Generative Label Enhancement with Gaussian Mixture and Partial Ranking

Yunan Lu[1], Liang He[1], Fan Min[2], Weiwei Li[3,4], Xiuyi Jia[1]*

[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
[2] School of Computer Science, Southwest Petroleum University, Chengdu, China
[3] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
[4] Ministry Key Laboratory for Safety-Critical Software Development and Verification, Nanjing University of Aeronautics and Astronautics, Nanjing, China
{luyn, heliang, jiaxy}@njust.edu.cn, minfan@swpu.edu.cn, liweiwei@nuaa.edu.cn

Abstract

Label distribution learning (LDL) is an effective learning paradigm for dealing with label ambiguity. When applying LDL, the datasets annotated with label distributions (i.e., the real-valued vectors like the probability distribution) are typically required. Unfortunately, most existing datasets only contain the logical labels, and manual annotating with label distributions is costly. To address this problem, we treat the label distribution as a latent vector and infer its posterior by variational Bayes. Specifically, we propose a generative label enhancement model to encode the process of generating feature vectors and logical label vectors from label distributions in a principled way. In terms of features, we assume that the feature vector is generated by a Gaussian mixture dominated by the label distribution, which captures the one-to-many relationship from the label distribution to the feature vector and thus reduces the feature generation error. In terms of logical labels, we design a probability distribution to generate the logical label vector from a label distribution, which captures partial label ranking in the logical label vector and thus provides a more accurate guidance for inferring the label distribution. Besides, to approximate the posterior of the label distribution, we design a inference model, and derive the variational learning objective. Finally, extensive experiments on real-world datasets validate our proposal.

Introduction

Label distribution learning (LDL) (Geng 2016) is a new learning paradigm, where each instance is annotated by a label distribution (a real-valued vector like the probability distribution). Each element in label distribution is called label description degree which indicates the relative importance degree of each label. Currently, LDL has been applied to a variety of real-world tasks, such as sentiment analysis (Jia et al. 2019; Li et al. 2021; He and Jin 2019), facial age estimation (Hou et al. 2017; Gao et al. 2018; Wen et al. 2020), head pose estimation (Liu et al. 2019; Zhang et al. 2020).

In the application of LDL, the datasets annotated with label distributions are typically required. Unfortunately, most existing datasets only contain the logical labels, and manual annotating with label distributions is costly. Therefore, Xu, Tao, and Geng (2018) proposed label enhancement (LE) to automatically recover label distributions from the dataset annotated with logical labels by mining the additional information underlying the feature space and the label space.

Most existing LE methods are based on the discriminative approach, which model the label distribution by a decision function or a conditional likelihood function, and use some additional information to learn the model parameters. For example, methods such as LESC (Tang et al. 2020), ML^2 (Hou, Geng, and Zhang 2016) and LEMLL (Shao, Geng, and Xu 2018) assume that each instance can be represented as a linear combination of other instances, and use such linear combination to learn the decision function. GLLE (Xu, Liu, and Geng 2021) and PLLE (Xu, Lv, and Geng 2019) use a similarity graph to describe the instance relationship and learn the decision function with the help of this graph. However, the superior predictive performance of the discriminative approach relies on the accurate supervisory information (Jebara 2012). Unfortunately, the LE task lacks a ground-truth label distribution as the supervisory information, which limits the capability of discriminative approach. The generative approach aims to model the joint distribution of observed data and is more concerned with the underlying patterns of the observed data, which is more important for unsupervised task (Jebara 2012). Hence, generative approach is more suitable for LE (an unsupervised task). To the best of our knowledge, LEVI (Xu et al. 2020) is the only generative LE method available. It treats the label distribution as a latent vector and infers its posterior by variational inference. The experimental results of LEVI also demonstrate the merits of generative approach in LE task.

However, LEVI has two drawbacks: (1) LEVI assumes that the feature vector $\boldsymbol{x}$ is generated from a Gaussian distribution $\mathcal{N}(\boldsymbol{x}|\varphi(\boldsymbol{d}))$ dominated by a nonlinear transformation $\varphi(\cdot)$ of the label distribution $\boldsymbol{d}$; in other words, if some instances are annotated with the same label distribution, then LEVI assumes that they can be generated from the same Gaussian distribution. However, in practical applications, the same label distribution is likely to correspond to multiple widely different instances; as shown in Fig. 1, in the Emotion6 dataset (Peng et al. 2015), the label distribution $[0, 0, 0, 0.67, 0, 0, 0.27, 0.06]$ corresponds to seven images with various styles, which are obviously difficult to generate from the the same Gaussian distribution. (2) LEVI uses the mean squared error (MSE) $\|\boldsymbol{y} - \boldsymbol{d}\|_2^2$ to penalize the dif-

*Corresponding author

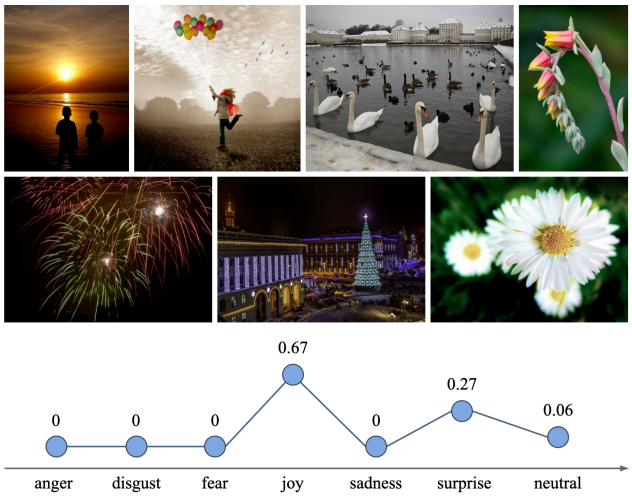


Figure 1: The examples from Emotion6 dataset that one label distribution corresponds to various images.

ference between the label distribution $\boldsymbol{d}$ and the logical label $\boldsymbol{y}$. On the one hand, minimizing MSE amounts to driving the recovered label distribution and the logical label vector to be numerically identical, yet they are actually different. On the other hand, MSE may corrupt the consistency of partial label ranking, i.e., the description degrees of irrelevant labels may exceed that of relevant labels (Jia, Lu, and Zhang 2021), while the accurate label ranking is crucial for LDL (Jia et al. 2021). Therefore, MSE is a defective guide for LE.

Therefore, in this paper, we propose GLEMR, a novel **G**enerative **L**abel **E**nhancement model with Gaussian **M**ixture and partial **R**anking. GLEMR contains a generative model and an inference model. The generative model describes the process of generating feature vectors and logical labels. First, we generate a real-valued vector $\boldsymbol{z}$ from a Gaussian prior and normalize it to a label distribution by the softmax function. Next, we generate the logical label vector by our designed probability distribution $p(\boldsymbol{y}|\boldsymbol{d})$. Inspired by the idea of partial label ranking, we want $p(\boldsymbol{y}|\boldsymbol{d})$ to be proportional to the degree of consistency between $\boldsymbol{y}$ and $\boldsymbol{d}$ w.r.t. partial label ranking; in other words, if the description degree of relevant labels exceeds the irrelevant labels by more, then the probability $p(\boldsymbol{y}|\boldsymbol{d})$ will be larger. Obviously, $p(\boldsymbol{y}|\boldsymbol{d})$ penalizes the ranking error between the relevant and irrelevant labels without affecting the relative importance among the relevant labels, thus overcoming the drawbacks of MSE. Besides, we employ a Gaussian mixture to represent the one-to-many mapping from the label distribution to the feature vector, thus overcoming the limitations of LEVI in feature generation. Based on the above generative model, we construct a corresponding inference model to approximate the posterior of the label distribution. In the inference model, we retain the exact posterior of the Gaussian component index (thus to some extent reducing the information loss from mean-field assumption), and derive the variational learning objective in the SGVB (Kingma and Welling 2014) framework. Finally, we compare our proposal with some state-of-the-art LE methods on several real-world datasets, and the experimental results validate of our proposal. Our contributions can be summarized as follows:

- We exploit Gaussian mixture to generate feature vectors from label distributions, which can model the one-to-many mapping from label distributions to feature vectors, thus reducing the error in the generation process of feature vectors.
- We design a probability distribution to generate logical label vectors from label distributions, which captures ranking information instead of numerical information in logical labels, thus providing a more accurate guidance to label distribution inference.
- We design an inference model according to the generation process and show how to perform efficient inference in the SGVB framework.

