathrm{ll}} = \{X, Y\}$, LE recovers the label distribution $\hat{d}_i = \hat{d}_{x_i}^{y_i}$ of x_i from the logical label y_i and converts $\mathcal{S}_{\mathrm{ll}}$ to the dataset with label distributions $\hat{\mathcal{S}}_{\mathrm{ld}} = \{X, \hat{D}\}$, where the estimated label distributions $\hat{D} = \left[\hat{d}_{x_1}^{y_1} \cdots \hat{d}_{x_n}^{y_n}\right]$ satisfy $\hat{d}_{x_i}^{y_{i,j}} \in [0,\ 1]$ and $\sum_{j=1}^{c} \hat{d}_{x_i}^{y_{i,j}} = 1$, $1 \le i \le n$ and $1 \le j \le c$. The goal is to make the estimated label distribution set $\hat{D}$ as close as possible to the unknown true label distribution set $D = [d_1 \cdots d_n]$. Fig. 2 depicts the overall framework of the two proposed LE algorithms. Both LEFR and LE-GCN first learn low-dimensional representations. Specifically, LEFR applies manifold learning on feature embeddings to implicitly correlate the feature and label spaces (Left part of Fig. 2), while LE-GCN constructs a graph representation of samples, labels and their relationships, and utilizes GCNs to propagate and transform features on this graph structure (Right part of Fig. 2). Next, they leverage LP on the learned embeddings to recover label distributions (Middle part of Fig. 2). In particular, LEFR incorporates LP as the loss function to obtain the estimated label distributions from the representations learned, while LE-GCN iterates LP to estimate label distributions on the representations extracted. Both the algorithms then use the enhanced training data to learn a predictive model (Middle part of Fig. 2), by employing a logistic regression classifier as the predictor for label distribution predictions. We now detail our two proposed approaches.

3.1. LEFR approach

(1) Sample correlation via low dimensional feature representation: In real-world data, the samples with similar features usually share the labels that are similar to each other. This indicates that the label estimation of a sample should not only be determined by its own feature-label annotation but also be influenced by its neighbors. The feature correlation between samples and its neighbors leads to sample correlation, which implicitly smooths their label annotation. In order to consider the sample correlation of label space and feature space, we build a new similarity matrix during the process of the LP in the

graph model. Our method learns the subspace of manifold learning to build the prediction target, i.e., label distribution. Therefore, we need to use low-dimensional feature representation to analyze the sample correlation.

(1.1) We obtain the low-dimensional feature representation by the LTSA algorithm [9], with the following two stages.

(1.1a) Extract local information: Build the local neighborhood $\boldsymbol{X}_i = \left[\boldsymbol{x}_{i_1} \cdots \boldsymbol{x}_{i_k}\right] \in \mathbb{R}^{m\times k}$ of each $\boldsymbol{x}_i$ with its k-nearest neighbors, according to the Euclidean distance metric. Then obtain the approximate representation of the local tangent space coordinates of each local neighborhood. Let $\boldsymbol{V}_i = \left[\boldsymbol{v}_1\ \boldsymbol{v}_2 \cdots \boldsymbol{v}_d\right] \in \mathbb{R}^{m\times d}$ be the matrix composed of the singular vectors corresponding to the first d maximum singular values of the covariance matrix of $\boldsymbol{X}_i\left(\boldsymbol{I}_k - \frac{1}{k}\mathbf{1}_k\mathbf{1}_k^{\mathrm{T}}\right)$. Then $\boldsymbol{\Theta}_i = \boldsymbol{V}_i^{\mathrm{T}}\boldsymbol{X}_i\left(\boldsymbol{I}_k - \frac{1}{k}\mathbf{1}_k\mathbf{1}_k^{\mathrm{T}}\right) \in \mathbb{R}^{d\times k}$ is the local tangent space matrix of the neighborhood.

(1.1b) Local coordinate alignment to global low-dimensional coordinates: Let $\boldsymbol{T}_i = \left[\boldsymbol{t}_{i_1}\ \boldsymbol{t}_{i_2} \cdots \boldsymbol{t}_{i_k}\right] \in \mathbb{R}^{d\times k}$ be the d-dimensional global embedding coordinates of $\boldsymbol{X}_i$. Align the local tangent space coordinates and the global low-dimensional coordinates by minimizing the alignment error:

$$\min\left\|\boldsymbol{T}_i\left(\boldsymbol{I}_k - \frac{1}{k}\mathbf{1}_k\mathbf{1}_k^{\mathrm{T}}\right) - \boldsymbol{L}_i\boldsymbol{\Theta}_i\right\| = \mathrm{tr}\left(\boldsymbol{T}\boldsymbol{Q}\boldsymbol{T}^{\mathrm{T}}\right), \tag{11}$$

where $\boldsymbol{L}_i = \boldsymbol{T}_i\left(\boldsymbol{I}_k - \frac{1}{k}\mathbf{1}_k\mathbf{1}_k^{\mathrm{T}}\right)\boldsymbol{\Theta}_i^{\dagger}$, $\boldsymbol{Q} = \sum_{i=1}^{n}\boldsymbol{S}_i\boldsymbol{B}_i\boldsymbol{B}_i^{\mathrm{T}}\boldsymbol{S}_i^{\mathrm{T}}$ 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 $\boldsymbol{B}_i$ is given by

$$\boldsymbol{B}_i = \left(\boldsymbol{I}_k - \frac{1}{k}\mathbf{1}_k\mathbf{1}_k^{\mathrm{T}}\right)\left(\boldsymbol{I}_k - \boldsymbol{\Theta}_i^{\dagger}\boldsymbol{\Theta}_i\right), \tag{12}$$

while the global low-dimensional coordinates $\boldsymbol{T} = [\boldsymbol{t}_1\ \boldsymbol{t}_2 \cdots \boldsymbol{t}_n]$ are composed of the d-dimensional eigenvectors of $\boldsymbol{Q}$. $\boldsymbol{T}$ reflects the information of the feature structure.

(1.2) We construct a new sample similarity matrix $\boldsymbol{A}$. In order to consider the sample correlation of label space through the topological information of feature space, $\boldsymbol{A}$ is constructed by using the low-dimensional coordinates $\boldsymbol{t}_i$.

Specifically, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represent the graph of the training set $\mathcal{S}_{\mathrm{ll}}$. The sample similarity of neighboring samples in the low-dimensional feature space can be represented as:

$$a_{i,j} = \begin{cases} \exp\left(-\frac{\|\boldsymbol{t}_i - \boldsymbol{t}_j\|^2}{2}\right), & i \neq j, \\ 0, & i = j, \end{cases} \quad 1 \le i, j \le n. \tag{13}$$

The sample similarity matrix $\boldsymbol{A} = \left[a_{i,j}\right]_{n\times n}$ is constructed via the topological information of feature space, i.e., the low-dimensional coordinates $\boldsymbol{t}_i$. According to the smooth property, the manifold structure in the feature space is retained in the label space. Thus we can use the similarity between the samples in the feature space to guide the label prediction in the label space.

(1.3) After obtaining $\boldsymbol{A}$, the LP matrix is constructed as $\boldsymbol{P} = \bar{\boldsymbol{A}}^{-\frac{1}{2}}\boldsymbol{A}\bar{\boldsymbol{A}}^{-\frac{1}{2}}$, where the diagonal matrix $\bar{\boldsymbol{A}} = \mathrm{diag}\left\{\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_n\right\}$ with $\bar{a}_i = \sum_{j=1}^{n} a_{i,j}$. Let $\boldsymbol{F} = \left[f_{i,j}\right]_{n\times c}$ denote the label importance matrix. Based on the training set, the initial label importance matrix is set to $\boldsymbol{F}^{(0)} = \boldsymbol{Y}^{\mathrm{T}}$. At the tth iteration, $\boldsymbol{F}$ is updated according to the iterative formula similar to (2), and $\boldsymbol{F}^{(t)}$ converges to $\boldsymbol{F}^{\star}$, which takes the form similar to (4). The solution $\boldsymbol{F}^{\star}$ is normalized to obtain the label distributions $\widehat{d}_{\boldsymbol{x}_i}^{y_{i,j}}$ according to the formula of (5).

It is worth emphasizing that although the LE based on LP [1] looks similar to the LEFR in converting the logical labels of the training set into label distributions, the two algorithms are fundamentally different. Firstly, the LP matrix $\boldsymbol{P}$ of our LEFR is very different from that of the LE based on LP. Secondly and more importantly, our LEFR builds a label distribution prediction model during the LE process, and it is capable of predicting the label distribution of a new sample.

(2) Construct label distribution prediction model via manifold structure: The label distribution estimate $\widehat{d}_{\boldsymbol{x}_i}^{y_{i,j}}$ for the training data together with the manifold structure are used to construct the label distribution prediction model. Define a linear regression function $\boldsymbol{f}(\boldsymbol{x}_i) = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i + \boldsymbol{b}$, where $\boldsymbol{W} \in \mathbb{R}^{m\times c}$ is the projection matrix and $\boldsymbol{b} \in \mathbb{R}^{c\times 1}$ is the bias. Our goal is to determine the optimal $(\boldsymbol{W}, \boldsymbol{b})$, which can generate an accurate label distribution according to the instance $\boldsymbol{x}_i$. For notational convenience, express $\boldsymbol{f}(\boldsymbol{x}_i) = \bar{\boldsymbol{W}}\bar{\boldsymbol{x}}_i$, where $\bar{\boldsymbol{W}} = \left[\boldsymbol{W}^{\mathrm{T}}\ \boldsymbol{b}\right]$ and $\bar{\boldsymbol{x}}_i^{\mathrm{T}} = \left[\boldsymbol{x}_i^{\mathrm{T}}\ 1\right]$. This leads to the optimization problem:

$$\min_{\bar{\boldsymbol{W}}} \Omega(\bar{\boldsymbol{W}}) = \sum_{i=1}^{n}\left\|\bar{\boldsymbol{W}}\bar{\boldsymbol{x}}_i - \boldsymbol{d}_i\right\|^2 + \lambda_1\left\|\bar{\boldsymbol{W}}\right\|^2 + \lambda_2\Psi(\bar{\boldsymbol{W}}), \tag{14}$$

where the regularization parameters λ_1 and λ_2 balance the regression error, the norm of $\bar{\boldsymbol{W}}$ and the manifold smoothness function $\Psi(\bar{\boldsymbol{W}})$. Since the ground-truth label distribution is unknown, we substitute $\boldsymbol{d}_i$ in (14) by the label distribution estimate $\widehat{\boldsymbol{d}}_i = \left[\widehat{d}_{\boldsymbol{x}_i}^{y_{i,1}}\ \widehat{d}_{\boldsymbol{x}_i}^{y_{i,2}} \cdots \widehat{d}_{\boldsymbol{x}_i}^{y_{i,c}}\right]^{\mathrm{T}}$ obtained in *(1.3)*. Then we use the topology information of feature space to mine the sample label distribution hidden in the manifold space to define the regularization term, $\Psi(\bar{\boldsymbol{W}})$, so as to enhance the logical label into the label distribution. More specifically, since the feature space and label space should have the similar local topological structure, by defining the feature matrix $\bar{\boldsymbol{X}} = \left[\bar{\boldsymbol{x}}_1\ \bar{\boldsymbol{x}}_2 \cdots \bar{\boldsymbol{x}}_n\right] \in \mathbb{R}^{(m+1)\times n}$ and noting the alignment matrix $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ given in (11), the manifold smoothness regularization function can be chosen as

$$\Psi(\bar{\boldsymbol{W}}) = \mathrm{tr}\left(\bar{\boldsymbol{W}}\bar{\boldsymbol{X}}\boldsymbol{Q}\bar{\boldsymbol{X}}^{\mathrm{T}}\bar{\boldsymbol{W}}^{\mathrm{T}}\right). \tag{15}$$

