Generative Calibration of Inaccurate Annotation for Label Distribution Learning

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

[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
[2]Guangxi Key Lab of Multi-source Information Mining & Security, Guangxi Normal University, Guilin, China
[3]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
{heliang, luyn, jiaxy}@njust.edu.cn, liweiwei@nuaa.edu.cn

Abstract

Label distribution learning (LDL) is an effective learning paradigm for handling label ambiguity. When applying LDL, it typically requires datasets annotated with label distributions. However, obtaining supervised data for LDL is a challenging task. Due to the randomness of label annotation, the annotator can produce inaccurate annotation results for the instance, affecting the accuracy and generalization ability of the LDL model. To address this problem, we propose a generative approach to calibrate the inaccurate annotation for LDL using variational inference techniques. Specifically, we assume that instances with similar features share latent similar label distributions. The feature vectors and label distributions are generated by Gaussian mixture and Dirichlet mixture, respectively. The relationship between them is established through a shared categorical variable, which effectively utilizes the label distribution of instances with similar features, and achieves a more accurate label distribution through the generative approach. Furthermore, we use a confusion matrix to model the factors that contribute to the inaccuracy during the annotation process, which captures the relationship between label distributions and inaccurate label distributions. Finally, the label distribution is used to calibrate the available information in the noisy dataset to obtain the ground-truth label distribution.

Introduction

Label polysemy, the situation where an instance can be described by multiple labels, is ubiquitous in real-world applications. To quantify label polysemy, two typical forms of labeling have been widely studied. The first is the logical label (Zhang and Zhou 2013), which directly assigns an instance a vector of logical values (0/1) to indicate whether each label is relevant to the instance. The second is the label distribution (Geng 2016), which assigns to the instance a real-valued vector with probability distribution form (called label distribution) to indicate the relative importance among labels. Since label distribution provides more semantic information about label polysemy, it has been widely applied in many practical applications, such as age estimation (Shen et al. 2019; Gao et al. 2018; Hou et al. 2017), head pose estimation (Liu et al. 2019; Geng and Xia 2014; Xu and Wang 2020), sentiment analysis (Jia et al. 2019; Li et al. 2021) and so on.

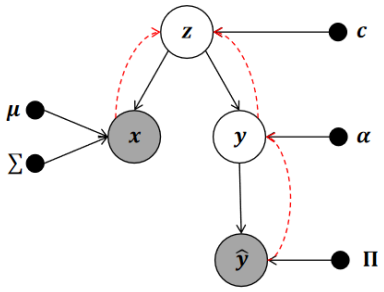


Figure 1: The GCIA Model. The feature vectors $\boldsymbol{x}$ and true label distributions $\boldsymbol{y}$ are generated by Gaussian mixture and Dirichlet mixture, respectively. The label given by the annotator is obtained from the ground-truth label distribution after concept confusion.

Label distribution learning (LDL) primarily focuses on the establishment of prediction function from features to label distributions, utilizing instances annotated by label distributions. However, generating large and high-quality labeled datasets is a challenging task due to the common presence of inaccurate annotation (Xu and Zhou 2017). Tasks with high subjectivity, like sentiment analysis and topic classification, have their annotation data quality significantly influenced by the skill set, experience, and knowledge level of the annotators (Frénay and Verleysen 2013). For instance, in the process of annotating a sentiment mining dataset, an annotator might mistakenly label "happy" as "surprised". This subjectivity can introduce erroneous annotations, resulting in inaccurate LDL datasets. Training on such inaccurate datasets may undermine the performance of LDL models.

LEVI (Xu et al. 2020) uses variational methods to solve the label enhancement (Xu, Liu, and Geng 2019) problem. Inspired by this, to alleviate the inaccurate annotations in LDL datasets, we propose a **G**enerative **C**alibration model of **I**naccurate **A**nnotation (GCIA) to infer the true label distribution. Specifically, given a LDL dataset, which lacks any additional information apart from the feature space and the inaccuracy label space, we can only infer the ground-

*Corresponding author

truth label distribution by exploring the relationship between these two aspects of data and the ground-truth label distribution. Therefore, we adopt a generative model to model two relationships, compared to discriminative models that focus on modeling decision boundaries, as shown in Figure 1: (1) The relationship between ground-truth label distributions and the feature vectors. We adhere to the smoothness assumption, i.e., similar instances in the feature vectors should also be similar in ground-truth label distributions. Based on this assumption, we use the indicator variable $\boldsymbol{z}$ to guide the generation of the feature vector $\boldsymbol{x}$ (with a Gaussian mixture as the prior) and the ground-truth label distribution $\boldsymbol{y}$ (with a Dirichlet mixture as the prior). (2) The relationship between the ground-truth label distribution and the inaccuracy label distribution. We believe that many factors that cause annotators to give an inaccurate label distribution (such as the subjective cognitive level of the annotator, the vagueness of the label concept itself, etc.) are manifested as confusion of label concepts, that is, a label has a certain probability of being mistaken for another label. Based on this, we use the confusion matrix $\mathbf{\Pi}$ to encode concept confusion information. Specifically, π_{mt} in $\mathbf{\Pi}$ represents the probability of labeling the m-th label as the t-th label. The label given by the annotator is obtained from the ground-truth label distribution after concept confusion (i.e., the true label distribution multiplied by the confusion matrix).

Related Work

LDL involves learning a predictive function from features to label distributions based on instances annotated by label distributions. Geng (2016) first introduced the maximum entropy model as a representation of the predictive function. While the maximum entropy model is rooted in probability theory, it has a limited model capacity w.r.t. label distributions (e.g., it is difficult in representing multi-modal label distributions). To enhance the model capacity, some research efforts have adapted traditional machine learning models to address the LDL problem. For instance, LDSVR (Geng and Hou 2015) extends the support vector machine, LDLogitBoost (Xing, Geng, and Xue 2016) extends the LogitBoost model, and LDLFs (Shen et al. 2017) extends the differentiable decision tree model. Other approaches aim to mine label correlations (Jia et al. 2018; Zhao and Zhou 2018; Zheng, Jia, and Li 2018) or learn label embedding (Peng, Tao, and Geng 2018; Wang and Geng 2018; Xu, Shang, and Shen 2019) to improve the generalization ability of LDL models. Despite the success achieved by these LDL methods, they often assume the label distribution data to be accurate. However, in practical applications, accurate annotation can be challenging for annotators, making the label distribution annotations prone to noise (Xu and Zhou 2017).