Related Work

The domain that is relevant to our work is label enhancement. Currently, most label enhancement methods are based on a discriminative approach. They use a conditional probability function (or decision function) to model the label distribution, and estimate that function by mining additional information from the matrices of features and logical labels.

Overall, there are two types of additional information: instance correlation and label correlation. Most of the methods mining instance correlation are based on the assumption that instances that are similar in the feature space are also similar in the label distribution space. To mine the instance correlation, some methods (Hou, Geng, and Zhang 2016; Shao, Geng, and Xu 2018; Zhang, Zhong, and Zhang 2018; Lv et al. 2019; Tang et al. 2020; Liu et al. 2021a; Zheng et al. 2021) reconstruct each instance's feature vector from its neighbors by a reconstruction matrix, and migrate this reconstructing process to label distribution. Some methods (Xu, Tao, and Geng 2018; Xu, Lv, and Geng 2019; Liu et al. 2021b; Zhang et al. 2021) quantify the correlation between each pair of instances through the feature matrix, which can be represented as the edge of a graph, and let the label distributions of each pair of instances also satisfy that correlation constraint as much as possible.

The methods mining label correlation are fundamentally based on the assumption that labels with similar semantics have similar description degree to the instance. For example, LELR (Jia, Lu, and Zhang 2021) quantifies pairwise label correlation by cosine similarity of logical label vectors. The extension of GLLE (Xu, Liu, and Geng 2021) automatically learns the label correlation from data. N-LDL (Luo et al. 2021) uses label propagation to pass the message about label correlation among all pairs of logical labels.

All of the above approaches are based on the discriminative approach, which excels in obtaining better predictive performance under the supervised task, and the predictive performance is dependent on the accurate supervision. However, the above LE methods all use logical labels as inaccurate supervision, which may introduce some additional error. Compared with the discriminative approach, the generative approach specializes in describing the underlying patterns

of observations, which is more essential for LE task. To the best of our knowledge, LEVI (Xu et al. 2020) is currently the only generative LE method. It treats the label distribution as the latent vector which generates features and logical labels, and adopts the variational Bayes to infer label distribution. Although LEVI is a successful application of generative approach to LE task, it still faces two drawbacks: (1) LEVI uses a single Gaussian distribution which can not model a one-to-many mapping from label distributions to instances; (2) LEVI still employs imprecise logical label vectors to quantitatively constrain label distributions. To solve both of these problems, we employ a Gaussian mixture to model one-to-many mapping, and define a new probability distribution motivated by partial label ranking.

Method

Notations

Let $\boldsymbol{x} \in \mathbb{R}^D$ denote the vector of feature variables, where D is the number of features. Let $\boldsymbol{y} \in \{0,1\}^M$ denote the vector of logical label variables, where M is the number of labels. Let $\boldsymbol{d} \in \{[d_1, d_2, \cdots, d_M]^\top \mid \forall m = 1, 2, \cdots, M, (d_m \geq 0 \text{ and } \sum_{m=1}^{M} d_m = 1)\}$ denote the label distribution. The i-th components in $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{d}$ are denoted as x_i, y_i, and d_i, respectively. The observations for $\boldsymbol{x}$ and $\boldsymbol{y}$ constitute a dataset $\mathcal{D} = \{(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)})\}_{n=1}^{N}$. Our goal is to infer the label distribution based on $\mathcal{D}$.

Probability Distribution for Logical Label Vector

Here we define a probability distribution $\mathrm{LD}(\boldsymbol{y}|\boldsymbol{d})$ to characterize the relationship between the **L**ogical label vector $\boldsymbol{y}$ and the label **D**istribution $\boldsymbol{d}$. The main idea of $\mathrm{LD}(\boldsymbol{y}|\boldsymbol{d})$ is that the probability will be higher if the relevant labels exceed the irrelevant labels by more w.r.t. the label description degree. Formally, we have:

$$\mathrm{LD}(\boldsymbol{y}|\boldsymbol{d}) \propto \sum_{i \in \mathcal{I}^+} \sum_{j \in \mathcal{I}^-} d_i - d_j = M\boldsymbol{y}^\top \boldsymbol{d} - \boldsymbol{y}^\top \mathbf{1}_M, \quad (1)$$

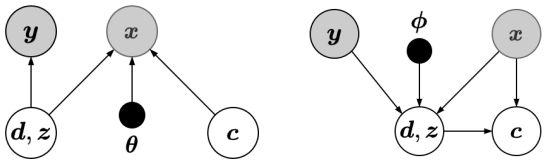
where $\mathcal{I}^+ \triangleq \{i \mid y_i = 1\}$, $\mathcal{I}^- \triangleq \{i \mid y_i = 0\}$, and $\mathbf{1}_M$ denotes an M-dimensional all-ones vector. We denote the right-hand side of Eq. (1) as $\widetilde{\mathrm{LD}}(\boldsymbol{y}|\boldsymbol{d})$. To ensure that the probability is nonnegative, we add a positive constant M (i.e., the number of labels) to Eq. (1). Then we have:

$$\widetilde{\mathrm{LD}}(\boldsymbol{y}|\boldsymbol{d}) = M\boldsymbol{y}^\top \boldsymbol{d} - \boldsymbol{y}^\top \mathbf{1}_M + M. \quad (2)$$

It is worth noting that while there are many distance metrics that can be used as surrogates to measure ranking consistency, such as replacing $d_i - d_j$ with $\exp(d_i - d_j)$, Eq. (2) has an essential property that the normalization constant is independent of $\boldsymbol{d}$, i.e., $\sum_{\boldsymbol{y} \in \{0,1\}^M} \widetilde{\mathrm{LD}}(\boldsymbol{y}|\boldsymbol{d}) = M(2^M - 1)$, which facilitates the gradient calculation. Finally, we can obtain the analytic form of $\mathrm{LD}(\boldsymbol{y}|\boldsymbol{d})$:

$$\mathrm{LD}(\boldsymbol{y}|\boldsymbol{d}) = (2^M - 1)^{-1} \cdot \left(\boldsymbol{y}^\top \boldsymbol{d} - M^{-1}\boldsymbol{y}^\top \mathbf{1}_M + 1\right). \quad (3)$$

Obviously, if we treat $\boldsymbol{d}$ and $\boldsymbol{y}$ as learnable parameters and observations, respectively, then the solution space for maximizing $\ln \mathrm{LD}(\boldsymbol{y}|\boldsymbol{d})$ is the set of the label distributions whose components for irrelevant labels are all zero, i.e., $\{\boldsymbol{d} \mid \forall i \in \mathcal{I}^-, d_i = 0\}$. The components corresponding to the relevant labels can be completely determined by other information.