Therefore, our method constructs the label distribution prediction model by minimizing the ridge regression errors while simultaneously preserving the manifold smoothness, namely, by minimizing the composite loss function:

$$\Omega(\bar{\boldsymbol{W}}) = \sum_{i=1}^{n}\left\|\bar{\boldsymbol{W}}\bar{\boldsymbol{x}}_i - \widehat{\boldsymbol{d}}_i\right\|^2 + \lambda_1\left\|\bar{\boldsymbol{W}}\right\|^2 + \lambda_2\mathrm{tr}\left(\bar{\boldsymbol{W}}\bar{\boldsymbol{X}}\boldsymbol{Q}\bar{\boldsymbol{X}}^{\mathrm{T}}\bar{\boldsymbol{W}}^{\mathrm{T}}\right). \tag{16}$$

Thus our method is a semi-supervised learning based on manifold regularization. The unlabeled instances help to identify the low-dimensional manifold structure, along which the labels can be assumed to change smoothly. In other words, we use the manifold regularization to exploit the geometry of the marginal distribution, as estimated by the unlabeled data.

(3) Prediction model optimization: We utilize the quasi-Newton method, BFGS [24], to minimize the objective function $\Omega(\bar{\boldsymbol{W}})$. Express $\Omega(\bar{\boldsymbol{W}})$ equivalently as

$$\begin{aligned}\Omega(\bar{\boldsymbol{W}}) =& \mathrm{tr}\left(\left(\bar{\boldsymbol{W}}\bar{\boldsymbol{X}} - \widehat{\boldsymbol{D}}\right)\left(\bar{\boldsymbol{W}}\bar{\boldsymbol{X}} - \widehat{\boldsymbol{D}}\right)^{\mathrm{T}}\right) \\ &+ \lambda_1\mathrm{tr}\left(\bar{\boldsymbol{W}}\bar{\boldsymbol{W}}^{\mathrm{T}}\right) + \lambda_2\mathrm{tr}\left(\bar{\boldsymbol{W}}\bar{\boldsymbol{X}}\boldsymbol{Q}\bar{\boldsymbol{X}}^{\mathrm{T}}\bar{\boldsymbol{W}}^{\mathrm{T}}\right),\end{aligned} \tag{17}$$

where $\widehat{\boldsymbol{D}} = \left[\widehat{\boldsymbol{d}}_1\ \widehat{\boldsymbol{d}}_2 \cdots \widehat{\boldsymbol{d}}_n\right]$. The gradient of $\Omega(\bar{\boldsymbol{W}})$ is

$$\nabla\Omega(\bar{\boldsymbol{W}}) = 2\bar{\boldsymbol{W}}\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}^{\mathrm{T}} - 2\widehat{\boldsymbol{D}}\bar{\boldsymbol{X}}^{\mathrm{T}} + 2\lambda_1\bar{\boldsymbol{W}} + 2\lambda_2\bar{\boldsymbol{W}}\bar{\boldsymbol{X}}\boldsymbol{Q}^{\mathrm{T}}\bar{\boldsymbol{X}}^{\mathrm{T}}. \tag{18}$$

BFGS iteratively optimizes $\Omega(\bar{\boldsymbol{W}})$ based on gradient descent.

When the optimal prediction model parameter $\bar{\boldsymbol{W}}^{*}$ is determined, the label distribution estimate for the new test instance $\boldsymbol{x}$ can be generated through the linear regression function $\boldsymbol{f}(\boldsymbol{x}) = \bar{\boldsymbol{W}}^{*}\bar{\boldsymbol{x}}$ with $\bar{\boldsymbol{x}}^{\mathrm{T}} = \left[\boldsymbol{x}^{\mathrm{T}}\ 1\right]$.

3.2. LE-GCN approach

Unlike the traditional graph model, where nodes and labels are not combined to construct the graph [10,11], we fully explore the correspondence (matching) between each instance and label by reformulating the LE task as an instance-label matching selection problem, and propose a novel LE deep learning model based on GCN, as illustrated in Fig. 3.

(1) We first represent each example feature as a node, and the feature nodes of the original samples are connected through a distance threshold $\varepsilon_{i,j} > \varepsilon$, calculated according to

$$\varepsilon_{i,j} = \exp\left(-\frac{\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2}{2}\right), \ i \neq j, \tag{19}$$

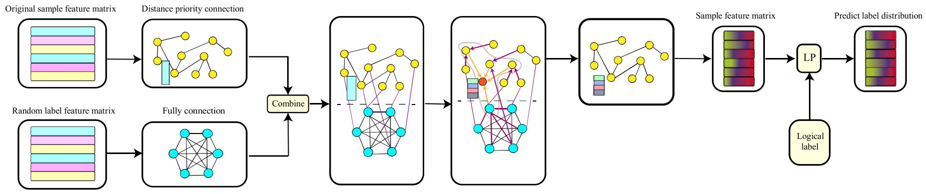
Original sample feature matrix
Distance priority connection
Random label feature matrix
Fully connection
Combine
Sample feature matrix
Predict label distribution
LP
Logical label

Fig. 3. Architecture of the proposed LE-GCN model.

to extract the hidden relationship between the feature nodes. Also the logical label information of each sample is used as a node. The label nodes are initialized with the random label feature matrix and they are fully connected. Then the sample feature nodes and their corresponding label nodes are connected, to fully mine the hidden relationship between the feature nodes and label nodes. Specifically, if the sample belongs to a certain class, we connect its feature node to the corresponding label node; otherwise the two nodes are not connected. Thus, the information between the feature space and label space is reconstructed via the node-label matrix, i.e., the connection matrix of nodes and labels.

(2) Next, the node-label matrix is injected into the GCN model and is processed as follows.

(2.1) The GCN preprocesses the node-label matrix, i.e., the node-label information, by adding self-connection to the node-label matrix, and hence the diagonal elements of the node-label matrix are all 1.

(2.2) A linear transformation transforms the input sample node features, which are subjected to structural dimension reduction. Specifically, the feature matrix of the nodes is linearly transformed, and the features are transformed to the output dimension. Then the node features are normalized.

(2.3) The GCN aggregates the neighbor node features according to

$$\boldsymbol{x}'_{\mathcal{N}(i)} = mean\left(\left\{\boldsymbol{x}_j, \forall j \in \mathcal{N}(i)\right\}\right), \tag{20}$$

where $\mathcal{N}(i)$ denotes the node i's neighbors, and hence $\boldsymbol{x}'_{\mathcal{N}(i)} \in \mathbb{R}^{m\times 1}$ is the average of the adjacent nodes' features of instance $\boldsymbol{x}_i \in \mathbb{R}^{m\times 1}$. A brand new feature matrix is obtained, which is the output of this layer of the GCN, according to

$$\boldsymbol{x}'_i = \boldsymbol{W}_1\boldsymbol{x}_i + \boldsymbol{W}_2\boldsymbol{x}'_{\mathcal{N}(i)}, \tag{21}$$

where $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^{c\times m}$ and both obey the Kaiming uniform [26], while $\boldsymbol{x}'_i \in \mathbb{R}^{c\times 1}$.

(2.4) Finally, a new feature matrix $\boldsymbol{L}' \in \mathbb{R}^{n\times c}$ is obtained, and each row of $\boldsymbol{L}'$ is constituted by $\boldsymbol{l}'^{\mathrm{T}}_i$ that is the output of this layer of the GCN given by

$$\boldsymbol{l}'_i = sigmoid\left(\frac{\boldsymbol{x}'_i}{\|\boldsymbol{x}'_i\|^2}\right). \tag{22}$$

The new feature matrix $\boldsymbol{L}'$ is then used in the following updating

$$\boldsymbol{L}' = \alpha\boldsymbol{P}\boldsymbol{L}' + (1-\alpha)\boldsymbol{Y}^{\mathrm{T}}, \tag{23}$$

where $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ is the probability transition matrix defined in Section 2.1. Inspired by the LP method of Section 2.1, the features are normalized to become the estimated label distributions $\widehat{\boldsymbol{d}}_i$, $1 \le i \le n$. The essence of this step is to map the features onto labels, and the theoretical basis of this is the smoothness property, i.e., the labels with similar features are also similar.

The GCN uses this reconstruction information to achieve the aggregation and updating of the training sample node information, to obtain new feature nodes and realize the LP, corresponding to the predicted label distribution. The distance between the predicted label distributions obtained by the GCN using the reshaping information and the LP constitutes the objective function, thereby obtaining the prediction score corresponding to each instance label. Steps *(1)* and *(2)* constitute the process of enhancing the logical labels into label distributions. Fig. 4 depicts the flow chart of this LE process.

(3) To predict the label distribution of new sample unseen in training, a label predictor model is constructed based on the enhanced training dataset $\{\boldsymbol{X}, \widehat{\boldsymbol{D}}\}$. We employ a logistic regression classifier, similar to $\boldsymbol{f}(\boldsymbol{x}) = \bar{\boldsymbol{W}}\bar{\boldsymbol{x}}$ used in LEFR, as the label predictor, and optimize the predictor's parameters [7]. However, we note that the gradient of the objective function is impacted by the LP process in LE-GCN, and this affects the performance of optimizing the label predictor. It is worth noting that the label predictor of LEFR is integrated/constructed/optimized within the LE process, while the label predictor of LE-GCN has to be constructed and optimized separately after the LE process.

