Adaptive-Grained Label Distribution Learning

Yunan Lu[1], Weiwei Li[2], Dun Liu[3], Huaxiong Li[4], Xiuyi Jia[1]*

[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
[2] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
[3] School of Economics and Management, Southwest Jiaotong University, Chengdu, China
[4] Department of Control Science and Intelligence Engineering, Nanjing University, Nanjing, China
{luyn, jiaxy}@njust.edu.cn

Abstract

Label polysemy, where an instance can be associated with multiple labels, is common in real-world tasks. LDL (label distribution learning) is an effective learning paradigm for handling label polysemy, where each instance is associated with a label distribution. Although numerous LDL algorithms have been proposed and achieved satisfactory performance on most existing datasets, they are typically trained directly on the collected label distributions which often lack quality guarantees in real-world tasks due to annotator subjectivity and algorithm assumptions. Consequently, direct learning from such uncertain label distributions can lead to unpredictable generalization performance. To address this problem, we propose an adaptive-grained label distribution learning framework whose main idea is to extract relatively reliable supervision information from unreliable label distributions, and thus the label distribution learning task can be decomposed into three subtasks: coarsening label distributions, learning coarse-grained labels and refining coarse-grained labels. In this framework, we design an adaptive label coarsening algorithm to extract an optimal coarsen-grained labels and a label refining function to enhance the coarse-grained label into the final label distributions. Finally, we conduct extensive experiments on real-world datasets to demonstrate the advantages of our proposal.

1 Introduction

Label polysemy, where an instance is relevant to multiple labels, is ubiquitous in the real world. The most direct method for addressing label polysemy is to assign a vector of logical values (either 0 or 1) to each instance, where each logical value indicates whether the instance is relevant to the corresponding label or not; the process of learning a mapping from features to a vector of logical values is called MLL (multi-label learning (Tsoumakas and Katakis 2006)). However, MLL only gives which labels can describe the instance but cannot answer a question with more polysemy, i.e., how much does each label describe the instance. Therefore, label distribution (Geng 2016) is introduced to answer this question. Label distribution is a real-valued vector analogous to a probability distribution, where each element, called LDD (label description degree), represents the degree to which the corresponding label describes the instance. The process of learning a mapping from features to a label distribution is called LDL (label distribution learning). So far, LDL has been applied in many practical tasks, such as sentiment analysis (Jia et al. 2019b; Machajdik and Hanbury 2010; Yang, She, and Sun 2017) and facial age estimation (Gao et al. 2018; Geng, Smith-Miles, and Zhou 2013; Wen et al. 2020).

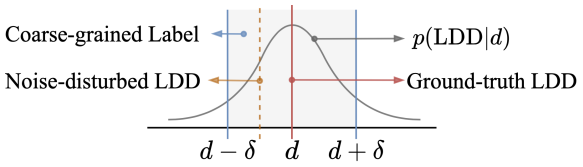


Figure 1: An example of why CGL mitigates noise.

Until now, a large number of algorithms have been proposed to handle the LDL task, such as (Geng 2016; Jia et al. 2023; Liu et al. 2021; Wang and Geng 2023). We refer the readers to Section 2 for more details. Although these algorithms have achieved satisfactory performance on various existing datasets, they typically learn a predictive model directly from the collected label distributions. However, the quality of the collected label distributions is unguaranteed in real-world tasks. On the one hand, the label distributions annotated by humans are susceptible to various factors, such as the subjectivity and expertise of the annotators, as well as the incentives for the annotation task. On the other hand, the label distributions generated by label enhancement algorithms (Xu, Tao, and Geng 2018), which aims to recover label distributions from logical labels by mining potential information, are susceptible to the algorithm assumptions or data distributions. These factors collectively contribute to the challenge of ensuring the quality of collected label distributions. Consequently, direct learning such collected label distributions can yield unreliable predictive performance.

Therefore, in this paper, we propose an **A**daptive-**G**rained **L**abel **D**istribution **L**earning framework called AGLDL. The main idea is to adaptively extract the coarse-grained label (abbr. CGL) information from unreliable label distributions. Since CGL contains a range of potential fine-grained label distributions, it is possible that the noise-disturbed label distributions and the ground truth label distribution correspond to the same CGL. As shown in Figure 1, if the value

*Corresponding author

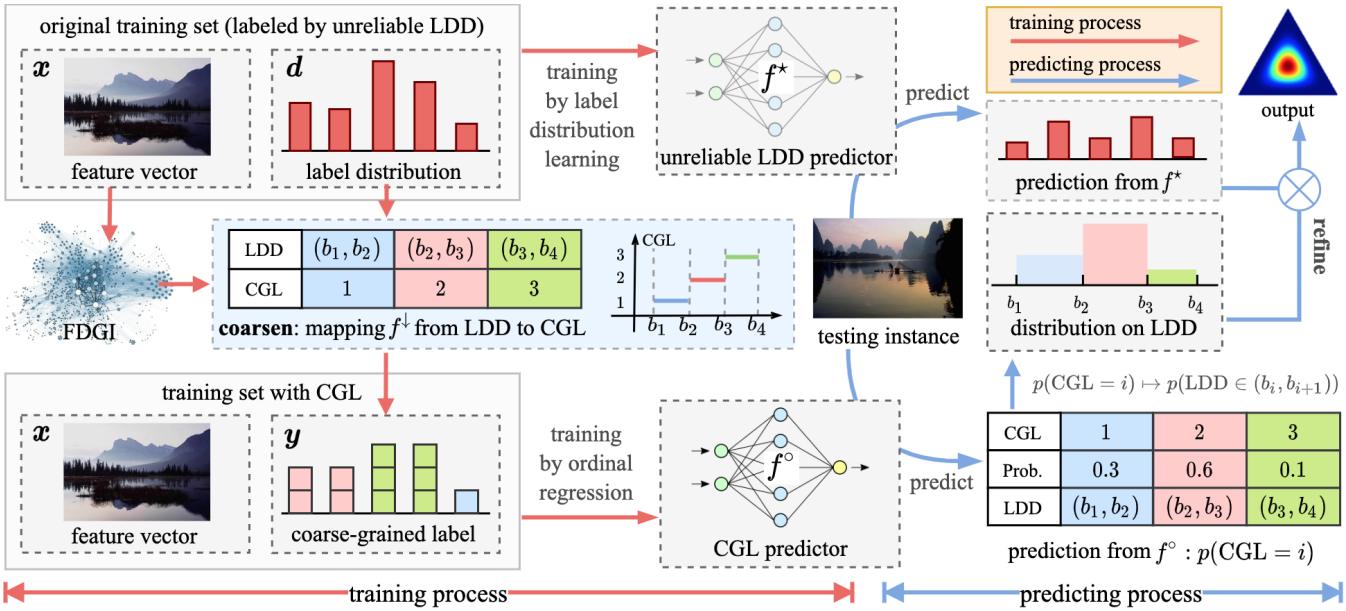


Figure 2: Overall diagram of AGLDL. In the training process, features are fed into two branches. The first branch trains an unreliable LDD predictor $f^\star$ using the collected (unreliable) label distribution through an off-the-shelf LDL algorithm; The second branch learns a partition of the set of training instances by FDGI, which transforming the LDD into ordinal discrete CGL; a CGL predictor f° can be trained on this CGL using any off-the-shelf classification algorithm. In the predicting process, the prediction from f° is mapped to a distribution of LDD bijectively, which is then refined by the prediction from $f^\star$.