(a) Generative model (b) Inference model

Figure 2: Graphical representation of the generative process and the inference process.

Generative Model

Here we explain the generation process of the observations. The generative model is visualized in Fig. 2(a).

1. Generate the vector of label distribution $\boldsymbol{d}$:
   (a) Generate the unnormalized label distribution $\boldsymbol{z}$ from a zero-mean isotropic Gaussian:
   $$\boldsymbol{z} \sim \mathcal{N}\left(\boldsymbol{z}|\mathbf{0}_M, \lambda_{\boldsymbol{z}}^{-1}\mathbf{I}_M\right), \quad (4)$$
   where $\mathbf{0}_M$ is an M-dimensional all-zeros vector, $\lambda_{\boldsymbol{z}} > 0$ is the precision of Gaussian, $\mathbf{I}_M$ is the $M \times M$ identity matrix.
   (b) Transform $\boldsymbol{z}$ into the label distribution $\boldsymbol{d}$ by the softmax function, i.e., $\boldsymbol{d} = \tau(\boldsymbol{z})$ whose m-th element is
   $$d_m = \frac{\exp(z_m)}{\sum_{i=1}^{M} \exp(z_i)}. \quad (5)$$
2. Generate the logical label $\boldsymbol{y}$ from the probability distribution $p(\boldsymbol{y}|\boldsymbol{d})$ defined by Eq. (3), i.e., $\boldsymbol{y} \mid \boldsymbol{z} \sim \mathrm{LD}(\boldsymbol{y}|\boldsymbol{d})$.
3. Generate a one-hot vector $\boldsymbol{c}$ from the categorical distribution to indicate which Gaussian is chosen:
   $$\boldsymbol{c} \sim \mathrm{Cat}(\boldsymbol{c}|K^{-1}\mathbf{1}_K), \quad (6)$$
   where K is the number of Gaussian components.
4. Generate the feature vector from a Gaussian distribution:
   $$\boldsymbol{x} \mid \boldsymbol{c}, \boldsymbol{z} \sim \prod_{k=1}^{K} \mathcal{N}\left(\boldsymbol{x}|\mu_{\boldsymbol{x},k}(\boldsymbol{d}), \mathrm{diag}(\sigma^2_{\boldsymbol{x},k}(\boldsymbol{d}))\right)^{c_k}, \quad (7)$$
   where $\mu_{\boldsymbol{x},k}(\boldsymbol{d})$ and $\sigma^2_{\boldsymbol{x},k}(\boldsymbol{d})$ are multilayer perceptrons with learnable parameters of $\boldsymbol{\theta}$; $\mathrm{diag}(\sigma^2_{\boldsymbol{x},k}(\boldsymbol{d}))$ is a diagonal matrix satisfying $\mathbf{1}_D^\top \mathrm{diag}(\sigma^2_{\boldsymbol{x},k}(\boldsymbol{d})) = \sigma^2_{\boldsymbol{x},k}(\boldsymbol{d})$.

Since $\boldsymbol{d}$ is the deterministic transformation of $\boldsymbol{z}$, we can omit $\boldsymbol{d}$ in the probabilistic model representation. Then, the joint density of the complete-data can be factorized as:

$$p(\boldsymbol{z}, \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{c}) = p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c})p(\boldsymbol{y}|\boldsymbol{z})p(\boldsymbol{c})p(\boldsymbol{z}). \quad (8)$$

Inference

Exact inference of the posterior distribution is intractable due to the nonlinear and nonconjugate dependence among random variables. Therefore, we adpot the variational inference, i.e., using a varitional posterior to approximate the

exact posterior of label distribution. Figure 2(b) shows the underlying dependence structure of the variational posterior. According to Fig. 2(b), variational posterior can be factorized as $q(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x},\boldsymbol{y}) = q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})p(\boldsymbol{c}|\boldsymbol{x},\boldsymbol{z})$, where the factor $q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{z}|\mu_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y}), \mathrm{diag}(\sigma^2_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y}))\right)$, $\mu_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y})$ and $\sigma^2_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y})$ are two multilayer perceptrons with learnable parameters of ϕ which map the feature vector and the logical label vector to the mean and the variance of Gaussian, respectively. It is worth noting that this factorization does not follow the mean-field approximation completely. Since $\boldsymbol{c}$ is a discrete variable, we can obtain $p(\boldsymbol{c}|\boldsymbol{x},\boldsymbol{z})$ analytically:

$$\gamma^{(k)}_{\boldsymbol{x},\boldsymbol{z}} \triangleq p(c_k = 1|\boldsymbol{x},\boldsymbol{z}) = \frac{p(\boldsymbol{x}|\boldsymbol{z},c_k=1)p(c_k=1)}{\sum_{i=1}^{K} p(\boldsymbol{x}|\boldsymbol{z},c_k=1)p(c_k=1)}. \quad (9)$$

We use $p(\boldsymbol{c}|\boldsymbol{x},\boldsymbol{z})$ to capture the dependence among $\boldsymbol{c}$, $\boldsymbol{x}$ and $\boldsymbol{z}$; thus the information loss induced by the mean-field assumption can be mitigated.

Next, we infer label distribution by maximizing the ELBO (**E**vidence **L**ower **BO**und). According to Eq. (8), Eq. (9) and the posterior factorization, and using the SGVB estimator, we can write the ELBO as:

$$\begin{aligned}&\mathbb{E}_{q(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x},\boldsymbol{y})}\left[\ln p(\boldsymbol{z},\boldsymbol{c},\boldsymbol{x},\boldsymbol{y}) - \ln q(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x},\boldsymbol{y})\right] = \\ &\underbrace{\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})p(\boldsymbol{c}|\boldsymbol{x},\boldsymbol{z})}\left[\ln p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{c}) + \ln p(\boldsymbol{y}|\boldsymbol{z})\right]}_{\text{reconstruction}} - \\ &\underbrace{\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}\left[D_{\mathrm{KL}}(p(\boldsymbol{c}|\boldsymbol{x},\boldsymbol{z})\|p(\boldsymbol{c}))\right]}_{\boldsymbol{c}\text{-prior}} - \underbrace{D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})\|p(\boldsymbol{z}))}_{\boldsymbol{z}\text{-prior}}.\end{aligned} \quad (10)$$

We refer to the terms in the ELBO as reconstruction term, $\boldsymbol{c}$-prior term, and $\boldsymbol{z}$-prior term, respectively.