Table 1
Complexity comparison of various LE algorithms for LE, where I_{irwls} is the number of iterations for the IRWLS algorithm [27], $I_{\mathrm{iteration}}$ is the number of iterations of LP [1], I_{SVM} is the number of iterations for the SVM algorithm and I_{bfgs} is the number of iterations for the BFGS algorithm [24].

LP [1]	$\mathrm{O}\left(n^2 \times m \times (1 + I_{\mathrm{iteration}}) + n^3\right)$
ML [16]	$\mathrm{O}\left(n + I_{\mathrm{irwls}} \times n^3\right)$
GLLE [7]	$\mathrm{O}\left(n^2 \times (m + I_{\mathrm{bfgs}})\right)$
LESC [17]	$\mathrm{O}\left(n^2 \times (m + I_{\mathrm{bfgs}})\right)$
PLEML [18]	$\mathrm{O}\left(n^2 \times (m + I_{\mathrm{bfgs}} + I_{\mathrm{SVM}})\right)$
LEFR	$\mathrm{O}\left(n^2 \times (m + I_{\mathrm{bfgs}}) + n^3\right)$
LE-GCN	$\mathrm{O}\left(n^3 + I_{\mathrm{iteration}} \times n^2 \times c\right)$

3.3. Complexity analysis

The computational complexity of the LEFR for LE consists of three parts as summarized below.

Step 1. The procedure of the incremental feature extraction has the complexity on the order of $\mathrm{O}\left(n^2 \times m\right)$.

Step 2. Construct the LP matrix has the complexity on the order of $\mathrm{O}\left(n^3\right)$.

Step 3. Let the number of iterations for the BFGS algorithm [24] be upper bounded by I_{bfgs}. Since the complexity per iteration of the BFGS optimization is $\mathrm{O}\left(n^2\right)$, the complexity of **Step 3** is $\mathrm{O}\left(I_{\mathrm{bfgs}} \times n^2\right)$.

The computational complexity of the LE-GCN for LE consists of two parts as summarized below.

Step 1. Construct the LP matrix has the complexity on the order of $\mathrm{O}\left(n^3\right)$.

Step 2. Let the number of iterations for (23) be upper bounded by $I_{\mathrm{iteration}}$. Since the complexity per iteration of the (23) is $\mathrm{O}\left(n^2 \times c\right)$, the complexity of **Step 2** is $\mathrm{O}\left(I_{\mathrm{iteration}} \times n^2 \times c\right)$.

Table 1 compares the LE complexity of our LEFR and LE-GCN with those of the five benchmarks.

4. Experimental evaluation

4.1. Description of experimental system

(1) Datasets: 13 real-world datasets from [2] are used in our experiments, including two facial expression datasets, one natural scene

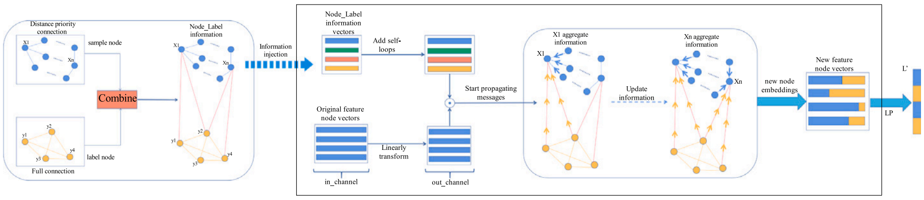

Fig. 4. Flow chart of the label enhancement process by the proposed LE-GCN algorithm.

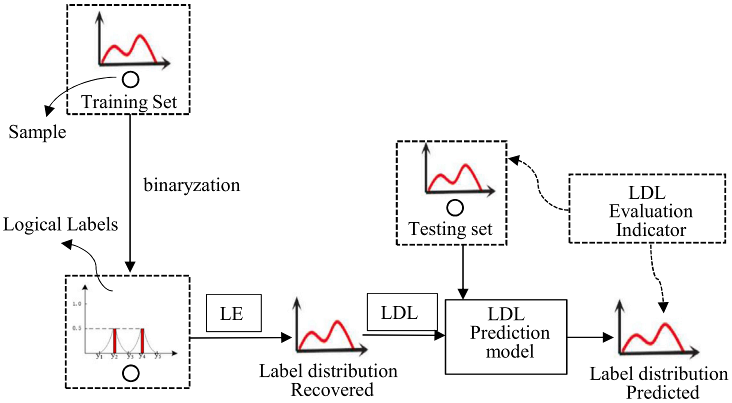

Fig. 5. Flow chart of the label distribution prediction process.

dataset and ten biological yeast experiment datasets. The two facial expression datasets are collected from facial expression images, and the natural scene dataset is collected from natural scene images and movies, while the ten datasets of Yeast-alpha to Yeast-spoem are collected from the records of 10 biological experiments on the budding yeast genes. These datasets are chosen because they all provide the ground-truth label distributions. As a basic task of LE learning is to estimate the label distributions from the logical labels, we need the true logical labels, which are obtained from the ground-truth label distributions via binarization.

(2) Evaluation measures: To evaluate the accuracy of estimated/predicted label distribution, a natural choice of measure is the average distance or similarity between the estimated label distribution and the ground-truth label distribution. We select six metrics to reflect LE and label prediction performance from various aspects in semantics [28], namely, Chebyshev distance (Cheb), Clark distance (Clark), Kullback–Leibler divergence (KL), Canberra distance (Canber), Cosine correlation coefficient (Cosine) and intersection similarity (Inter). The first four metrics are distance metrics and the last two are similarity metrics.

(3) Benchmarks: We compare our LEFR and LE-GCN with the five state-of-the-art LE algorithms, and they are: the LE algorithm based on label propagation (LP) [1], the LE algorithm based on manifold learning (ML) [16], the graph Laplacian-based LE algorithm (GLLE) [7], the LE algorithm using privileged information (PLEML) [18], and the LE algorithm with sample correlations via low-rank representation (LESC) [17].

(4) Experiment procedures: There are two sets of experiments to evaluate an LE method's label distribution recovery performance and label predictive performance, respectively.

(4.1) In the first set of experiments for recovery performance, there is no need to divide the dataset into training set and test set. We apply LE algorithms to the whole dataset to recover the label distributions from the logical labels. Then the recovered or estimated label distributions are compared with the ground-truth label distributions using the six evaluation metrics.

(4.2) The second set of experiments is for label predictive performance to further test the effectiveness of LDL after the LE preprocessing on the logical-labeled datasets. To predict the label distribution of new sample unseen in training, a label predictor is required. Most LE algorithms considered already train such parametric label predictor models during or after the LE process. The LP [1] and ML [16] only recover the label distributions from the logical labels for the training set during the LE process, and therefore label predictors are trained for them using the enhanced training dataset based on the maximum entropy model optimization [2]. The trained label predictor models are tested on the new test dataset, and the label distribution predictions are compared with the ground-truth label distributions. Ten-fold cross validation is conducted for each algorithm and the average results are recorded. Fig. 5 depicts the label distribution prediction or LDL flow chart, where the connection with the label distribution recovery or LE can be clearly seen.

(4.3) We now list the algorithmic parameters. For our LEFR, $\lambda_1 = 0.5$ and λ_2 is selected from $\{0.0001, 0.001, \dots, 10\}$. For our LE-GCN, the distance threshold $\varepsilon = 0.8226$, and $\alpha = 0.7560$. For LP and ML, $\alpha = 0.5$, and for ML the number of nearest neighbors is set to $k = c + 1$. For GLLE, λ is selected from $\{0.01, 0.1, \dots, 100\}$, and the number of nearest neighbors is set to $k = c + 1$. For PLEML, λ_1 and λ_2 are selected from $\{2^{-4}, 2^{-3}, \dots, 2^{8}\}$, $\gamma = 0.1$ and $C = 0.1$. For LESC, λ_1 and λ_2 are selected from $\{0.0001, 0.001, \dots, 10\}$. The best algorithmic parameter combination is used. Source codes with data are available.[1]

4.2. Label recovery experimental results

For the label distribution recovery experiments, we present the quantitative results of the 7 LE algorithms using the 6 evaluation metrics in Tables 2 to 7, respectively, where the notation '↓' after a metric indicates 'the smaller the better', while '↑' after a metric means 'the larger the better'. The rank of every algorithm for each dataset is also listed in the bracket, and the last row of each table presents the average ranking performance over the 13 datasets, where the numerical value before the bracket is the average ranking value and the number in the bracket is again the rank. The experimental results of Tables 2 to 7 show that our LE-GCN attains the best recovery performance on average, and our LEFR achieves the second best performance, while the existing

[1] https://github.com/code-opensource/LEFR-and-LE-GCN

Table 2
Label distribution recovery performance measured by Chebyshev distance ↓.

dataset	LP	ML	GLLE	PLEML	LESC	LEFR	LE-GCN
Yeast-alpha	0.0400(7)	0.0387(6)	0.0192(5)	0.0137(1.5)	0.0169(4)	0.0137(1.5)	0.0146(3)
Yeast-cdc	0.0420(6)	0.0475(7)	0.0217(5)	0.0167(3)	0.0198(4)	0.0166(2)	0.0149(1)
Yeast-cold	0.1370(7)	0.1207(6)	0.0650(5)	0.0540(3)	0.0572(4)	0.0518(2)	0.0472(1)
Yeast-diau	0.0990(6)	0.2011(7)	0.0530(5)	0.0415(2)	0.0419(3)	0.0404(1)	0.0451(4)
Yeast-dtt	0.1280(7)	0.1073(6)	0.0518(5)	0.0372(2)	0.0466(4)	0.0381(3)	0.0308(1)
Yeast-elu	0.0440(6)	0.0499(7)	0.0221(5)	0.0165(2.5)	0.0208(4)	0.0165(2.5)	0.0154(1)
Yeast-heat	0.0860(6)	0.0915(7)	0.0478(5)	0.0433(2)	0.0466(4)	0.0440(3)	0.0314(1)
Yeast-spo	0.0900(6)	0.0953(7)	0.0608(4)	0.0603(3)	0.0609(5)	0.0596(2)	0.0429(1)
Yeast-spo5	0.1140(6)	0.1514(7)	0.0980(5)	0.0921(2)	0.0933(4)	0.0924(3)	0.0718(1)
Yeast-spoem	0.1630(7)	0.1319(6)	0.0870(2.5)	0.1170(5)	0.0870(2.5)	0.0885(4)	0.0527(1)
SBU_3DFE	0.1230(4.5)	0.1868(7)	0.1230(4.5)	0.1228(2.5)	0.1231(6)	0.1228(2.5)	0.1142(1)
SJAFFE	0.1070(6)	0.2188(7)	0.0845(3)	0.0885(4)	0.0692(1)	0.0824(2)	0.0940(5)
Natural_Scene	0.2750(2)	0.2990(3)	0.3353(4.5)	0.3384(6)	0.3417(7)	0.2679(1)	0.3353(4.5)
Average Rank	5.8846(6)	6.3846(7)	4.5000(5)	2.9615(3)	4.0385(4)	2.2692(2)	1.9615(1)

