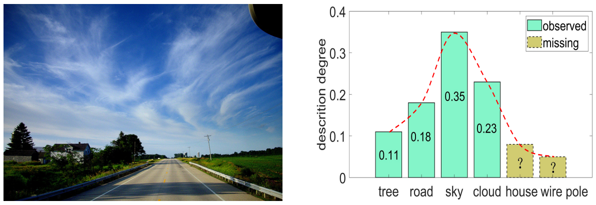
No Regularization Is Needed: Efficient and Effective Incomplete Label Distribution Learning

Xiang Li[1,2] and **Songcan Chen**[1,2]

[1]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics
[2]MIIT Key Laboratory of Pattern Analysis and Machine Intelligence
{lx90, s.chen}@nuaa.edu.cn

Abstract

In reality, it is laborious to obtain complete label degrees, giving birth to Incomplete Label Distribution Learning (InLDL), where some degrees are missing. Existing InLDL methods often assume that degrees are uniformly random missing. However, it is often not the case in practice, which arises the first issue. Besides, they often adopt explicit regularization to compensate the incompleteness, leading to burdensome parameter tuning and extra computation, causing the second issue. To address the first issue, we adopt a more practical setting, i.e., small degrees are more prone to be missing, since large degrees are likely to catch more attention. To tackle the second issue, we argue that label distribution itself already contains abundant knowledge, such as label correlation and ranking order, thus it may have provided enough prior for learning. It is precisely because existing methods overlook such a prior that leads to the forced adoption of explicit regularization. By directly utilizing the label degrees prior, we design a properly weighted objective function, exempting the need from explicit regularization. Moreover, we provide rigorous theoretical analysis, revealing in principle that the weighting plays an implicit regularization role. To sum up, our method has four advantages, it is 1) *model selection free*; 2) with closed-form solution (sub-problem) and easy-to-implement (a few lines of codes); 3) with *linear* computational complexity in the number of samples, thus scalable to large datasets; 4) competitive with state-of-the-arts in both random and non-random missing scenarios.



(a) Image from real-world.

(b) Incomplete label distribution in practice.

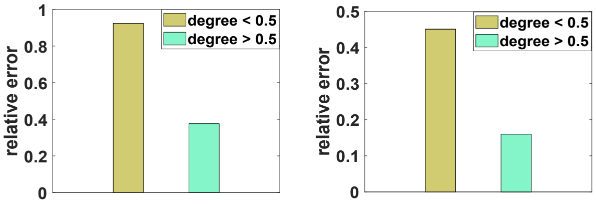
Figure 1: An illustration of incomplete label distribution in practice. Figure 1(a) is an image from real-world and Figure 1(b) is the corresponding label distribution. Note that, labels with small degrees, like "house" and "wire pole" are harder to recognize compared to those with large degrees. This suggests that label degrees are not randomly missing but small degrees are more likely to be missing.

1 Introduction

In real applications, labels may be associated with a sample to some degree, thus soft labels are preferred rather than the hard ones to describe the label ambiguity [Rupprecht *et al.*, 2017], where Label Distribution Learning (LDL) [Geng, 2016] originated from. LDL is a learning paradigm that assigns a sample with different label description degrees, *i.e.*, the relevance of a sample belonging to different labels, which satisfy the probability simplex constraint. However, obtaining complete label degrees is always laborious and challenging in real-world, thus the desire to get rid of such predicament drives the emergence of Incomplete LDL (InLDL) [Xu and Zhou, 2017]. To date, InLDL has wide applications in facial expression recognition [Yang *et al.*, 2017; Chen *et al.*, 2020], age estimation [Hou *et al.*, 2017; Gao *et al.*, 2018], and multi-label ranking [Geng *et al.*, 2021; Lu *et al.*, 2023].

In the pioneer InLDL work [Xu and Zhou, 2017], the authors assume that label degrees are uniformly random missing, and subsequent works [Jia *et al.*, 2019a; Li *et al.*, 2022; Wang and Geng, 2023] just simply follow this assumption. However, it is often not the case in practice. As shown in Figure 1, when annotating such a picture from real-world, labels with small degrees such as "house" and "wire pole" are much more difficult to recognize compared to those with large degrees such as "sky" and "cloud". This suggests that label degrees are not randomly missing but small degrees are more likely to be missing in practice, which arises the first issue.

Besides, to compensate the degree incompleteness, existing InLDL methods always have to make various assumptions, which are then translated into one or more explicit regularization terms in their objective functions. For example, in [Xu and Zhou, 2017], the authors assume that the label degree matrix is low-rank in characterizing the correlation between labels, and adopt the trace norm as a regularization term; in



(a) Flickr.

