Learning Label Distribution with Dirichlet Process Mixture Model

Minglong Wang[1], Weiwei Li[1]*, Yunan Lu[2,3], Xiuyi Jia[3]

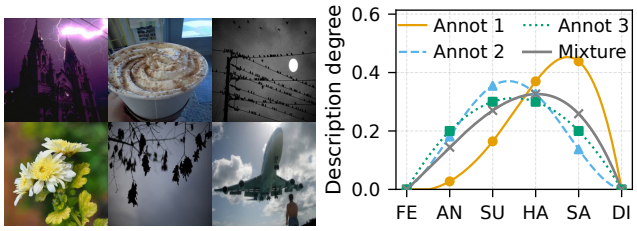
[1]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
[2]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[3]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
{mlwang777, liweiwei}@nuaa.edu.cn, yunan.lu@polyu.edu.hk, jiaxy@njust.edu.cn

Abstract

Label Distribution Learning (LDL) is an effective machine learning paradigm for addressing label ambiguity, where each sample is annotated with a distribution that conveys rich semantic information. However, during the actual annotation process of label distributions, annotators often exhibit divergent labeling preferences for the same sample. Most existing LDL methods overlook this heterogeneity, assuming that the observed label distribution originates from a single labeling pattern. Such an assumption limits their capacity to manage inter-annotator disagreement and constrains the generalization of the resulting models. To address this issue, we propose, for the first time, a Dirichlet process mixture model (DPMM)-based framework for LDL. This framework leverages nonparametric Bayesian methods to adaptively uncover diverse latent labeling patterns from the data and to accurately model annotator heterogeneity. Specifically, the ground-truth label distribution of each sample is modeled as a weighted mixture of multiple latent components, where a feature-conditioned gating mechanism adaptively controls the contribution of each component. Experimental results demonstrate that the proposed model consistently achieves competitive performance on several widely-used benchmark datasets.

1 Introduction

In traditional supervised learning paradigms such as Single-Label learning (SLL) and Multi-Label learning (MLL), the relationship between a sample and its label(s) is assumed to be deterministic (Zhang and Zhou 2014). However, in many real-world scenarios, label ambiguity is widespread (Gao et al. 2017), i.e., the correspondence between samples and labels usually contains inherent uncertainty (Rupprecht et al. 2017). To capture the diverse perceptions of annotators in realistic settings, Geng (2016) proposed a new learning paradigm called Label Distribution Learning (LDL). Unlike SLL and MLL, which assign a discrete label or a set of labels, LDL assigns to each sample a non-negative real-valued vector, called a label distribution, whose elements quantify the relative importance of the corresponding labels and collectively sum to one. Since label distribution conveys richer information about label uncertainty, LDL has been widely applied in tasks such as age estimation (Gao et al. 2018; Wen et al. 2020), head-pose estimation (Liu et al. 2019; Zhang et al. 2020), and sentiment analysis (Jia et al. 2019; Li et al. 2021), demonstrating a strong capability to represent label ambiguity.



(a) Visual sentiment images (b) Annotator heterogeneity

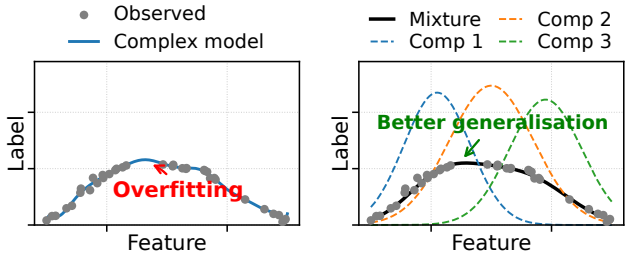
Figure 1: Illustration of annotator heterogeneity in label distribution: (a) a collage of example images sampled from the Emotion6 dataset; (b) label-probability curves from three different annotators (colored) and their weighted mixture (grey), highlighting the divergence among annotators.

During the annotation process of label distributions, a single sample is often labeled by multiple annotators with significantly different opinions, reflecting the inherent heterogeneity in the annotation process. Taking visual sentiment assessment[1] as an example, different types of annotators may evaluate the emotional attitude in an image differently. Such heterogeneity suggests that the observed label distribution arises from a mixture of annotator opinions, as illustrated in Figure 1.

Most existing LDL methods assume that the observed label distribution arises from a single underlying labeling pattern. This simplifying assumption limits their ability to model annotator heterogeneity, which is prevalent in many real-world scenarios. In practice, due to the diversity of annotation opinions, the target label distribution often exhibits multi-modal structures. However, a single model is typically capable of capturing only the overall trend, thereby reducing its performance when applied to highly subjective or heterogeneous data.

*Corresponding author.

[1]FE, AN, SU, HA, SA, and DI represent 6 common emotions in sentiment analysis datasets, namely fear, anger, surprise, happiness, surprise, and disgust, respectively.



(a) Single complex model (b) Mixture of simple components

Figure 2: Comparison between a single complex flexible model (prone to over-fitting) and a weighted mixture of several simple components that achieves better generalisation.

A mixture model, in contrast, is capable of capturing diverse heterogeneous labeling patterns through multiple components and weighting them appropriately to better fit the multi-modal distribution. As illustrated in Figure 2, combining multiple simple components provides a better fit to the data, mitigates the risk of overfitting, and enhances generalization, compared to using a single complex model. Although previous studies (Lu and Jia 2022; Lu et al. 2023) have introduced the idea of using mixture model to represent label distributions, a key limitation is that the number of mixture component is predetermined. Given the inherently unpredictable nature of annotator heterogeneity, assuming a fixed number of mixture components restricts the model's capacity to adaptively capture diverse labeling behaviors. Hence, a more flexible modeling framework is required.

Motivated by the above observations, we propose a novel **D**irichlet **p**rocess **m**ixture model (DPMM)-based framework for **L**abel **D**istribution **L**earning, termed LDL-DPM. As a classical nonparametric Bayesian method, the DPMM can adaptively determine the number of mixture components based on the underlying distributional complexity, thereby addressing the inherent limitations of conventional mixture models with fixed component numbers. However, directly applying standard DPMMs to LDL remains challenging, as traditional DPMMs assign global component weights, thereby inducing a single shared mixture structure across all samples. This limits their capacity to capture sample-specific feature variations. To address this limitation, a feature-conditioned gating mechanism is further introduced to dynamically assign sample-specific component weights, enabling fine-grained adaptation to heterogeneous annotation behaviors and improving generalization.