Table 3
Label distribution recovery performance measured by Clark distance ↓.

dataset	LP	ML	GLLE	PLEML	LESC	LEFR	LE-GCN
Yeast-alpha	0.4322(6)	0.6025(7)	0.3304(5)	0.2147(1)	0.2823(4)	0.2158(2)	0.2372(3)
Yeast-cdc	0.3803(6)	0.5593(7)	0.3018(5)	0.2191(3)	0.2727(4)	0.2189(2)	0.2072(1)
Yeast-cold	0.1805(6)	0.3224(7)	0.1738(5)	0.1465(3)	0.1552(4)	0.1413(2)	0.1236(1)
Yeast-diau	0.2841(5)	0.7276(7)	0.2964(6)	0.2222(2)	0.2302(3)	0.2139(1)	0.2482(4)
Yeast-dtt	0.1902(6)	0.2953(7)	0.1413(5)	0.1012(2)	0.1278(4)	0.1036(3)	0.0839(1)
Yeast-elu	0.3642(6)	0.5340(7)	0.2845(5)	0.2042(2)	0.2617(4)	0.2044(3)	0.1872(1)
Yeast-heat	0.2144(6)	0.3823(7)	0.2082(5)	0.1871(2)	0.2037(4)	0.1893(3)	0.1362(1)
Yeast-spo	0.5585(7)	0.4030(6)	0.2618(5)	0.2558(3)	0.2596(4)	0.2536(2)	0.1841(1)
Yeast-spo5	0.2741(6)	0.3015(7)	0.1943(5)	0.1855(2.5)	0.1871(4)	0.1855(2.5)	0.1435(1)
Yeast-spoem	0.2718(7)	0.2036(6)	0.1321(4)	0.1757(5)	0.1295(2)	0.1311(3)	0.0818(1)
SBU_3DFE	0.5810(6)	0.7861(7)	0.3818(5)	0.3689(3)	0.3785(4)	0.3628(2)	0.3194(1)
SJAFFE	0.3140(3)	0.8055(7)	0.3633(5)	0.3775(6)	0.2763(1)	0.3091(2)	0.3171(4)
Natural_Scene	2.4828(7)	2.4520(2)	2.4609(4)	2.4659(6)	2.4649(5)	2.3839(1)	2.4603(3)
Average Rank	5.9231(6)	6.4615(7)	4.9231(5)	3.1154(3)	3.6154(4)	2.1923(2)	1.7692(1)

Table 4
Label distribution recovery performance measured by Canberra metric ↓.

dataset	LP	ML	GLLE	PLEML	LESC	LEFR	LE-GCN
Yeast-alpha	1.7068(6)	2.0181(7)	1.1135(5)	0.6981(2)	0.9514(4)	0.6812(1)	0.7500(3)
Yeast-cdc	1.3532(6)	1.7591(7)	0.9442(5)	0.6545(3)	0.8405(4)	0.6544(2)	0.6367(1)
Yeast-cold	0.3241(6)	0.5598(7)	0.3016(5)	0.2527(3)	0.2680(4)	0.2434(2)	0.2138(1)
Yeast-diau	0.6425(5)	1.6538(7)	0.6734(6)	0.4772(2)	0.5021(3)	0.4556(1)	0.5523(4)
Yeast-dtt	0.3560(6)	0.5070(7)	0.2458(5)	0.1747(3)	0.2229(4)	0.1742(2)	0.1430(1)
Yeast-elu	1.2612(6)	1.6263(7)	0.8692(5)	0.6014(2)	0.7906(4)	0.6026(3)	0.5563(1)
Yeast-heat	0.4706(6)	0.7826(7)	0.4203(5)	0.3741(3)	0.4110(4)	0.3734(2)	0.2734(1)
Yeast-spo	1.2341(7)	0.8440(6)	0.5422(5)	0.5281(3)	0.5329(4)	0.5236(2)	0.3696(1)
Yeast-spo5	0.4013(6)	0.4664(7)	0.3018(5)	0.2849(3)	0.2884(4)	0.2848(2)	0.2214(1)
Yeast-spoem	0.3655(7)	0.2800(6)	0.1840(5)	0.1837(4)	0.1801(2)	0.1827(3)	0.1123(1)
SBU_3DFE	1.2463(6)	1.6593(7)	0.8409(5)	0.7866(3)	0.8039(4)	0.7817(2)	0.6813(1)
SJAFFE	1.0708(6)	1.6894(7)	0.7518(4)	0.7876(5)	0.5606(1)	0.6279(2)	0.6430(3)
Natural_Scene	6.7810(3)	6.7217(2)	6.8511(4)	6.8717(6)	6.8780(7)	6.6936(1)	6.8566(5)
Average Rank	5.8462(6)	6.4615(7)	4.9231(5)	3.2308(3)	3.7692(4)	1.9231(2)	1.8462(1)

Table 5
Label distribution recovery performance measured by Kullback–Leibler divergence ↓.

dataset	LP	ML	GLLE	PLEML	LESC	LEFR	LE-GCN
Yeast-alpha	0.1210(7)	0.0550(6)	0.0130(5)	0.0057(2)	0.0080(4)	0.0053(1)	0.0063(3)
Yeast-cdc	0.1110(7)	0.0609(6)	0.0140(5)	0.0073(3)	0.0100(4)	0.0057(1)	0.0058(2)
Yeast-cold	0.1030(6)	0.5560(7)	0.0190(5)	0.0135(3)	0.0150(4)	0.0121(2)	0.0090(1)
Yeast-diau	0.1270(6)	0.1934(7)	0.0270(5)	0.0158(2)	0.0170(3)	0.0155(1)	0.0178(4)
Yeast-dtt	0.1030(7)	0.0648(6)	0.0130(5)	0.0066(2)	0.0100(4)	0.0073(3)	0.0044(1)
Yeast-elu	0.1090(7)	0.0567(6)	0.0130(5)	0.0064(2)	0.0090(4)	0.0071(3)	0.0053(1)
Yeast-heat	0.0890(7)	0.0656(6)	0.0170(5)	0.0134(2)	0.0155(4)	0.0150(3)	0.0070(1)
Yeast-spo	0.0840(6)	0.5320(7)	0.0290(5)	0.0271(3)	0.0280(4)	0.0154(2)	0.0128(1)
Yeast-spo5	0.0420(6)	0.0811(7)	0.0340(5)	0.0299(3)	0.0310(4)	0.0157(1)	0.0172(2)
Yeast-spoem	0.0670(6)	0.5030(7)	0.0270(3.5)	0.0459(5)	0.0270(3.5)	0.0206(2)	0.0110(1)
SBU_3DFE	0.1050(6)	0.2489(7)	0.0690(4)	0.0659(3)	0.0692(5)	0.0610(2)	0.0572(1)
SJAFFE	0.0770(6)	0.2513(7)	0.0500(5)	0.0494(4)	0.0290(1)	0.0400(2)	0.0424(3)
Natural_Scene	1.5950(5)	2.2757(6)	2.6630(7)	0.9787(1)	1.1663(4)	1.1032(3)	0.9941(2)
Average Rank	6.3077(6)	6.5385(7)	4.9615(5)	2.6923(3)	3.7308(4)	2.0000(2)	1.7692(1)

Table 6
Label distribution recovery performance measured by Cosine coefficient ↑.

dataset	LP	ML	GLLE	PLEML	LESC	LEFR	LE-GCN
Yeast-alpha	0.9814(6)	0.9530(7)	0.9876(5)	0.9944(2)	0.9905(4)	0.9945(1)	0.9940(3)
Yeast-cdc	0.9828(6)	0.9468(7)	0.9875(5)	0.9930(3)	0.9896(4)	0.9932(2)	0.9945(1)
Yeast-cold	0.9847(5)	0.9429(7)	0.9827(6)	0.9873(3)	0.9859(4)	0.9882(2)	0.9913(1)
Yeast-diau	0.9805(5)	0.8427(7)	0.9750(6)	0.9854(2)	0.9844(3)	0.9864(1)	0.9836(4)
Yeast-dtt	0.9835(6)	0.9515(7)	0.9884(5)	0.9937(3)	0.9901(4)	0.9938(2)	0.9960(1)
Yeast-elu	0.9829(6)	0.9489(7)	0.9879(5)	0.9906(3)	0.9896(4)	0.9980(1)	0.9948(2)
Yeast-heat	0.9861(4)	0.9454(7)	0.9845(6)	0.9872(3)	0.9851(5)	0.9878(2)	0.9934(1)
Yeast-spo	0.9386(6)	0.8397(7)	0.9747(4)	0.9747(4)	0.9747(4)	0.9754(2)	0.9883(1)
Yeast-spo5	0.9686(6)	0.9359(7)	0.9713(5)	0.9736(2)	0.9732(4)	0.9734(3)	0.9847(1)
Yeast-spoem	0.9503(6)	0.8530(7)	0.9782(3)	0.9620(5)	0.9780(4)	0.9785(2)	0.9916(1)
SBU_3DFE	0.9220(6)	0.8435(7)	0.9304(5)	0.9344(2)	0.9319(3)	0.9305(4)	0.9425(1)
SJAFFE	0.9410(6)	0.8231(7)	0.9594(3)	0.9576(5)	0.9731(1)	0.9614(2)	0.9588(4)
Natural_Scene	0.7264(5)	0.6610(7)	0.7789(2)	0.7738(3)	0.7602(4)	0.7792(1)	0.6705(6)
Average Rank	5.6154(6)	7.0000(7)	4.6154(5)	3.0769(3)	3.6923(4)	1.9231(1)	2.0769(2)