(b) Fbp5500.

Figure 2: The mean relative errors for degrees less than 0.5 and degrees greater than 0.5 on two datasets Flickr and Fbp5500.

[Wang and Geng, 2023], the authors assume that the predictions of their model lie in the same manifold, and exploit both the global and local label correlations with three different regularization terms. Obviously, each imposed regularization term associates with a hyper-parameter, which leads to burdensome parameter tuning and extra computation cost for model selection, causing the second issue.

To address the first issue, we consider a more practical setting in this paper, i.e., small degrees are more likely to be missing, instead of adopting the uniformly random missing assumption as in existing InLDL methods. Our setting is more reasonable since labels with smaller degrees are harder to recognize as shown in Figure 1, rendering them more susceptible to being missing, which is more in line with practicality. To our best knowledge, this is the first work that considers non-random missing scenario in the InLDL field.

To tackle the second issue, we argue that label distribution itself already contains abundant knowledge including label correlation and ranking order, thus it may have provided enough prior for learning. Existing methods mainly focus on mining label correlations while overlooking such a useful prior that leads to the forced adoption of explicit regularization. Contrary to these methods, we highlight the benefits and importance of the label degrees prior in this paper. Intriguingly, we discover that when this prior is appropriately leveraged, the InLDL problem can be resolved even without any explicit regularization, thereby exempting from parameter tuning and additional computational overhead.

Subsequently, a pivotal question naturally arises: how to effectively leverage the aforementioned prior? Our solution is straightforward: we design a properly weighted objective function by directly integrating the label degrees into a weighting matrix. Reasons of this solution lie in four aspects. (1) Intuitively, labels with large degrees are likely to receive more concern, while those with small degrees are easily overwhelmed or even overlooked, making them more difficult to be recognized. (2) Empirically, we conduct experiments to validate the above intuition. From Figure 2, it is evident that the mean relative errors are higher for degrees below 0.5 compared to those degrees above 0.5. This observation indicates that small degrees are underfitted in comparison to those large degrees. (3) Generally, in label distribution learning, the goal is to learn a distribution of all labels. Given that small degrees are underfitted, it is imperative to allocate more attention on them so as to learn a more accurate label distribution. A natural and straightforward way is to design a weighting scheme that puts more emphasis on small degrees. (4) By directly integrating the label degrees into a weighting matrix, we not only make a rational use of the label degree prior, but also save additional overhead of learning a weighted matrix.

Based on the above analyses and with the aim to learn an accurate label distribution for all labels, we impose larger weights on the losses of the smaller observed degrees. For missing degrees, we gradually increase the weights on their losses, since these degrees should become more reliable as the training progresses. To make theoretical explanations of why the proposed method could work without any explicit regularization, we first define a weighted empirical risk and derive data-dependent upper bounds between the expected risk and the weighted empirical risk. These upper bounds *explicitly* depend on the weighted matrix, and the generalization error can be bounded by this weighted matrix, implying that such weighting plays a role of *implicit* regularization as no explicit regularization is *really* imposed. Note that, by utilizing the Alternating Direction Method of Multipliers (ADMM) [Boyd *et al.*, 2011] for optimization, we derive the closed form solution of each sub-problem and implement the codes in just a few lines. Interestingly, the computational complexity is linear in the number of samples, making our method fast and scalable to large datasets.

To sum up, our main contributions are threefold:

(1) We propose an efficient, effective, and easy-to-implement Weighted method for InLDL, abbreviated as WInLDL, which is free of any explicit regularization, to the best of our knowledge, this is the first time in the InLDL field.

(2) We theoretically derive a data-dependent upper bound between the expected risk and the weighted empirical risk with the help of Rademacher complexity, which contains the classical risk bounds under single-label and unweighted settings as our special cases.

(3) We empirically verify the effectiveness of WInLDL on ten real-world datasets, and experiments show that it is competitive with state-of-the-art methods in both random and non-random missing scenarios.