The main contributions of this work are summarized as follows:

- We analyze the heterogeneity inherent in the label distribution generation process, which is prevalent. Ignoring such heterogeneity may undermine model performance, particularly when handling highly heterogeneous or strongly subjective data.
- We propose LDL-DPM, which addresses heterogeneity through mixture modelling and improves generalization under heterogeneous annotations. In addition, it adaptively determines the number of mixture components, thereby enhancing model flexibility.
- We design a feature-conditioned gating mechanism that enhances the representational capacity of DPMM-based label distribution predictors.

2 Related Work

Our work primarily focuses on LDL, a learning paradigm that aims to learn a mapping function from input features to label distributions, where each sample is annotated with a distribution over labels. Initially introduced to address the problem of age estimation (Geng, Yin, and Zhou 2013), LDL has since evolved into a distinct machine learning paradigm (Geng 2016) and has been supported by solid theoretical foundations (Wang and Geng 2019a).

A wide range of LDL algorithms have been proposed in the literature. According to (Geng 2016), existing LDL methods can generally be categorized into three groups: Problem Transformation (PT), Algorithm Adaptation (AA), and Specialized Algorithms (SA). PT methods convert the original LDL task into an SLL problem by transforming the label distribution into SLL-style data, which can then be learned using conventional SLL algorithms. For instance, PT-Bayes (Geng and Hou 2015) applies Bayes classifier. AA methods, in contrast, directly adapt existing machine learning algorithms to handle label distributions. Specifically, CPNN (Geng, Yin, and Zhou 2013) utilizes a three layer neural network to approximate label distribution. SA methods are designed from scratch, specifically tailored to the unique properties of LDL. Geng (2016) first proposed the maximum entropy-based SA-BFGS as a predictive model. It employs the maximum entropy model and KL divergence to learn the label distribution and BFGS to solve the optimization problem. Although grounded in probabilistic theory, this approach suffers from limited model capacity. To address this, subsequent research has incorporated additional modeling strategies, such as exploiting label correlations (Jia et al. 2018; Zheng, Jia, and Li 2018), learning label embeddings (Wang and Geng 2019b; Xu, Shang, and Shen 2019), or modeling label ranking relationships via advanced algorithms like LDL-LRR (Jia et al. 2023) and LDL-DPA (Jia et al. 2024). Other approaches have focused on feature extraction, metric learning, and representation learning to further improve the generalization performance of LDL models. Despite the success of these approaches, most existing methods implicitly assume that the label distribution is generated from a single underlying labeling pattern and thus overlook annotator heterogeneity.

Our work is also related to DPMM. The Dirichlet process (DP) is a stochastic process defined over a space of distributions or random measures. It can be viewed as an infinite-dimensional generalization of the finite-dimensional Dirichlet distribution. Later, Sethuraman (1994) introduced a constructive representation known as the stick-breaking construction. When a group of observations is assumed to be generated from a mixture distribution with parameters drawn from a Dirichlet process prior, the resulting model is

referred to as a DPMM. DPMM has found widespread applications in clustering, density estimation, and nonparametric regression (Ya et al. 2007; Wang and Dunson 2011; Hannah, Blei, and Powell 2011). Due to its ability to automatically infer the number of components without requiring it to be specified in advance, the DPMM is particularly well-suited for modeling heterogeneous labeling patterns in LDL. Nevertheless, how to effectively integrate DPMMs with LDL to improve the predictive accuracy of label distribution learning remains an open and challenging problem.

3 Method

3.1 Notations

Let $\mathcal{X} = \mathbb{R}^P$ be the input space and $\mathcal{Y} = \{y_1, y_2, \cdots, y_C\}$ be the label space with C candidate labels. Let $\boldsymbol{x} \in \mathcal{X}$ denote the sample variable. The particular i-th sample is denoted by $\boldsymbol{x}_i$. Let $y \in \mathcal{Y}$ denote the label variable, the particular j-th label value is denoted by y_j. In the settings of LDL, the description degree of y to $\boldsymbol{x}$ is denoted by $d_{\boldsymbol{x}}^{y}$, and the label distribution of $\boldsymbol{x}_i$ is denoted by $\boldsymbol{d}_i = \left[d_{\boldsymbol{x}_i}^{y_1}, d_{\boldsymbol{x}_i}^{y_2}, \cdots, d_{\boldsymbol{x}_i}^{y_C}\right]^\top$, where $d_{\boldsymbol{x}_i}^{y_j}$ is the label description degree that tells the relevance degree of y_j to $\boldsymbol{x}_i$ and satisfies $d_{\boldsymbol{x}_i}^{y_j} \geqslant 0$ and $\sum_{j=1}^{C} d_{\boldsymbol{x}_i}^{y_j} = 1$. We represent an LDL training set with N samples as $\mathcal{Q} = \{(\boldsymbol{x}_1, \boldsymbol{d}_1), (\boldsymbol{x}_2, \boldsymbol{d}_2), \cdots, (\boldsymbol{x}_N, \boldsymbol{d}_N)\}$, where $\boldsymbol{x}_i$ is the i-th training sample with the label distribution $\boldsymbol{d}_i$. Our goal is to learn the conditional probability distribution $p(\boldsymbol{d} \mid \boldsymbol{x})$ based on $\mathcal{Q}$.