Reconstruction term The first term in Eq. (10) is also known as reconstruction term, which encourages the label distribution to explain the observations well. Obviously, the reconstruction term, which involves integral of conditional likelihood, is intractable. Therefore, we draw Monte Carlo samples from $q(\boldsymbol{z},\boldsymbol{c}|\boldsymbol{x},\boldsymbol{y})$ to estimate it, where the differentiability can be ensured by the standard reparameterisation trick. Besides, it is worth noting that the likelihood functions for the feature variables and logistic labels are different, which leads to a significant difference in their values. In this case, the contribution of each likelihood to the ELBO differs significantly, resulting in challenging optimization problems (Kendall, Gal, and Cipolla 2018), where the observations for some variables (e.g., logical label vectors) may be poorly reconstructed. Therefore, we use a trade-off parameter $\lambda_{\boldsymbol{y}}$ to re-weight the reconstruction errors for features and logical labels. Formally, the reconstruction term can be rewritten as:

$$\begin{aligned}&\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})p(\boldsymbol{c}|\boldsymbol{x},\boldsymbol{z})}\left[\frac{\ln p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{c})}{1+\lambda_{\boldsymbol{y}}} + \frac{\lambda_{\boldsymbol{y}}\ln p(\boldsymbol{y}|\boldsymbol{z})}{1+\lambda_{\boldsymbol{y}}}\right] \approx \frac{1}{S}\cdot \\ &\sum_{s=1}^{S}\left(\sum_{k=1}^{K}\gamma^{(k)}_{\boldsymbol{x},\boldsymbol{z}^{(s)}}\ln\mathcal{N}\left(\boldsymbol{x}|\mu_{\boldsymbol{x},k}(\boldsymbol{d}^{(s)}), \mathrm{diag}(\sigma^2_{\boldsymbol{x},k}(\boldsymbol{d}^{(s)}))\right)\right. \\ &\left.+\lambda_{\boldsymbol{y}}\ln\left(\boldsymbol{y}^\top\boldsymbol{d}^{(s)} - M^{-1}\boldsymbol{y}^\top\mathbf{1}_M + 1\right)\right)\frac{1}{1+\lambda_{\boldsymbol{y}}} + \mathrm{const},\end{aligned} \quad (11)$$

where S is the number of Monte Carlo samples, $\boldsymbol{d}^{(s)} = \tau(\boldsymbol{z}^{(s)})$, $\boldsymbol{z}^{(s)} = \mu_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y}) + \sigma^2_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y}) \odot \boldsymbol{\epsilon}^{(s)}$ ($\odot$ denotes the element-wise product), and $\boldsymbol{\epsilon}^{(s)} \sim \mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}_M, \mathbf{I}_M)$.

$\boldsymbol{c}$-prior and $\boldsymbol{z}$-prior terms The $\boldsymbol{c}$-prior and $\boldsymbol{z}$-prior terms can encourage the posteriors of label distribution and mixture coefficients to approach their priors. The $\boldsymbol{c}$-prior term can be estimated by Monte Carlo samples $\{\boldsymbol{d}^{(s)}\}_{s=1}^{S}$:

$$\begin{aligned}&\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}\left[D_{\mathrm{KL}}(p(\boldsymbol{c}|\boldsymbol{x},\boldsymbol{z})\|p(\boldsymbol{c}))\right] \\ &\approx \frac{1}{S}\sum_{s=1}^{S}\sum_{k=1}^{K}\gamma^{(k)}_{\boldsymbol{x},\boldsymbol{z}^{(s)}}\ln\gamma^{(k)}_{\boldsymbol{x},\boldsymbol{z}^{(s)}} + \mathrm{const}.\end{aligned} \quad (12)$$

The $\boldsymbol{z}$-prior term can be calculated analytically:

$$\begin{aligned}&D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})\|p(\boldsymbol{z})) = \frac{1}{2}\Big(\lambda_{\boldsymbol{z}}\|\mu_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y})\|_2^2 \\ &+\lambda_{\boldsymbol{z}}\mathbf{1}_M^\top\sigma^2_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y}) - \mathbf{1}_M^\top\ln\sigma^2_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y})\Big) + \mathrm{const}.\end{aligned} \quad (13)$$

Optimization objective The maximization objective can be obtained by combining Eq. (11), Eq. (12), and Eq. (13).

$$\begin{aligned}&\mathrm{ELBO} = \sum_{n=1}^{N}\frac{1}{1+\lambda_{\boldsymbol{y}}}\left(\lambda_{\boldsymbol{y}}\ln\left(1 + (\boldsymbol{d}^{(n)} - \frac{\mathbf{1}_M}{M})^\top\boldsymbol{y}^{(n)}\right) + \right. \\ &\left.\sum_{k=1}^{K}\gamma^{(k)}_{\boldsymbol{x}^{(n)},\boldsymbol{z}^{(n)}}\ln\mathcal{N}\left(\boldsymbol{x}^{(n)}|\mu_{\boldsymbol{x},k}(\boldsymbol{d}^{(n)}), \mathrm{diag}(\sigma^2_{\boldsymbol{x},k}(\boldsymbol{d}^{(n)}))\right)\right) \\ &- \sum_{k=1}^{K}\gamma^{(k)}_{\boldsymbol{x}^{(n)},\boldsymbol{z}^{(n)}}\ln\gamma^{(k)}_{\boldsymbol{x}^{(n)},\boldsymbol{z}^{(n)}} - \frac{1}{2}\lambda_{\boldsymbol{z}}\left\|\mu_{\boldsymbol{d}}(\boldsymbol{x}^{(n)},\boldsymbol{y}^{(n)})\right\|_2^2 \\ &- \frac{1}{2}\mathbf{1}_M^\top\left(\lambda_{\boldsymbol{z}}\sigma^2_{\boldsymbol{d}}(\boldsymbol{x}^{(n)},\boldsymbol{y}^{(n)}) - \ln\sigma^2_{\boldsymbol{d}}(\boldsymbol{x}^{(n)},\boldsymbol{y}^{(n)})\right) + \mathrm{const},\end{aligned} \quad (14)$$

where $\boldsymbol{d}^{(n)}$ is the softmax transformation of $\boldsymbol{z}^{(n)}$, $\boldsymbol{z}^{(n)} = \mu_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y}) + \sigma^2_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y}) \odot \boldsymbol{\epsilon}^{(n)}$, where $\boldsymbol{\epsilon}^{(n)} \sim \mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}_M, \mathbf{I}_M)$. The parameters of $\mu_{\boldsymbol{x}}(\cdot), \mu_{\boldsymbol{d}}(\cdot), \sigma^2_{\boldsymbol{x}}(\cdot)$ and $\sigma^2_{\boldsymbol{d}}(\cdot)$ need to be learned. As suggested by Kingma and Welling (2014), we generate only one Monte Carlo sample for each observation, i.e., $S = 1$.

Recovering label distributions Once the parameters are learned, we can recover label distributions according to $q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})$. To evaluate our model, we take the expectation of $\boldsymbol{d}$ based on $q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})$ as the deterministic output. Since $\boldsymbol{d}$ is the softmax normalization of Gaussian random vector $\boldsymbol{z}$, it is difficult to obtain the analytic form of $\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}[\boldsymbol{d}]$. Hence, we use the following formula as an approximation:

$$\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}[d_m] \approx \left(\sum_{i=1}^{M}\exp\left(\frac{\mu_i - \mu_m}{\sqrt{1 + 3\pi^{-2}(\sigma_m^2 + \sigma_i^2)}}\right)\right)^{-1}, \quad (15)$$

where μ_i and σ_i are the i-th elements of $\mu_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y})$ and $\sigma_{\boldsymbol{d}}(\boldsymbol{x},\boldsymbol{y})$, respectively. More technical details can be found in the literature (Daunizeau 2017).

Experiments

Experimental Configuration

Datasets The datasets we used is shown in Table 1. These datasets come from the tasks including emotion mining (No. 1-5), natural scene predicting (No. 6), movie rating predicting (No. 7), and bioinformatics (No. 8-14)[1]. In terms of the

[1]Although there are 10 Yeast datasets released by Geng (2016), we only select those with more than 4 labels due to page limitation.