2 Related Work

In this section, we review the most relevant works to ours. InLDL was first proposed by [Xu and Zhou, 2017] to address the problem with incomplete annotations. They assume that the matrix formed by the label degrees is low-rank to combat the incompleteness, and adopt the trace norm as a regularization term to formulate the label correlations. Later, in [Jia *et al.*, 2019b], the authors assume the clusters of samples are low-rank and also utilize the trace norm to characterize such local label correlations. Recently, in [Wang and Geng, 2023], the authors argue that the low-rank assumption may not hold, instead, they assume that the predictions of their model lie on the same manifold whose structure may encode the correlations among labels. Further, they exploit both the global and local correlations to learn the label distribution in the InLDL setting. All the above methods focus on mining label correlations while ignoring the fact that label distribution itself provides useful even sufficient prior knowledge, as we dissected in the Introduction. We contend that the degree prior should not be overlooked but rather utilized rationally. To the best of our knowledge, [Wang *et al.*, 2022] is the only work that considers the label degrees to do the weighting. Compared with our work, there are three main differences, 1) their task focuses on classification, the goal is to learn the top label(s), thus they put large weights on large degrees, which will be verified in Table 5 that in the InLDL setting, such a weighting performs worse than our WInLDL; 2) they discard the weighting scheme in their theoretical analysis, thus they do not provide a theoretical guarantee for weighting, which is a main contribution of this paper; 3) they use the product between the entropy ($E_{\mathbf{x}} = -\sum_{y\in\mathcal{Y}} d_{\mathbf{x}}^{y} \ln d_{\mathbf{x}}^{y}$) of degree and degree itself as the weights. While in the InLDL setting, the missing degrees are set to 0, and $\ln d_{\mathbf{x}}^{y}$ will be meaningless in such a situation. Therefore, their weighting scheme cannot be applied to the InLDL setting. There are other works in the field of InLDL [Jia *et al.*, 2021; Teng and Jia, 2021; Zhang *et al.*, 2022; Qian *et al.*, 2022a; Qian *et al.*, 2022b] that either adopt different settings such as semi-supervised paradigm, or focus on different tasks such as feature selection. Consequently, they are not highly relevant to the scope of this work. Due to the page limitations, we have omitted discussing these works here, and interested readers are referred to these literatures for further details.

3 Proposed Method

3.1 Problem Setting

Let $\mathcal{X} \subseteq \mathbb{R}^k$ be the feature space and $\mathcal{Y} = \{y_1, y_2, \cdots, y_C\}$ be the label space, where k is the dimension of the feature, and C is the number of labels. In LDL, each sample $\mathbf{x} \in \mathcal{X}$ is assigned with a label distribution $\mathbf{d}_{\mathbf{x}} = [d_{\mathbf{x}}^{y_1}, d_{\mathbf{x}}^{y_2}, \cdots, d_{\mathbf{x}}^{y_C}]^\top$, where $d_{\mathbf{x}}^{y_i}$ is called the label description degree, which indicates the relevance of sample $\mathbf{x}$ belonging to the i-th label. Note that, $\mathbf{d}_{\mathbf{x}}$ satisfies the probability simplex constraints, *i.e.*, $d_{\mathbf{x}}^{y_j} \geq 0$ and $\sum_{j=1}^{C} d_{\mathbf{x}}^{y_j} = 1$. Given a training dataset $S = \{(\mathbf{x}_i, \mathbf{d}_{\mathbf{x}_i})\}_{i=1}^{N}$, where N is the number of samples, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]^\top \in \mathbb{R}^{N\times k}$ is the feature matrix, $\mathbf{D} = [\mathbf{d}_{\mathbf{x}_1}, \mathbf{d}_{\mathbf{x}_2}, \cdots, \mathbf{d}_{\mathbf{x}_N}]^\top \in \mathbb{R}^{N\times C}$ is the label distribution matrix. The goal of LDL is to learn a function $f : \mathcal{X} \mapsto \mathbb{R}^C$, which minimizes the difference between the prediction of f and the ground-truth label distribution, *i.e.*, $\min_f \mathcal{L}(f(\mathbf{X}), \mathbf{D})$, where $\mathcal{L}$ is the loss function.

In the scenario of InLDL, label degrees may be incomplete. In this paper, we consider a more practical scenario, i.e., small degrees are more prone to be missing. Formally, let $\Omega \in [N] \times [C]$ and $U \in [N] \times [C]$ denote the indices of the observed and the unobserved entries from $\mathbf{D}$, respectively. The unobserved entries of the label distribution matrix are set to 0, *i.e.*, the observed label matrix $\widetilde{\mathbf{D}}$ can be defined as, $\forall (i, j) \in [N] \times [C]$,

$$\widetilde{\mathbf{D}}_{ij} = \begin{cases} \mathbf{D}_{ij} & \text{if } (i,j) \in \Omega \\ 0 & \text{if } (i,j) \in U. \end{cases} \tag{1}$$

Then in InLDL, the goal is to $\min_f \mathcal{L}(f(\mathbf{X}), \widetilde{\mathbf{D}})$. Note that, to eliminate any potential bias for our WInLDL, we also conduct experiments in the random missing scenario as in the compared methods for a fair comparison.