of the ground-truth LDD is d and the CGL corresponds to $d-\delta < \text{LDD} < d+\delta$, where δ is a positive real number, then both the ground-truth LDD d and any LDD disturbed by noise with an amplitude smaller than δ will be mapped to the same CGL, which means that the noise with an amplitude smaller than δ can be mitigated. Therefore, learning on the coarse-grained labels to some extent can alleviate the impact of noise. Following the above idea, the LDL task can be decomposed into three sub-tasks: coarsening, learning and refining, whose workflow is shown in Figure 2. 1) In the coarsening process, the objective is to learn a set of adjacent intervals that partition the range of LDD; the ordinal number of the interval to which each LDD value belongs is then taken as its corresponding CGL value. To achieve this, we propose an adaptive label coarsening algorithm which is inspired by a basic idea from fuzzy rough feature selection, i.e., finding the smallest feature subset that maintains the fuzzy dependency (or fuzzy positive region) of the instance set. In the adaptive label coarsening algorithm, we design a metric called FDGI (fuzzy dependency gain reweighted by inclusion probability) based on the original fuzzy dependency metric, and aim to maximize FDGI in the coarsening process. 2) In the learning process, we employ an off-the-shelf classification algorithm to train a CGL predictor. 3) In the refining process, we first train an unreliable LDD predictor $f^\star$ directly on the collected (unreliable) label distributions using an off-the-shelf LDL algorithm. Then, within the intervals specified by CGL, we identify the label distribution that is closest to the output of $f^\star$, which serves as the fine-grained prediction of AGLDL. Finally, we validate AGLDL through extensive experiments on real-world datasets. The merits of AGLDL can be summarized as follows.

- *More stable performance*: Our framework is able to work stably in the environments where the quality of label distributions is uncertain or unreliable.
- *More informative predictions*: Our framework is able to output both coarse-grained and fine-grained predictions and their respective uncertainties.
- *More compatible architecture*: Our framework can seamlessly integrate with any off-the-shelf classification algorithm for learning CGL; it can also serve as a post-processor to enhance the existing LDL algorithms.

2 Related Work

Our work is mainly related to LDL. Existing works on LDL algorithms can be broadly divided into two topics, i.e., model representation and learning criterion.

The focus of model representation is how to represent the mapping from features to a label distribution. For example, Geng (2016) derive a maximum entropy model for LDL. AAkNN (Geng 2016), GMM-kLDL (Zhai and Dai 2019), and LDL-LCR (Xu et al. 2020) are the instance-based models which use the training instances to represent the mapping from the features to label distribution. These works focus on how to find the neighbors of an instance and calculate the similarities. LDLogitBoost (Xing, Geng, and Xue 2016), LDLFs (Shen et al. 2017), ENN-LDL (Zhai, Dai, and Shi 2018), StructRF (Chen et al. 2018), and BC-LDL (Wang and Geng 2018) extend the standard ensemble models for LDL.

The focus of learning criterion is how to design a loss function to improve the generalization ability of the LDL algorithm. The most popular idea is to mine the label relation underlying the label distribution and design regularization terms to improve the generalization performance (Jia et al. 2018, 2024, 2023, 2019b; Ren et al. 2019a,b; Wang and Geng 2023; Zhao and Zhou 2018; Zheng, Jia, and Li 2018). Besides, some works focus on designing loss functions for special scenarios (such as incomplete and unbalanced labels). For example, IncomLDL (Xu and Zhou 2017) complements the incomplete label distributions by the low rank assumption. WSLDL-MCSC (Jia et al. 2019a) is a weakly supervised LDL method by matrix completion. IncomLDL-LR (Zeng et al. 2019) is an incomplete LDL method with local reconstruction.

3 Preliminaries

3.1 Fuzzy Dependency

The main idea of fuzzy rough feature selection is to find a smallest subset of features that can maintain the fuzzy dependency (or fuzzy positive region) of the original feature set. Some concepts are introduced below.

Definition 1 (Triangular norm) *A function $\mathcal{T} : [0,1]^2 \to [0,1]$ is called a triangular norm if the following three conditions hold for any $u, v, w \in [0,1]$:*

1. *Exchangeability, i.e., $\mathcal{T}(u,v) = \mathcal{T}(u,v)$.*
2. *Associativity, i.e., $\mathcal{T}(u, \mathcal{T}(v,w)) = \mathcal{T}(\mathcal{T}(u,v), w)$.*
3. *Monotonicity, i.e., $u \le v, \mathcal{T}(w,u) \le \mathcal{T}(w,v)$.*

Definition 2 (Fuzzy similarity relation) *Given a finite nonempty set of objects U, a function $R : U^2 \to [0,1]$ is called a fuzzy similarity relation if the following two conditions hold for any $u, v \in U$: 1) Reflexibity, i.e., $R(u,u) = 1$. 2) Symmetry, i.e., $R(u,v) = R(v,u)$.*

Definition 3 (Fuzzy $\mathcal{T}$-equivalence relation) *Given a finite nonempty set of objects U, a fuzzy similarity relation R is called a fuzzy $\mathcal{T}$-equivalence relation if $\mathcal{T}$-transitivity is additionally imposed, i.e., $\mathcal{T}(R(u,v), R(v,w)) \le R(u,w)$ holds for any $u, v, w \in U$, where $\mathcal{T}$ is a triangular norm.*

Definition 4 (Fuzzy implicator) *A mapping $\mathcal{I} : [0,1]^2 \to [0,1]$ is called fuzzy implicator if it satisfies that $\mathcal{I}(0,0) = 1$, $\forall x \in [0,1], \mathcal{I}(1,x) = x$, and $\mathcal{I}$ is decreasing in its first component, and increasing in its second component.*

Definition 5 (Fuzzy dependency) *Given a finite nonempty set of objects U and a fuzzy subset V, and π is a partition on U. The fuzzy dependency γ_π^R are defined as*

$$\gamma_\pi^R = \frac{1}{|U|} \sum_{u \in U} \sup_{V \in \pi} \inf_{v \in U} \mathcal{I}\big(R(u,v), V(v)\big). \tag{1}$$