3.2 Overall Framework

Here, we determine the distribution form of $\boldsymbol{d}$ conditioned on $\boldsymbol{x}$. We choose the Dirichlet distribution as the base distribution because it is a probability distribution in the real number field supported by a standard simplex, which satisfies our assumption of a positive simplex for label distribution. We employ DPMM to model $p(\boldsymbol{d} \mid \boldsymbol{x})$:

$$p(\boldsymbol{d} \mid \boldsymbol{x}) = \sum_{k=1}^{K} \pi_k(\boldsymbol{x}; \boldsymbol{\Gamma}_k)\ \mathrm{Dir}\left(\boldsymbol{d} \mid \boldsymbol{\alpha}_k(\boldsymbol{x}; \boldsymbol{\Theta}_k)\right), \quad (1)$$

where $\pi_k(\boldsymbol{x})$, $\boldsymbol{\alpha}_k(\boldsymbol{x})$ are related to $\boldsymbol{x}$ and contain learnable parameters. K is not a hyperparameter, do not specify it in advance, and it is adaptively determined by the Dirichlet process. $\boldsymbol{\pi}(\boldsymbol{x})$ outputs a K-dimensional positive realvalued vector with a sum of 1, and $\pi_k(x)$ is its k-th value. $\boldsymbol{\alpha}_k(\boldsymbol{x})$ outputs a c-dimensional positive real-valued vector with a sum of 1. K is the possible number of different annotators, that is, the number of potential patterns. $\mathrm{Dir}(\cdot)$ denotes Dirichlet distribution whose details are in the appendix.

In the following, we provide a detailed explanation of the variables involved.

$$\pi_k(\boldsymbol{x}; \boldsymbol{\Gamma}_k) = \frac{w_k \exp g_k(\boldsymbol{x}; \boldsymbol{\Gamma}_k)}{\sum_j w_j \exp g_k(\boldsymbol{x}; \boldsymbol{\Gamma}_k)}, \quad (2)$$

$$\begin{aligned} g_k(\boldsymbol{x}; \boldsymbol{\Gamma}_k) &= \boldsymbol{m}_k^\top \boldsymbol{x} + c_k = \boldsymbol{\Gamma}_k^\top \tilde{\boldsymbol{x}}, \\ \boldsymbol{\alpha}_k(\boldsymbol{x}; \boldsymbol{\Theta}_k) &= \mathrm{softplus}(\boldsymbol{W}_k^\top \boldsymbol{x} + \boldsymbol{b}_k) \\ &= \mathrm{softplus}(\boldsymbol{\Theta}_k^\top \tilde{\boldsymbol{x}}), \end{aligned} \quad (3)$$

where $\boldsymbol{\Gamma}_k = (\boldsymbol{m}_k; c_k)$, $\boldsymbol{\Theta}_k = \left[\boldsymbol{W}_k; \boldsymbol{b}_k^\top\right]$, $\tilde{\boldsymbol{x}} = [\boldsymbol{x}; 1]$. $g_k(\boldsymbol{x}; \boldsymbol{\Gamma}_k)$ serves as the gating function, enabling the weights of the mixture components to be modeled as functions of the input rather than as fixed constants. Gating mechanism eliminates the reliance on fixed global weights w_k, enabling the model to assign sample-specific importance to each mixture component. $\alpha_k(\boldsymbol{x}; \boldsymbol{\Theta}_k)$ contains a linear predictor that models the relationship between input features and label distributions.

The overall learning process is summarized in Algorithm 1.

Algorithm 1: LDL-DPM

Input: training set $\mathcal{Q} = \{(\boldsymbol{x}_n, \boldsymbol{d}_n)\}_{n=1}^{N}$, testing sample $\boldsymbol{x}^*$.
Parameter: concentration α, strength S, regularization parameter θ, γ, max iteration T.
Output: predicted label distribution $\boldsymbol{d}^*$ for $\boldsymbol{x}^*$.

```
Initialize v_k, w_k in Eqs. (4) and (5) and Θ_k, Γ_k;
for t = 1 to T do
    u_i ← sample slice variables in Eq. (6);
    Θ_k, Γ_k ← update component parameters in Eq. (7);
    ν_k, w_k ← update via adaptive variation of the number of components in Eqs. (8) and (9);
    z_i ← sample indicator variables in Eq. (12);
end for
d* ← predict label distribution for x* in Eq. (13);
return the label distribution d*.
```

3.3 Adaptive Determination of K

In this section, we will introduce how to adaptively determine the number of mixture model through the Dirichlet process. There are many ways to construct the Dirichlet process. We utilize the stick-breaking process and learn the model through slice sampling method (Walker 2007) whose details are in the appendix.

For each component of DPMM, there is a weight w_k, which is generated by the stick-breaking process. The stick-breaking process is as follows:

$$\nu_k \sim \mathrm{Beta}(1, \alpha), \quad (4)$$

$$w_k = v_k \prod_{j<k} (1 - v_j), \quad (5)$$

where v_k is an intermediate variable used to generate w_k, generated by the Beta distribution. $\mathrm{Beta}(\cdot)$ denotes Beta Distribution.

The DPMM is an infinite model, but we cannot sample in infinity. According to the slice sampling method, we assign a slice variable u_i to each sample to convert infinite candidate components into finite ones, and also assign an indicator variable z_i to each sample to indicate which component the sample is assigned to. Below is the full slice sampling procedure for LDL, which is repeated for T iterations.

i. Generate slice variable u_i. The slice variable u_i is sampled as follows:

$$u_i \sim \mathrm{Uni}\,(0, w_{z_i})\,, \tag{6}$$

where $\mathrm{Uni}\,(\cdot)$ denotes the uniform distribution. The slice variable u_i is sampled from the uniform distribution over the weight w_{z_i} of the current component to which it belongs. This ensures that, during the resampling of z_i, sample i always has at least one valid candidate component, which is the component it currently occupies.

ii. Update of component parameters $\boldsymbol{\Theta}_k$, $\boldsymbol{\Gamma}_k$. The prediction model is defined in Eq. (1). In this section, we will explain in detail how the variables involved are optimized. We formulate the optimization objective as the maximization of the negative log-likelihood, augmented with regularization terms to mitigate overfitting. The complete objective function is defined as follows:

$$\begin{aligned}\underset{\boldsymbol{\Gamma},\,\boldsymbol{\Theta}}{\arg\min}\; &-\sum_{i=1}^{N}\log\left[\sum_{k=1}^{K}\pi_k(\boldsymbol{x}_i;\boldsymbol{\Gamma}_k)\,\mathrm{Dir}(\boldsymbol{d}_i \mid \boldsymbol{\alpha}_k(\boldsymbol{x}_i;\boldsymbol{\Theta}_k))\right] \\ &+\theta\sum_{k=1}^{K}\|\boldsymbol{\Theta}_k\|_F^2+\gamma\sum_{k=1}^{K}\|\boldsymbol{\Gamma}_k\|_F^2\,.\end{aligned} \tag{7}$$