In recent years, the issue of noise in LDL has received limited attention. An LDL approach called 3WD-LDL (Li et al. 2022) has been proposed to address the problem of noisy labeled data by utilizing the three-way decisions theory to remove amplified noise. However, 3WD-LDL relies on training a base model on accurately annotated trustworthy samples and then calibrating the noise using the correlation between the trustworthy samples and the noisy samples. The key challenge lies in obtaining a set of accurate trustworthy samples, which is difficult to identify within the annotated data without introducing additional expert knowledge for discerning trustworthiness. In our study, our goal is to address the limitations mentioned earlier by examining the causes of inaccurate label distributions and developing a preprocessing method tailored specifically for noise elimination in label grade recognition. This preprocessing method serves as an initial step that can be applied to any LDL algorithm, thereby improving its effectiveness in handling inaccurate label distributions.

Method

Notations

Let $\boldsymbol{x} \in \mathbb{R}^F$ denote the vector of feature variable, where F is the number of features. We denote $\{1, 2, \ldots, N\}$ as $[N]$. Let $\hat{\boldsymbol{y}} \in \{[\hat{y}^{(1)}, \hat{y}^{(2)}, \ldots, \hat{y}^{(M)}] \mid \forall m \in [M], \hat{y}^{(m)} \geq 0, \hat{\boldsymbol{y}}\, \mathbf{1}_M^{\mathsf{T}} = 1\}$ denote the label distribution with noise where M is the number of labels, and $\mathbf{1}_M$ is an all-ones vector. We use $\boldsymbol{y}$ to denote the latent label distribution. The i-th components in $\boldsymbol{x}, \boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ are denoted as $x^{(i)}, y^{(i)}$ and $\hat{y}^{(i)}$. The observations for $\boldsymbol{x}$ and $\hat{\boldsymbol{y}}$ constitute a dataset $\boldsymbol{D} = \{(\boldsymbol{x}_n, \hat{\boldsymbol{y}}_n)\}_{n=1}^N$. Our goal is to obtain the ground-truth label distributions based on $\boldsymbol{D}$.

Generative Model

The causal connections of GCIA encoding can be represented as the structure shown in Figure 1.

1. Generate a one-hot vector $\boldsymbol{z}$ from the category distribution to indicate which Gaussian/Dirichlet distribution has been selected:

$$\boldsymbol{z} \sim \mathrm{Cat}(\boldsymbol{z}|\boldsymbol{c}) \tag{1}$$

where K is the number of sub-models in the mixture model.

2. Generate feature vector $\boldsymbol{x}$ from Gaussian mixture:

$$\boldsymbol{x}|\boldsymbol{z} \sim \prod_{k=1}^{K} \mathcal{N}\left(\boldsymbol{x}|\mu_{\boldsymbol{x},k}, \Sigma_{\boldsymbol{x},k}\right)^{c_k} \tag{2}$$

where $\mu_{\boldsymbol{x},k}$ and $\Sigma_{\boldsymbol{x},k}$ represent the mean and variance of the k-th Gaussian mixture sub-model, while c_k represents the probability that the instance belongs to the k-th sub-model and $\sum_{k=1}^{K} c_k = 1$. $\boldsymbol{\mu_x}$ and $\boldsymbol{\Sigma_x}$ are multi-layer perceptrons (MLPs) with learnable parameters $\boldsymbol{\theta}_1$.

3. Generate latent ground-truth label distribution $\boldsymbol{y}$ from the Dirichlet distribution.:

$$\boldsymbol{y}|\boldsymbol{z} \sim \sum_{k=1}^{K} \mathrm{Dir}\left(\boldsymbol{y}|\boldsymbol{\alpha}_{\boldsymbol{y},k}\right)^{c_k} \tag{3}$$

where $\boldsymbol{\alpha}_{\boldsymbol{y},k}$ represents the concentration parameter of the k-th Dirichlet mixture component model, and $\boldsymbol{\alpha}_{\boldsymbol{y},k} \in \mathbb{R}^M$. M is the number of labels. $\boldsymbol{\alpha_y}$ is a MLP with learnable parameters $\boldsymbol{\theta}_2$.

4. Generate noise label distribution $\hat{\boldsymbol{y}}$ from the latent ground-truth label distribution:

   The confusion matrix is a powerful tool that can comprehensively depict the distribution of an annotator's capability over all pairs of labels, providing fine-grained information. In this context, a confusion matrix $\mathbf{\Pi}$ is used to model the reliability of M labels, and $\mathbf{\Pi} \in \mathbb{R}^{M\times M}$. Each column element of $\mathbf{\Pi}, \boldsymbol{\pi}_m = [\pi_{1m}, \pi_{2m}, \ldots, \pi_{tm}, ..., \pi_{Mm}]^{\mathsf{T}}$ represents the impact of other labels on the annotation label m, i.e., π_{tm} represents the impact of the t-th label on the annotation result when annotating the m-th label, $\sum_{t=1}^{M} \pi_{tm} = 1$. Using $\mathbf{\Pi}$ to confuse $\boldsymbol{y}$ yields the following equation:

$$\hat{\boldsymbol{y}}|\boldsymbol{y} \sim \mathrm{Dir}\left(\boldsymbol{\tau}_{\boldsymbol{y}}\right) \tag{4}$$

   where$\boldsymbol{\tau}_{\boldsymbol{y}} = \boldsymbol{y} \cdot \mathbf{\Pi}_{\hat{\boldsymbol{y}}} = \left[\tau_{\boldsymbol{y}}^{(1)}, \tau_{\boldsymbol{y}}^{(2)}, ..., \tau_{\boldsymbol{y}}^{(M)}\right]$,and $\mathbf{\Pi}_{\hat{\boldsymbol{y}}}$ is the learnable parameter, .

The joint density of the complete-data can be factorized as:

$$p\left(\boldsymbol{x}, \boldsymbol{y}, \hat{\boldsymbol{y}}, \boldsymbol{z}\right) = p\left(\boldsymbol{z}\right) p_{\boldsymbol{\theta}_1}\left(\boldsymbol{x}|\boldsymbol{z}\right) p_{\boldsymbol{\theta}_2}\left(\boldsymbol{y}|\boldsymbol{z}\right) p_{\boldsymbol{\theta}_2}\left(\hat{\boldsymbol{y}}|\boldsymbol{y}\right) \tag{5}$$

Inference

Since exact inference is intractable due to the nonconjugate dependence among variables, we adopt the variational inference, i.e., approximating the exact posterior of label distribution by a variational posterior. As shown in Figure 1, variational posterior, according to the mean-field assumption, can be factorized as:

$$q_{\phi}\left(\boldsymbol{z}, \boldsymbol{y}|\boldsymbol{x}, \hat{\boldsymbol{y}}\right) = q_{\phi_2}\left(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}\right) q_{\phi_1}\left(\boldsymbol{y}|\hat{\boldsymbol{y}}\right) \tag{6}$$

Since $\boldsymbol{z}$ is a discrete variable, we can analytically obtain $p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})$:

$$\begin{aligned}\gamma_{\boldsymbol{x},\boldsymbol{y}}^{(k)} &\triangleq p(z_k = 1|\boldsymbol{x}, \boldsymbol{y}) \\ &= \frac{p(z_k = 1)p(\boldsymbol{y}|z_k = 1)p(\boldsymbol{x}|z_k = 1)}{\sum_{k=1}^{K} p(z_k = 1)p(\boldsymbol{y}|z_k = 1)p(\boldsymbol{x}|z_k = 1)}\end{aligned} \tag{7}$$

We use $p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})$ to capture the dependency relationship between $\boldsymbol{z}$, $\boldsymbol{x}$ and $\boldsymbol{y}$, thereby alleviating the information loss caused by the mean-field assumption.