3.2 Proposed WInLDL

In this subsection, we first design a weighted method named WInLDL and then apply the ADMM to solve the objective, by which an efficient algorithm is derived.

Specifically, given a feature matrix $\mathbf{X}$ and an observed label matrix $\widetilde{\mathbf{D}}$, let $f(\mathbf{X}) = \mathbf{XW}$ and $\mathcal{L}$ be the ℓ_2 loss, then we can define the following weighted function,

$$g(\mathbf{W}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{C}\mathbf{P}_{ij}((\mathbf{XW})_{ij} - \tilde{\mathbf{D}}_{ij})^2 \tag{2}$$

$$= \frac{1}{2}\left\|\mathbf{P}^{\frac{1}{2}} \odot (\mathbf{XW} - \tilde{\mathbf{D}})\right\|_F^2, \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{k\times C}$ is the transformation matrix to be optimized, and $\odot$ is the Hadamard product. Since the label distribution satisfies the probability simplex constraint, then $\mathbf{XW1}_C = \mathbf{1}_N$ and $\mathbf{XW} \geq \mathbf{0}_{N\times C}$ should hold, where $\mathbf{1}_C$ and $\mathbf{1}_N$ are column vectors of size C and N with all ones, "$\geq$" here means that all elements are greater than or equal to 0. For simplicity of notation, let $\mathbf{Q} = \mathbf{P}^{\frac{1}{2}}$, and $ProS(\mathbf{Z}) := \{\mathbf{Z} \in \mathbb{R}^{N\times C} | \mathbf{Z1}_C = \mathbf{1}_N, \mathbf{Z} \geq \mathbf{0}_{N\times C}\}$. By incorporating the probability simplex constraint, the final objective function of WInLDL can be written as:

$$g(\mathbf{W}) = \frac{1}{2}\left\|\mathbf{Q} \odot (\mathbf{XW} - \tilde{\mathbf{D}})\right\|_F^2, \tag{4}$$

$$s.t. \quad \mathbf{XW} \in ProS(\mathbf{XW}). \tag{5}$$

Remark 1. Eq. (5) is not an additional regularization that we impose on our method, but a constraint inherent in label distribution learning by its definition, and all label distribution learning algorithms must satisfy such a probability simplex constraint. To deal with it, we can utilize off-the-shelf projection methods[Wang and Carreira-Perpinán, 2013; Condat, 2016] to avoid introducing extra model hyper-parameters.

With WInLDL in hand, our main concern in the following is how to design the weighting matrix $\mathbf{Q}$. Motivated by the intuition mentioned in the Introduction, to learn an accurate label distribution for InLDL, it is crucial not to ignore the small observed degrees but to impose large weights on them, while gradually increasing the weights of the missing degrees. In order to directly exploit the degree prior of the label distribution, we subtract the observed degrees from 1 to give large weights on small degrees, formally, the weighting matrix $\mathbf{Q}$ composed of $\mathbf{Q}_\Omega$ and $\mathbf{Q}_U$ is defined as:

$$\mathbf{Q}_{ij} = \begin{cases} 1 - \widetilde{\mathbf{D}}_{ij} & \text{if } (i,j) \in \Omega \\ 1 - \mathbf{D}_{U_{ij}} & \text{if } (i,j) \in U, \end{cases} \tag{6}$$

where $\mathbf{D}_{U_{ij}} = \frac{1}{N}\sum_{i=1}^{N} \widetilde{\mathbf{D}}_{ij}$, that is, the missing label degrees are estimated by the mean value of the observed degrees in the corresponding column.

Moreover, to gradually increase the weights of the missing degrees, we take a number greater than 1 that increases monotonically with the number of iterations as the base of the power function. Since the observed degrees are more reliable than the missing ones, the base of its power function is set at 2, which is larger than the number "a" during the whole iterations.

$$\mathbf{Q}_{ij} = \begin{cases} 2^{(1-\tilde{\mathbf{D}}_{ij})} & \text{if } (i,j) \in \Omega \\ a^{(1-\mathbf{D}_{U_{ij}})} & \text{if } (i,j) \in U, a = 1 + \frac{iter}{maxIter}, \end{cases} \tag{7}$$

where $maxIter$ is the maximum iterations, in this paper, fixed at 50. By such a design, three benefits can be obtained, 1) smaller degrees are imposed with larger weights, 2) the weights of the observed degrees are larger than the missing ones, 3) the weights of the missing degrees are gradually increased. Note that the above three benefits can also be regarded as three principles for designing the weighting matrix. Any matrix that satisfies three principles can be utilized as a weighting matrix. In the experimental section, we report the performance of various weighting matrices in Table 5. It is important to highlight that the main focus of this paper is to leverage the useful prior knowledge of label distribution to create an efficient and effective method, rather than extensively exploring the design of an optimal weighting matrix.