3.2 Problem Formulation

We cope with the training datasets that appear as pairs $\{(\boldsymbol{x}_i, \boldsymbol{d}_i)\}_{i=1}^N$, where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \cdots, x_{iC}]$ and $\boldsymbol{d}_i = [d_{i1}, d_{i2}, \cdots, d_{iM}]$ denote the i-th feature vector and the i-th label distribution (i.e., the vector of LDD), respectively. We denote the CGL vector corresponding to the label distribution $\boldsymbol{d}_i$ as $\boldsymbol{y}_i$, and its element y_{ij} is an ordinal number. $\boldsymbol{X} = [\boldsymbol{x}_1; \cdots; \boldsymbol{x}_N] \in \mathbb{R}^{N \times C}$, $\boldsymbol{D} = [\boldsymbol{d}_1; \cdots; \boldsymbol{d}_N] \in \mathbb{R}^{N \times M}$ and $\boldsymbol{Y} = [\boldsymbol{y}_1; \cdots; \boldsymbol{y}_N]$ denote the matrices of features, LDD and CGL, respectively. We denote the i-th column of $\boldsymbol{X}$ and $\boldsymbol{D}$ as $\boldsymbol{x}_{\cdot i}$ and $\boldsymbol{d}_{\cdot i}$, respectively, i.e., $\boldsymbol{x}_{\cdot i} = [x_{1i}, x_{2i}, \cdots, x_{Ni}]^\top$ and $\boldsymbol{d}_{\cdot i} = [d_{1i}, d_{2i}, \cdots, d_{Ni}]^\top$. Our goal is to learn a mapping from a feature vector to a label distribution according to training set $(\boldsymbol{X}, \boldsymbol{D})$.

4 Methodology

4.1 General Framework of AGLDL

Here we show the general framework of AGLDL in detail. The computational flow of AGLDL includes three phases.

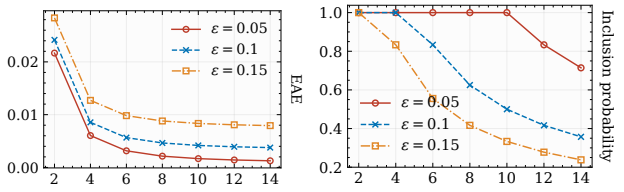
- The first phase is to obtain the CGL $\boldsymbol{Y}$ and multiple partitions of training set π by the label coarsening function $f^\downarrow$, i.e., $\boldsymbol{Y}, \pi \leftarrow f^\downarrow(\boldsymbol{X}, \boldsymbol{D})$; each CGL is a discrete value which corresponds to an interval of LDD; each partition is derived from the description degree of a certain label. The details will be illustrated in Section 4.2.
- The second phase is to train an unreliable LDD predictor and a CGL predictor with the training sets $(\boldsymbol{X}, \boldsymbol{Y})$ and $(\boldsymbol{X}, \boldsymbol{D})$, respectively:

$$\begin{aligned} f^\star &\leftarrow \arg\min_f \mathcal{L}_{\text{LDD}}([f(\boldsymbol{x}_i)]_{i=1}^N, \boldsymbol{D}), \\ f^\circ &\leftarrow \arg\min_f \mathcal{L}_{\text{CGL}}([f(\boldsymbol{x}_i)]_{i=1}^N, \boldsymbol{Y}), \end{aligned} \tag{2}$$

  where $\mathcal{L}_{\text{LDD}}(\cdot)$ and $\mathcal{L}_{\text{CGL}}(\cdot)$ are the loss functions of LDD predictor and CGL predictor, respectively. These two processes correspond to two existing learning paradigms (i.e., LDL tasks and classification tasks), so they can be implemented with the help of off-the-shelf techniques. $f^\star$ is a mapping from a vector of features to a label distribution; $f^\circ(\boldsymbol{x}) = \{f_1^\circ(\boldsymbol{x}), f_2^\circ(\boldsymbol{x}), \cdots, f_M^\circ(\boldsymbol{x})\}$ is a set of mappings from features to a discrete probability distribution whose components represent the probabilities that the instance belongs to the corresponding CGL.
- The third phase is to integrate the outputs of the CGL predictor and unstable LDD predictor by the label refining function $f^\uparrow$ to obtain the final label distribution, i.e., $\{\hat{\boldsymbol{d}}_i\}_{i=1}^{\hat{N}} \leftarrow \{f^\uparrow(f^\star(\hat{\boldsymbol{x}}_i), f^\circ(\hat{\boldsymbol{x}}_i))\}_{i=1}^{\hat{N}}$, and the details is illusrated in Section 4.3.

The algorithm of AGLDL is shown in Appendix. The above mechanism converts the unreliable LDD into reliable CGL. In this way, the CGL prediction can provide a reliable range for the LDD prediction. There are two questions of interest in this process: how well CGL approximates the true LDD and how probable is the LDD range specified by CGL to include the true LDD. We answer them by two theorems.

Theorem 1 *Given an unreliable LDD d, its corresponding true value $\tilde{d}$ is subject to a uniform distribution on the interval $(d - \epsilon^l, d + \epsilon^r)$, and d is mapped to the interval $(d - \delta^l, d + \delta^r)$ by a coarsening process. $\epsilon^l \in (0, d), \epsilon^r \in (0, 1-d), \delta^l \in (0, d), \delta^r \in (0, 1-d)$ are four independent real values. Then, the expected error of approximating*



(a) Theorem 1 (b) Theorem 2

Figure 3: Visualization of theorems. The horizontal axis denotes the number of identical-width sub-intervals.

the true LDD by any value $\hat{d}$ uniformly generated from the interval $(d-\delta^l, d+\delta^r)$ is given by:

$$\mathbb{E}[(\tilde{d}-\hat{d})^2] = 3^{-1}(\epsilon^l-\epsilon^r)^2 + 3^{-1}(\delta^l-\delta^r)^2 + 3^{-1}(\epsilon^l\epsilon^r+\delta^l\delta^r) - 2^{-1}(\epsilon^l-\epsilon^r)(\delta^l-\delta^r). \tag{3}$$

Theorem 1 quantifies how well CGL approximates the true LDD with the help of the EAE (expected approximation error) defined in (Lu and Jia 2022) and (Lu et al. 2023). The expected value $\mathbb{E}[(\tilde{d}-\hat{d})^2]$ can be easily obtained by the integration of $\tilde{d}$ and $\hat{d}$.