The output $\boldsymbol{\alpha}_k\,(\boldsymbol{x}_i;\boldsymbol{\Theta}_k)$ is first normalized and then scaled by a fixed strength parameter S. The choice of S is critical to model training, as it controls the overall concentration of the resulting Dirichlet distribution. Besides, the parameters $\boldsymbol{\Theta}_k$ are initialized from a multivariate normal distribution with zero mean and a standard deviation of 0.1, while $\boldsymbol{\Gamma}_k$ is initialized as a zero vector. We employ the Adam optimizer (Kingma and Ba 2017) to optimize Eq. (7).

iii. Adaptive learning of the number of components. We begin by resampling v_k and w_k, conditioned on the current sample's affiliation with each component.

$$v_k \sim \mathrm{Beta}\left(1+n_k,\,\alpha+\sum_{j>k}n_j\right), \tag{8}$$

where $n_k=\sum_i \mathbf{1}\{z_i=k\}$. We subsequently update w_k using Eq. (5) based on the newly sampled v_k.

To control the dynamic variation in the number of components K, we determine whether to introduce new components based on the following criteria. The number of components is incrementally increased until these conditions are no longer satisfied,

$$\sum_{k=1}^{K} w_k < 1-\min\,(u_i). \tag{9}$$

To prevent excessive component generation and control model complexity, we restrict the algorithm to generate at most one new component per iteration. When the model decides to introduce a new component, the associated variable v is sampled based on Eq. (4). Subsequently, the complete weight w is updated via the stick-breaking construction defined in Eq. (5).

iv. Sample of the indicator variables z_i. Only components whose weights w_i exceed the slice variable u_i of the sample are considered candidate components. Accordingly, we first construct a set of feasible components $\mathcal{F}_i$,

$$\mathcal{F}_i=\{k|w_k>u_i\}\,. \tag{10}$$

Subsequently, the relative probabilities within the feasible component set are normalized, and used to sample a new value of z_i. Specifically, the log-probability weight $\log\rho_{ik}$ for each feasible component is first computed, followed by normalization to obtain the posterior distribution, from which the updated assignment is sampled,

$$\log\rho_{ik}=g_k(\boldsymbol{x}_i)+\log\mathrm{Dir}\,(\boldsymbol{d}_i\mid\boldsymbol{\alpha}_k(\boldsymbol{x}_i))\,,\quad (k\in\mathcal{F}_i) \tag{11}$$

$$z_i\sim\mathrm{Cat}\left(\frac{\exp(\log\rho_{ik})}{\sum_{j\in\mathcal{F}_i}\exp(\log\rho_{ij})}\right), \tag{12}$$

where $\mathrm{Cat}\,(\cdot)$ denotes categorical distribution. A new round of z_i assignment is performed, determining the active components according to Eq. (12). Throughout the iterative process, components that consistently attract more samples will accumulate higher weights w.

3.4 Predictive Process

After training, the learned model parameters are used to predict the label distribution $\boldsymbol{d}^\star$ for a test sample $\boldsymbol{x}^\star$ according to Eq. (1). In the prediction phase, each component yields a deterministic output by taking its expected value. The final predicted label distribution is then formed by a weighted average of these expected values, using the output of the gating function as the weights. The detailed prediction procedure is as follows:

$$\begin{aligned}\hat{\boldsymbol{d}}_k(\boldsymbol{x}^\star)&=\mathbb{E}_{\mathrm{Dir}(\boldsymbol{\alpha}_k(\boldsymbol{x}^\star))}[\boldsymbol{d}]=\frac{\boldsymbol{\alpha}_k(\boldsymbol{x}^\star)}{\sum_{j=1}^{C}\alpha_{k,j}(\boldsymbol{x}^\star)},\\ \pi_k(\boldsymbol{x}^\star)&=\frac{w_k\exp\left(g_k(\boldsymbol{x}^\star)\right)}{\sum_{j=1}^{K}w_j\exp\left(g_j(\boldsymbol{x}^\star)\right)},\\ \boldsymbol{d}^\star&=\sum_{k=1}^{K}\pi_k(\boldsymbol{x}^\star)\cdot\hat{\boldsymbol{d}}_k(\boldsymbol{x}^\star).\end{aligned} \tag{13}$$

3.5 Complexity Analysis

For a dataset with sample size N, feature dimension P, and label dimension C, the overall training complexity with K components and T iterations is $\mathcal{O}(TNK(P+C))$. This complexity grows linearly with respect to both the sample size and label dimensionality. Therefore, the proposed method is scalable to large-scale datasets. The prediction cost is negligible, and the memory footprint remains low.

4 Experiments

4.1 Experimental Configuration

Dataset. We evaluate our method on multiple heterogeneous label distribution learning datasets in real-world, including SBU_3DFE and Movie (Geng 2016), Emotion6 (Yang, Sun, and Sun 2017), fbp5500 (Liang et al. 2018), M^{2}B (Nguyen et al. 2012), and SCUT-FBP (Xie et al. 2015). These datasets are all associated with subjective perception tasks, such as emotion recognition and aesthetic evaluation. A summary of dataset statistics is provided in Table 1.

No	Dataset	Samples	Features	Labels
1	SBU_3DFE	2500	243	6
2	Movie	7755	1869	5
3	Emotion6	1980	168	7
4	fbp5500	5500	512	5
5	M^{2}B	1240	250	5
6	SCUT-FBP	1500	300	5

Table 1: Datasets statistics.