3.3 Optimization

In this subsection, we apply ADMM to design an efficient algorithm for solving the objective of WInLDL. Let $\mathbf{Z} = \mathbf{XW}$, the augmented Lagrangian function can be written as:

$$\Phi = \frac{1}{2}\left\|\mathbf{Q} \odot (\mathbf{Z} - \tilde{\mathbf{D}})\right\|_F^2 + tr(\mathbf{\Lambda}^\top(\mathbf{XW} - \mathbf{Z})) + \frac{\mu}{2}\|\mathbf{XW} - \mathbf{Z}\|_F^2), \tag{8}$$

$$s.t. \quad \mathbf{Z} \in ProS(\mathbf{Z}). \tag{9}$$

where tr is the trace operator, and μ is a penalty factor. Note that, μ is NOT a model hyper-parameter BUT a parameter of the ADMM algorithm, it is introduced for the convenience of optimization. In this paper, μ is fixed at 2, which does not need to be tuned. We also conduct experiments in section 4.6 to verify that μ only affects the convergence rate and does not affect performance.

Sub-problem of W. With $\mathbf{Z}$ and $\mathbf{\Lambda}$ fixed, $\mathbf{W}$ can be updated by,

$$\mathbf{W} = (\mathbf{X}^\top\mathbf{X})^{-1}(\mathbf{X}^\top(\mathbf{Z} - \frac{\mathbf{\Lambda}}{\mu})). \tag{10}$$

Sub-problem of Z. With $\mathbf{W}$ and $\mathbf{\Lambda}$ fixed, $\mathbf{Z}$ can be updated by,

$$\mathbf{Z} = \frac{\mu\mathbf{XW} + \mathbf{\Lambda} + \mathbf{Q} \odot \mathbf{Q} \odot \tilde{\mathbf{D}}}{\mathbf{Q} \odot \mathbf{Q} + \mu\mathbf{I}_{N\times C}}, \tag{11}$$

$$\mathbf{Z} = proj(\mathbf{Z}), \tag{12}$$

where $\mathbf{I}_{N\times C}$ is a matrix with all ones, and the division in Eq. (11) is element wise. Eq. (12) projects $\mathbf{Z}$ onto the probability simplex to satisfy the constraint of Eq. (9), and the $proj$ is a projection operator that can be found in [Wang and Carreira-Perpinán, 2013; Condat, 2016].

Sub-problem of Λ. With $\mathbf{W}$ and $\mathbf{Z}$ fixed, $\mathbf{\Lambda}$ can be updated by,

$$\mathbf{\Lambda} \longleftarrow \mathbf{\Lambda} + \mu(\mathbf{XW} - \mathbf{Z}). \tag{13}$$

Complexity Analysis. The computational complexity of the ADMM algorithm is dominated by matrix multiplication and inverse operations. In each iteration, the complexity of updating $\mathbf{W}$ in Eq. (10) is $\mathcal{O}(Nk^2) + \mathcal{O}(k^3) + \mathcal{O}(NkC)$, the complexities of updating $\mathbf{Z}$ in Eq. (11) and Eq. (12) are $\mathcal{O}(NkC) + \mathcal{O}(NC)$ and $\mathcal{O}(NC)$ [Condat, 2016], respectively, the complexity of updating $\mathbf{\Lambda}$ in Eq. (13) is $\mathcal{O}(NkC)$, and the complexity of updating $\mathbf{Q}_U$ in Eq. (7) is $\mathcal{O}(|U|)$, where $|U|$ is the cardinality of set U, usually smaller than NC. Thus, the total computational complexity is $\mathcal{O}(max(Nk^2, NkC) + k^3)$, which is linear in the number of samples N. In Table 1, we list computational complexities of different methods, where 'g' of LDM is the number of the clusters. While the computational complexity of LDM is also linear in the number of samples, it involves clustering, thus in practice, its running time is much longer than our WInLDL, details can be referred to Figure 3.