Theorem 2 *Given an unreliable LDD d, its corresponding true value $\tilde{d}$ is subject to a uniform distribution on the interval $(d-\epsilon^l, d+\epsilon^r)$, and d is mapped to the interval $(d-\delta^l, d+\delta^r)$ by a coarsening process. $\epsilon^l \in (0,d), \epsilon^r \in (0, 1-d), \delta^l \in (0,d), \delta^r \in (0, 1-d)$ are four independent real values. Then we can obtain the inclusion probability:*

$$p(\tilde{d} \in (d-\delta^l, d+\delta^r)) = \min\left\{\frac{\epsilon^l+\delta^r}{\epsilon^l+\epsilon^r}, \frac{\epsilon^r+\delta^l}{\epsilon^l+\epsilon^r}, \frac{\delta^l+\delta^r}{\epsilon^l+\epsilon^r}, 1\right\}. \tag{4}$$

Theorem 2 quantifies how probable is the LDD range specified by CGL to incorporate the true LDD, which can be easily measured by the proportion of the intersection of $(d-\delta^l, d+\delta^r)$ and the interval of $\tilde{d}$ in $(d-\delta^l, d+\delta^r)$. In Figure 3, we show how the granularity of CGL affects the expected approximation error and the inclusion probability. For simplicity, we in Figure 3 assume that $\delta^l = \delta^r$, $\epsilon^l = \epsilon^r$, and the interval $(0,1)$ is divided into K intervals of identical width, i.e., $\delta^l = \delta^r = (2K)^{-1}$. As can be seen from Figure 3(a), if K exceeds 6, then refining the granularity of CGL (i.e., increasing K) cannot significantly reduce the expected approximation error. Figure 3(b) shows that refining the granularity of CGL significantly reduces the inclusion probability if ϵ is large, i.e., the collected LDD is unreliable.

4.2 Coarsening Label Description Degree

In fuzzy rough feature selection, fuzzy dependency is an important metric to quantify the association strength between features and labels. Therefore, in the process of coarsening the label description degree, we desire the coarse-grained labels and features to have a large fuzzy dependency. Nevertheless, it should be noted that a coarser label can easily lead to a large value of fuzzy dependency. For example, if all instances belong to the same coarse-grained label (i.e., the elements of $\boldsymbol{Y}$ are identical), the fuzzy dependency will be 1. Obviously, this situation is meaningless since it leads to a large expected approximation error to the true LDD and provides an uninformative range of LDD, i.e., LDD$\in (0,1)$. Therefore, we design a metric called **F**uzzy **D**ependency **G**ain reweighted by **I**nclusion probability (abbr. FDGI) to strike a balance between the fuzzy dependency and expected approximation error. FDGI is composed of the fuzzy dependency reweighted by error probability and expected error of approximating true LDD.

Fuzzy Dependency Reweighted by Inclusion Probability Given a training set $(\boldsymbol{X}, \boldsymbol{D})$ and a CGL-based instance partition π, we first calculate the fuzzy dependency. According to Eq. (1), the fuzzy dependency relies on the fuzzy $\mathcal{T}$-equivalence relation and fuzzy similarity relation. We first calculate the fuzzy similarity relation. The fuzzy similarity relation between the i-the instance and the j-th instance based on the c-th feature can be defined as:

$$R_c^S(i,j) = 1 - |\max(\boldsymbol{x}_{\cdot c}) - \min(\boldsymbol{x}_{\cdot c})|^{-1}|x_{ic} - x_{jc}|. \tag{5}$$

Since it has been shown that the fuzzy intersection of all fuzzy similarity relations based on individual feature can preserve $\mathcal{T}$-transitivity (Wallace, Avrithis, and Kollias 2006), we derive the fuzzy $\mathcal{T}$-equivalence relation between the i-th instance and the j-th instance:

$$R^{\mathcal{T}}(i,j) = \bigcap\nolimits_{c\in[C]}\{R_c^S(i,j)\}, \quad [C] = \{1,2,\ldots,C\}. \tag{6}$$

Besides, we take the fuzzy intersection as $u, v \mapsto \min(u,v)$, and Lukasiewicz implicator $\mathcal{I}: u, v \mapsto \min\{1, 1-u+v\}$ is used as the fuzzy implicator in this paper. According to Theorem 2, it can be seen that the CGL of each instance probably does not contain the true LDD. Hence, we use Eq. (4) to reweight each instance, and then we can obtain the fuzzy dependency reweighted by inclusion probability:

$$\mathcal{F}(\boldsymbol{X}, \pi_i, \boldsymbol{d}_{\cdot i}) = \frac{1}{|U|}\sum_{u\in U}\frac{1}{\epsilon_{ui}^l+\epsilon_{ui}^r}\min\{\epsilon_{ui}^l + \overline{d_{ui}} - d_{ui}, \epsilon_{ui}^r + d_{ui} - \underline{d_{ui}}, \overline{d_{ui}} - \underline{d_{ui}}, \epsilon_{ui}^l+\epsilon_{ui}^r\} \sup_{V\in\pi_i}\inf_{v\in U}\min\left\{1, 1+\mathbb{I}(v\in V) - \min_{c\in[C]}\left\{1 - \frac{|x_{uc}-x_{vc}|}{|\max(\boldsymbol{x}_{\cdot c}) - \min(\boldsymbol{x}_{\cdot c})|}\right\}\right\}, \tag{7}$$

where $\mathbb{I}(\cdot)$ is the indicator function, $U = \{1,2,\ldots,N\}$, $\overline{d_{ui}} = \max_{j\in\pi_i^u} d_{ji}$, $\underline{d_{ui}} = \min_{j\in\pi_i^u} d_{ji}$, and π_i^u denotes the group (in π_i) containing the instance u. ϵ_{ui}^l and ϵ_{ui}^r measure the unreliability of the training LDD, which are set as $\epsilon_{ui}^l = \min\{d_{ui}, \zeta\}$ and $\epsilon_{ui}^r = \min\{1-d_{ui}, \zeta\}$ where ζ is set to 0.05 in this paper.

