Label Enhancement for Label Distribution Learning

Ning Xu^{1,2}, **An Tao**^{3} and **Xin Geng**^{1,2,*}
^{1}MOE Key Laboratory of Computer Network and Information Integration, China
^{2}School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
^{3}School of Information Science and Engineering, Southeast University, Nanjing 210096, China
{xning, taoan, xgeng}@seu.edu.cn

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 the 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 datasets show clear advantages of GLLE over several existing LE algorithms.

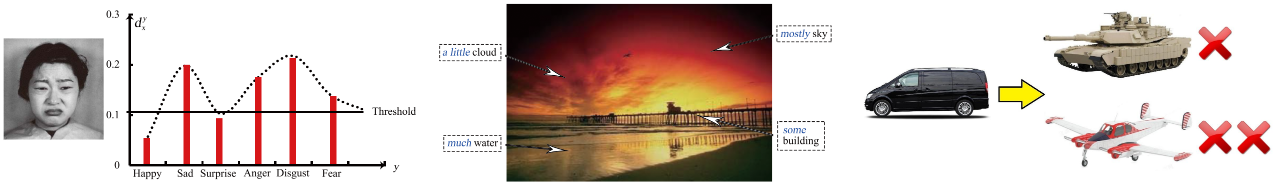
1 Introduction

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) [Tsoumakas and Katakis, 2006] 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 [Gibaja and Ventura, 2015; Zhang and Zhou, 2014]. During the past decade, multi-label learning techniques have been widely employed to learn from data with rich semantics, such as text [Rubin *et al.*, 2012], image [Cabral *et al.*, 2011], audio [Lo *et al.*, 2011], video [Wang *et al.*, 2011], 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) [Geng, 2016]. 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) [Zhou *et al.*, 2015]. As shown in Fig. 1a, 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. 1b, 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.

[*] Corresponding author



(a) Emotion (b) Nature scene (c) Target

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

- The "irrelevance" of each irrelevant label may be very different. For example, in Fig. 1c, 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 to recover the label distributions from the logical labels in the training set via leveraging the topological information in the feature space and the correlation among the labels. This process is called *label enhancement* (LE) in this paper. LE reinforces the supervision information in the training sets by exploiting the relative importance of each label. After the label distributions are recovered, more effective supervised learning can be achieved by leveraging the label distributions [Li *et al.*, 2015; Hou *et al.*, 2016].

Note that although there is no explicit concept of LE defined in existing literatures, 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 [Geng and Xia, 2014] and facial age estimation [Geng *et al.*, 2014]. Some works [Gayar *et al.*, 2006; Jiang *et al.*, 2006] build the membership degrees to the labels, which can constitute a label distribution. Some works [Li *et al.*, 2015; Hou *et al.*, 2016] establish the relationship between instances and labels by graph and transfer logical labels into label distributions.

The rest of this paper is organized as follows. First, the formulation of LE and the details of the LE algorithms are proposed in Section 2. After that, the results of the comparative experiments are reported in Section 3. Finally, conclusions are drawn in Section 4.

2 Label Enhancement

2.1 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 $\boldsymbol{l}_i = (l_{\boldsymbol{x}_i}^{y_1}, l_{\boldsymbol{x}_i}^{y_2}, ..., l_{\boldsymbol{x}_i}^{y_c})^\top$, where c is the number of possible labels. The description degree of y to $\boldsymbol{x}$ is denoted by $d_{\boldsymbol{x}}^y$, and the label distribution of $\boldsymbol{x}_i$ is denoted by $\boldsymbol{d}_i = (d_{\boldsymbol{x}_i}^{y_1}, d_{\boldsymbol{x}_i}^{y_2}, ..., d_{\boldsymbol{x}_i}^{y_c})^\top$. 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 $\mathcal{E} = \{(\boldsymbol{x}_i, \boldsymbol{d}_i) | 1 \leq i \leq n\}$.