Evaluation metrics. We adopt six evaluation metrics recommended by (Geng 2016), including Chebyshev distance (Cheby.), Clark distance (Clark), Canberra metric (Can.), Kullback–Leibler divergence (KL), cosine similarity (Cosine), and intersection similarity (Int.). Among them, the first four are distance-based metrics, where lower values indicate better performance, while the last two are similarity-based metrics, where higher values are preferred. $\uparrow$ ($\downarrow$) indicates "the higher (lower) the better". The above metrics[2] can be summarized in Table 2.

Name	Formula
Cheby. $\downarrow$	$Dis_1(\boldsymbol{a},\boldsymbol{b}) = \max_j \lvert\boldsymbol{a}_j - \boldsymbol{b}_j\rvert$
Clark $\downarrow$	$Dis_2(\boldsymbol{a},\boldsymbol{b}) = \sqrt{\sum_{j=1}^{c}\left(\frac{\boldsymbol{a}_j-\boldsymbol{b}_j}{\boldsymbol{a}_j+\boldsymbol{b}_j}\right)^2}$
Can. $\downarrow$	$Dis_3(\boldsymbol{a},\boldsymbol{b}) = \sum_{j=1}^{c}\frac{\lvert\boldsymbol{a}_j-\boldsymbol{b}_j\rvert}{\boldsymbol{a}_j+\boldsymbol{b}_j}$
KL $\downarrow$	$Dis_4(\boldsymbol{a},\boldsymbol{b}) = \sum_{j=1}^{c}\boldsymbol{a}_j \ln\frac{\boldsymbol{a}_j}{\boldsymbol{b}_j}$
Cosine $\uparrow$	$Sim_1(\boldsymbol{a},\boldsymbol{b}) = \frac{\sum_{j=1}^{c}\boldsymbol{a}_j\boldsymbol{b}_j}{\sqrt{\sum_{j=1}^{c}\boldsymbol{a}_j^2}\sqrt{\sum_{j=1}^{c}\boldsymbol{b}_j^2}}$
Int. $\uparrow$	$Sim_2(\boldsymbol{a},\boldsymbol{b}) = \sum_{j=1}^{c}\min(\boldsymbol{a}_j,\boldsymbol{b}_j)$

Table 2: Summary of the metrics.

Comparison methods. We compare our method LDL-DPM against five methods, PT-Bayes (Geng and Hou 2015), CPNN (Geng, Yin, and Zhou 2013), SA-BFGS (Geng 2016), LDL-LRR (Jia et al. 2023), LDL-DPA (Jia et al. 2024). The hyperparameter configuration for each method follows their respective literature. Specifically, for SA-BFGS, we follow the same settings with (Geng 2016). For LDL-LRR, λ and β are selected from $10^{\{-6,-5,\cdots,-2,-1\}}$ and $10^{\{-3,-2,\cdots,1,2\}}$, respectively. For LDL-DPA, λ_1, λ_2, and λ_3 are selected from $10^{\{-9,-8,\cdots,-1,0\}}$, $10^{\{-7,-6,\cdots,-3,-2\}}$, and $10^{\{-4,-3,\cdots,0,1\}}$, respectively. The maximum iteration is set to 400 and the initialization of other variables is all zero. For all methods in the deep regime, the learning rate is chosen from $\{1,\ 2,\ 5\}\times 10^{\{-4,-3,-2\}}$, and the selection of the number of epochs is nested into a 10-fold cross validation. Finally, all method are examined on 6 datasets with 10-fold cross-validation, and mean performance and the standard deviation are reported.

4.2 Predictive Experiment

Methodology. For LDL-DPM, α, S, θ and γ are selected from $\{1,2,5,8,10\}$, $\{1,5,10,15,30\}$, $10^{\{-6,-5,-4,-3,-2,-1\}}$ and $10^{\{-6,-5,-4,-3,-2,-1\}}$, respectively. The default value of T is 20. We first tune the parameters of each method via 10-fold cross-validation, and then evaluate each method with the best parameters on six datasets, also using 10-fold cross-validation. Besides, to accelerate the convergence, we utilize min-max normalization to preprocess the feature matrices of all datasets. The mean performance and the standard deviation are reported. Additional experimental details[3], such as computing infrastructure and final hyperparameter choices, are provided in the appendix.

Performance. The results of the predictive experiment are shown in Table 3. 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, where •/○ indicates whether LDL-DPM is statistically superior/inferior to the comparing method and there is no significant if neither • nor ○ is shown. We also presented the performance rankings of various methods on different datasets and metrics. The best performance is highlighted by boldface. Overall, our method achieves significant advantages. From the table, we can see that our method achieved the best performance in 77.78% of cases. These experimental results are sufficient to demonstrate that our method can achieve good performance on these subjective datasets, which also proves the effectiveness of our method.

Visualization. To qualitatively demonstrate the effectiveness of our proposed method, we provide visualizations of predicted distributions for real samples selected from the SBU_3DFE datasets shown in Figure 3. Despite the discrete label space, in the field of LDL, the label distribution is intentionally plotted as a curve, to distinguish it from the logical labels.

4.3 Parameter Sensitivity

Here, we demonstrate how the regularization parameters θ and γ, the concentration parameter α, and the strength pa-

[2]Note that these metrics are not as intuitive as accuracy or error rate, i.e., small changes can mean large performance differences.

[3]Computer resources have a negligible effect on both the experimental results and the main claims of this paper.