Fuzzy Dependency Gain Reweighted by Inclusion Probability To define FDGI, it is necessary to obtain the expected error of approximating LDD based on Theorem 1:

$$\mathcal{E}(\pi_i, \boldsymbol{d}_{\cdot i}) = \frac{1}{|U|}\sum_{u\in U}\frac{\epsilon_{ui}^l\epsilon_{ui}^r - \underline{\Delta}_{ui}\overline{\Delta}_{ui}}{3} + \frac{(\epsilon_{ui}^r-\epsilon_{ui}^l)^2}{3} + \frac{(\underline{\Delta}_{ui}+\overline{\Delta}_{ui})^2}{3} + \frac{(\epsilon_{ui}^l-\epsilon_{ui}^r)(\underline{\Delta}_{ui}+\overline{\Delta}_{ui})}{2}, \tag{8}$$

Algorithm 1: Label coarsening function $f^{\downarrow}$

Input: training set $(\boldsymbol{X}, \boldsymbol{D})$, a set distance function $d_{\text{set}}(\cdot,\cdot)$;
Output: coarse-grained labels $\boldsymbol{Y}$ and partitions of training set π;

$\boldsymbol{Y} \leftarrow \boldsymbol{1}$ ▷ Initilize a matrix of coarse-grained labels.;
for $i = 1, 2, \cdots, M$ **do**
  $\pi_i \leftarrow \{\{u\}\}_{u=1}^{N}$; ▷ Initialize a partition of training set.
  $\pi_i^{\star} \leftarrow \pi_i$; ▷ Initialize the optimal partition.
  $\gamma^{\star} \leftarrow 0$; ▷ Initialize the maximum FDGI.
  while $|\pi_i| > 1$ **do**
    $v_1^{\star}, v_2^{\star} \leftarrow \arg\min_{v_1 \neq v_2 \in \pi_i \times \pi_i} d_{\text{set}}(v_1, v_2)$; ▷ Find two groups closest to each other in terms of $\boldsymbol{d}_{\cdot i}$.
    $\pi_i \leftarrow \pi_i \backslash v_1 \backslash v_2 \cup (v_1 \cup v_2)$; ▷ Merge the two closest groups to update the trainint set partition.
    if $\gamma(\boldsymbol{X}, \pi_i, \boldsymbol{d}_{\cdot i}) > \gamma^{\star}$ **then**
      $\gamma^{\star} \leftarrow \gamma(\boldsymbol{X}, \pi_i, \boldsymbol{d}_{\cdot i})$; ▷ Update FDGI.
      $\pi_i^{\star} \leftarrow \pi_i$; ▷ Update the optimal partition.
  for $k = 1, 2, \cdots, |\pi_i^{\star}|$ **do**
    $\pi_{ik}^{\star} \leftarrow$ the group in $\pi_i^{\star}$ which ranks the k-th position in ascending order w.r.t. the average LDD;
    for $u \in \pi_{ik}^{\star}$ **do**
      $y_{ui} \leftarrow k$; ▷ Assign the coarse-grained label.
return $\boldsymbol{Y}$, π.

where $\overline{\Delta}_{ui} \triangleq d_{ui} - \overline{d_{ui}}$, $\underline{\Delta}_{ui} \triangleq d_{ui} - \underline{d_{ui}}$, other symbols are defined as in Eq. (7). Finally, we define the FDGI as the quotient of the reweighted fuzzy dependency by the expected approximating error, i.e., the quotient of Eq. (7) by Eq. (8):

$$\gamma(\boldsymbol{X}, \pi_i, \boldsymbol{d}_{\cdot i}) = \mathcal{E}(\pi_i, \boldsymbol{d}_{\cdot i})^{-1} \mathcal{F}(\boldsymbol{X}, \pi_i, \boldsymbol{d}_{\cdot i}). \tag{9}$$

The process of coarsening the label description degree can be summarized in Algorithm 1.

4.3 Refining Coarse-Grained Label

Here we investigate how to refine the CGL predictions by the unreliable LDD predictions. The main idea is to find the label distribution closest to the predicted LDD within the range of LDD specified by CGL as the result, i.e.,

$$\begin{aligned} &f^{\uparrow}\big(f^{\star}(\hat{\boldsymbol{x}}), f^{\circ}(\hat{\boldsymbol{x}}), \pi\big) \\ =&\mathbb{E}_{\hat{\boldsymbol{y}} \sim p^{\circ}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{x}})} \Big[\arg\min_{d_i' \in B(i, \hat{y}_i, \pi), i \in [M]} \|\boldsymbol{d}' - f^{\star}(\hat{\boldsymbol{x}})\| \Big], \end{aligned} \tag{10}$$

where the operation $\arg\min_{d_i' \in B(i, \hat{y}_i, \pi), i \in [M]} \|\boldsymbol{d}' - f^{\star}(\hat{\boldsymbol{x}})\|$ returns the label distribution closest to $f^{\star}(\hat{\boldsymbol{x}})$ while satisfying that $d_i' \in B(i, \hat{y}_i, \pi)$ holds for any $i \in \{1, 2, \cdots, M\}$; the interval B, which provides the LDD range corresponding to the CGL, and the probability distribution of coarse-grained labels $p^{\circ}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{x}})$ are defined as follows:

$$\begin{aligned} B(m, k, \pi) &= (\min\{d_{im} | i \in \pi_{mk}\}, \max\{d_{im} | i \in \pi_{mk}\}), \\ p^{\circ}(\hat{\boldsymbol{y}}|\hat{\boldsymbol{x}}) &= \prod_{m=1}^{M} \text{Cat}(\hat{y}_m | f_m^{\circ}(\hat{\boldsymbol{x}})), \end{aligned} \tag{11}$$

where d_{im} is the description degree of the m-th label to the i-th training instance, π_{mk} is the group in π_m, which ranks the k-th position in ascending order w.r.t. the mean LDD value of the instance group. $\text{Cat}(\hat{y}_m | f_m^{\circ}(\hat{\boldsymbol{x}}))$ is a categorical distribution with the parameter of $f_m^{\circ}(\hat{\boldsymbol{x}})$. It should be noted that the time complexity of computing the exact solution of the expectation in Eq. (10) is at least $\mathcal{O}(K^M)$, where K is the number of possible values for CGL, so we use Monte Carlo to approximate it:

$$f^{\uparrow}\big(f^{\star}(\hat{\boldsymbol{x}}), f^{\circ}(\hat{\boldsymbol{x}}), \pi\big) \approx \frac{1}{L} \sum_{l=1}^{L} \arg\min_{d_i' \in B(i, \hat{y}_i^{(l)}, \pi), i \in [M]} \|\boldsymbol{d}' - f^{\star}(\hat{\boldsymbol{x}})\|, \tag{12}$$

where $\hat{y}^{(l)}$ is a Monte Carlo sample generated from the categorical distribution $\text{Cat}(\hat{y}_m | f_m^{\circ}(\hat{\boldsymbol{x}}))$, and the number of Monte Carlo samples L is set to 20 in this paper.

5 Experiments

5.1 Datasets and Evaluation Measures

We adopt six datasets from several representative real-world tasks, including JAFFE (Lyons et al. 1998) from a facial emotion recognition task, Movie (Geng 2016) from a movie rating prediction task, Emotion6 (Peng et al. 2015) and Painting (Machajdik and Hanbury 2010) from image sentiment recognition tasks, M2B (Nguyen et al. 2012) and FBP5500 (Liang et al. 2018) from facial beauty perception tasks. More details of the datasets can be found in Appendix.