Next, we infer the label distribution by maximizing the **E**vidence **L**ower **BO**und (ELBO). According to Eq.(5) and Eq.(6), by integrating our decoder and encoder networks, we can express ELBO as follows:

$$\begin{aligned}&\mathbb{E}_{q(\boldsymbol{z},\boldsymbol{y}|\boldsymbol{x},\hat{\boldsymbol{y}})}\left[\log p(\boldsymbol{x}, \boldsymbol{y}, \hat{\boldsymbol{y}}, \boldsymbol{z}) - \log q(\boldsymbol{z}, \boldsymbol{y}|\boldsymbol{x}, \hat{\boldsymbol{y}})\right] \\ &= \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{y}|\hat{\boldsymbol{y}})}\log\frac{p\left(\boldsymbol{y}|\boldsymbol{z}\right)p\left(\hat{\boldsymbol{y}}|\boldsymbol{y}\right)p\left(\boldsymbol{x}|\boldsymbol{z}\right)p\left(\boldsymbol{z}\right)}{q\left(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}\right)q\left(\boldsymbol{y}|\hat{\boldsymbol{y}}\right)} \\ &= \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{y}|\hat{\boldsymbol{y}})}\left[\log p\left(\hat{\boldsymbol{y}}|\boldsymbol{y}\right) + \log p\left(\boldsymbol{x}|\boldsymbol{z}\right)\right] \\ &- \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{y}|\hat{\boldsymbol{y}})}\left[\log\frac{q\left(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}\right)}{p\left(\boldsymbol{z}\right)} + \log\frac{q\left(\boldsymbol{y}|\hat{\boldsymbol{y}}\right)}{p\left(\boldsymbol{y}|\boldsymbol{z}\right)}\right] \\ &= \underbrace{\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{y}|\hat{\boldsymbol{y}})}\left[\log p\left(\boldsymbol{x}|\boldsymbol{z}\right) + \log p\left(\hat{\boldsymbol{y}}|\boldsymbol{y}\right)\right]}_{\text{reconstruction term}} \\ &- \underbrace{\mathbb{E}_{q(\boldsymbol{y}|\hat{\boldsymbol{y}})}D_{KL}\left[q\left(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}\right)\|p\left(\boldsymbol{z}\right)\right]}_{\text{z-prior}} \\ &- \underbrace{\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}D_{KL}\left[q(\boldsymbol{y}|\hat{\boldsymbol{y}})||p\left(\boldsymbol{y}|\boldsymbol{z}\right)\right]}_{\text{y-prior}}\end{aligned} \tag{8}$$

It can be seen that the ELBO consists of two parts: reconstruction term and prior regularization. Once these two items are specified, we can learn the model by maximizing the ELBO.

Optimization Objective

Reconstruction term The first term of Eq.(8) is referred to as the reconstruction term. Due to the difficulty in dealing with the reconstruction term that involves integrating over the conditional likelihood, we extract Monte Carlo samples from $q_{\phi}(\boldsymbol{z}, \boldsymbol{y}|\boldsymbol{x}, \hat{\boldsymbol{y}})$ to estimate it. The differentiability of this term can be ensured by using standard reparameterization techniques for backpropagation:

$$\begin{aligned}&\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{y}|\hat{\boldsymbol{y}})}\left[\log p\left(\boldsymbol{x}|\boldsymbol{z}\right) + \log p\left(\hat{\boldsymbol{y}}|\boldsymbol{y}\right)\right] \\ &\approx \frac{1}{S}\sum_{s=1}^{S}\Big(\nu * \sum_{k=1}^{K}\gamma_{\boldsymbol{x},\boldsymbol{y}^{(s)}}^{(k)}\log\mathcal{N}\left(\boldsymbol{x}|\mu_{\boldsymbol{x},k}(\boldsymbol{z}), \Sigma_{\boldsymbol{x},k}(\boldsymbol{z})\right) \\ &+ \beta * \log\frac{\Gamma\left(\sum_{m=1}^{M}\tau_{\boldsymbol{y}^{(s)}}^{(m)}\right)}{\prod_{m=1}^{M}\Gamma(\tau_{\boldsymbol{y}^{(s)}}^{(m)})}\prod_{m=1}^{M}{\hat{y}^{(m)}}^{\tau_{\boldsymbol{y}^{(s)}}^{(m)}-1}\Big) + \mathrm{const}\end{aligned} \tag{9}$$

where S is the number of Monte Carlo samples. $\boldsymbol{\tau}_{\boldsymbol{y}^{(s)}} = \boldsymbol{y}^{(s)} \cdot \mathbf{\Pi}_{\boldsymbol{y}^{(s)}}$, where $\boldsymbol{y}^{(s)}$ is the reconstructed output, and $\boldsymbol{y}^{(s)} = \mathrm{Dir}(\boldsymbol{\alpha}_{\boldsymbol{y}^{(s)}})$. $\boldsymbol{\alpha}_{\boldsymbol{y}^{(s)}}$ represents the expanded concentration parameters for Dirichlet, and the expansion method is $\boldsymbol{\alpha}_{\boldsymbol{y}^{(s)}} = \xi(\underline{\boldsymbol{\alpha}_{\boldsymbol{y}}}(\hat{\boldsymbol{y}}); -1) \bigoplus \xi(-1; \underline{\boldsymbol{\alpha}_{\boldsymbol{y}}}(\hat{\boldsymbol{y}}))$, where $\xi(;-1)$ is used to obtain all dimensions except the last one, and $\xi(-1;)$ obtains the length of the last dimension. $\bigoplus$ denotes element-wise addition.