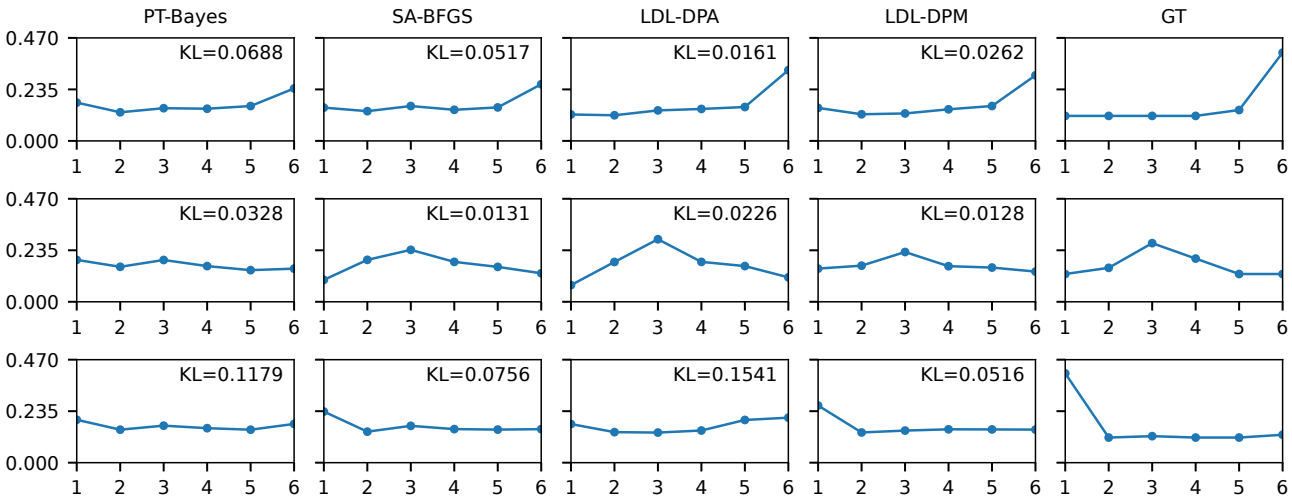


Figure 3: LDL curves for three randomly selected test samples (rows) under five methods (columns) from the SBU_3DFE dataset. Each panel also reports the KL divergence to the ground truth (GT); a smaller value indicates a closer match. Visually, LDL-DPM stays nearer to GT than the other baselines.

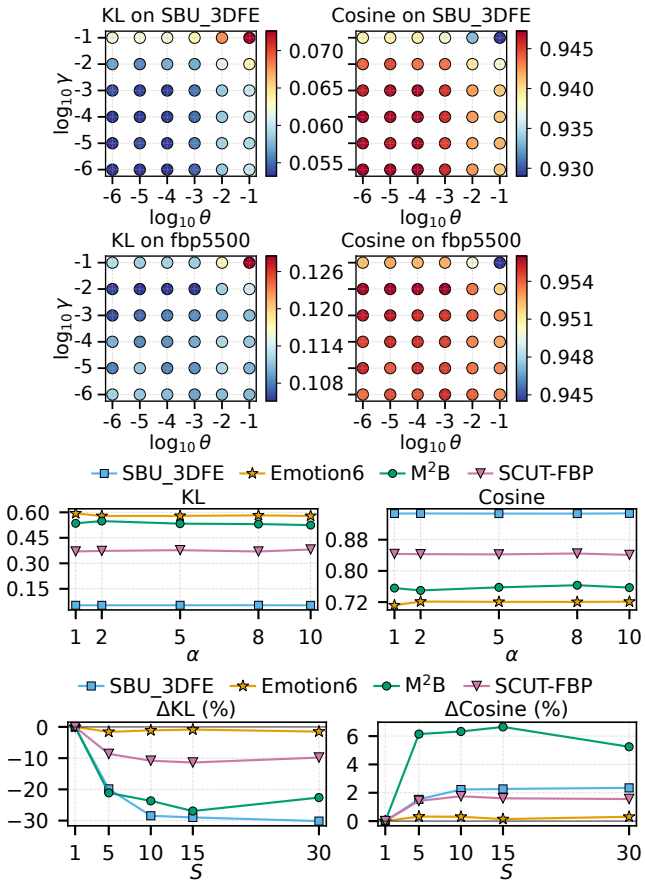


Figure 4: Parameter sensitivity of LDL-DPM. Rows 1-2: grid search over (θ, γ) on SBU_3DFE and fbp5500 (left: KL ↓, right: Cosine ↑). Row 3: sensitivity to the DP concentration $\alpha \in \{1, 2, 5, 8, 10\}$. Row 4: sensitivity to the Dirichlet strength $S \in \{1, 5, 10, 15, 30\}$ (shown as relative changes).

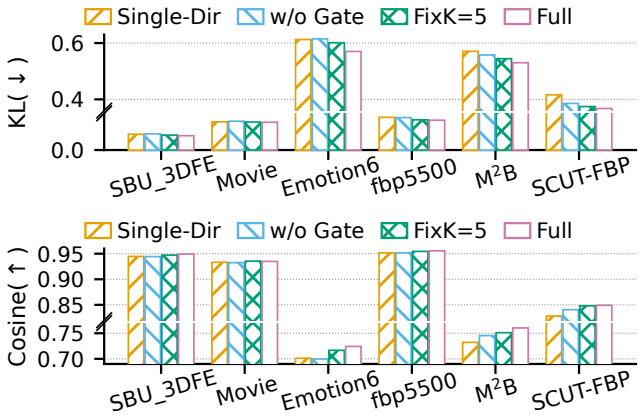


Figure 5: Ablation results of four variants. Four variants are compared: *Single-Dir* (mixture removed, single Dirichlet), *w/o Gate* (gating disabled), *FixK=5* (component number fixed to 5 instead of being learned), and the complete model *Full*.

rameter S affect performance. Results are shown in Figure 4. The first two rows of Figure 4 report the sensitivity of θ and γ on SBU_3DFE and fbp5500. Observing the color trends within each row, we find that performance improves as θ and γ increase from very small values, then plateaus or slightly degrades. This behavior is expected: as regularization coefficients, moderate θ and γ penalize overly complex models and thus enhance performance, whereas excessively large values may over-regularize and hurt performance. The last two rows of Figure 4 present the sensitivity of α and S across all six datasets. Changes in α lead to only minor performance variation, indicating that the Dirichlet process is fairly robust to its concentration parameter. In contrast, increasing S generally raises performance at first and then