2.2 Existing Label Enhancement Algorithms

Fuzzy Label Enhancement

The LE algorithm based on fuzzy clustering (FCM) [Gayar *et al.*, 2006] employs fuzzy C-means clustering [Castillo and Melin, 2005] 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

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

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

$$\boldsymbol{A}_j = \boldsymbol{A}_j + \boldsymbol{m}_{\boldsymbol{x}_i}, if l_{\boldsymbol{x}_i}^{y_j} = 1, \tag{2}$$

where $\boldsymbol{m}_{\boldsymbol{x}_i} = (m_{\boldsymbol{x}_i}^1, m_{\boldsymbol{x}_i}^2, ..., m_{\boldsymbol{x}_i}^p)$. Then, the membership degree vector of $\boldsymbol{x}_i$ to the labels is calculated by using fuzzy composition $\tilde{\boldsymbol{d}}_i = \boldsymbol{A} \circ \boldsymbol{m}_{\boldsymbol{x}_i}^\top$. Finally, the label distribution corresponding to each instance is generated via the softmax normalization $d_{\boldsymbol{x}_i}^y = \frac{e^{\tilde{d}_{\boldsymbol{x}_i}^y}}{\sum_y e^{\tilde{d}_{\boldsymbol{x}_i}^y}}$.

For each label y_j, the LE algorithm based on kernel method (KM) [Jiang *et al.*, 2006] 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 $l_{\boldsymbol{x}_i}^{y_j} = 1$ and $C_-^{y_j}$ contains such sample point $\boldsymbol{x}_i$ with $l_{\boldsymbol{x}_i}^{y_j} = 0$. Then, the center of $C_+^{y_j}$ in the feature space is defined by $\boldsymbol{\psi}^{y_j} = \frac{1}{n_+} \sum_{\boldsymbol{x}_i \in C_+^{y_j}} \varphi(\boldsymbol{x}_i)$, 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

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

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

Graph-based Label Enhancement

The LE algorithm based on label propagation (LP) [Li *et al.*, 2015] recovers the label distributions from logical labels by using iterative label propagation technique [Zhu and Goldberg, 2009]. 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

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

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 $\hat{a}_{ii} = \sum_{j=1}^{n} a_{ij}$. 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 we normalize the label distributions by using the softmax normalization.

The LE algorithm based on manifold learning (ML) [Hou *et al.*, 2016] 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 [Roweis and Saul, 2000; Wang and Zhang, 2008]. Then, the approximation of the feature manifold is to induce the minimization of

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

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 [Zhu *et al.*, 2005], 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

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

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

2.3 The GLLE Algorithm

This section proposes a new LE algorithm named Graph Laplacian Label Enhancement (GLLE). Given a training set $\mathcal{S}$, we construct the feature matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n]$ and the logical label matrix $\boldsymbol{L} = [\boldsymbol{l}_1, \boldsymbol{l}_2, ..., \boldsymbol{l}_n]$. 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, we consider the 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, we set $\hat{\boldsymbol{W}} = [\boldsymbol{W}^\top, \boldsymbol{b}]$ and $\boldsymbol{\phi}_i = [\varphi(\boldsymbol{x}_i); 1]$. 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 $\boldsymbol{x}_i$. Then, the optimization problem becomes

$$\min_{\hat{\boldsymbol{W}}} L(\hat{\boldsymbol{W}}) + \lambda \Omega(\hat{\boldsymbol{W}}), \tag{11}$$

where L is a loss function, Ω is the function to mine hidden label importance, and λ is the parameter trading off the two terms. 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, we choose the least squares (LS) loss function as

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

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

In order to mine the hidden label importance from the training examples via leveraging the topological information of the feature space, we specify the local similarity matrix $\boldsymbol{A}$ whose elements are calculated by

$$a_{ij} = \begin{cases} \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right) & \text{if } \boldsymbol{x}_j \in N(i) \\ 0 & \text{otherwise} \end{cases}, \tag{13}$$

where $N(i)$ means the set of $\boldsymbol{x}_i$'s K-nearest neighbors, and $\sigma > 0$ is the width parameter for similarity calculation which is fixed to be 1 in this paper. According to the smoothness assumption [Zhu *et al.*, 2005], the points close to each other are more likely to share a label. Intuitively, if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ have

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

Table 1: Statistics of the 15 Datasets Used in the Experiments

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 which we wish to minimize:

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

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

Formulating the LE problem into an optimization framework over Eq. (12) and Eq. (14) yields the target function of $\hat{\boldsymbol{W}}$