The z-prior and y-prior The z-prior and y-prior can encourage the posterior of the mixture coefficients and label distributions to be close to their respective priors. The prior on $\boldsymbol{z}$ can be estimated using Monte Carlo samples $\{\boldsymbol{y}^{(s)}\}_{s=1}^{S}$:

$$\begin{aligned}&\mathbb{E}_{q(\boldsymbol{y}|\hat{\boldsymbol{y}})}D_{KL}\left[p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})||p(\boldsymbol{z})\right] \\ &\approx \frac{1}{S}\sum_{s=1}^{S}\sum_{k=1}^{K}\gamma_{\boldsymbol{x},\boldsymbol{y}^{(s)}}^{(k)}\log\gamma_{\boldsymbol{x},\boldsymbol{y}^{(s)}}^{(k)} + \mathrm{const}\end{aligned} \tag{10}$$

The prior of y-prior can be calculated as:

$$\begin{aligned}&\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}D_{KL}\left[q(\boldsymbol{y}|\hat{\boldsymbol{y}})||p\left(\boldsymbol{y}|\boldsymbol{z}\right)\right] \\ &\approx \frac{1}{S}\sum_{s=1}^{S}\sum_{k=1}^{K}\gamma_{\boldsymbol{x},\boldsymbol{y}^{(s)}}^{(k)}\Big(\sum_{m=1}^{M}\log\Gamma\left(\alpha_{\boldsymbol{y},k}^{(m)}(\boldsymbol{z})\right) \\ &- \sum_{m=1}^{M}\left(\alpha_{\boldsymbol{y}^{(s)}}^{(m)} - \alpha_{\boldsymbol{y},k}^{(m)}(\boldsymbol{z})\right)\phi\left(\alpha_{\boldsymbol{y}^{(s)}}^{(m)}\right) \\ &+ \sum_{m=1}^{M}\log\Gamma\left(\alpha_{\boldsymbol{y}^{(s)}}^{(m)}\right)\Big) + \mathrm{const}\end{aligned} \tag{11}$$

where $\Gamma(\cdot)$ and $\phi(\cdot)$ are the gamma and digamma functions, respectively.

Optimization objective By combining Eq.(9), Eq.(10), and Eq.(11), we can obtain an approximate value for the optimization objective:

$$\begin{aligned}\text{ELBO} \approx & \sum_{n=1}^{N}\Big(\beta * \log\frac{\Gamma\left(\sum_{m=1}^{M}\tau_{\boldsymbol{y}_n}^{(m)}\right)}{\prod_{m=1}^{M}\Gamma(\tau_{\boldsymbol{y}_n}^{(m)})}\prod_{m=1}^{M}\hat{y}^{(m)^{\tau_{\boldsymbol{y}_n}^{(m)}-1}} \\ & + \nu * \sum_{k=1}^{K}\gamma_{\boldsymbol{x}_n,\boldsymbol{y}_n}^{(k)}\log\mathcal{N}\left(\boldsymbol{x}|\mu_{\boldsymbol{x},k}(\boldsymbol{z}),\Sigma_{\boldsymbol{x},k}(\boldsymbol{z})\right) \\ & - \sum_{k=1}^{K}\gamma_{\boldsymbol{x}_n,\boldsymbol{y}_n}^{(k)}\Big(\sum_{m=1}^{M}\log\Gamma\left(\alpha_{\boldsymbol{y}_n,k}^{(m)}(\boldsymbol{z})\right) + \sum_{m=1}^{M}\log\Gamma\left(\alpha_{\boldsymbol{y}_n}^{(m)}\right) \\ & - \sum_{m=1}^{M}\left(\alpha_{\boldsymbol{y}_n}^{(m)} - \alpha_{\boldsymbol{y}_n,k}^{(m)}(\boldsymbol{z})\right)\phi\left(\alpha_{\boldsymbol{y}_n}^{(m)}\right)\Big) \\ & - \sum_{k=1}^{K}\gamma_{\boldsymbol{x}_n,\boldsymbol{y}_n}^{(k)}\log\gamma_{\boldsymbol{x}_n,\boldsymbol{y}_n}^{(k)}\Big) + \text{const} \end{aligned} \tag{12}$$

where $\boldsymbol{\alpha}_{\boldsymbol{y}_n}$ represents the expanded concentration parameters for Dirichlet, and the expansion method is $\boldsymbol{\alpha}_{\boldsymbol{y}_n} = \xi(\boldsymbol{\alpha}_{\underline{\boldsymbol{y}}}(\hat{\boldsymbol{y}_n}); -1) \bigoplus \xi(-1; \boldsymbol{\alpha}_{\underline{\boldsymbol{y}}}(\hat{\boldsymbol{y}_n}))$, where $\xi(;-1)$ is used to obtain all dimensions except the last one, and $\xi(-1;)$ obtains the length of the last dimension. $\bigoplus$ denotes element-wise addition. The parameters of $\boldsymbol{\mu}_{\boldsymbol{x}}(\cdot)$, $\boldsymbol{\Sigma}_{\boldsymbol{x}}(\cdot)$, $\boldsymbol{\alpha}_{\boldsymbol{y}}(\cdot)$ and $\boldsymbol{\alpha}_{\underline{\boldsymbol{y}}}(\cdot)$ need to be learned. We set $S = 1$, which means generating only one Monte Carlo sample for each observation (Kingma and Welling 2014).

Recovering label distributions In the optimization process, we employ standard stochastic gradient-based optimization methods. Once $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \phi_1, \phi_2\}$ are determined, we sample the latent label distribution $\boldsymbol{y}_i$ of each sample $\boldsymbol{x}_i, \hat{\boldsymbol{y}}_i$ from $q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{y}|\boldsymbol{x}, \hat{\boldsymbol{y}}) = q_{\phi_2}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})q_{\phi_1}(\boldsymbol{y}|\hat{\boldsymbol{y}})$. Eventually, we use softmax normalization to normalize $\boldsymbol{y}_i$. Notably, as $\boldsymbol{y}_i$ being a latent variable generated by other latent variables, in order to avoid relying merely on the sampled values of the latent variables and considering the collected inaccurate annotation data contains some useful base fact information, such as correct logical labels or label rankings, we consider retaining part of the information of $\hat{\boldsymbol{y}}_i$. The final inferred ground-truth label representation is $\tilde{\boldsymbol{y}} = (1-\lambda)\boldsymbol{y}_i + \lambda\hat{\boldsymbol{y}}_i$, where λ represents the credibility of the dataset.

Experiments

Experimental Configuration

Datasets Based on the annotation methodology of datasets, we classify the LDL datasets into three distinct groups. The specific datasets we use are presented in Table 1.

Subjective annotation datasets: These datasets are sourced from subjective annotation tasks, including emotion mining (No.1-2) and movie rating prediction (No.3). They were meticulously generated through manual annotation processes that assigned specific grades or ratings to the data instances.