We use five LDL measures, which are Cheb (Chebyshev distance), Clark (Clark distance), KL (Kullback-Leibler divergence), Cosine (cosine coefficient), and Intersec (intersection similarity), to evaluate the performance. The formula of these measures can be found in paper (Geng 2016). The lower value of distance-based measures (i.e., Cheb, Clark, KL) or the higher value of similarity-based measures (i.e., Cosine and Intersec) indicates the better performance, which are denoted by $\downarrow$ and $\uparrow$, respectively.

5.2 Predictive Experiments

Experimental Procedure Given a dataset, it is randomly divided into two chunks (70% for training and 30% for testing). In the noise-free scenario, we train an LDL model directly through the training set and evaluate the performance on the test set; in the noisy scenario, we first inject noise into the label distribution of the training instance according to the following equation:

$$\boldsymbol{d} \leftarrow \alpha \boldsymbol{\epsilon} + (1 - \alpha)\boldsymbol{d}, \tag{13}$$

where $\boldsymbol{\epsilon}$ is a random label distribution, α is the proportion of noise in supervision information, then train an LDL model on such a noise-disturbed training set and evaluate the predictive performance on the test set. We repeat the above process ten times and record the mean and standard deviation.

Comparison Methods We use SABFGS (Geng 2016), AAkNN (Geng 2016), BD-LDL (Liu et al. 2021), LDL-LRR (Jia et al. 2023), and LDL-LDM (Wang and Geng 2023) for comparison. The hyperparameter K in AAkNN is selected from $\{5, 6, \cdots, 10\}$; the hyperparameters λ_1 and λ_2 in BD-LDL are both selected from $\{10^{-3}, 10^{-2}, \cdots, 10^{3}\}$; the hyperparameters λ and β in LDL-LRR are selected from $\{10^{-6}, 10^{-5}, \cdots, 10^{-1}\}$

Method	Cheb (↓)	Clark (↓)	KL (↓)	Cosine (↑)	Intersec (↑)
			JAFFE		
AGLDL	(1) 0.090 ± 0.005	(3) 0.346 ± 0.011	(3) 0.049 ± 0.003	(3) 0.955 ± 0.003	(3) 0.881 ± 0.004
AAkNN	(5) 0.102 ± 0.006 •	(4) 0.358 ± 0.013 •	(5) 0.053 ± 0.004 •	(5) 0.949 ± 0.004 •	(5) 0.871 ± 0.006 •
SABFGS	(6) 0.117 ± 0.007 •	(6) 0.463 ± 0.028 •	(6) 0.086 ± 0.011 •	(6) 0.926 ± 0.009 •	(6) 0.841 ± 0.009 •
BD-LDL	(4) 0.097 ± 0.005 •	(5) 0.362 ± 0.011 •	(4) 0.050 ± 0.003 •	(4) 0.953 ± 0.003 •	(4) 0.874 ± 0.004 •
LDL-LDM	(1) 0.090 ± 0.004	(1) 0.332 ± 0.009 ○	(1) 0.044 ± 0.003 ○	(1) 0.958 ± 0.003 ○	(1) 0.884 ± 0.005 ○
LDL-LRR	(1) 0.090 ± 0.004	(1) 0.332 ± 0.009 ○	(1) 0.044 ± 0.003 ○	(1) 0.958 ± 0.003 ○	(1) 0.884 ± 0.005 ○
			Movie		
AGLDL	(3) 0.117 ± 0.001	(3) 0.521 ± 0.005	(3) 0.101 ± 0.002	(3) 0.934 ± 0.001	(3) 0.835 ± 0.002
AAkNN	(5) 0.123 ± 0.001 •	(5) 0.552 ± 0.003 •	(4) 0.113 ± 0.003 •	(5) 0.925 ± 0.002 •	(5) 0.823 ± 0.001 •
SABFGS	(6) 0.137 ± 0.002 •	(6) 0.592 ± 0.008 •	(6) 0.143 ± 0.005 •	(6) 0.911 ± 0.003 •	(6) 0.808 ± 0.003 •
BD-LDL	(3) 0.117 ± 0.001	(4) 0.539 ± 0.004 •	(5) 0.117 ± 0.007 •	(3) 0.934 ± 0.001	(4) 0.833 ± 0.002 •
LDL-LDM	(1) 0.115 ± 0.001 ○	(1) 0.515 ± 0.004 ○	(1) 0.098 ± 0.002 ○	(1) 0.936 ± 0.001 ○	(1) 0.837 ± 0.002 ○
LDL-LRR	(1) 0.115 ± 0.001 ○	(1) 0.515 ± 0.004 ○	(1) 0.098 ± 0.002 ○	(1) 0.936 ± 0.001 ○	(1) 0.837 ± 0.002 ○
			Emotion6		
AGLDL	(1) 0.252 ± 0.006	(1) 1.622 ± 0.013	(1) 0.427 ± 0.012	(1) 0.805 ± 0.006	(1) 0.659 ± 0.005
AAkNN	(2) 0.260 ± 0.007 •	(4) 1.654 ± 0.013 •	(3) 0.482 ± 0.022 •	(2) 0.786 ± 0.008 •	(3) 0.649 ± 0.007 •
SABFGS	(5) 0.300 ± 0.009 •	(6) 1.762 ± 0.017 •	(5) 0.621 ± 0.030 •	(5) 0.733 ± 0.010 •	(5) 0.612 ± 0.009 •
BD-LDL	(4) 0.282 ± 0.005 •	(2) 1.638 ± 0.015 •	(4) 0.517 ± 0.019 •	(3) 0.775 ± 0.006 •	(4) 0.620 ± 0.005 •
LDL-LDM	(6) 0.336 ± 0.013 •	(3) 1.651 ± 0.031 •	(6) 0.647 ± 0.287 •	(6) 0.715 ± 0.016 •	(6) 0.601 ± 0.057 •
LDL-LRR	(3) 0.281 ± 0.006 •	(5) 1.678 ± 0.032 •	(2) 0.476 ± 0.067 •	(4) 0.764 ± 0.013 •	(2) 0.655 ± 0.011
			Painting		
AGLDL	(1) 0.255 ± 0.007	(3) 1.718 ± 0.034	(1) 0.553 ± 0.010	(1) 0.727 ± 0.005	(1) 0.599 ± 0.006
AAkNN	(2) 0.258 ± 0.011	(5) 1.754 ± 0.035 •	(4) 0.590 ± 0.035 •	(5) 0.707 ± 0.014 •	(5) 0.587 ± 0.013 •
SABFGS	(6) 0.321 ± 0.014 •	(6) 1.911 ± 0.037 •	(6) 1.164 ± 0.319 •	(6) 0.634 ± 0.022 •	(6) 0.535 ± 0.016 •
BD-LDL	(4) 0.261 ± 0.010 •	(3) 1.718 ± 0.034	(5) 0.594 ± 0.100 •	(3) 0.719 ± 0.008 •	(3) 0.592 ± 0.009 •
LDL-LDM	(5) 0.264 ± 0.011 •	(1) 1.714 ± 0.037	(3) 0.570 ± 0.026 •	(4) 0.716 ± 0.011 •	(4) 0.588 ± 0.012 •
LDL-LRR	(3) 0.260 ± 0.008 •	(1) 1.714 ± 0.034	(2) 0.557 ± 0.016	(2) 0.723 ± 0.007	(2) 0.594 ± 0.008 •
			M2B		
AGLDL	(1) 0.356 ± 0.007	(1) 1.205 ± 0.011	(1) 0.561 ± 0.019	(1) 0.709 ± 0.006	(1) 0.581 ± 0.007
AAkNN	(4) 0.371 ± 0.008 •	(2) 1.216 ± 0.014 •	(5) 0.664 ± 0.051 •	(5) 0.678 ± 0.015 •	(3) 0.564 ± 0.009 •
SABFGS	(6) 0.392 ± 0.008 •	(3) 1.254 ± 0.015 •	(6) 0.815 ± 0.048 •	(6) 0.645 ± 0.010 •	(6) 0.551 ± 0.008 •
BD-LDL	(5) 0.373 ± 0.008 •	(6) 1.310 ± 0.005 •	(4) 0.636 ± 0.206 •	(4) 0.705 ± 0.006 •	(5) 0.558 ± 0.008 •
LDL-LDM	(3) 0.370 ± 0.009 •	(5) 1.298 ± 0.033 •	(3) 0.572 ± 0.019 •	(3) 0.706 ± 0.009 •	(4) 0.559 ± 0.010 •
LDL-LRR	(2) 0.365 ± 0.007 •	(4) 1.273 ± 0.034 •	(2) 0.562 ± 0.055	(2) 0.708 ± 0.022	(2) 0.570 ± 0.006 •
			FBP5500		
AGLDL	(2) 0.144 ± 0.004	(1) 1.259 ± 0.007	(2) 0.134 ± 0.010	(2) 0.953 ± 0.002	(2) 0.832 ± 0.006
AAkNN	(2) 0.151 ± 0.003 •	(4) 1.315 ± 0.008 •	(4) 0.170 ± 0.009 •	(5) 0.942 ± 0.003 •	(2) 0.832 ± 0.004
SABFGS	(6) 0.161 ± 0.002 •	(6) 1.334 ± 0.009 •	(5) 0.221 ± 0.004 •	(6) 0.935 ± 0.001 •	(3) 0.829 ± 0.002
BD-LDL	(5) 0.160 ± 0.002 •	(5) 1.331 ± 0.006 •	(6) 0.226 ± 0.015 •	(4) 0.943 ± 0.002 •	(6) 0.822 ± 0.003 •
LDL-LDM	(4) 0.155 ± 0.002 •	(3) 1.294 ± 0.005 •	(3) 0.139 ± 0.004	(2) 0.945 ± 0.002 •	(5) 0.823 ± 0.003 •
LDL-LRR	(1) 0.136 ± 0.002 ○	(2) 1.278 ± 0.005 •	(1) 0.107 ± 0.004 ○	(1) 0.954 ± 0.002	(1) 0.850 ± 0.002 ○