No	Dataset	# Instances	# Features	# Labels
1	SJAFFE (sj)	213	243	6
2	SBU-3DFE (3dfe)	2500	243	6
3	Emotion6 (emo6)	1980	168	7
4	Twitter-LDL (twit)	10045	168	8
5	Flickr-LDL (flic)	11150	168	8
6	Natural-Scene (ns)	2000	294	9
7	Movie (mov)	7755	1869	5
8	Human Gene (gene)	30542	36	68
9	Yeast-alpha (alpha)	2465	24	18
10	Yeast-heat (heat)	2465	24	6
11	Yeast-spo (spo)	2465	24	6
12	Yeast-diau (diau)	2465	24	7
13	Yeast-cdc (cdc)	2465	24	15
14	Yeast-elu (elu)	2465	24	14

Table 1: Datasets statistics

"twit", "flic" (Yang, Sun, and Sun 2017) and "emo6" (Peng et al. 2015) datasets, we extract 168-dimensional feature vectors for each image by the preprocessing method adopted by Ren et al. (2019). Besides, to accelerate the convergence, we use min-max normalization to preprocess the feature matrices of all datasets.

Evaluation measures We consider six measures suggested by Geng (2016): Chebyshev distance (Cheb), Clark distance (Clark), Canberra metric (Canber), Kullback-Leibler divergence (KL), cosine coefficient (Cosine), and intersection similarity (Intersec). The first four measures are based on distance (whose lower values indicate better performances), and the last two measures are based on similarity (whose higher values indicate better performances). Due to page limitations, we only show the results on Cheb, KL, Cosine and Intersec; results on other measures are similar.

Comparison methods We compare our method with six recently proposed LE methods, i.e., LELR (Jia, Lu, and Zhang 2021), gLESC (Zheng et al. 2021), LEVI (Xu et al. 2020), PLEA (Tan et al. 2021), BDLE (Liu et al. 2021b), and FLE (Wang et al. 2021). The hyperparameter configuration for each comparison method follows their respective literature. Specifically, for LELR, λ is selected among $\{10^{-2}, 10^{-1}, \cdots, 10^{2}\}$. For gLESC, λ_1 and λ_2 are selected among $\{10^{-4}, 10^{-3}, \cdots, 10^{1}\}$. For LEVI, the MLPs are constructed with two hidden layers, each with 500 hidden units and softplus activation function, i.e., $\ln(1+\exp(\cdot))$. For BDLE, α and λ are both set to 10^{-2}. For FLE, α, β, λ, γ are selected among $\{10^{-4}, 5\times10^{-4}, 10^{-3}, \cdots, 5\times10^{2}, 10^{3}\}$. For our method, we set $K = 3$, $\lambda_{\boldsymbol{y}} = 10^{5}$, and $\lambda_{\boldsymbol{z}}$ is selected in $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$; both $\mu_{\boldsymbol{x}}(\cdot)$ and $\mu_{\boldsymbol{d}}(\cdot)$ are modeled as linear functions; both $\sigma^2_{\boldsymbol{x}}(\cdot)$ and $\sigma^2_{\boldsymbol{d}}(\cdot)$ are modeled as linear functions with the softplus activation function.

Recovery Experiment

Methodology The procedure for recovery experiment is as follows. First, we transform the label distribution in the dataset into logical labels by the binarization process suggested by Xu, Tao, and Geng (2018). Then, we run different LE methods on the examples with logical labels, and use them to recover the label distributions. Finally, we compare the recovered label distributions with the ground-truth label distributions, and record the performances on four measures.

Peformance The recovery prformance is shown in Table 2. It can be seen that our method outperforms others on most datasets, and has the best average ranks. In particular, on the "emo6", "twit", "flic", and "mov" datasets, our method is well ahead of the second place. Besides, although our method does not achieve the best performance on some datasets (e.g., "sj", "3dfe" and "ns"), the difference between our method and the first place is negligible.

Predictive Experiment

Methodology In order to test the effectiveness of GLEMR on LDL task, we use the label distribution recovered by each LE method to train the LDL model, and test the predictive performance of the LDL model. First, we randomly partition dataset (90% for training and 10% for testing). Then, we use LE methods to recover the label distributions for training instances and train an LDL model (SABFGS (Geng 2016) is used in this paper) according to these recovered distributions. Next, we record the predictive performance of the SABFGS on test instances. Finally, we repeat the above process ten times and report the mean performance.

Performance The results of the predictive experiment are shown in Fig. 3, where the "Ground-Truth" denotes the predictive performance of SABFGS directly trained on the ground-truth label distributions. Besides, to perform comparative analysis in more well-founded ways, we conduct a pairwise two-tailed t-test with 0.05 significance level, whose results are summarized in Table 3. Overall, SABFGS trained with the label distribution recovered by GLEMR is superior to other LE methods on most of the datasets.

Parameter Sensitivity

Here, we demonstrate how hyperparameters $\lambda_{\boldsymbol{y}}$ and $\lambda_{\boldsymbol{z}}$ affect the recovery performance in Fig. 4.

Precision of Gaussian prior $\lambda_{\boldsymbol{z}}$ The value of $\lambda_{\boldsymbol{z}}$ varies in $\{10^{-4}, 10^{-3}, \cdots, 10^{1}\}$. To understand the effect of $\lambda_{\boldsymbol{z}}$ on the recovery performance, we can observe the trend of the dot color in each row of the subplots in Fig. 4. It can be seen that with the increase of $\lambda_{\boldsymbol{z}}$, the recovery performance first becomes better and then stays stable (or gets slightly worse). This is in line with our expectations. As the precision (or strength) of the prior of $\boldsymbol{z}$, a moderate $\lambda_{\boldsymbol{z}}$ can effectively penalize over-complex models and thus improve the recovery performance, while a too large $\lambda_{\boldsymbol{z}}$ may cause $\boldsymbol{z}$ to be over-regularized and thus reduce the recovery performance.

Trade-off parameter $\lambda_{\boldsymbol{y}}$ in reconstruction term The value of $\lambda_{\boldsymbol{y}}$ varies in $\{10^{1}, 10^{2}, \cdots, 10^{6}\}$. To understand the effect of $\lambda_{\boldsymbol{y}}$, we can observe the trend of the dot color in each column of the subfigures in Fig. 4. It can be seen that with the increase of $\lambda_{\boldsymbol{y}}$, the recovery performance first becomes better and then stays stable. The reason is that a sufficiently large $\lambda_{\boldsymbol{y}}$ allows consistency between the logical