No	Datasets	Instance	Features	Labels
1	SJAFFE (sj)	213	243	6
2	Twitter-LDL (twit)	10045	168	8
3	Movie (mov)	7755	1869	5
4	Yeast-heat (heat)	2465	24	6
5	Yeast-cold (cold)	2465	24	4
6	Yeast-spo (spo)	2465	24	6
7	Yeast-spo5 (spo5)	2465	24	3
8	Nature-Scence (ns)	2000	294	9

Table 1: Datasets statistics

Biological experiment datasets: This category of datasets comprises three Yeast datasets (No.4-7). These datasets are real-world datasets collected from the biological experiments.

Ranking dataset: This type of dataset includes a natural scene dataset (No.8). This type of dataset is created through a manual annotation process focused on label ranking.

According to our definition of inaccurate annotations (in highly subjective annotation tasks, inaccurate annotations are prone to occur), the data characteristics of the first type of datasets are more suitable for our model experiments. Considering the data characteristics of the second type of datasets, it can be used as toy datasets. The third type of datasets is obtained through label ranking, and its data characteristics differ from those of directly annotated data based on label distribution. We expect limited model performance on this type of datasets.

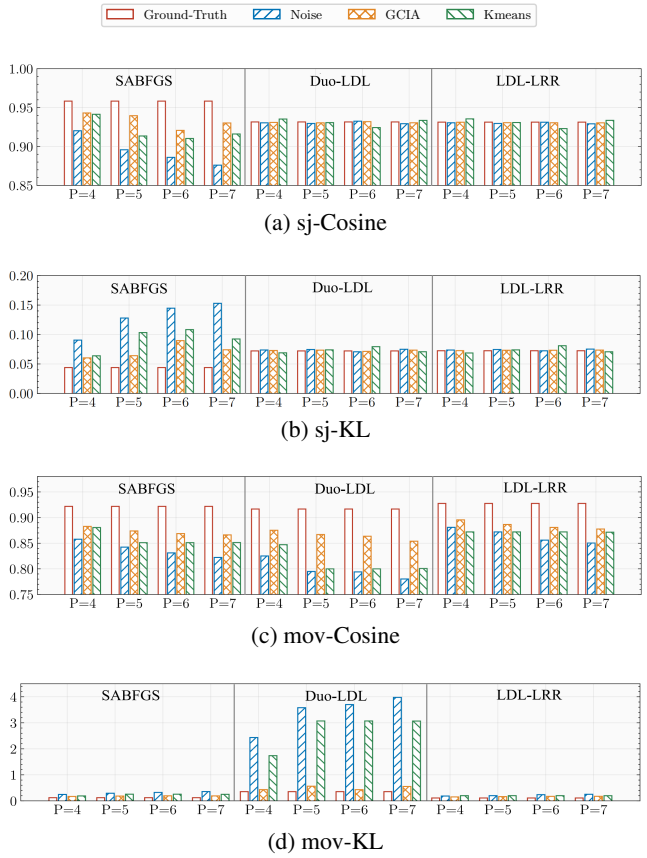
Evaluation measures We take into account a total of seven evaluation metrics. Six of these metrics were suggested by Geng (2016), including Chebyshev distance (Cheb), Clark distance (Clark), Canberra metric (Canber), Kullback-Leibler divergence (KL), cosine coefficient (Cosine), and intersection similarity (Intersec). Additionally, we also include a ranking metric known as Spearman's rank (Rho), which was suggested by Jia et al. (2023). These evaluation indicators provide comprehensive measures to assess the performance of our model. The first four measures are based on distance (whose lower values indicate better performances). The next two measures are based on similarity of distribution and the last measure is based on similarity of ranking (whose higher values indicate better performances).

Prediction methods We have chosen three LDL prediction methods that leverage different information: SABFGS (Geng 2016), Duo-LDL (Żychowski and Mańdziuk 2021) and LDL-LRR (Jia et al. 2021). The hyperparameter configuration for each method follows their respective literature. Specifically, for Duo-LDL, the learning rate set to 0.05 and the weight decay cost set to 0.5. For LDL-LRR, λ and β are selected from $10^{\{-6,-5,\ldots,-2,-1\}}$ and $10^{\{-3,-2,\ldots,1,2\}}$, respectively.

Simulation scene settings We argue the annotation process as akin to a voting task, which aligns with the Central Limit Theorem. In this context, the mean of the Gaussian distribution represents the ground-truth of the label,