Methods	Computational complexity
InLDL-a(p)	$\mathcal{O}(N^2C + C^3)$
EDL-LRL	$\mathcal{O}(N^2C + NkC + C^3 + k^2C^2)$
LDM	$\mathcal{O}(NC^3 + NkC + gC^4)$
WInLDL(ours)	$\mathcal{O}(max(Nk^2, NkC) + k^3)$

Table 1: Computational complexities of different methods.

3.4 Theoretical Analyses

To make theoretical explanations of why the proposed WInLDL could work without any explicit regularization, we first define a weighted empirical risk. Formally, the definition is detailed as follows.

Definition 1. Given a training dataset $S = \{(\mathbf{x}_i, \mathbf{d}_{\mathbf{x}_i})\}_{i=1}^N$, a class of functions $\mathcal{F}$, a loss function $\mathcal{L}$, and a weighted matrix $\mathbf{P} \in \mathbb{R}^{N\times C}$, the weighted empirical risk of function $f \in \mathcal{F}$ can be defined as:

$$\hat{\mathcal{R}}_S(f) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}\mathbf{P}_{ij}\mathcal{L}(f(\mathbf{x}_i)_j, \mathbf{D}_{ij}), \tag{14}$$

where $f(\mathbf{x}_i)_j$ denotes the j-th element of $f(\mathbf{x}_i)$. Suppose the population data follow an underlying probability distribution $\mathcal{D}$, then the expected risk can be written as:

$$\mathcal{R}_{\mathcal{D}}(f) = \mathbb{E}_{S\sim\mathcal{D}}[\hat{\mathcal{R}}_S(f)] \tag{15}$$

Before deriving the risk bound, we also provide the definition of empirical Rademacher complexity and Rademacher complexity for self-containment.

Definition 2. [Koltchinskii, 2001; Bartlett and Mendelson, 2002] Let $\mathcal{F}$ be a class of functions, $S = \{(\mathbf{x}_i, \mathbf{d}_{\mathbf{x}_i})\}_{i=1}^N$ be a fixed size dataset with N samples, and $\mathcal{L}$ be a loss function. Then, the empirical Rademacher complexity of $\mathcal{F}$ with respect to the sample set S is defined as:

$$\hat{\mathfrak{R}}_S(\mathcal{F}) = \mathbb{E}_{\boldsymbol{\sigma}}[\sup_{f\in\mathcal{F}}\frac{1}{N}\sum_{i=1}^{N}\sigma_i\mathcal{L}(f(\mathbf{x}_i), \mathbf{d}_{\mathbf{x}_i})], \tag{16}$$

where $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \cdots, \sigma_N)^\top$, with independent rademacher random variables σ_is uniformly taking values in $\{-1, +1\}$. Then the Rademacher complexity of $\mathcal{F}$ is the expectation of the empirical Rademacher complexity over all samples of size N drawn according to the distribution $\mathcal{D}$:

$$\mathfrak{R}_N(\mathcal{F}) = \mathop{\mathbb{E}}_{S \sim \mathcal{D}^N} [\hat{\mathfrak{R}}_S(\mathcal{F})]. \tag{17}$$

In the following, we will derive an upper bound between the expected risk and the weighted empirical risk with the help of Rademacher complexity.

Theorem 1. Let $\mathcal{F}$ be a class of functions, $\mathcal{L} = \sum_{j=1}^{C} \ell(f(\mathbf{x})_j, d_{\mathbf{x}}^{y_j})$ is a loss function, where ℓ is bounded by a constant B, and $\|\mathbf{P}\|_\infty = \max_{ij} \mathbf{P}_{ij}$. Then, $\forall \delta > 0$, with probability at least $1 - \delta$ over the draw of an *i.i.d.* sample S of size N from the distribution $\mathcal{D}$, the following bound holds for all $f \in \mathcal{F}$:

$$\mathcal{R}_{\mathcal{D}}(f) \leq \hat{\mathcal{R}}_S(f) + 2\|\mathbf{P}\|_\infty \mathfrak{R}_N(\mathcal{F}) + CB\|\mathbf{P}\|_\infty \sqrt{\frac{log\frac{1}{\delta}}{2N}}. \tag{18}$$

The proof of this theorem primarily relies on the McDiarmid inequality [McDiarmid and others, 1989], and details can be found in the Appendix. Notably, the above bound can be seen as a generalization of the single-label and unweighted cases. When $C = 1$ (single-label), and $\mathbf{P}$ is a matrix with all ones (unweighted), Eq. (18) degenerates into the classical form in [Bartlett and Mendelson, 2002; Shalev-Shwartz and Ben-David, 2014; Mohri *et al.*, 2018]. Moreover, the derived upper bound between the expected risk and the weighted empirical risk explicitly depends on the weighted matrix $\mathbf{P}$, and the generalization error can be bounded by this weighted matrix, which implies that the weighting plays an implicit regularization role since no explicit regularization is really imposed. Theorem 1 provides a rational explanation why our WInLDL could work even without any explicit regularization.