Table 7
Label distribution recovery performance measured by intersection similarity ↑.

dataset	LP	ML	GLLE	PLEML	LESC	LEFR	LE-GCN
Yeast-alpha	0.9074(6)	0.8898(7)	0.9386(5)	0.9615(2)	0.9473(4)	0.9616(1)	0.9587(3)
Yeast-cdc	0.9122(6)	0.8836(7)	0.9376(5)	0.9569(3)	0.9445(4)	0.9570(2)	0.9583(1)
Yeast-cold	0.9213(6)	0.8646(7)	0.9250(5)	0.9376(3)	0.9338(4)	0.9400(2)	0.9467(1)
Yeast-diau	0.9128(5)	0.7557(7)	0.9052(6)	0.9335(2)	0.9301(3)	0.9367(1)	0.9234(4)
Yeast-dtt	0.9134(6)	0.8779(7)	0.9393(5)	0.9570(2)	0.9448(4)	0.9560(3)	0.9649(1)
Yeast-elu	0.9120(6)	0.8839(7)	0.9383(5)	0.9575(3)	0.9439(4)	0.9576(2)	0.9605(1)
Yeast-heat	0.9237(6)	0.8718(7)	0.9310(5)	0.9385(2)	0.9324(4)	0.9380(3)	0.9553(1)
Yeast-spo	0.8184(6)	0.7614(7)	0.9105(5)	0.9130(3)	0.9121(4)	0.9137(2)	0.9402(1)
Yeast-spo5	0.8855(6)	0.7486(7)	0.9020(5)	0.9079(2)	0.9067(4)	0.9076(3)	0.9282(1)
Yeast-spoem	0.8367(6)	0.7681(7)	0.9130(2.5)	0.9109(5)	0.9130(2.5)	0.9115(4)	0.9473(1)
SBU_3DFE	0.8096(6)	0.7414(7)	0.8531(5)	0.8570(3)	0.8542(4)	0.8589(2)	0.8745(1)
SJAFFE	0.8361(6)	0.7251(7)	0.8757(4)	0.8718(5)	0.9050(1)	0.8922(2)	0.8901(3)
Natural_Scene	0.4512(5)	0.3307(7)	0.5226(2)	0.5214(3)	0.5107(4)	0.5656(1)	0.4151(6)
Average Rank	5.8462(6)	7.0000(7)	4.5769(5)	2.9231(3)	3.5769(4)	2.1538(2)	1.9231(1)

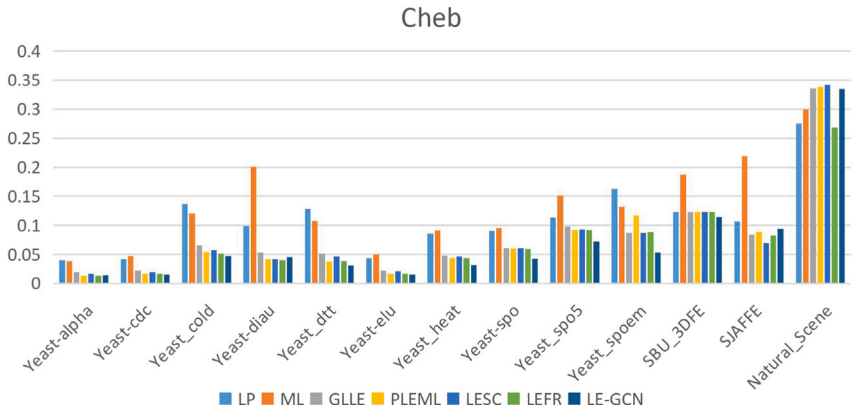

Fig. 6. Comparison of label distribution recovery results measured by Cheb↓.

state-of-the-art PLEML is only the third best. Specifically, of the total of 78 cases, our LE-GCN ranks the 1st in 65.4% and the 2nd in 5.1%, while our LEFR ranks the 1st in 24.4% cases and the 2nd in 50.0%.

Figs. 6 to 11 depict the histograms of the LE recovery performance measured by the 6 metrics, which again show that our LEFR and LE-GCN have the second highest average ranking and the highest average ranking, respectively.

4.3. Label predictive experimental results

For the LDL prediction experiments, we present the quantitative results of the 7 LE algorithms as measured by the 6 metrics in Tables 8 to 13, respectively. In terms of LDL predictive performance, our LEFR is the clear winner, ranking the best, and PLEML ranks the second best, while our LE-GCN ranks the third best. Specifically, of the total 78 cases, our LEFR ranks the 1st in 82.1% and the 2nd in 9.0%, and our LE-GCN ranks the 1st in 15.4% and the 2nd in 1.3%, while PLEML ranks the 1st in 3.8% and the 2nd in 87.2%. Later the reliable statistical test results will demonstrate that our LE-GC actually achieves better label distribution prediction performance than PLEML.

4.4. Statistical validation of experimental results

In the label distribution recovery experimental results of Tables 2 to 7, our LE-GCN exhibits the best performance over our LEFR and the five benchmarks, while in the label distribution prediction experimental results of Tables 8 to 13, our LEFR achieves the best performance, in comparison with our LE-GCN and the five benchmarks. We now apply statistical tests to validate the statistical significance of these results.

(1) Wilcoxon signed-rank test: To show the overall statistical relationships among the 7 LE algorithms on 13 datasets in the label distribution

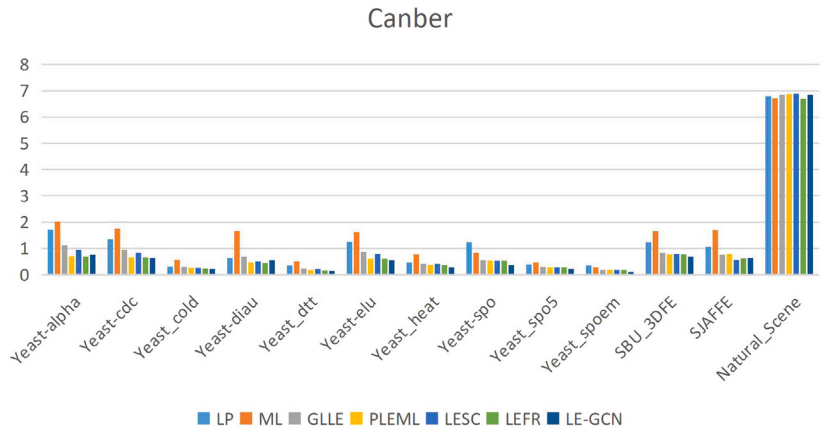

Fig. 7. Comparison of label distribution recovery results measured by Canber↓.

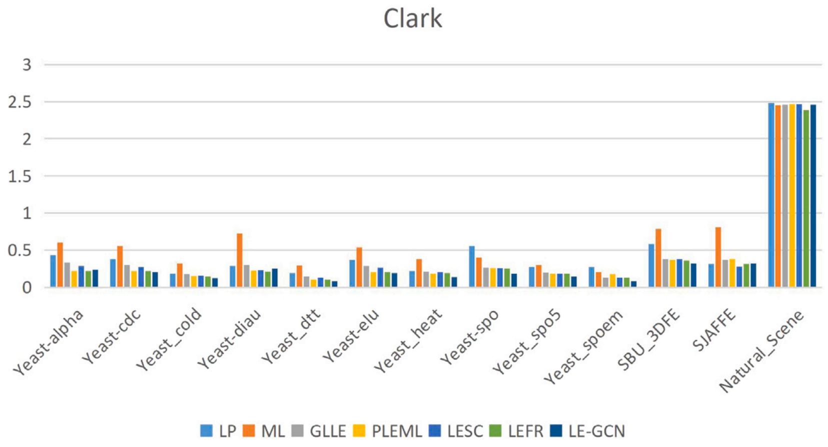

Fig. 8. Comparison of label distribution recovery results measured by Clark↓.

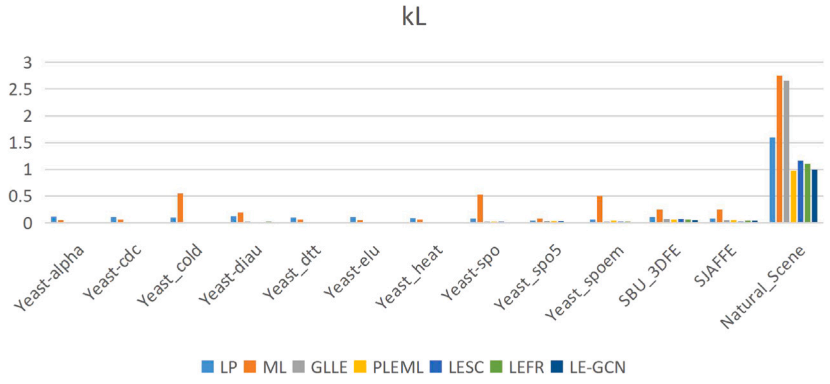

Fig. 9. Comparison of label distribution recovery results measured by KL↓.

recovery experiments, Wilcoxon signed-rank test [29] is employed as the statistical test to validate whether our LE-GCN algorithm performs significantly better than the other five existing LE algorithms and our LERF, in terms of each evaluation metric. Table 14 summarizes the statistical test results for the label distribution recovery experiments, where the *p*-values for the corresponding tests are shown in the brackets. As validated by Wilcoxon signed-rank test results of Tables 14, it is statistically significant that our LE-GCN outperforms the other six algorithms, in terms of all the six metrics, with one exception, namely, for the Cosine coefficient metric, the performance of our LE-GCN and our LEFR are statistically tied. The statistical test results hence clearly validate the effectiveness of our LE-GCN algorithm in enhancing logical labels into label distributions.

We also employ Wilcoxon signed-rank test for validating the statistical relationship between our LEFR and the other six algorithms in the label distribution prediction experiments, and the test results are summarized in Table 15. Our LEFR achieves statistically superior performance over all the other six algorithms in all the metrics, with one exception that for the KL divergence, our LEFR and PLEML are statistically tied. Therefore, the statistical test results convincingly confirm that our LEFR algorithm achieves the best label distribution prediction performance.