	P=4	GCIA	Kmeans	P=5	GCIA	Kmeans	P=6	GCIA	Kmeans	P=7	GCIA	Kmeans
						sj						
Cheb (↓)	0.1649	**0.1248** •	**0.1373**	0.2015	**0.1287** •	**0.1722**	0.2292	**0.1595** •	**0.2102**	0.2350	**0.1363** •	**0.1767**
Clark (↓)	0.8651	**0.5609** •	**0.6019**	1.0721	**0.5695** •	**0.6778**	1.1870	**0.7077** •	**0.7703**	1.2717	**0.5689** •	**0.6695**
Canber (↓)	1.6629	**1.1315** •	**1.2050**	2.1086	**1.1661** •	**1.3805**	2.3258	**1.4210** •	**1.5861**	2.5752	**1.1733** •	**1.3783**
KL (↓)	0.5583	**0.1247** •	**0.1444**	0.9452	**0.1253** •	**0.1899**	1.2150	**0.2040** •	**0.2482**	1.5298	**0.1311** •	**0.1987**
Cosine (↑)	0.8746	**0.9148** •	**0.9047**	0.8285	**0.9068** •	**0.8660**	0.8024	**0.8751** •	**0.8363**	0.7806	**0.8970** •	**0.8522**
Intersec (↑)	0.7654	**0.8227** •	**0.8102**	0.7088	**0.8142** •	**0.7735**	0.6803	**0.7770** •	**0.7391**	0.6525	**0.8075** •	**0.7664**
Rho (↑)	0.3402	0.3320	**0.3643** •	0.3147	0.2801	**0.3343** •	0.2658	**0.3006** •	**0.2793**	0.2423	0.1846	0.2182
						mov						
Cheb (↓)	0.2230	**0.1676** •	**0.1986**	0.2591	**0.1799** •	**0.2566**	0.2841	**0.1888** •	**0.2566**	0.3019	**0.1873** •	**0.2710**
Clark (↓)	0.9269	**0.6349**	**0.6232** •	1.0725	**0.6830** •	**0.7729**	1.1699	**0.7115** •	**0.7729**	1.2400	**0.7206** •	**0.7882**
Canber (↓)	1.6688	**1.1829** •	**1.1871**	1.9613	**1.2808** •	**1.4858**	2.1618	**1.3389** •	**1.4858**	2.3137	**1.3688** •	**1.5229**
KL (↓)	0.8784	**0.1707** •	**0.2004**	1.2584	**0.2035** •	**0.3437**	1.5343	**0.2261** •	**0.3437**	1.7582	**0.2080** •	**0.3419**
Cosine (↑)	0.8506	**0.8959** •	**0.8725**	0.8114	**0.8796** •	**0.8133**	0.7876	**0.8691** •	**0.8133**	0.7673	**0.8617** •	**0.7955**
Intersec (↑)	0.7320	**0.7930** •	**0.7696**	0.6870	**0.7750** •	**0.7064**	0.6574	**0.7637** •	**0.7064**	0.6339	**0.7570** •	**0.6907**
Rho (↑)	0.5356	**0.6118** •	**0.6082**	0.4477	**0.5649** •	**0.4644**	0.4464	**0.5444** •	**0.4644**	0.4321	0.3369	0.4270
						twit						
Cheb (↓)	0.1272	0.1426	0.1298	0.1675	0.1774	0.1680	0.1995	0.2061	**0.1991** •	0.0563	0.0842	0.0625
Clark (↓)	2.3326	**2.3262** •	**2.3294**	2.3619	**2.3506** •	**2.3552**	2.3859	**2.3703** •	**2.3779**	2.2940	**2.2940** •	2.2943
Canber (↓)	5.7762	**5.7509** •	**5.7654**	5.9300	**5.8884** •	**5.9068**	6.0497	**5.9942** •	**6.0241**	5.4955	5.5072	5.5006
KL (↓)	0.2178	**0.1912**	**0.1532** •	0.4493	**0.3031**	**0.2734** •	0.6581	**0.3968**	**0.3830** •	0.0171	0.0790	0.0367
Cosine (↑)	0.9475	**0.9489** •	**0.9484**	0.9142	**0.9170** •	**0.9162**	0.8867	**0.8907** •	**0.8887**	0.9902	0.9885	0.9899
Intersec (↑)	0.8692	0.8407	0.8623	0.8274	0.8039	0.8224	0.7944	0.7743	0.7903	0.9418	0.9021	0.9309
Rho (↑)	0.0880	**0.7934**	**0.8033** •	0.0850	**0.7659**	**0.7848** •	0.8648	0.7443	0.7706	0.9921	0.8183	0.8185
						heat						
Cheb (↓)	0.1592	**0.1437** •	**0.1493**	0.1861	**0.1505** •	0.1918	0.2166	**0.1746** •	**0.1918**	0.2339	**0.1882** •	**0.2089**
Clark (↓)	0.8650	**0.7391** •	**0.7656**	1.0435	**0.7547** •	**0.9273**	1.1907	**0.8575** •	**0.9273**	1.2715	**0.9097** •	**0.9914**
Canber (↓)	1.6647	**1.4486** •	**1.5011**	2.0166	**1.5094** •	**1.8848**	2.3442	**1.7429** •	**1.8848**	2.5405	**1.8711** •	**2.0369**
KL (↓)	0.5845	**0.2255** •	**0.2422**	0.9170	**0.2162** •	**0.3278**	1.2691	**0.2807** •	**0.3278**	1.4773	**0.3144** •	**0.3756**
Cosine (↑)	0.8684	**0.8873** •	**0.8812**	0.8340	**0.8781** •	0.8307	0.7976	**0.8490** •	**0.8307**	0.7772	**0.8328** •	**0.8118**
Intersec (↑)	0.7612	**0.7846** •	**0.7771**	0.7184	**0.7723** •	0.7173	0.6751	**0.7377** •	**0.7173**	0.6496	**0.7182** •	**0.6954**
Rho (↑)	0.2033	0.2000	0.2029	0.1655	0.1638	0.1479	0.1477	0.1427	**0.1479** •	0.1100	**0.1121**	**0.1181** •
						spo						
Cheb (↓)	0.1643	**0.1485** •	**0.1544**	0.1914	**0.1726** •	0.2056	0.2175	**0.1757** •	**0.1948**	0.2363	**0.1901** •	**0.2110**
Clark (↓)	0.8896	**0.7581** •	**0.7852**	1.0463	**0.8838** •	**1.0418**	1.1868	**0.8472** •	**0.9203**	1.2684	**0.9029** •	**0.9907**
Canber (↓)	1.7004	**1.4775** •	**1.5298**	2.0266	**1.7467** •	2.0873	2.3356	**1.7202** •	**1.8708**	2.5343	**1.8573** •	**2.0361**
KL (↓)	0.6276	**0.2373** •	**0.2561**	0.9201	**0.3211** •	**0.4536**	1.2356	**0.2754** •	**0.3267**	1.4654	**0.3162** •	**0.3833**
Cosine (↑)	0.8658	**0.8842** •	**0.8782**	0.8313	**0.8538** •	0.8170	0.8014	**0.8510** •	**0.8319**	0.7799	**0.8335** •	**0.8122**
Intersec (↑)	0.7578	**0.7811** •	**0.7736**	0.7160	**0.7440** •	0.6986	0.6778	**0.7406** •	**0.7187**	0.6516	**0.7197** •	**0.6952**
Rho (↑)	0.2622	**0.2624**	**0.2640** •	0.2014	**0.2059** •	0.2012	0.2006	**0.2040** •	**0.2023**	0.1569	**0.1606** •	0.1539
						cold						
Cheb (↓)	0.2707	**0.2440** •	**0.2644**	0.3043	**0.2746** •	**0.2979**	0.3277	**0.2953** •	**0.3197**	0.1555	**0.1414**	**0.1401** •
Clark (↓)	0.9198	**0.7801** •	**0.8473**	1.0310	**0.8732** •	**0.9504**	1.1013	**0.9284** •	**1.0042**	0.4971	**0.4341**	**0.4298** •
Canber (↓)	1.5177	**1.3080** •	**1.4161**	1.7290	**1.4850** •	**1.6153**	1.8578	**1.5905** •	**1.7155**	0.8089	**0.7172**	**0.7098** •
KL (↓)	1.2431	**0.4198** •	**0.5117**	1.6209	**0.5286** •	**0.6584**	1.8821	**0.5692** •	**0.6915**	0.2663	**0.1251**	**0.1201** •
Cosine (↑)	0.8124	**0.8371** •	**0.8203**	0.7821	**0.8090** •	**0.7901**	0.7634	**0.7917** •	**0.7724**	0.9179	**0.9295**	**0.9306** •
Intersec (↑)	0.6848	**0.7161** •	**0.6962**	0.6452	**0.6806** •	**0.6572**	0.6215	**0.6589** •	**0.6361**	0.8204	**0.8365**	**0.8381** •
Rho (↑)	0.1217	**0.1317** •	**0.1262**	0.1102	**0.1338** •	**0.1168**	0.0112	**0.1302** •	**0.1120**	0.2503	0.2383	**0.2542** •

Table 2: Recovery performance on 6 datasets. Every three columns from left to right represent one group of experiments, with the same P-value used to generate noise. 'P=4/5/6/7' represents the difference between the unprocessed noise dataset and the ground-truth. 'GCIA/KMeans' represents the difference between the dataset calibrated by the calibrationn method and the ground-truth. Bold indicates that the calibrated dataset is superior to the noise dataset. A dot following the data indicates that the calibrationn method is better.