Furthermore, let $\mathcal{F}$ be a linear function class and assume that $\mathcal{L}$ is Lipschitz continuous, then we can derive the following theorem.

Theorem 2. Let $\mathcal{F}$ be a linear function class with a bounded linear transformation $\mathbf{W}$, defined as $\mathcal{F} = \{\mathbf{x} \mapsto \mathbf{W}\mathbf{x} : \|\mathbf{W}\|_F \leq B_0\}$, where $\|\bullet\|_F$ is the Frobenius norm. Assume $\mathcal{L} = \sum_{j=1}^{C} \ell(f(\mathbf{x})_j, d_{\mathbf{x}}^{y_j})$, where ℓ is a Lipschitz continuous loss function with Lipschitz constant L, and ℓ is bounded by a constant B. $\|\mathbf{P}\|_\infty = \max_{ij} \mathbf{P}_{ij}$. Then, $\forall \delta > 0$, with probability at least $1 - \delta/2$ over the draw of an *i.i.d.* sample S of size N from the distribution $\mathcal{D}$, the following bound holds for all $f \in \mathcal{F}$:

$$\begin{aligned}\mathcal{R}_{\mathcal{D}}(f) \leq \hat{\mathcal{R}}_S(f) &+ \frac{2\sqrt{2}LB_0C^{\frac{1}{2}}}{\sqrt{N}} \max_i \|\mathbf{x}_i\|_2 \|\mathbf{P}\|_\infty \\ &+ 3CB\|\mathbf{P}\|_\infty \sqrt{\frac{log\frac{2}{\delta}}{2N}}.\end{aligned} \tag{19}$$

Theorem 2 can be proven by leveraging the main results of [Maurer, 2016], and the detailed proof is provided in the Appendix. By further assuming $\mathcal{L}$ is ℓ_∞ continuous, another relevant bound can be obtained, interested readers can refer to [Foster and Rakhlin, 2019] for details.

4 Experiments

4.1 Experiments Settings

Datasets. In this paper, we use 10 real datasets covering fields of biology, natural scene recognition, facial expression, movie-rating, and image visual sentiment. The statistics of the datasets are summarized in Table 2. The first five datasets are collected by Geng [Geng, 2016], the sixth to tenth datasets are from [Peng *et al.*, 2015], [Liang *et al.*, 2018], [Yang *et al.*, 2017], [Li and Deng, 2019], [Xie *et al.*, 2015], respectively.

Datasets	#Samples(N)	#Features(k)	#Labels(C)
Gene	17892	36	68
Movie	7755	1869	5
Scene	2000	294	9
SBU3DFE	2500	243	6
SJAFFE	213	243	6
Emotion6	1980	1000	7
Fbp5500	5500	512	5
Flickr	11150	200	8
RAF_ML	4908	200	6
SCUTFBP	1500	300	5

Table 2: Statistics of the datasets.

Compared methods. We compare WInLDL with six methods, including two baselines named BFGS-LDL [Geng, 2016] and IIS-LDL [Geng, 2016], and four SOTA methods named InLDL-p [Xu and Zhou, 2017], InLDL-a [Xu and Zhou, 2017], EDL-LRL [Jia *et al.*, 2019b], and LDM [Wang and Geng, 2023]. BFGS-LDL and IIS-LDL are two maximum entropy models optimized with the BFGS [Fletcher, 2013] and the IIS [Della Pietra *et al.*, 1997] algorithm. InLDL-p and InLDL-a are two InLDL models that optimized by the proximal gradient descend and ADMM algorithms, respectively. EDL-LRL assumes local low-rank structure on clusters of samples, and LDM exploits both the global and local label correlations. All the compared methods consider the incomplete setting in the LDM paper. Codes of the compared methods are shared by the original authors, and the best parameters suggested by their papers are used.