	sj	3dfe	emo6	twit	flic	ns	mov	gene	alpha	heat	spo	diau	cdc	elu	AR
							Cheb (↓)								
Ours	.073 (3)	.103 (3)	**.188 (1)**	**.274 (1)**	**.233 (1)**	.270 (2)	**.095 (1)**	**.051 (1)**	**.010 (1)**	**.030 (1)**	**.041 (1)**	**.028 (1)**	**.012 (1)**	**.011 (1)**	1.4
LELR	.093 (6)	.134 (6)	.317 (3)	.470 (5)	.399 (4)	.335 (4)	.121 (3)	.053 (3)	.018 (3)	.046 (3)	.059 (4)	.050 (5)	.020 (5)	.021 (5)	4.2
gLESC	**.070 (1)**	.123 (4)	.332 (6)	.505 (7)	.419 (6)	.344 (6)	.133 (4)	.053 (4)	.018 (4)	.047 (4)	.061 (5)	.040 (3)	.017 (4)	.019 (4)	4.4
LEVI	.073 (2)	**.092 (1)**	.271 (2)	.468 (4)	.374 (3)	.324 (3)	.109 (2)	.052 (2)	.018 (5)	.053 (6)	.054 (3)	.044 (4)	.016 (3)	.018 (3)	3.1
PLEA	.122 (7)	.145 (7)	.320 (4)	.308 (2)	.331 (2)	.363 (7)	.162 (6)	.054 (6)	.033 (7)	.052 (5)	.065 (6)	.083 (7)	.026 (6)	.027 (6)	5.6
BDLE	.078 (4)	.130 (5)	.324 (5)	.478 (6)	.405 (5)	.339 (5)	.167 (7)	.059 (7)	.018 (6)	.073 (7)	.081 (7)	.050 (6)	.034 (7)	.037 (7)	6.0
FLE	.083 (5)	.092 (2)	.360 (7)	.325 (3)	.447 (7)	**.268 (1)**	.153 (5)	.053 (5)	.014 (2)	.032 (2)	.048 (2)	.032 (2)	.015 (2)	.017 (2)	3.4
							KL (↓)								
Ours	**.027 (1)**	.044 (3)	**.322 (1)**	**.558 (1)**	**.474 (1)**	.668 (2)	**.064 (1)**	**.200 (1)**	**.003 (1)**	**.007 (1)**	**.012 (1)**	**.008 (1)**	**.004 (1)**	**.003 (1)**	1.2
LELR	.042 (6)	.085 (5)	.595 (3)	.998 (5)	.848 (5)	.963 (4)	.098 (3)	.236 (4)	.011 (3)	.015 (3)	.026 (4)	.025 (6)	.011 (4)	.011 (4)	4.2
gLESC	.028 (2)	.068 (4)	.638 (6)	1.13 (7)	.924 (7)	1.01 (7)	.111 (4)	.223 (3)	.011 (4)	.016 (4)	.027 (5)	.015 (3)	.008 (3)	.008 (3)	4.4
LEVI	.032 (3)	.041 (2)	.474 (2)	.987 (4)	.783 (4)	.924 (3)	.081 (2)	.207 (2)	.011 (5)	.028 (6)	.025 (3)	.024 (4)	.013 (5)	.014 (5)	3.6
PLEA	.083 (7)	.115 (7)	.624 (5)	.719 (3)	.753 (3)	1.01 (6)	.204 (7)	.244 (6)	.031 (7)	.017 (5)	.032 (6)	.071 (7)	.019 (6)	.019 (6)	5.8
BDLE	.033 (4)	.085 (6)	.618 (4)	1.05 (6)	.903 (6)	.971 (5)	.171 (6)	.316 (7)	.012 (6)	.037 (7)	.047 (7)	.024 (5)	.031 (7)	.033 (7)	5.9
FLE	.033 (5)	**.039 (1)**	.995 (7)	.662 (2)	.612 (2)	**.648 (1)**	.143 (5)	.236 (5)	.006 (2)	.007 (2)	.017 (2)	.009 (2)	.005 (2)	.006 (2)	2.9
							Cosine (↑)								
Ours	**.974 (1)**	.955 (3)	**.909 (1)**	**.904 (1)**	**.900 (1)**	.828 (2)	**.965 (1)**	**.861 (1)**	**.997 (1)**	**.994 (1)**	**.989 (1)**	**.993 (1)**	**.996 (1)**	**.997 (1)**	1.2
LELR	.959 (6)	.919 (5)	.725 (3)	.676 (5)	.681 (5)	.686 (4)	.938 (3)	.835 (4)	.990 (3)	.986 (3)	.976 (4)	.977 (5)	.990 (4)	.990 (4)	4.1
gLESC	.973 (2)	.933 (4)	.695 (6)	.598 (7)	.632 (7)	.656 (6)	.929 (4)	.845 (3)	.989 (5)	.985 (4)	.975 (5)	.986 (3)	.992 (3)	.992 (3)	4.4
LEVI	.969 (3)	.958 (2)	.814 (2)	.683 (4)	.725 (4)	.713 (3)	.954 (2)	.857 (2)	.990 (4)	.976 (6)	.978 (3)	.981 (4)	.988 (5)	.987 (5)	3.5
PLEA	.923 (7)	.895 (7)	.696 (5)	.802 (3)	.733 (3)	.632 (7)	.886 (6)	.829 (6)	.977 (7)	.983 (5)	.970 (6)	.942 (7)	.982 (6)	.982 (6)	5.8
BDLE	.968 (4)	.919 (6)	.708 (4)	.642 (6)	.644 (6)	.676 (5)	.883 (7)	.778 (7)	.989 (6)	.966 (7)	.957 (7)	.977 (6)	.970 (7)	.969 (7)	6.1
FLE	.967 (5)	**.959 (1)**	.601 (7)	.875 (2)	.821 (2)	**.849 (1)**	.905 (5)	.835 (5)	.995 (2)	.993 (2)	.984 (2)	.992 (2)	.995 (2)	.994 (2)	2.9
							Intersec (↑)								
Ours	**.909 (1)**	.887 (2)	**.713 (1)**	**.612 (1)**	**.653 (1)**	.555 (2)	**.871 (1)**	**.814 (1)**	**.968 (1)**	.954 (2)	**.942 (1)**	.952 (2)	**.969 (1)**	**.970 (1)**	1.3
LELR	.887 (6)	.844 (5)	.569 (3)	.418 (4)	.478 (5)	.429 (5)	.834 (3)	.786 (4)	.945 (3)	.934 (3)	.913 (3)	.909 (4)	.944 (4)	.943 (4)	4.0
gLESC	.905 (2)	.856 (4)	.550 (7)	.381 (7)	.450 (7)	.409 (7)	.818 (4)	.797 (3)	.944 (4)	.932 (4)	.912 (4)	.936 (3)	.954 (3)	.951 (3)	4.4
LEVI	.896 (5)	.884 (3)	.621 (2)	.416 (5)	.494 (4)	.442 (3)	.850 (2)	.810 (2)	.933 (6)	.895 (6)	.904 (5)	.909 (5)	.927 (5)	.925 (5)	4.1
PLEA	.845 (7)	.827 (7)	.564 (4)	.612 (2)	.561 (2)	.438 (4)	.784 (6)	.778 (6)	.922 (7)	.930 (5)	.900 (6)	.844 (7)	.921 (6)	.924 (6)	5.4
BDLE	.898 (4)	.840 (6)	.559 (5)	.404 (6)	.464 (6)	.426 (6)	.774 (7)	.734 (7)	.941 (5)	.895 (7)	.882 (7)	.909 (6)	.904 (7)	.902 (7)	6.1
FLE	.900 (3)	**.892 (1)**	.558 (6)	.546 (3)	.553 (3)	**.573 (1)**	.786 (5)	.786 (5)	.962 (2)	**.957 (1)**	.937 (2)	**.953 (1)**	.967 (2)	.958 (2)	2.6

Table 2: Recovery performance (value (rank)) on 14 datasets. "↓" indicates "the smaller the better", "↑" indicates "the larger the better". The last column "AR" denotes the average rank of each method. The best performance is marked in bold.