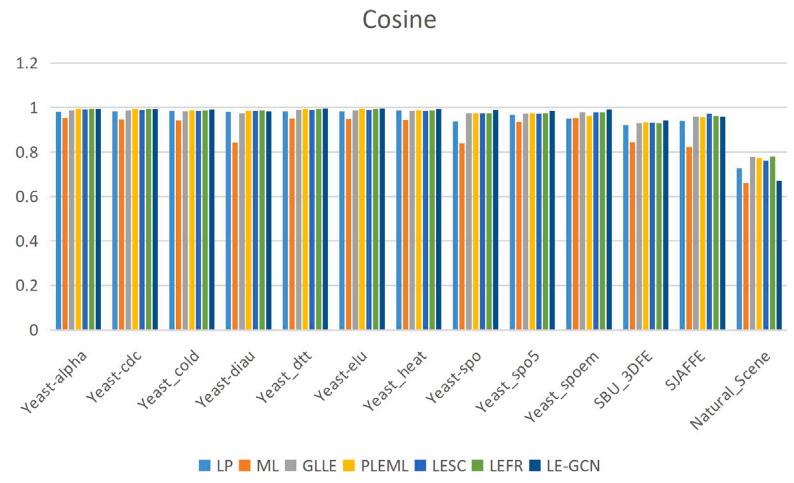

Fig. 10. Comparison of label distribution recovery results measured by Cosine↑.

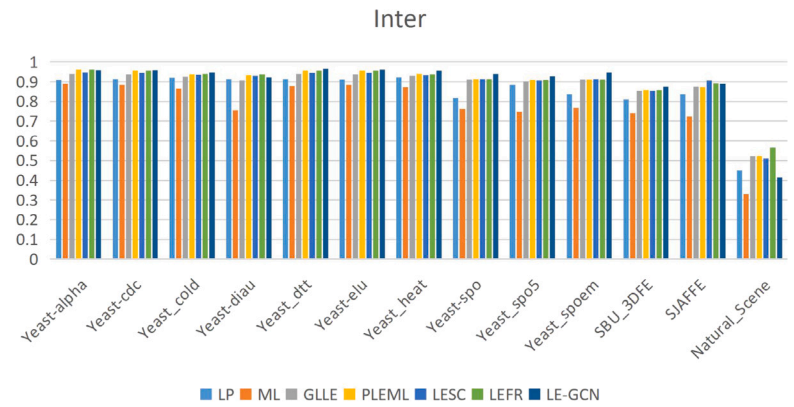

Fig. 11. Comparison of label distribution recovery results measured by Inter↑.

Table 8
Label distribution predictive performance measured by Chebyshev distance ↓.

dataset	LP	ML	GLLE	PLEML	LESC	LEFR	LE-GCN
Yeast_alpha	0.0871(4)	0.0885(7)	0.0874(5.5)	0.0202(2)	0.0874(5.5)	0.0133(1)	0.0333(3)
Yeast_cdc	0.0937(6)	0.0951(7)	0.0935(5)	0.0232(2)	0.0934(4)	0.0165(1)	0.0304(3)
Yeast_cold	0.1558(4)	0.1605(7)	0.1561(5)	0.0649(2)	0.1562(6)	0.0554(1)	0.0933(3)
Yeast_diau	0.1283(4)	0.1408(7)	0.1292(6)	0.0484(2)	0.1285(5)	0.0409(1)	0.0909(3)
Yeast_dtt	0.1478(4)	0.1537(7)	0.1494(5)	0.0500(2)	0.1498(6)	0.0372(1)	0.0621(3)
Yeast_elu	0.0944(4)	0.0972(7)	0.0949(6)	0.0237(2)	0.0948(5)	0.0164(1)	0.0319(3)
Yeast_heat	0.1351(5)	0.1391(7)	0.1350(4)	0.0532(2)	0.1353(6)	0.0431(1)	0.0584(3)
Yeast_spo	0.1445(7)	0.1435(4)	0.1442(5.5)	0.0683(2)	0.1442(5.5)	0.0613(1)	0.0710(3)
Yeast_spo5	0.1742(6)	0.1791(7)	0.1741(5)	0.1001(2)	0.1738(4)	0.0955(1)	0.1108(3)
Yeast_spoem	0.1492(6)	0.1521(7)	0.1488(5)	0.0936(2)	0.1486(4)	0.0923(1)	0.0955(3)
SBU_3DFE	0.1637(6)	0.1677(7)	0.1630(5)	0.1382(2)	0.1622(4)	0.1379(1)	0.1429(3)
SJAFFE	0.2393(5)	0.2425(7)	0.2394(6)	0.1168(2)	0.2383(4)	0.1169(3)	0.1146(1)
Natural_scene	0.3916(4)	0.4035(7)	0.3922(5)	0.3702(2)	0.3923(6)	0.3712(3)	0.3658(1)
Average Rank	5.0000(4.5)	6.7692(7)	5.2308(6)	2.0000(2)	5.0000(4.5)	1.3077(1)	2.6923(3)

(2) Bayesian signed-rank test: To further discover to what extent our LE-GCN performs better than our LEFR and the other five LE algorithms in the label distribution recovery experiments, Bayesian signed-rank test [30] is employed as the statistical test. Table 16 summarizes the statistical test results, where numerical values a, b, and c in the brackets [a, b, c] are the probabilities of the control algorithm being [WIN, TIE, LOSE] over the comparing algorithm. The prior default is that the performance of the two algorithms are the same. Prior strength is the strength of this null hypothesis, which means that this null hypothesis is established with a probability of 0.6. The performance of two algorithms are similar if the difference between two algorithms' results is less than rope = 0.0001. As validated by Bayesian signed-rank test results of Tables 16, it is statistically significant that our LE-GCN algorithm outperforms the other six algorithms in all the six metrics. Observe that the winning probabilities of LE-GCN over PLEML are bigger than those of LE-GCN over LEFR, with one exception in the KL divergence. This implicitly validates that statistically our LEFR outperforms PLEML. Compared with Wilcoxon signed-rank test, Bayesian signed-rank test provides more statistical details.

Bayesian signed-rank test is also employed to validate the statistical relationship between our LEFR and the other six algorithms in the label distribution prediction experiments, and Table 17 lists the test results.

Table 9
Label distribution predictive performance measured by Clark distance ↓.

Dataset	LP	ML	GLLE	PLEML	LESC	LEFR	LE-GCN
Yeast_alpha	0.8263(4)	0.8468(7)	0.8304(5)	0.3051(2)	0.8312(6)	0.2128(1)	0.5715(3)
Yeast_cdc	0.7659(5)	0.7863(7)	0.7667(6)	0.2914(2)	0.7654(4)	0.2160(1)	0.4074(3)
Yeast_cold	0.3867(6)	0.4049(7)	0.3866(5)	0.1734(2)	0.3863(4)	0.1504(1)	0.2298(3)
Yeast_diau	0.5429(5)	0.5896(7)	0.5444(6)	0.2536(2)	0.5424(4)	0.2185(1)	0.5364(3)
Yeast_dtt	0.3656(4)	0.3789(7)	0.3694(5)	0.1326(2)	0.3701(6)	0.1011(1)	0.1695(3)
Yeast_elu	0.7320(4)	0.7577(7)	0.7365(6)	0.2773(2)	0.7361(5)	0.2033(1)	0.3916(3)
Yeast_heat	0.4885(4)	0.5123(7)	0.4932(5)	0.2263(2)	0.4944(6)	0.1859(1)	0.2398(3)
Yeast_spo	0.5254(5)	0.5276(7)	0.5253(4)	0.2867(2)	0.5258(6)	0.2597(1)	0.3094(3)
Yeast_spo5	0.3444(6)	0.3559(7)	0.3441(5)	0.2021(2)	0.3433(4)	0.1936(1)	0.2171(3)
Yeast_spoem	0.2311(5.5)	0.2363(7)	0.2311(5.5)	0.1389(2)	0.2307(4)	0.1370(1)	0.1418(3)
SBU_3DFE	0.5457(5)	0.5822(7)	0.5464(6)	0.4102(2)	0.5445(4)	0.4047(1)	0.4340(3)
SJAFFE	0.9152(6)	0.9348(7)	0.9148(5)	0.4453(3)	0.9107(4)	0.4336(2)	0.4274(1)
Natural_scene	2.5017(4)	2.5184(7)	2.5025(6)	2.4822(1)	2.5023(5)	2.4832(2)	2.4845(3)
Average Rank	4.8846(5)	7.0000(7)	5.3462(6)	2.0000(2)	4.7692(4)	1.1538(1)	2.8462(3)

Table 10
Label distribution predictive performance measured by Canberra metric ↓.

Dataset	LP	ML	GLLE	PLEML	LESC	LEFR	LE-GCN
Yeast_alpha	2.8993(4)	2.9761(7)	2.9151(5)	1.0221(2)	2.9180(6)	0.6940(1)	1.6690(3)
Yeast_cdc	2.4757(5)	2.5471(7)	2.4791(6)	0.8936(2)	2.4751(4)	0.6422(1)	1.3134(3)
Yeast_cold	0.6893(4)	0.7184(7)	0.6901(6)	0.2994(2)	0.6899(5)	0.2594(1)	0.4032(3)
Yeast_diau	1.2345(4)	1.3368(7)	1.2391(6)	0.5482(2)	1.2349(5)	0.4670(1)	1.2020(3)
Yeast_dtt	0.6576(4)	0.6848(7)	0.6663(5)	0.2302(2)	0.6680(6)	0.1734(1)	0.2935(3)
Yeast_elu	2.2941(4)	2.3732(7)	2.3084(6)	0.8324(2)	2.3061(5)	0.6001(1)	1.1890(3)
Yeast_heat	1.0314(4)	1.0798(7)	1.0398(5)	0.4613(2)	1.0427(6)	0.3716(1)	0.4822(3)
Yeast_spo	1.1121(5.5)	1.1151(7)	1.1113(4)	0.5935(2)	1.1121(5.5)	0.5367(1)	0.6486(3)
Yeast_spo5	0.5415(6)	0.5588(7)	0.5412(5)	0.3098(2)	0.5403(4)	0.2967(1)	0.3392(3)
Yeast_spoem	0.3176(6)	0.3245(7)	0.3173(5)	0.1935(2)	0.3168(4)	0.1907(1)	0.1974(3)
SBU_3DFE	1.1285(5)	1.2042(7)	1.1305(6)	0.8944(2)	1.1257(4)	0.8827(1)	0.9046(3)
SJAFFE	1.9610(6)	1.9755(7)	1.9585(5)	0.9116(3)	1.9512(4)	0.8948(2)	0.8666(1)
Natural_scene	7.0606(4)	7.1244(7)	7.0628(6)	6.9858(2)	7.0618(5)	6.9913(3)	6.8938(1)
Average Rank	4.7308(4)	7.0000(7)	5.3846(6)	2.0769(2)	4.8846(5)	1.2308(1)	2.6923(3)

Table 11
Label distribution predictive performance measured by Kullback–Leibler divergence ↓.