Table 1: Prediction performance formatted as (rank) mean±std statistical significance.

and $\{10^{-3}, 10^{-2}, \cdots, 10^{2}\}$, respectively; in LDL-LDM, the hyperparameters λ_1, λ_2, and λ_3 are selected from $\{10^{-3}, 10^{-2}, \cdots, 10^{3}\}$, the hyperparameter g is selected from $\{1, 2, \cdots, 14\}$. In our framework, we use the ordinal logistic model (All-Threshold variant) proposed in (Rennie and Srebro 2005), abbreviated as LogisticAT, as the CGL predictor whose L_2 regularization weight is selected from $\{1, (2n)^{-1}, 2n\}_{n=1}^{5}$. To prevent the performance of AGLDL from benefiting from any specific LDL algorithm, we do not use any LDD predictor to refine the predictions of the CGL predictor in experiments. The label refining function can be formalized as $f^{\uparrow}\big(f^{\star}(\hat{\boldsymbol{x}}), f^{\circ}(\hat{\boldsymbol{x}}), \pi\big) \approx \frac{1}{L}\sum_{l=1}^{L} \boldsymbol{d}'$, where $d'_i \sim \mathrm{Uni}(d'_i | B(i, \hat{y}_i^{(l)}, \pi))$. To accelerate the convergence,

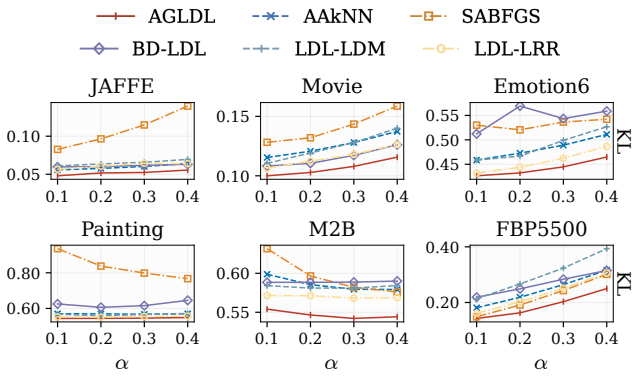


Figure 4: KL performance under varying extent of noise.

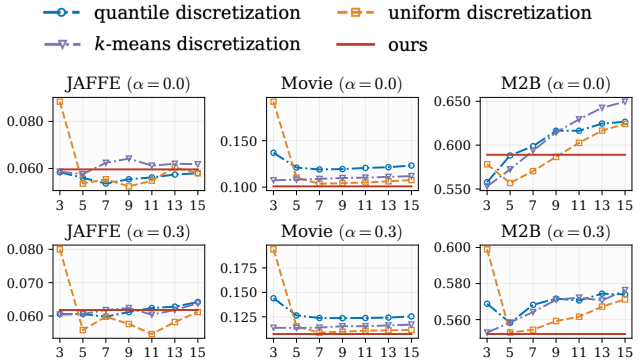


Figure 5: KL performance of varying label coarsening methods. The horizontal axis denotes the number of possible values of CGL obtained by the label coarsening method, and the vertical axis denotes the KL performance.

we preprocess the feature by min-max normalization.