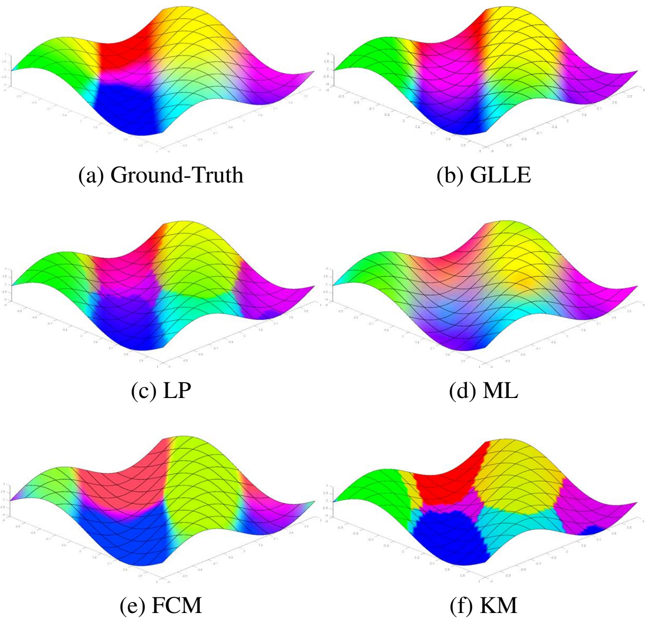
$$\begin{aligned}T(\hat{\boldsymbol{W}}) = \mathrm{tr}[(\hat{\boldsymbol{W}}\boldsymbol{\Phi} - \boldsymbol{L})^\top(\hat{\boldsymbol{W}}\boldsymbol{\Phi} - \boldsymbol{L})] \\ +\lambda\mathrm{tr}(\hat{\boldsymbol{W}}\boldsymbol{\Phi}\boldsymbol{G}\boldsymbol{\Phi}^\top\hat{\boldsymbol{W}}^\top).\end{aligned} \tag{15}$$

The optimization of Eq. (15) uses an effective quasi-Newton method BFGS [Nocedal and Wright, 2006]. 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

$$\begin{aligned}\frac{\partial T}{\partial \hat{\boldsymbol{W}}} = 2\hat{\boldsymbol{W}}\boldsymbol{\Phi}\boldsymbol{\Phi}^\top - 2\boldsymbol{L}\boldsymbol{\Phi}^\top + \lambda\hat{\boldsymbol{W}}\boldsymbol{\Phi}\boldsymbol{G}^\top\boldsymbol{\Phi}^\top \\ +\lambda\hat{\boldsymbol{W}}\boldsymbol{\Phi}\boldsymbol{G}\boldsymbol{\Phi}^\top.\end{aligned} \tag{16}$$

When the best parameter $\hat{\boldsymbol{W}}^*$ is determined, the label distribution $\boldsymbol{d}_i$ can be generated through Eq. (10). Finally, we normalize $\boldsymbol{d}_i$ by using the softmax normalization.

According to the representor's theorem [Smola, 1999], 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 we replace this expression into Eq. (15) and Eq. (16), it will generate the inner product$< \varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j) >$, and then the kernel trick can be applied.


(a) Ground-Truth

(b) GLLE

(c) LP

(d) ML

(e) FCM

(f) KM

Figure 2: Comparison between the ground-truth and recovered label distributions (regarded as RGB colors) on the artificial manifold.

3 Experiments

3.1 Datasets

There are in total 15 datasets used in the experiments including an artificial toy dataset and 14 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 $\boldsymbol{d} = (d_{\boldsymbol{x}}^{y_1}, d_{\boldsymbol{x}}^{y_2}, d_{\boldsymbol{x}}^{y_3})^\top$ of $\boldsymbol{x} = (x_1, x_2, x_3)^\top$ is created in the following way.

$$t_i = ax_i + bx_i^2 + cx_i^3 + d, i = 1, ..., 3, \tag{17}$$

$$\psi_1 = (\boldsymbol{h}_1^\top\boldsymbol{t})^2, \psi_2 = (\boldsymbol{h}_2^\top\boldsymbol{t} + \beta_1\psi_1)^2, \psi_3 = (\boldsymbol{h}_3^\top\boldsymbol{t} + \beta_2\psi_2)^2, \tag{18}$$

$$d_{\boldsymbol{x}}^{y_i} = \frac{\psi_i}{\psi_1 + \psi_2 + \psi_3}, i = 1, ..., 3, \tag{19}$$

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

$$x_3 = \sin((x_1 + x_2) \times \pi). \tag{20}$$

Then, the label distribution $\boldsymbol{d}$ corresponding to each is calculated via Eq. (17)-(19).