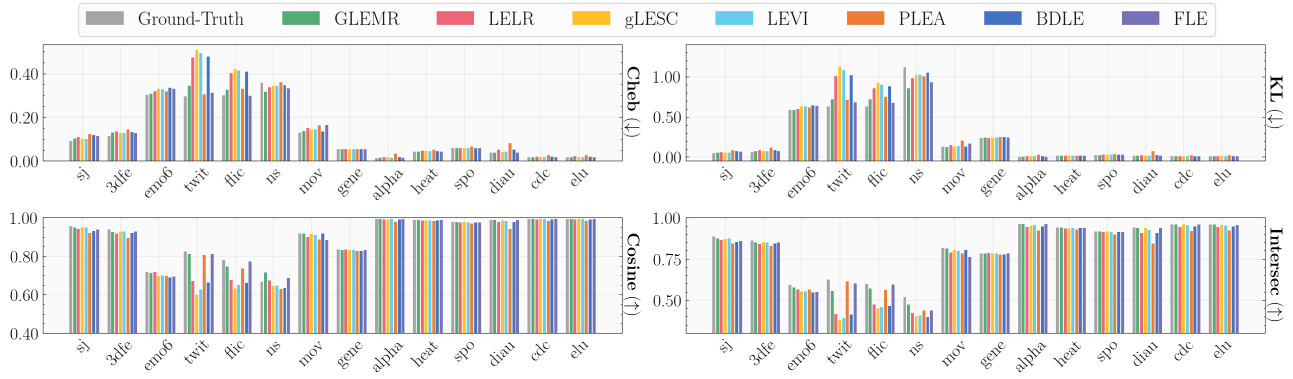


Figure 3: Predictive performance on 14 datasets. The horizontal and vertical axes denote datasets and performance, respectively.

label vector and the label distribution to be fully satisfied w.r.t. partial label ranking, so that the recovery performance can be improved. After that, further increases in λ_y barely affect results as ranking consistency has been satisfied.

Ablation Study

Here, we show the effectiveness of each module proposed in our method. To verify the Gaussian mixture, we replace the Gaussian mixture in GLEMR with a Gaussian distribution,

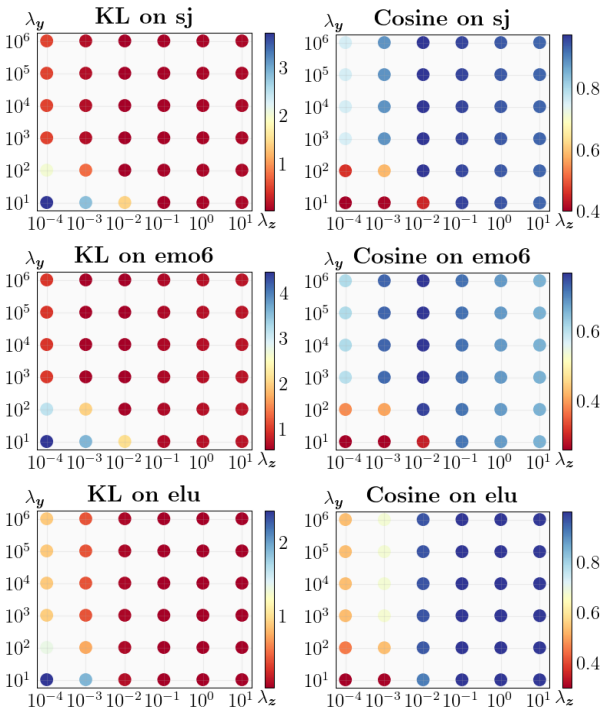


Figure 4: Recovery performance with varying λ_y and λ_z. The horizontal and vertical axes are λ_z and λ_y, respectively.

	LELR	gLESC	LEVI	PLEA	BDLE	FLE
Cheb	13/0/1	12/1/1	11/1/2	12/0/2	12/1/1	10/1/3
KL	13/0/1	12/1/1	12/1/1	13/0/1	14/0/0	11/0/3
Cosine	13/0/1	12/1/1	12/1/1	13/1/0	13/1/0	11/0/3
Intersec	13/0/1	12/2/0	12/1/1	13/0/1	14/0/0	10/1/3

Table 3: Counts of win/tie/loss for predictive experiments under pairwise two-tailed t-test with 0.05 significance level.

i.e., Eq. (7) becomes $\boldsymbol{x}|\boldsymbol{z} \sim \mathcal{N}\left(\boldsymbol{x}|\mu_{\boldsymbol{x}}(\boldsymbol{d}), \mathrm{diag}(\sigma^2_{\boldsymbol{x}}(\boldsymbol{d}))\right)$. To verify our proposed probability distribution for generating logical labels, we replace the $\mathrm{LD}(\boldsymbol{y}|\boldsymbol{d})$ with a Bernoulli distribution and a Gaussian distribution, respectively, i.e., $\mathrm{LD}(\boldsymbol{y}|\boldsymbol{d})$ becomes $\mathrm{Ber}(\boldsymbol{y}|\boldsymbol{d})$ or $\mathcal{N}(\boldsymbol{y}|\boldsymbol{d}, \lambda')$, respectively. To obtain the best performance of the modified models, the hyperparameter $\lambda_{\boldsymbol{z}}$ of the modified models is re-tuned, and the hyperparameter λ' is selected among $\{10^{-2}, \cdots, 10^{2}\}$. The experimental results are shown in Fig. 5. It can be seen that our proposed $\mathrm{LD}(\boldsymbol{y}|\boldsymbol{d})$ always outperforms the other two probability distributions for modeling $p(\boldsymbol{y}|\boldsymbol{d})$. Besides, it can be seen that the merits offered by using the Gaussian mixture seem to be less pronounced on some datasets. We believe the reason is that those datasets contain a simple relationship between the label distribution and the feature vector, which can be adequately encoded by a single Gaussian.

Conclusion

In this paper, we propose a novel generative label enhancement model. This model makes innovations in two aspects:

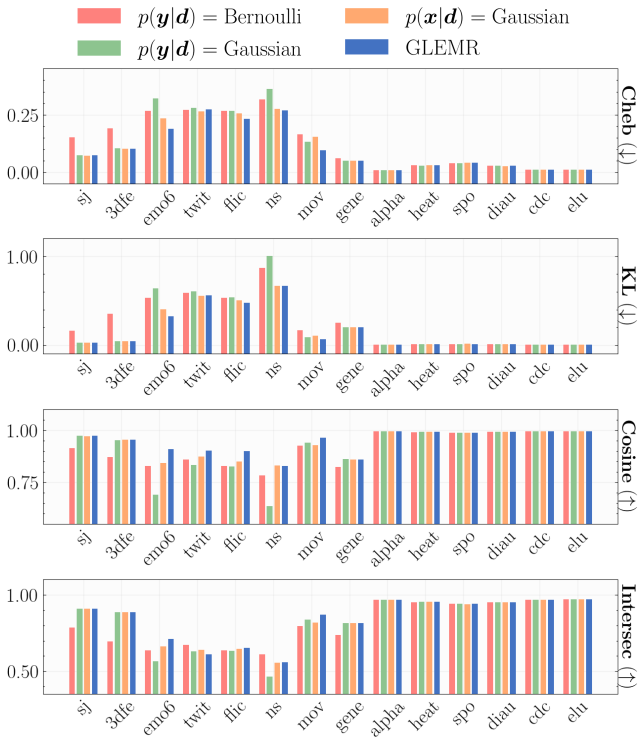


Figure 5: Ablation study. The pink and green bars indicate the versions that replace the $\mathrm{LD}(\boldsymbol{y}|\boldsymbol{d})$ in GLEMR with Bernoulli and Gaussian distributions, respectively. The orange bars indicate the version that replaces the Gaussian mixture in GLEMR with a single Gaussian distribution, i.e., $K = 1$. The blue bars is GLEMR without any modification.

(1) We employ Gaussian mixtures to model the one-to-many relationship from label distributions to instance features; (2) We design a novel probability distribution to encode the process of generating logical labels from label distributions, which penalizes the ranking inconsistency between label distributions and logical labels, and thus can provide reliable supervised information for model inference. In addition, we design an inference model and show how to efficiently infer the posterior of the label distribution in the SGVB framework. Finally, we compare our proposal with some state-of-the-art LE methods on 14 real-world datasets, and the experimental results show the superiority of our proposal.