Results and Discussions Table 1 shows the predictive performance of each comparison method on seven real-world datasets. Each experimental result is formatted as "(rank) mean±std t-test"; "(rank)" is the rank of each comparison method among all methods; •/◦ denotes whether AGLDL is statistically superior/inferior to the corresponding comparison method under the pairwise two-tailed t-test at 0.05 significance level; if neither • nor ◦ is shown, there is no significant difference between the corresponding method and AGLDL; It can be observed that our method AGLDL performs weakly on the "JAFFE" and "Movie" datasets, but performs well on the "Emotion6", "Painting", and "M2B" datasets; the performance on the "FBP5500" dataset is moderate. Furthermore, we compute average performance ranking of AGLDL and the fuzzy dependency of each dataset; the results are shown in the following table:

Emotion6	M2B	Painting	FBP5500	JAFFE	Movie
0.34 / 1	0.38 / 1	0.23 / 1.4	0.41 / 1.8	0.58 / 2.6	0.86 / 3

where each entry is formatted as "fuzzy dependency / average performance ranking". It can be seen that AGLDL performs weakly on the datasets with fuzzy dependency over FBP5500, and AGLDL performs well on the datasets with fuzzy dependency below FBP5500. We believe the reason is that the fuzzy dependency is positively correlated with the quality of label information. Specifically, a lower fuzzy dependency indicates that the label information may contain more noise, and our proposed AGLDL specializes in noisy environments, which is why our method performs well. Nevertheless, we find that our method performs comparably to the top-performing method on "JAFFE" and "Movie" datasets, which indicates that our method can achieve good performance even under high-quality label information.

Furthermore, we test the performance of comparison algorithms under conditions where the labels contain varying degrees of noise. As shown in Figure 4, where different styles of lines represent different comparison algorithms, the horizontal axis represents the proportion of noise. It is evident that our algorithm performs the best in most cases and is relatively insensitive to the increasing noise.

5.3 Experiments on Label Coarsening Function

As shown in Figure 5, we use uniform discretization, quantile discretization and k-means discretization as alternatives to our proposed label coarsening function. Uniform discretization divides the interval of label description degree into multiple groups with identical width; quantile discretization divides the interval of label description degree into groups containing identical number of elements; k-means discretization divides the interval of label description degree into multiple groups by the k-means clustering algorithm. It can be found that in most cases the predictive performance of our proposal is better than that of uniform discretization and quantile discretization. In addition, we observe that the predictive performance obtained by the comparison discretization methods gets better first and then worse as the label gets finer. This phenomenon is consistent with our intuition: too coarse a label can lead to a large error in approximating the true label description degree, and too fine a label can easily lead to a noise-sensitive model.

6 Conclusion

This paper proposes an adaptive-grained label distribution learning framework to cope with the ubiquitous situation that the quality of the collected label distributions cannot be guaranteed. On the one hand, we novelly propose a label coarsening algorithm to transform unreliable label distributions into relatively reliable ordinal discrete labels, which can be mined by any off-the-shelf ordinal regression model. In the label coarsening algorithm, we design FDGI to obtain the best coarse-grained label. On the other hand, we design a label refining algorithm to enhance the coarse-grained labels into label distributions by any existing LDL algorithm. The experimental results on multiple real-world datasets show that our framework can significantly improve the performance of LDL in the cases where the quality of the collected label distributions is not guaranteed.

Acknowledgements

This work was partially supported by the Natural Science Foundation of Jiangsu Province (BK20242045), and the National Natural Science Foundation of China (62176123, 62476130, 62276217, 62176116).

References

Chen, M.; Wang, X.; Feng, B.; and Liu, W. 2018. Structured random forest for label distribution learning. *Neurocomputing*, 320: 171–182.

Gao, B.; Zhou, H.; 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.; Smith-Miles, K.; and Zhou, Z.-H. 2013. Facial Age Estimation by Learning from Label Distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35: 2401–2412.

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.; Ren, T.; Chen, L.; Wang, J.; Zhu, J.; and Long, X. 2019a. Weakly supervised label distribution learning based on transductive matrix completion with sample correlations. *Pattern Recognition Letters*, 125: 453–462.

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. 2019b. Facial Emotion Distribution Learning by Exploiting Low-Rank Label Correlations Locally. In *IEEE Conference on Computer Vision and Pattern Recognition*, 9833–9842.

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

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.

Lyons, M.; Akamatsu, S.; Kamachi, M.; and Gyoba, J. 1998. Coding Facial Expressions with Gabor Wavelets. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 200–205.

Machajdik, J.; and Hanbury, A. 2010. Affective Image Classification Using Features Inspired by Psychology and Art Theory. In *ACM International Conference on Multimedia*, 83–92.

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.

Peng, K.-C.; Chen, T.; Sadovnik, A.; and Gallagher, A. 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. 2019a. Label Distribution Learning with Label-specific Features. In *International Joint Conference on Artificial Intelligence*, 3318–3324.

Ren, T.; Jia, X.; Li, W.; and Zhao, S. 2019b. Label Distribution Learning with Label Correlations via Low-Rank Approximation. In *International Joint Conference on Artificial Intelligence*, 3325–3331.

Rennie, J. D.; and Srebro, N. 2005. Loss Functions for Preference Levels: Regression with Discrete Ordered Labels. In *IJCAI Multidisciplinary Workshop on Advances in Preference Handling*, 180–186.

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

Tsoumakas, G.; and Katakis, I. 2006. Multi-Label Classification: An Overview. *International Journal of Data Warehousing and Mining*, 3(3): 1–13.

Wallace, M.; Avrithis, Y.; and Kollias, S. 2006. Computationally Efficient Sup-T Transitive Closure for Sparse Fuzzy Binary Relations. *Fuzzy Sets and Systems*, 157(3): 341–372.

Wang, J.; and Geng, X. 2023. Label Distribution Learning by Exploiting Label Distribution Manifold. *IEEE Transactions on Neural Networks and Learning Systems*, 34(2): 839–852.

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

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.

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

Xu, S.; Ju, H.; Shang, L.; Pedrycz, W.; Yang, X.; and Li, C. 2020. Label distribution learning: A local collaborative mechanism. *International Journal of Approximate Reasoning*, 121: 59–84.

Yang, J.; She, D.; and Sun, M. 2017. Joint Image Emotion Classification and Distribution Learning via Deep Convolutional Neural Network. In *International Joint Conference on Artificial Intelligence*, 3266–3272.

Zeng, X.-Q.; Chen, S.-F.; Xiang, R.; Wu, S.-X.; and Wan, Z.-Y. 2019. Filling missing values by local reconstruction for incomplete label distribution learning. *International Journal of Wireless and Mobile Computing*, 16(4): 314–321.

Zhai, Y.; and Dai, J. 2019. Geometric Mean Metric Learning for Label Distribution Learning. In *International Conference on Neural Information Processing*, 260–272.

Zhai, Y.; Dai, J.; and Shi, H. 2018. Label Distribution Learning Based on Ensemble Neural Networks. In *International Conference on Neural Information Processing*, 593–602.

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.