(a) sj-Cosine

(b) sj-KL

(c) mov-Cosine

(d) mov-KL

Figure 2: Prediction performance. "Noise" refers to directly using a dataset that contains noise for training an LDL model. "GCIA/KMeans" represents using GCIA/KMeans as a preprocessing method for the dataset before utilizing it for model training. The processed dataset is then employed in the training process. "P=4/5/6/7" indicates different values of professionalism P used to generate the noisy dataset.

while the variance determines the potential interval of values. Concerning particular label within a specific instance, noisy labels exhibit a relationship with two key factors: the quantity of labels M and the size of the ground-truth y_m. In relation to the entire dataset, annotation is affected by the proficiency P of the annotator. Therefore, we utilize the Gauss$(y_m, \frac{y_m}{M} * P)$ resampling to simulate the annotation process and generate datasets with added noise. This approach is based on the assumption that the data in the standard dataset is clean, and aims to reproduce the annotation scenario by introducing synthetic noise. The professional level P of the annotator is represented on a scale ranging from 1 to 10, where lower values indicate greater levels of professionalism (extremely professional $P \in [1, 3]$) and higher values signify lower levels of expertise (extremely unprofessional $P \in [8, 10]$). For the experiment, professional levels P of 4, 5, 6, and 7 were chosen to represent a centralized and moderately professional annotator.

Recovery Experiment

Methodology The procedure for recovery experiment is as follows. First, we simulate scenario by varying the values of P, generating noisy datasets. Then, we apply our GCIA model to recover noisy datasets, resulting in the GCIA-recovered datasets. Secondly, for conducting the comparative experiment, we cluster the noisy datasets using the KMeans algorithm, and then use the clustering center to calibrate the noisy datasets, resulting in the KMeans-recovered datasets. Ensure that the clustering parameters K and datasets feasibility λ have the same values as those used in our model. Finally, we calculate the seven evaluation measures for each of the three datasets(Noisy, GCIA, and KMeans) compared to the ground-truth datasets.

Performance The results of the recovery experiment are shown in Table 2. Our model performs well on the 'sj', 'mov', 'heat', 'spo', 'spo5' and 'cold' datasets. However, in some cases, it struggles to improve the Rho indicator. We argue that when the differences in label values are small, even if the overall label distributions show similarity, the specific ranking of the labels may vary. The performance on the 'twit' dataset is average due to the deliberate removal of images with highly similar features during its construction (Yang, Sun, and Sun 2017). Especially when the noise level is high (P=7), there are significant differences in both feature vectors and label distributions between instances. Overall, the experimental results align with our expectations. For more information, please refer to the supplementary materials.

Predictive Experiment

Methodology To evaluate the effectiveness of GCIA in LDL tasks, we conduct the following experiment. Firstly, we randomly partition the dataset into a training set (90%) and a testing set (10%). Next, we introduce noise to the training set, resulting in a noisy dataset. Subsequently, we apply the GCIA and the KMeans method to recover the label distributions for each training instance. We use these recovered label distributions to train the LDL model. For comparison, we also train the LDL model directly using the noisy dataset. To assess the predictive performance of the LDL model, we record the performance of the SABFGS, Duo-LDL, and LDL-LRR algorithms on the test instances. Finally, we repeat this process 10 times and report the average performance across the repetitions.

Performance The accompanying Figure 2 illustrates some of the experimental results obtained from our model. It is important to note that when the indicator values are closer to the ground-truth values, the model performs better. Merely improving the indicators without achieving proximity to the ground-truth would hold little significance. Clearly, our model demonstrates denoising capabilities, resulting in prediction outcomes that closely resemble those obtained through training on ground-truth data.

Parameter Sensitivity

Here, we demonstrate how hyperparameters λ and K affect the recovery performance and prediction performance for

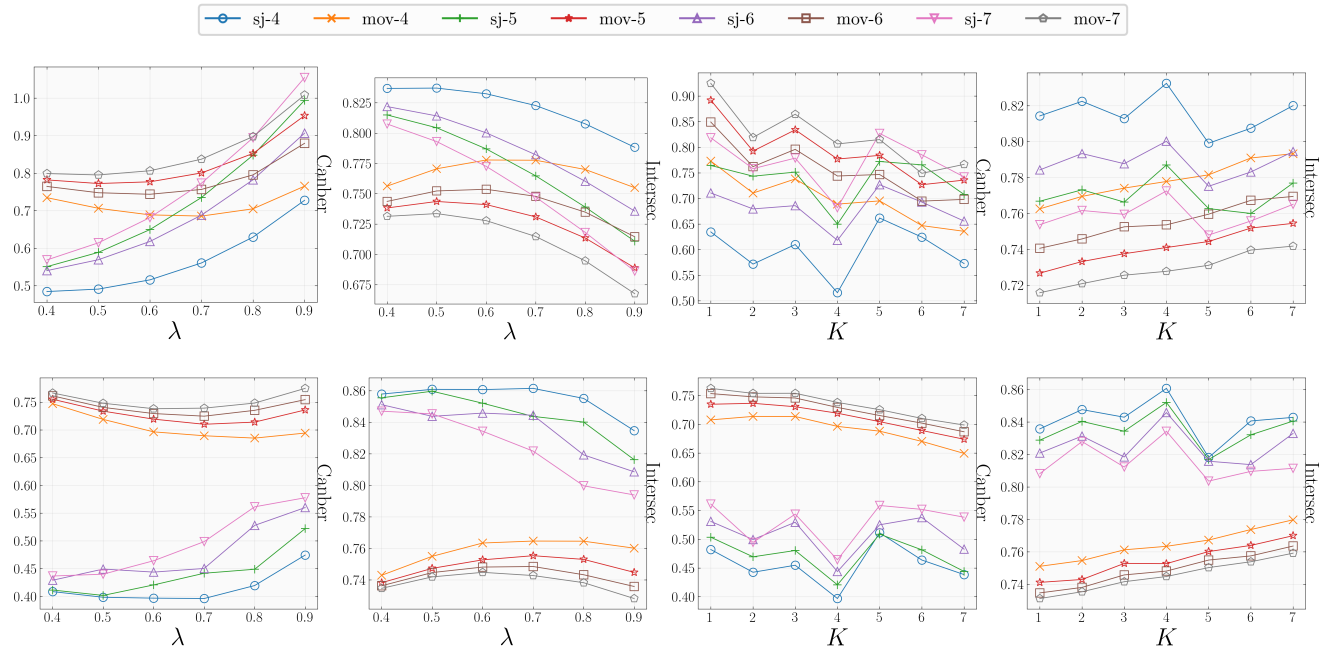
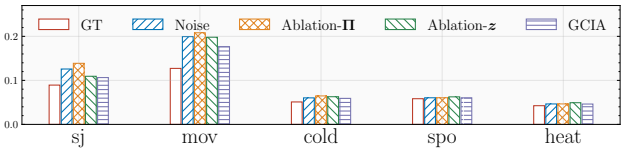
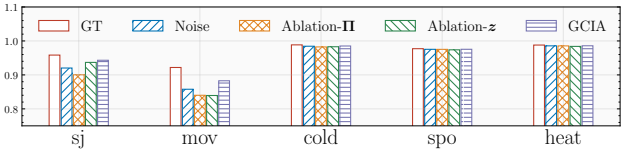