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

Datasets	FCM	KM	LP	ML	GLLE
Artificial	0.188(3)	0.260(5)	0.130(2)	0.227(4)	**0.108(1)**
SJAFFE	0.132(3)	0.214(5)	0.107(2)	0.190(4)	**0.100(1)**
Natural Scene	0.368(5)	0.306(4)	**0.275(1)**	0.295(2)	0.296(3)
Yeast-spoem	0.233(3)	0.408(5)	0.163(2)	0.400(4)	**0.108(1)**
Yeast-spo5	0.162(3)	0.277(5)	0.114(2)	0.273(4)	**0.092(1)**
Yeast-dtt	0.097(2)	0.257(5)	0.128(3)	0.244(4)	**0.065(1)**
Yeast-cold	0.141(3)	0.252(5)	0.137(2)	0.242(4)	**0.093(1)**
Yeast-heat	0.169(4)	0.175(5)	0.086(2)	0.165(3)	**0.056(1)**
Yeast-spo	0.130(3)	0.175(5)	0.090(2)	0.171(4)	**0.067(1)**
Yeast-diau	0.124(3)	0.152(5)	0.099(2)	0.148(4)	**0.084(1)**
Yeast-elu	0.052(3)	0.078(5)	0.044(2)	0.072(4)	**0.030(1)**
Yeast-cdc	0.051(3)	0.076(5)	0.042(2)	0.071(4)	**0.038(1)**
Yeast-alpha	0.044(3)	0.063(5)	0.040(2)	0.057(4)	**0.033(1)**
SBU_3DFE	0.135(2)	0.238(5)	**0.123(1)**	0.233(4)	0.141(3)
Movie	0.230(4)	0.234(5)	0.161(2)	0.164(3)	**0.160(1)**
Avg. Rank	3.13	4.93	1.93	3.73	1.27

Table 2: Recovery Results (value(rank)) Measured by Cheb ↓

Datasets	FCM	KM	LP	ML	GLLE
Artificial	0.933(3)	0.918(5)	0.974(2)	0.925(4)	**0.980(1)**
SJAFFE	0.906(3)	0.827(5)	0.941(2)	0.857(4)	**0.946(1)**
Natural Scene	0.593(5)	0.748(4)	**0.860(1)**	0.818(2)	0.769(3)
Yeast-spoem	0.878(3)	0.812(5)	0.950(2)	0.815(4)	**0.968(1)**
Yeast-spo5	0.922(3)	0.882(5)	0.969(2)	0.884(4)	**0.974(1)**
Yeast-dtt	0.959(2)	0.759(5)	0.921(3)	0.763(4)	**0.983(1)**
Yeast-cold	0.922(3)	0.779(5)	0.925(2)	0.784(4)	**0.969(1)**
Yeast-heat	0.883(3)	0.779(5)	0.932(2)	0.783(4)	**0.980(1)**
Yeast-spo	0.909(3)	0.800(5)	0.939(2)	0.803(4)	**0.968(1)**
Yeast-diau	0.882(3)	0.799(5)	0.915(2)	0.803(4)	**0.939(1)**
Yeast-elu	0.950(2)	0.758(5)	0.918(3)	0.763(4)	**0.978(1)**
Yeast-cdc	0.929(2)	0.754(5)	0.916(3)	0.759(4)	**0.959(1)**
Yeast-alpha	0.922(2)	0.751(5)	0.911(3)	0.756(4)	**0.973(1)**
SBU_3DFE	0.912(2)	0.812(5)	**0.922(1)**	0.815(4)	0.900(3)
Movie	0.773(5)	0.880(4)	**0.929(1)**	0.919(2)	0.900(3)
Avg. Rank	2.93	4.87	2.07	3.73	1.40

Table 3: Recovery Results (value(rank)) Measured by Cosine ↑

The second to the fourteen datasets are real-world LDL datasets [Geng, 2016] collected from biological experiments on the yeast genes, facial expression images, natural scene images and movies, respectively.

3.2 Evaluation Measures

In order to compare the recovered label distribution with the ground-truth, a natural choice of the evaluation measure is the average distance or similarity between the recovered label distribution and the ground-truth label distribution. According to Geng's suggestion [Geng, 2016], we select six LDL measures, i.e., Chebyshev distance (Cheb), Clark distance (Clark), Canberra metric (Canber), Kullback-Leibler divergence (KL), cosine coefficient (Cosine) and intersection similarity (Intersec). The first four are distance measures and the last two are similarity measures. Due to page limitation, we only show representative results on Cheb and Cosine. Those results on other evaluation measures are similar.

3.3 Methodology

The four algorithms described in Section 2.2, i.e., FCM [Gayar *et al.*, 2006], KM [Jiang *et al.*, 2006], LP [Li *et al.*, 2015], ML [Hou *et al.*, 2016], and our GLLE are all applied to the 15 datasets shown in Table 1.