Dataset	LP	ML	GLLE	PLEML	LESC	LEFR	LE-GCN
Yeast_alpha	0.1221(4)	0.1262(7)	0.1228(5)	0.0115(2)	0.1231(6)	0.0055(1)	0.0342(3)
Yeast_cdc	0.1266(5.5)	0.1305(7)	0.1266(5.5)	0.0128(2)	0.1263(4)	0.0071(1)	0.0233(3)
Yeast_cold	0.0410(5)	0.0839(7)	0.0411(6)	0.0106(1)	0.0384(4)	0.0135(2)	0.0319(3)
Yeast_diau	0.1257(5)	0.1431(7)	0.1266(6)	0.0202(2)	0.1255(4)	0.0152(1)	0.0809(3)
Yeast_dtt	0.1071(4)	0.1115(7)	0.1083(5)	0.0112(2)	0.1089(6)	0.0065(1)	0.0169(3)
Yeast_elu	0.1215(4)	0.1263(7)	0.1226(6)	0.0121(2)	0.1224(5)	0.0063(1)	0.0234(3)
Yeast_heat	0.1216(4)	0.1302(7)	0.1230(5)	0.0196(2)	0.1235(6)	0.0132(1)	0.0221(3)
Yeast_spo	0.1114(7)	0.1076(5)	0.1102(6)	0.0532(4)	0.0524(3)	0.0384(2)	0.0365(1)
Yeast_spo5	0.1190(6)	0.1266(7)	0.1186(5)	0.0347(2)	0.1181(4)	0.0318(1)	0.0395(3)
Yeast_spoem	0.0559(7)	0.0374(4)	0.0546(6)	0.0537(5)	0.0223(2)	0.0220(1)	0.0292(3)
SBU_3DFE	0.1398(5.5)	0.1570(7)	0.1398(5.5)	0.0832(1)	0.1388(4)	0.0833(2)	0.1095(3)
SJAFFE	0.3648(6)	0.3745(7)	0.3643(5)	0.0731(2)	0.3595(4)	0.0735(3)	0.0720(1)
Natural_scene	1.3053(4)	1.3671(7)	1.3079(5.5)	1.1630(2)	1.3079(5.5)	1.1749(3)	1.0148(1)
Average Rank	5.1538(5)	6.6154(7)	5.5000(6)	2.2308(2)	4.4231(4)	1.5385(1)	2.5385(3)

Table 12
Label distribution predictive performance measured by Cosine coefficient ↑.

Dataset	LP	ML	GLLE	PLEML	LESC	LEFR	LE-GCN
Yeast_alpha	0.8824(4)	0.8801(7)	0.8820(5)	0.9881(2)	0.8818(6)	0.9945(1)	0.9729(3)
Yeast_cdc	0.8844(6)	0.8822(7)	0.8846(5)	0.9869(2)	0.8848(4)	0.9931(1)	0.9768(3)
Yeast_cold	0.9151(5)	0.9128(7)	0.9151(5)	0.9817(2)	0.9151(5)	0.9866(1)	0.9682(3)
Yeast_diau	0.9002(5)	0.8888(7)	0.8997(6)	0.9806(2)	0.9003(4)	0.9858(1)	0.9345(3)
Yeast_dtt	0.9218(4)	0.9181(7)	0.9208(5)	0.9888(2)	0.9205(6)	0.9937(1)	0.9840(3)
Yeast_elu	0.8894(4)	0.8856(7)	0.8886(6)	0.9875(2)	0.8888(5)	0.9938(1)	0.9772(3)
Yeast_heat	0.9062(4)	0.9021(7)	0.9060(5)	0.9807(2)	0.9057(6)	0.9873(1)	0.9782(3)
Yeast_spo	0.8929(7)	0.8936(4)	0.8933(5)	0.9683(2)	0.8931(6)	0.9739(1)	0.9654(3)
Yeast_spo5	0.9172(6)	0.9138(7)	0.9174(5)	0.9691(2)	0.9176(4)	0.9720(1)	0.9645(3)
Yeast_spoem	0.9381(6)	0.9365(7)	0.9384(5)	0.9745(3)	0.9386(4)	0.9754(1)	0.9751(2)
SBU_3DFE	0.8717(6)	0.8648(7)	0.8722(5)	0.9189(2)	0.8732(4)	0.9191(1)	0.8987(3)
SJAFFE	0.7689(7)	0.7734(4)	0.7698(6)	0.9306(3)	0.7719(5)	0.9325(1.5)	0.9325(1.5)
Natural_scene	0.5429(4)	0.5209(7)	0.5419(5.5)	0.5787(2)	0.5419(5.5)	0.5745(3)	0.6294(1)
Average Rank	5.2308(5)	6.5385(7)	5.2692(6)	2.1538(2)	4.9615(4)	1.1923(1)	2.6538(3)

Table 13
Label distribution predictive performance measured by intersection similarity ↑.

Dataset	LP	ML	GLLE	PLEML	LESC	LEFR	LE-GCN
Yeast_alpha	0.8298(4)	0.8259(7)	0.8290(5)	0.9423(2)	0.8289(6)	0.9616(1)	0.9122(3)
Yeast_cdc	0.8269(5)	0.8226(7)	0.8268(6)	0.9399(2)	0.8271(4)	0.9577(1)	0.9124(3)
Yeast_cold	0.8288(4)	0.8230(7)	0.8286(5.5)	0.9253(2)	0.8286(5.5)	0.9359(1)	0.8965(3)
Yeast_diau	0.8212(4)	0.8065(7)	0.8205(6)	0.9227(2)	0.8211(5)	0.9350(1)	0.8380(3)
Yeast_dtt	0.8372(4)	0.8304(7)	0.8350(5)	0.9424(2)	0.8346(6)	0.9571(1)	0.9274(3)
Yeast_elu	0.8288(4)	0.8232(7)	0.8278(6)	0.9399(2)	0.8280(5)	0.9576(1)	0.9151(3)
Yeast_heat	0.8269(4)	0.8201(7)	0.8261(5)	0.9229(2)	0.8257(6)	0.9388(1)	0.9192(3)
Yeast_spo	0.8117(7)	0.8118(6)	0.8121(4)	0.9010(2)	0.8119(5)	0.9114(1)	0.8926(3)
Yeast_spo5	0.8257(6)	0.8209(7)	0.8258(5)	0.8998(2)	0.8261(4)	0.9044(1)	0.8892(3)
Yeast_spoem	0.8507(6)	0.8479(7)	0.8511(5)	0.9063(2)	0.8513(4)	0.9076(1)	0.9045(3)
SBU_3DFE	0.7953(5.5)	0.7864(7)	0.7953(5.5)	0.8397(2)	0.7964(4)	0.8417(1)	0.8326(3)
SJAFFE	0.6759(7)	0.6788(4)	0.6765(6)	0.8455(3)	0.6780(5)	0.8494(2)	0.8531(1)
Natural_scene	0.3549(5)	0.3513(7)	0.3549(5)	0.3658(2)	0.3549(5)	0.3636(3)	0.4357(1)
Average Rank	5.0385(5)	6.6923(7)	5.3077(6)	2.0769(2)	4.9615(4)	1.2308(1)	2.6923(3)

Table 14
Wilcoxon signed-rank test for label recovery performance of LE-GCN versus five benchmarks and LEFR (significance level $\alpha = 0.05$, p-values shown in the brackets).

LE-GCN versus	Evaluation metric					
	Chebyshev	Clark	Canberra	KL divergence	Cosine	Intersection similarity
LP	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]
ML	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]
GLLE	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]
PLEML	WIN[0.00341796875]	WIN[0.00244140625]	WIN[0.021484375]	WIN[0.03417047269]	WIN[0.039794921875]	WIN[0.039794921875]
LESC	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000732421875]	WIN[0.000244140625]	WIN[0.000244140625]
LEFR	WIN[0.00244140625]	WIN[0.001220703125]	WIN[0.006103515625]	WIN[0.039794921875]	TIE[0.339599609375]	WIN[0.026611328125]

Table 15
Wilcoxon signed-ranks test for label prediction performance of LEFR versus five benchmarks and LE-GCN (significance level $\alpha = 0.05$, p-values shown in the brackets).

LEFR versus	Evaluation metric					
	Chebyshev	Clark	Canberra	KL divergence	Cosine	Intersection similarity
LP	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]
ML	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]
GLLE	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]
PLEML	WIN[0.001708984375]	WIN[0.00048828125]	WIN[0.000732421875]	TIE[0.057373046875]	WIN[0.00244140625]	WIN[0.001220703125]
LESC	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]	WIN[0.000244140625]
LE-GCN	WIN[0.00244140625]	WIN[0.001220703125]	WIN[0.006103515625]	WIN[0.039794921875]	WIN[0.034170472692]	WIN[0.026611328125]

Table 16
Bayesian signed-rank test for label recovery performance of LE-GCN versus five benchmarks and LEFR (rope=0.0001, default prior strength is 0.6).

LE-GCN versus	Evaluation metric					
	Chebyshev	Clark	Canberra	KL divergence	Cosine	Intersection similarity
LP	[0.99624, 2e−05, 0.00374]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[0.98732, 0.0, 0.01268]	[0.99998, 0.0, 2e−05]
ML	[0.99998, 0.0, 2e−05]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]
GLLE	[0.99932, 0.0, 0.00068]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[0.98564, 0.0, 0.01436]	[0.98726, 0.0, 0.01274]
PLEML	[0.99662, 0.0, 0.00338]	[0.99666, 0.0, 0.00334]	[0.99226, 0.0, 0.00774]	[0.98162, 0.0, 0.01838]	[0.97052, 0.0, 0.02948]	[0.95934, 0.0, 0.04066]
LESC	[0.98772, 0.0, 0.01228]	[0.9998, 0.0, 0.0002]	[0.99912, 0.0, 0.00088]	[0.99788, 0.0, 0.00212]	[0.90586, 0.0, 0.09414]	[0.96962, 0.0, 0.03038]
LEFR	[0.91436, 0.0, 0.08564]	[0.90826, 0.0, 0.09174]	[0.85384, 0.0, 0.14616]	[0.99824, 0.0, 0.00176]	[0.92958, 0.0, 0.07042]	[0.92752, 0.0, 0.07248]

Table 17
Bayesian signed-rank test for label prediction performance of LEFR versus five benchmarks and LE-GCN (rope=0.0001, default prior strength is 0.6).