Dataset	Method	Cheby.↓	Clark↓	Can.↓	KL↓	Cosine↑	Int.↑
SBU_3DFE	PT-Bayes	(6) .1290±.004•	(6) .4034±.009•	(6) .8634±.020•	(6) .0775±.003•	(6) .9244±.003•	(6) .8447±.003•
	CPNN	(4) .1123±.002•	(4) .3555±.007•	(4) .7512±.017•	(4) .0599±.002•	(4) .9408±.002•	(4) .8652±.003•
	SA-BFGS	(5) .1144±.003•	(5) .3691±.007•	(5) .7794±.016•	(5) .0623±.002•	(5) .9389±.002•	(5) .8606±.003•
	LDL-LRR	(2) .1046±.002•	(2) .3510±.007•	(2) .7322±.015•	(2) .0546±.002•	(2) .9466±.002•	(2) .8702±.002•
	LDL_DPA	(3) .1048±.002•	(3) .3523±.007•	(3) .7350±.015•	(3) .0550±.002•	(3) .9462±.002•	(3) .8699±.002•
	LDL-DPM	(1) **.1026**±.002	(1) **.3224**±.005	(1) **.6798**±.014	(1) **.0510**±.002	(1) **.9492**±.002	(1) **.8778**±.002
Movie	PT-Bayes	(6) .8057±.009•	(6) 2.0263±.015•	(6) 4.4489±.040•	(6) 8.5468±.514•	(6) .2805±.007•	(6) .1768±.007•
	CPNN	(5) .1624±.004•	(5) .6968±.015•	(5) 1.3605±.031•	(5) .2178±.012•	(5) .8744±.006•	(5) .7712±.006•
	SA-BFGS	(4) .1277±.002•	(4) .5579±.011•	(4) 1.0745±.024•	(4) .1224±.006•	(4) .9219±.003•	(4) .8205±.004•
	LDL-LRR	(3) .1192±.002•	(3) .5231±.009	(3) 1.0044±.020	(3) .1035±.005•	(3) .9318±.003•	(3) .8324±.004•
	LDL_DPA	(2) .1169±.002	(1) **.5170**±.007	(1) **.9906**±.015	(2) .0995±.004	(2) .9344±.002	(2) .8350±.003
	LDL-DPM	(1) **.1154**±.001	(2) .5222±.005	(2) .9968±.012	(1) **.0986**±.003	(1) **.9349**±.002	(1) **.8355**±.002
Emotion6	PT-Bayes	(5) .3283±.015•	(5) 1.7606±.030•	(5) 4.0203±.091•	(5) .7455±.057•	(5) .6603±.021•	(5) .5565±.014•
	CPNN	(6) .5076±.012•	(6) 2.1367±.019•	(6) 5.2589±.063•	(6) 2.5044±.087•	(6) .4404±.017•	(6) .3710±.010•
	SA-BFGS	(3) .3036±.008	(4) 1.6760±.020	(4) 3.7676±.050	(4) .5837±.015	(4) .7204±.007	(3) .5904±.006
	LDL-LRR	(1) **.3024**±.009	(3) 1.6750±.023	(3) 3.7637±.059	(3) .5818±.018	(2) .7210±.009	(1) **.5918**±.007○
	LDL_DPA	(2) .3034±.008	(2) 1.6667±.025	(2) 3.7396±.064	(2) .5784±.017	(3) .7205±.009	(2) .5905±.008
	LDL-DPM	(4) .3061±.010	(1) **1.6561**±.026	(1) **3.7114**±.067	(1) **.5702**±.022	(1) **.7243**±.009	(4) .5851±.007
fbp5500	PT-Bayes	(6) .6437±.013•	(6) 1.6606±.006•	(6) 3.0807±.016•	(6) 12.0672±.354•	(6) .4852±.019•	(6) .3532±.013•
	CPNN	(5) .1533±.004•	(5) 1.3349±.007•	(5) 2.3358±.022•	(5) .1445±.007•	(5) .9395±.003•	(5) .8302±.004•
	SA-BFGS	(4) .1365±.003	(4) 1.2932±.007•	(4) 2.2115±.023•	(4) .1082±.004	(4) .9536±.001•	(4) .8497±.003
	LDL-LRR	(3) .1347±.003	(3) 1.2860±.007•	(3) 2.1916±.024•	(2) .1052±.004	(3) .9548±.002	(2) .8517±.003
	LDL_DPA	(2) .1345±.003	(2) 1.2839±.006•	(2) 2.1862±.021	(1) **.1045**±.003	(2) .9553±.001	(1) **.8519**±.003
	LDL-DPM	(1) **.1343**±.003	(1) **1.2776**±.007	(1) **2.1773**±.023	(3) .1059±.002	(1) **.9556**±.001	(3) .8505±.003
M^2B	PT-Bayes	(6) .5290±.054•	(6) 1.7288±.066•	(6) 3.5677±.183•	(6) 1.6941±.548•	(6) .5414±.067•	(6) .4587±.054•
	CPNN	(2) .3785±.020	(5) 1.2934±.052•	(5) 2.5347±.120•	(2) .6407±.094•	(2) .7338±.025•	(4) .6089±.021•
	SA-BFGS	(5) .3865±.019•	(4) 1.2269±.025	(3) 2.3621±.069	(5) .7071±.076•	(5) .7189±.023•	(5) .6049±.019•
	LDL-LRR	(4) .3822±.018•	(2) 1.1953±.023○	(2) 2.3000±.063○	(4) .6795±.075•	(4) .7241±.022•	(3) .6099±.018•
	LDL_DPA	(3) .3814±.017•	(1) **1.1714**±.021○	(1) **2.2606**±.051○	(3) .6749±.077•	(3) .7247±.022•	(2) .6110±.018•
	LDL-DPM	(1) **.3604**±.024	(3) 1.2214±.030	(4) 2.4232±.073	(1) **.5302**±.077	(1) **.7604**±.029	(1) **.6315**±.024
SCUT-FBP	PT-Bayes	(6) .9278±.009•	(6) 2.1823±.010•	(6) 4.8523±.030•	(6) 26.3679±3.185•	(6) .1021±.011•	(6) .0706±.008•
	CPNN	(2) .2712±.011•	(2) 1.4070±.020•	(2) 2.6418±.052•	(2) .5784± .100•	(2) .8153±.012•	(2) .6785±.011•
	SA-BFGS	(5) .4188±.012•	(5) 1.5685±.016•	(5) 3.1314±.052•	(5) 1.7446± .318•	(5) .6034±.016•	(5) .4743±.014•
	LDL-LRR	(4) .4095±.012•	(4) 1.5509±.015•	(4) 3.0807±.049•	(4) 1.3460± .220•	(4) .6180±.016•	(4) .4832±.013•
	LDL_DPA	(3) .4038±.010•	(3) 1.5417±.014•	(3) 3.0570±.047•	(3) 1.1534± .161•	(3) .6252±.014•	(3) .4871±.011•
	LDL-DPM	(1) **.2555**±.008	(1) **1.3810**±.015	(1) **2.5724**±.041	(1) **.3676**± .025	(1) **.8443**±.012	(1) **.6912**±.010