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, $T = 0.5$.

There are two parts in the experiments. In the first part, 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. In the second part, 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 model directly trained on the ground-truth label distributions. Ten-fold cross validation is conducted for each algorithm.

For GLLE, the parameter λ is 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. 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. The kernel function in KM is Gaussian kernel. The LDL algorithm used in this paper is SA-BFGS [Geng, 2016].

3.4 Experimental Results

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. 2. 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.

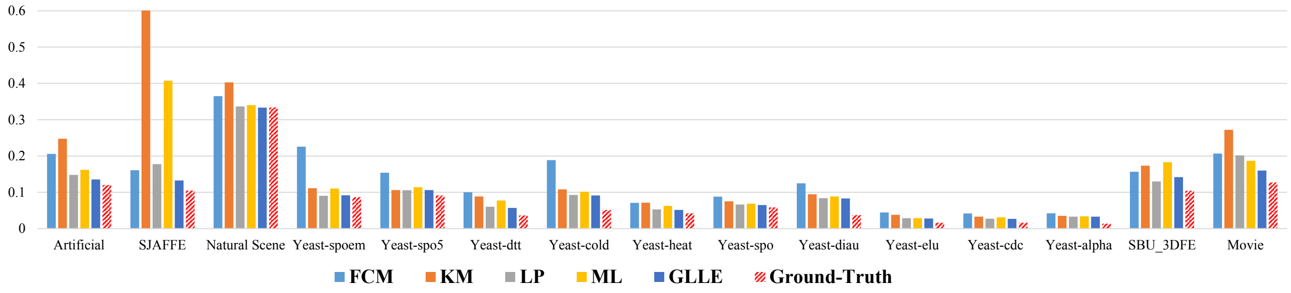


Figure 3: Comparison of the LDL after the LE pre-process against the direct LDL measured by Cheb ↓.

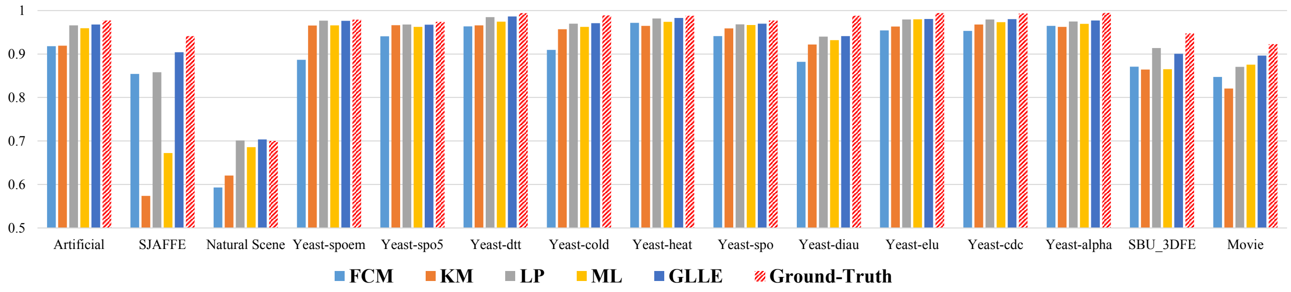


Figure 4: Comparison of the LDL after the LE pre-process against the direct LDL measured by Cosine ↑.

Table 4: The Average Ranks of Five Algorithms on Six Measures

Criterion	FCM	KM	LP	ML	GLLE
Cheb	4.40	4.20	2.00	3.13	1.27
Clark	4.33	4.07	2.27	3.07	1.27
Canber	4.20	4.13	2.27	3.13	1.27
KL	4.37	4.30	2.00	3.13	1.20
Cosine	4.53	4.27	1.93	3.07	1.20
Intersec	4.40	4.20	1.93	3.13	1.33

For quantitative analysis, Table 2 and Table 3 tabulate 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. The performances of the five LE algorithms evaluated by six measures are ranked as GLLE≻LP≻FCM≻ML≻KM. GLLE ranks *1st* in 86.7% cases and ranks *2nd* in 6.7% cases across six evaluation measures. Thus, GLLE generally performs better than other LE algorithms.

LDL Predictive Performance

In this experiment, ‘Ground-Truth’ represents the predictions made by the LDL model directly trained on the ground-truth label distributions. Then, ‘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. All the algorithms are tested via ten-fold cross validation. The histograms of the LDL predictive performances are given in Fig. 3 and Fig. 4. The average rank of each algorithm over all the datasets is shown in Table 4. Note that since ‘Ground-Truth’ is regarded as a upper bound performance in this experiment, we rank FCM, KM, LP, ML and GLLE without considering Ground-Truth.