Figure 3: Recovery performance with varying K and λ. When conducting the parameter sensitivity of predictive experiment for K and λ, we keep λ and K fixed at 0.6 and 4, respectively. '-4/5/6/7' represents different values of P. The four figures above the result depict the parameter variation results of the recovery experiment, while the four figures below display the parameter variation results of the prediction experiment (using SABFGS as an example).



(a) Cheb (↓)



(b) Cosine (↑)

Figure 4: Ablation study. By setting K=1, we obtained the prediction experiment results without the categorical variable z (using SABFGS as an example). The figures depict the experimental results at $P = 4$. 'Ablation-' indicates the removal of this component. 'GCIA' represents a complete model.

the datasets 'sj' and 'mov' in Figure 3. The value of λ varies in $\{0.4, 0.5, \ldots, 0.9\}$. We can see that as the noise level increases, the optimal value of the λ parameter shifts towards smaller values. This indicates that higher noise levels result in reduced confidence in the information contained within the original dataset. As for the K, The value of K varies in $\{1, 2, \ldots, 7\}$. The increase in noise did not significantly affect the selection of the optimal value for the parameter K. Regardless of the noise level, the trend of the change in K value exhibits similar patterns and has a consistent impact on the results.

Ablation Study

Here, we show the effectiveness of each module proposed in our method. We remove the categorical variable $\boldsymbol{z}$ and confusion matrix $\boldsymbol{\Pi}$ and examine the performance of the prediction experiments. To obtain the best performance of the modified models, the hyperparameter λ of the modified models is re-tuned. It can be observed that our model exhibits better performance when the categorical variable $\boldsymbol{z}$ and $\boldsymbol{\Pi}$ are added. This is because by incorporating the categorical variable $\boldsymbol{z}$ and $\boldsymbol{\Pi}$, we can effectively consider the information in the feature space and simulate the scenarios of annotators during the annotation process.

Conclusion

This paper highlights the issue of inaccurate annotation during the annotation process and proposes the GCIA model to address this problem. Specifically, the model assumes the existence of a latent ground-truth label distribution and employs a generative approach to construct the model. We designed recovery experiments and prediction experiments. Our model demonstrates calibration capabilities, resulting in prediction outcomes that closely resemble those obtained through training on ground-truth data.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (62176123, 61906090), Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security (MIMS23-M-03), and the Fund of Prospective Layout of Scientific Research for Nanjing University of Aeronautics and Astronautics.

References

Frénay, B.; and Verleysen, M. 2013. Classification in the Presence of Label Noise: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 25(5): 845–869.

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.

Geng, X.; and Hou, P. 2015. Pre-release Prediction of Crowd Opinion on Movies by Label Distribution Learning. In *International Joint Conference on Artificial Intelligence*, 3511–3517.

Geng, X.; and Xia, Y. 2014. Head Pose Estimation Based on Multivariate Label Distribution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1837–1842.

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

Jia, X.; Li, W.; Liu, J.; and Zhang, Y. 2018. Label Distribution Learning by Exploiting Label Correlations. In *AAAI Conference on Artificial Intelligence*, 3310–3317.

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*, 35(2): 1695 – 1707.

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/CVF Conference on Computer Vision and Pattern Recognition*, 9841–9850.

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

Li, W.; Lu, Y.; Chen, L.; and Jia, X. 2022. Label Distribution Learning with Noisy Labels via Three-Way Decisions. *International Journal of Approximate Reasoning*, 150: 19–34.

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

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 Workshops*, 1232–1240.

Peng, C.-L.; Tao, A.; and Geng, X. 2018. Label Embedding Based on Multi-Scale Locality Preservation. In *International Joint Conference on Artificial Intelligence*, 2623–2629.

Shen, W.; Guo, Y.; Wang, Y.; Zhao, K.; Wang, B.; and Yuille, A. 2019. Deep Differentiable Random Forests for Age Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(2): 404–419.

Shen, W.; Zhao, K.; Guo, Y.; and Yuille, A. 2017. Label Distribution Learning Forests. In *International Conference on Neural Information Processing Systems*, 834–843.

Wang, K.; and Geng, X. 2018. Binary Coding based Label Distribution Learning. In *International Joint Conference on Artificial Intelligence*, 2783–2789.

Xing, C.; Geng, X.; and Xue, H. 2016. Logistic Boosting Regression for Label Distribution Learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 4489–4497.

Xu, M.; and Zhou, Z.-H. 2017. Incomplete Label Distribution Learning. In *International Joint Conference on Artificial Intelligence*, 3175–3181.

Xu, N.; Liu, Y.-P.; and Geng, X. 2019. Label Enhancement for Label Distribution Learning. *IEEE Transactions on Knowledge and Data Engineering*, 33(4): 1632–1643.

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

Xu, S.; Shang, L.; and Shen, F. 2019. Latent Semantics Encoding for Label Distribution Learning. In *International Joint Conference on Artificial Intelligence*, 3982–3988.

Xu, Y.; and Wang, X. 2020. 3D Hand Pose Estimation from Single Depth Images with Label Distribution Learning. In *IEEE International Conference on Embedded Software and Systems*, 1–5.

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.; and Zhou, Z.-H. 2013. A Review on Multi-Label Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8): 1819–1837.

Zhao, P.; and Zhou, Z.-H. 2018. Label Distribution Learning by Optimal Transport. In *AAAI Conference on Artificial Intelligence*, 4506–4513.

Zheng, X.; Jia, X.; and Li, W. 2018. Label Distribution Learning by Exploiting Sample Correlations Locally. In *AAAI Conference on Artificial Intelligence*, 4556–4563.

Żychowski, A.; and Mańdziuk, J. 2021. Duo-LDL Method for Label Distribution Learning Based on Pairwise Class Dependencies. *Applied Soft Computing*, 110(3): 107585.