Table 3: Results on six LDL benchmark datasets (mean±std; rank in parentheses; first four metrics ↓ smaller-better, last two ↑ larger-better. •: LDL-DPM significantly better; ○: significantly worse (p=0.05).

saturates. Moreover, the optimal S differs across datasets, underscoring its role in achieving a well-converged Dirichlet distribution.

4.4 Ablation Study

We demonstrate the effectiveness of each module in our method through an ablation study. To verify the usefulness of the mixture model, we remove all mixture components and reduce the full LDL-DPM to a single Dirichlet distribution, denoted as *Single-Dir*. To assess the gating mechanism, we disable it, yielding *w/o Gate*. To examine the benefit of adaptively determining the number of components, we fix the component number to an arbitrary value of 5, resulting in *FixK=5*. For fairness, we re-tune the hyperparameters of each modified variant to achieve its best performance. As shown in Figure 5, every degraded variant performs worse than the complete model (*Full*), indicating that each module contributes to the overall performance.

5 Conclusion

This paper proposes LDL-DPM. We revisit LDL from the perspective of heterogeneous label generation. Our main contributions are threefold: (1) We analyze the inherent heterogeneity in the label distribution generation process; (2) We propose LDL-DPM to effectively address this heterogeneity through mixture modeling and adaptive component selection; (3) We design a feature-conditioned gating module that enhances the representational capacity of DPMM-based LDL models. Extensive experiments demonstrate that LDL-DPM achieves competitive performance across six evaluation metrics. Parameter sensitivity analyses the choice of hyperparameters and ablation studies further confirm the effectiveness of each proposed module.

Acknowledgments

This research is partially supported by the National Natural Science Foundation of China (62576166, 62176123, 62476130), and the Natural Science Foundation of Jiangsu Province (BK20242045).

References

Gao, B.-B.; Xing, C.; Xie, C.-W.; Wu, J.; and Geng, X. 2017. Deep Label Distribution Learning with Label Ambiguity. *IEEE Transactions on Image Processing*, 26(6): 2825–2838.

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.; Yin, C.; and Zhou, Z.-H. 2013. Facial Age Estimation by Learning from Label Distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10): 2401–2412.

Hannah, L. A.; Blei, D. M.; and Powell, W. B. 2011. Dirichlet Process Mixtures of Generalized Linear Models. *Journal of Machine Learning Research*, 12(54): 1923–1953.

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.; Qin, T.; Lu, Y.; and Li, W. 2024. Adaptive Weighted Ranking-Oriented Label Distribution Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(8): 11302–11316.

Jia, X.; Shen, X.; Li, W.; Lu, Y.; and Zhu, J. 2023. 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 Conference on Computer Vision and Pattern Recognition*, 9841–9850.

Kingma, D. P.; and Ba, J. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980.

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.

Liang, L.; Lin, L.; Jin, L.; Xie, D.; and Li, M. 2018. SCUT-FBP5500: A Diverse Benchmark Dataset for Multi-Paradigm Facial Beauty Prediction. In *International Conference on Pattern Recognition*, 1598–1603.

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

Lu, Y.; and Jia, X. 2022. Predicting Label Distribution from Multi-Label Ranking. In *Advances in Neural Information Processing Systems*, 36931–36943.

Lu, Y.; Li, W.; Li, H.; and Jia, X. 2023. Predicting Label Distribution From Tie-Allowed Multi-Label Ranking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12): 15364–15379.

Nguyen, T. V.; Liu, S.; Ni, B.; Tan, J.; Rui, Y.; and Yan, S. 2012. Sense Beauty via Face, Dressing, and/or Voice. In *ACM International Conference on Multimedia*, 239–248.

Rupprecht, C.; Laina, I.; DiPietro, R.; Baust, M.; Tombari, F.; Navab, N.; and Hager, G. D. 2017. Learning in an Uncertain World: Representing Ambiguity Through Multiple Hypotheses. In *IEEE International Conference on Computer Vision*, 3611–3620.

Sethuraman, J. 1994. A Constructive Definition of Dirichlet Priors. *Statistica Sinica*, 4(2): 639–650.

Walker, S. G. 2007. Sampling the Dirichlet Mixture Model with Slices. *Communications in Statistics-Simulation and Computation*, 36(1): 45–54.

Wang, J.; and Geng, X. 2019a. Theoretical Analysis of Label Distribution Learning. In *AAAI Conference on Artificial Intelligence*, 5256–5263.

Wang, K.; and Geng, X. 2019b. Discrete Binary Coding based Label Distribution Learning. In *International Joint Conference on Artificial Intelligence*, 3733–3739.

Wang, L.; and Dunson, D. B. 2011. Fast Bayesian Inference in Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics*, 20(1): 196–216.

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 Conferenceon on Computer Vision*, 379–395.

Xie, D.; Liang, L.; Jin, L.; Xu, J.; and Li, M. 2015. SCUT-FBP: A Benchmark Dataset for Facial Beauty Perception. In *IEEE International Conference on Systems, Man, and Cybernetics*, 1821–1826.

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

Ya, X.; Xuejun, L.; Carin, L.; and Krishnapuram, B. 2007. Multi-Task Learning for Classification with Dirichlet Process Priors. *Journal of Machine Learning Research*, 8: 35–63.

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. 2014. A Review on Multi-Label Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8): 1819–1837.

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