Based on the experimental results, GLLE ranks *1st* in 78.9% cases and ranks *2nd* in 16.7% 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 Ground-Truth, especially on the Nature Scene and Yeast-spoem datasets. But the difference between them is relatively larger on a few datasets (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 and Yeast-spoem 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 Ground-Truth is closer by 12.9% on Cheb, 16.9% on Clark, 17.0% on Canber, 18.0% on KL, 23.1% on Cosine, and 18.9% 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.

4 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 via leveraging 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. In the future, we will explore if there exist better ways to recover the label distributions.

Acknowledgements

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

[Cabral *et al.*, 2011] Ricardo S. Cabral, Fernando De la Torre, João P. Costeira, and Alexandre Bernardino. Matrix completion for multi-label image classification. In *Advances in Neural Information Processing Systems*, pages 190–198, Granada SPAIN, 2011.

[Castillo and Melin, 2005] Oscar Castillo and Patricia Melin. *Hybrid intelligent systems for pattern recognition using soft computing: An evolutionary approach for neural networks and fuzzy systems*. Springer, New York, 2005.

[Gayar *et al.*, 2006] Neamat El Gayar, Friedhelm Schwenker, and Günther 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*, pages 67–80, Ulm, Germany, 2006.

[Geng and Xia, 2014] Xin Geng and Yu Xia. Head pose estimation based on multivariate label distribution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1837–1842, Columbus, OH, 2014.

[Geng *et al.*, 2014] Xin Geng, Qin Wang, and Yu Xia. Facial age estimation by adaptive label distribution learning. In *Proceedings of the 22nd International Conference on Pattern Recognition*, pages 4465–4470, Stockholm, Sweden, 2014.

[Geng, 2016] Xin Geng. Label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1734–1748, 2016.

[Gibaja and Ventura, 2015] Eva Gibaja and Sebastian Ventura. A tutorial on multilabel learning. *ACM Computing Surveys*, 47(3):1–38, 2015.

[Hou *et al.*, 2016] Peng Hou, Xin Geng, and Min-Ling Zhang. Multi-label manifold learning. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 1680–1686, Phoenix, AZ, 2016.

[Jiang *et al.*, 2006] Xiufeng Jiang, Zhang Yi, and Jian Cheng Lv. Fuzzy svm with a new fuzzy membership function. *Neural Computing & Applications*, 15(3-4):268–276, 2006.

[Li *et al.*, 2015] Yu-Kun Li, Min-Ling Zhang, and Xin Geng. Leveraging implicit relative labeling-importance information for effective multi-label learning. In *Proceedings of the 15th IEEE International Conference on Data Mining*, pages 251–260, Atlantic City, NJ, 2015.

[Lo *et al.*, 2011] Hung-Yi Lo, Ju-Chiang Wang, Hsin-Min Wang, and Shou-De Lin. Cost-sensitive multi-label learning for audio tag annotation and retrieval. *IEEE Transactions on Multimedia*, 13(3):518–529, 2011.

[Nocedal and Wright, 2006] Jorg Nocedal and Stephen J Wright. *Numerical optimization*. Springer, New York, 2006.

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

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

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

[Tsoumakas and Katakis, 2006] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2006.

[Wang and Zhang, 2008] Fei Wang and Changshui Zhang. Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering*, 20(1):55–67, 2008.

[Wang *et al.*, 2011] Jingdong Wang, Yinghai Zhao, Xiuqing Wu, and Xian-Sheng Hua. A transductive multi-label learning approach for video concept detection. *Pattern Recognition*, 44(10):2274–2286, 2011.

[Zhang and Zhou, 2014] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2014.

[Zhou *et al.*, 2015] Ying Zhou, Hui Xue, and Xin Geng. Emotion distribution recognition from facial expressions. In *Proceedings of the 23rd ACM International Conference on Multimedia*, pages 1247–1250, Brisbane, Australia, 2015.

[Zhu and Goldberg, 2009] Xiaojin Zhu and Andrew B. Goldberg. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–130, 2009.

[Zhu *et al.*, 2005] Xiaojin Zhu, John Lafferty, and Ronald Rosenfeld. *Semi-supervised learning with graphs*. Carnegie Mellon University, language technologies institute, school of computer science, 2005.