LEFR versus	Evaluation metric					
	Chebyshev	Clark	Canberra	KL divergence	Cosine	Intersection similarity
LP	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]
ML	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]
GLLE	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]
PLEML	[0.9999, 0.0, 0.0001]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[0.98122, 0.0, 0.01878]	[0.99978, 0.0, 0.00022]	[0.99998, 0.0, 2e−05]
LESC	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]	[1.0, 0.0, 0.0]
LE-GCN	[0.9999, 0.0, 0.0001]	[1.0, 0.0, 0.0]	[0.99946, 0.0, 0.00054]	[0.9808, 0.0, 0.0192]	[0.98574, 0.0, 0.01426]	[0.98886, 0.0, 0.01114]

As confirmed clearly by the test results of Table 17, it is statistically significant that our LEFR outperforms the other six algorithms. The test results also show that our LE-GCN statistically outperforms PLEML, since the winning probabilities of LEFR over PLEML are either equal to or bigger than those of LEFR over LE-GCN. Therefore, the statistical test results convincingly demonstrate that our LEFR achieves the best label distribution prediction performance, while our LE-GCN is the second best.

5. Conclusions

First we have proposed a new LE method based on feature representation, called LEFR. Specifically, we have developed a framework that excavates the underlying reduced-dimensional feature information via a manifold learning and obtains the predicted label distribution during the intermediate procedure via label propagation. A new sample similarity matrix has been proposed to mine the correlation between the sample feature space and the sample label space. By combining the rich label information in low-dimensional feature space and the label distribution information estimated, an enhanced label distribution prediction model has been established, which can be trained through gradient descent optimization to yield the accurate label distribution prediction. Second, a novel LE method based on the graph convolutional network, named LE-GCN, has been designed. According to the similarity property, the connection threshold of the feature node has been determined, and the information reshaping of feature space and label space has been achieved. This has enabled LE-GCN to produce the accurate label distribution estimation. Experiments involving 13 real-word datasets have demonstrated the superior performance of our proposed LEFR and LE-GCN algorithms over several existing state-of-the-art LE algorithms, in terms of label enhancement learning and label distribution prediction.

Of particularly interesting observation is as follows. Our LE-GCN attains the best label distribution recovery performance with our LEFR as a close second best. In the label distribution prediction experiments by contrast, our LEFR is a clear winner, while our LE-GCN can only attain the second best. The superior LE capability of the proposed LE-GCN comes from its deep information mining structure which fully exploits the hidden relationships between feature nodes and labels. However, the optimization of the logistic regression based label predictor model for LE-GCN exhibits suboptimal behavior. Further research is warranted to investigate alternative label predictor form in order to fulfill the full potential of this new LE-GCN structure.

CRediT authorship contribution statement

Chao Tan: Writing – review & editing, Writing – original draft, Validation, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Sheng Chen:** Writing – review & editing, Supervision, Formal analysis. **Xin Geng:** Resources, Funding acquisition, Conceptualization. **Yunyao Zhou:** Visualization, Software, Data curation. **Genlin Ji:** Resources, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by National Natural Science Foundation of China under Grant 61702270, 41971343 and 62076063. This work was also supported by Industry Projects in Jiangsu S & T Pillar Program under Grant BE2023089 and Key Project of Natural Science Research in Jiangsu Provincial Colleges and Universities under Grant 21KJA520002. Dr Tan would like to thank the sponsorship of Chinese Scholarship Council for funding her research at School of Electronics and Computer Science, University of Southampton, UK.

References

[1] M.L. Zhang, et al., Leveraging implicit relative labeling-importance information for effective multi-label learning, IEEE Trans. Knowl. Data Eng. 33 (5) (2021) 2057–2070.

[2] X. Geng, Label distribution learning, IEEE Trans. Knowl. Data Eng. 28 (7) (2016) 1734–1748.

[3] X. Geng, Q. Wang, Y. Xia, Facial age estimation by adaptive label distribution learning, in: Proc. 22nd Int. Conf. Pattern Recognition, Stockholm, Sweden, 2014, pp. 4465–4470.

[4] Y. Zhou, H. Xue, X. Geng, Emotion distribution recognition from facial expressions, in: Proc. 23rd ACM Int. Conf. Multimedia, Brisbane, Australia, 2015, pp. 1247–1250.

[5] X. Geng, M. Ling, Soft video parsing by label distribution learning, in: Proc. AAAI 2017, San Francisco, CA, USA, 2017, pp. 1331–1337.

[6] L. Qi, et al., Label distribution learning for generalizable multisource person re-identification, IEEE Trans. Inf. Forensics Secur. 17 (2022) 3139–3150.

[7] N. Xu, Y.P. Liu, X. Geng, Label enhancement for label distribution learning, IEEE Trans. Knowl. Data Eng. 33 (4) (2021) 1632–1643.

[8] G. Tsoumakas, I. Katakis, Multi-label classification: An overview, Int. J. Data Warehous. Min. 3 (3) (2009) 1–13.

[9] Z. Zhang, H. Zha, Principal manifolds and nonlinear dimension reduction via local tangent space alignment, SIAM J. Sci. Comput. 26 (1) (2002) 313–338.

[10] T. Chen, et al., Learning semantic-specific graph representation for multi-label image recognition, in: Proc. ICCV 2019, Vol. 2, Seoul, South Korea, 2019, pp. 522–531.

[11] R. You, et al., Cross-modality attention with semantic graph embedding for multi-label classification, in: Proc. AAAI 2020, New York, NY, USA, 2020, pp. 12709–12716.

[12] Y. Wu, et al., GM-MLIC: Graph matching based multi-label image classification, in: Proc. IJCAI 2021, Montreal, QC, Canada, 2021, pp. 1179–1185.

[13] Z. Chen, X. Wei, P. Wang, Y. Guo, Multi-label image recognition with graph convolutional networks, in: Proc. CVPR 2019, Long Beach, CA, USA, 2019, pp. 5177–5186.

[14] N.E. Gayar, F. Schwenker, G. Palm, A study of the robustness of knn classifiers trained using soft labels, in: Proc. ANNPR 2006, Vol. 2, Ulm, Germany, 2006, pp. 67–80.

[15] X. Jiang, Z. Yi, J.C. Lv, Fuzzy SVM with a new fuzzy membership function, Neural Comput. Appl. 15 (3) (2006) 268–276.

[16] P. Hou, X. Geng, M.-L. Zhang, Multi-label manifold learning, in: Proc. AAAI 2016, Phoenix, AZ, USA, 2016, pp. 1680–1686.

[17] H. Tang, et al., Label enhancement with sample correlations via low-rank representation, in: Proc. AAAI 2020, New York, NY, USA, 2020, pp. 5932–5939.

[18] W. Zhu, X. Jia, W. Li, Privileged label enhancement with multi-label learning, in: Proc. IJCAI-PRICAI 2020, Yokohama, Japan, 2021, pp. 2376–2382.

[19] N. Xu, et al., Variational label enhancement, IEEE Trans. Pattern Anal. Mach. Intell. 45 (5) (2023) 6537–6551.

[20] Y. Lu, et al., Generative label enhancement with gaussian mixture and partial ranking, in: Proc. AAAI 2023, Washington, DC, USA, 2023, pp. 8975–8983.

[21] Y. Wang, et al., Contrastive label enhancement, in: Proc. IJCAI 2023, Macao, SAR, China, 2023, pp. 4353–4361.

[22] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.

[23] X. Zhu, Semi-Supervised Learning with Graphs (Ph.D. thesis) CMU-LTI-05-192, School of Computer Science, Carnegie Mellon University, USA, 2005.

[24] J. Nocedal, S.J Wright, Numerical Optimization, Springer, New York, NY, USA, 2006.

[25] V. Vapnik, A. Vashist, A new learning paradigm: Learning using privileged information, Neural Netw. 22 (5–6) (2009) 544–557.

[26] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, in: Proc. ICCV 2015, Santiago, Chile, 2015, pp. 1026–1034.

[27] F. Párez-Cruz, A. Navia-Vázquez, P.L. Alarcón-Diana, A. Artés-Rodríguez, An IRWLS procedure for SVR, in: Proc. 10th European Signal Processing Conf, Tampere, Finland, 2000, pp. 1–4.

[28] S. Cha, Comprehensive survey on distance/similarity measures between probability density functions, Int. J. Math. Models Methods Appl. Sci. 1 (4) (2007) 300–307.

[29] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (1) (2006) 1–30.

[30] A. Benavoli, G. Corani, J. Demšar, M. Zaffalon, Time for a change: A tutorial for comparing multiple classifiers through Bayesian analysis, J. Mach. Learn. Res. 18 (77) (2017) 1–36.

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 Ph.D. 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 in the School of Computer and Electronic Information/School of Artificial Intelligence at present. She has worked as a postdoctoral researcher in Southeast University. Her research interests generally focus on machine learning, multi-label manifold learning and data mining.

Sheng Chen received his BEng degree from the East China Petroleum Institute, Dongying, China, in 1982, and his Ph.D. 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, wireless communications, modeling and identification of nonlinear systems, evolutionary computation methods and optimization. He has published over 700 research papers. Dr. Chen is a Fellow of the United Kingdom Royal Academy of Engineering, a Fellow of IEEE, 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 20,500+ Google Scholar citations with h-index 75.

Xin Geng received the B.Sc. and M.Sc. degrees in Computer Science from Nanjing University, China, in 2001 and 2004, respectively, and the Ph.D. degree from Deakin University, Australia in 2008. He is currently a professor in the school of Computer Science and Engineering and the dean of the graduate school 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.

Yunyao Zhou is currently an undergraduate student in the School of Computer and Electronic Information/School of Artificial Intelligence at Nanjing Normal University. His research interests include machine learning and graph mining.

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 Ph.D. degree in Computer Science and Technology from Southeast University in 2004. He is now a professor in the School of Computer and Electronic Information/School of Artificial Intelligence at Nanjing Normal University. His research interests generally focus on data mining and its application.