Acknowledgements

This work was partially supported by the National Key Research and Development Program of China under Grant 2019YFB1706900, the National Natural Science Foundation of China (62176123, 61906090), the Fundamental Research Funds for the Central Universities (30920021131), the Found of Nanjing University of Aeronautics and Astronautics Research Base Innovation (Science and Technology) Project under Grant NJ2020022, and the Fund of Prospective Layout of Scientific Research for Nanjing University of Aeronautics and Astronautics.

References

Daunizeau, J. 2017. Semi-Analytical Approximations to Statistical Moments of Sigmoid and Softmax Mappings of Normal Variables. *arXiv*, 1703.00091.

Gao, B.-B.; Zhou, H.-Y.; Wu, J.; and Geng, X. 2018. Age Estimation Using Expectation of Label Distribution Learning. In *International Joint Conference on Artificial Intelligence*, 712–718.

Geng, X. 2016. Label Distribution Learning. *IEEE Transactions on Knowledge and Data Engineering*, 28(7): 1734–1748.

He, T.; and Jin, X. 2019. Image Emotion Distribution Learning with Graph Convolutional Networks. In *International Conference on Multimedia Retrieval*, 382–390.

Hou, P.; Geng, X.; Huo, Z.-W.; and Lv, J. 2017. Semi-Supervised Adaptive Label Distribution Learning for Facial Age Estimation. In *AAAI Conference on Artificial Intelligence*, 2015–2021.

Hou, P.; Geng, X.; and Zhang, M.-L. 2016. Multi-Label Manifold Learning. In *AAAI Conference on Artificial Intelligence*, 1680–1686.

Jebara, T. 2012. *Machine Learning: Discriminative and Generative*, volume 755. Springer Science & Business Media.

Jia, X.; Lu, Y.; and Zhang, F. 2021. Label Enhancement by Maintaining Positive and Negative Label Relation. *IEEE Transactions on Knowledge and Data Engineering*, PP: 1–1.

Jia, X.; Shen, X.; Li, W.; Lu, Y.; and Zhu, J. 2021. Label Distribution Learning by Maintaining Label Ranking Relation. *IEEE Transactions on Knowledge and Data Engineering*, PP: 1–1.

Jia, X.; Zheng, X.; Li, W.; Zhang, C.; and Li, Z. 2019. Facial Emotion Distribution Learning by Exploiting Low-Rank Label Correlations Locally. In *IEEE Conference on Computer Vision and Pattern Recognition*, 9841–9850.

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *IEEE/CVF International Conference on Computer Vision*, 7482–7491.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*.

Li, Z.; Xie, H.; Cheng, G.; and Li, Q. 2021. Word-level Emotion Distribution with Two Schemas for Short Text Emotion Classification. *Knowledge-Based System*, 227: 107163.

Liu, X.; Zhu, J.; Li, Z.; Tian, Z.; Jia, X.; and Chen, L. 2021a. Unified Framework for Learning with Label Distribution. *Information Fusion*, 75: 116–130.

Liu, X.; Zhu, J.; Zheng, Q.; Li, Z.; Liu, R.; and Wang, J. 2021b. Bidirectional Loss Function for Label Enhancement and Distribution Learning. *Knowledge-Based System*, 213: 106690.

Liu, Z.; Chen, Z.; Bai, J.; Li, S.; and Lian, S. 2019. Facial Pose Estimation by Deep Learning from Label Distributions. In *IEEE/CVF International Conference on Computer Vision Workshop*, 1232–1240.

Luo, J.; Wang, Y.; Ou, Y.; He, B.; and Li, B. 2021. Neighbor-Based Label Distribution Learning to Model Label Ambiguity for Aerial Scene Classification. *Remote Sensing*, 13(4): 755.

Lv, J.; Xu, N.; Zheng, R.; and Geng, X. 2019. Weakly Supervised Multi-Label Learning via Label Enhancement. In *International Joint Conference on Artificial Intelligence*, 3101–3107.

Peng, K.-C.; Chen, T.; Sadovnik, A.; and Gallagher, A. C. 2015. A Mixed Bag of Emotions: Model, Predict, and Transfer Emotion Distributions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 860–868.

Ren, T.; Jia, X.; Li, W.; Chen, L.; and Li, Z. 2019. Label Distribution Learning with Label-Specific Features. In *International Joint Conference on Artificial Intelligence*, 3318–3324.

Shao, R.; Geng, X.; and Xu, N. 2018. Multi-Label Learning with Label Enhancement. In *IEEE International Conference on Data Mining*, 437–446.

Tan, C.; Chen, S.; Ji, G.; and Geng, X. 2021. A Novel Probabilistic Label Enhancement Algorithm for Multi-Label Distribution Learning. *IEEE Transactions on Knowledge and Data Engineering*, PP: 1–1.

Tang, H.; Zhu, J.; Zheng, Q.; Wang, J.; Pang, S.; and Li, Z. 2020. Label Enhancement with Sample Correlations via Low-Rank Representation. In *AAAI Conference on Artificial Intelligence*, 5932–5939.

Wang, K.; Xu, N.; Ling, M.; and Geng, X. 2021. Fast Label Enhancement for Label Distribution Learning. *IEEE Transactions on Knowledge and Data Engineering*, PP: 1–1.

Wen, X.; Li, B.; Guo, H.; Liu, Z.; Hu, G.; Tang, M.; and Wang, J. 2020. Adaptive Variance Based Label Distribution Learning for Facial Age Estimation. In *European Conference on Computer Vision*, 379–395.

Xu, N.; Liu, Y.-P.; and Geng, X. 2021. Label Enhancement for Label Distribution Learning. *IEEE Transactions on Knowledge and Data Engineering*, 33: 1632–1643.

Xu, N.; Lv, J.; and Geng, X. 2019. Partial Label Learning via Label Enhancement. In *AAAI Conference on Artificial Intelligence*, 5557–5564.

Xu, N.; Shu, J.; Liu, Y.-P.; and Geng, X. 2020. Variational Label Enhancement. In *International Conference on Machine Learning*, 10597–10606.

Xu, N.; Tao, A.; and Geng, X. 2018. Label Enhancement for Label Distribution Learning. In *International Joint Conference on Artificial Intelligence*, 1632–1643.

Yang, J.; Sun, M.; and Sun, X. 2017. Learning Visual Sentiment Distributions via Augmented Conditional Probability Neural Network. In *AAAI Conference on Artificial Intelligence*, 224–230.

Zhang, M.-L.; Zhang, Q.-W.; Fang, J.-P.; Li, Y.; and Geng, X. 2021. Leveraging Implicit Relative Labeling-Importance Information for Effective Multi-Label Learning. *IEEE Transactions on Knowledge and Data Engineering*, 33: 2057–2070.

Zhang, Q.-W.; Zhong, Y.; and Zhang, M.-L. 2018. Feature-Induced Labeling Information Enrichment for Multi-Label Learning. In *AAAI Conference on Artificial Intelligence*, 4446–4453.

Zhang, Y.; Fu, K.; Wang, J.; and Cheng, P. 2020. Learning from Discrete Gaussian Label Distribution and Spatial Channel-aware Residual Attention for Head Pose Estimation. *Neurocomputing*, 407: 259–269.

Zheng, Q.; Zhu, J.; Tang, H.; Liu, X.; Li, Z.; and Lu, H. 2021. Generalized Label Enhancement with Sample Correlations. *IEEE Transactions on Knowledge and Data Engineering*, PP: 1–1.