Evaluation metrics. Five commonly used metrics are applied to evaluate the performance in this paper, including *Cosine*, *Intersection*, *Chebyshev*, *Clark*, and *Canberra*. The first two compute the similarity between two vectors, thus they are the higher the better, whereas the last three quantify the distance between two vectors, thus they are the lower the better. For two vectors $\mathbf{p}, \mathbf{q} \in \mathbb{R}^C$, the definitions of the five metrics are listed as below. 1) *Cosine* ↑: $\boldsymbol{p}^\top \boldsymbol{q}/(\|\boldsymbol{p}\|_2\|\boldsymbol{q}\|_2)$; 2) *Intersection* ↑: $\sum_i \min(p_i, q_i)$; 3) *Chebyshev* ↓: $\max_i |p_i - q_i|$; 4) *Clark* ↓: $\sqrt{\sum_i (p_i - q_i)^2 / (p_i + q_i)^2}$; 5) *Canberra* ↓: $\sum_i |p_i - q_i| / (p_i + q_i)$, where ↑ means the higher the better, and ↓ means the lower the better. Here we omit the KL-divergence metric just as in [Xu and Zhou, 2017], since KL-divergence is calculated by $log(d_{\mathbf{x}}^y / \hat{d}_{\mathbf{x}}^y)$, and in InLDL, the $\hat{d}_{\mathbf{x}}^y$

Datasets	WInLDL	LDM	EDL-LRL	InLDL-p	InLDL-a	BFGS-LDL	IIS-LDL
Gene	**0.8359(.0044)**	0.8303(.0044)	0.8082(.0080)	0.8297(.0044)	0.8285(.0044)	0.8322(.0044)	0.8314(.0044)
Movie	**0.9351(.0012)**	0.9031(.0040)	0.8428(.0064)	0.9078(.0028)	0.9050(.0028)	0.8956(.0011)	0.8924(.0020)
Scene	**0.7513(.0080)**	0.6159(.0132)	0.5598(.0265)	0.7033(.0084)	0.6958(.0057)	0.6359(.0073)	0.6620(.0065)
SBU3DFE	**0.9430(.0021)**	0.9174(.0022)	0.9161(.0016)	0.9249(.0035)	0.9239(.0033)	0.9165(.0020)	0.9145(.0020)
SJAFFE	**0.9494(.0021)**	0.9312(.0037)	0.9326(.0038)	0.9078(.0047)	0.9344(.0020)	0.9299(.0064)	0.9262(.0047)
Emotion6	**0.8025(.0033)**	0.6711(.0168)	0.2485(.0142)	0.6654(.0051)	0.6561(.0067)	0.4103(.0111)	0.5465(.0100)
Fbp5500	0.9455(.0011)	0.9443(.0017)	0.6482(.0255)	**0.9512(.0014)**	0.9500(.0015)	0.7950(.0031)	0.8881(.0024)
Flickr	**0.8335(.0032)**	0.8142(.0015)	0.8224(.0017)	0.8319(.0024)	0.8306(.0025)	0.8286(.0028)	0.7602(.0033)
RAF_ML	0.8874(.0021)	0.8797(.0033)	**0.9229(.0017)**	0.8892(.0013)	0.8975(.0014)	0.8300(.0048)	0.7200(.0061)
SCUTFBP	**0.8151(.0062)**	0.6607(.0046)	0.6119(.0444)	0.5822(.0120)	0.5629(.0091)	0.6523(.0073)	0.6529(.0065)

Table 3: *Cosine*↑ (the higher the better) results in the **non-random missing scenario** (at 50% missing rate). Values in parentheses are standard deviations. The best result is in bold and the second best result is underlined.

Datasets	WInLDL	LDM	EDL-LRL	InLDL-p	InLDL-a	BFGS-LDL	IIS-LDL
Gene	**0.8356(.0044)**	0.8355(.0038)	0.8350(.0040)	0.8352(.0043)	0.8353(.0043)	0.8331(.0044)	0.8338(.0044)
Movie	**0.9351(.0013)**	0.9329(.0017)	0.8517(.0125)	0.8824(.0019)	0.8886(.0017)	0.8475(.0016)	0.8536(.0056)
Scene	**0.7418(.0085)**	0.7291(.0040)	0.6577(.0062)	0.6955(.0052)	0.6956(.0069)	0.6320(.0065)	0.6613(.0064)
SBU3DFE	**0.9417(.0021)**	0.9224(.0018)	0.9191(.0022)	0.9329(.0035)	0.9335(.0036)	0.9170(.0020)	0.9189(.0020)
SJAFFE	**0.9517(.0043)**	0.9341(.0018)	0.9343(.0019)	0.9344(.0020)	0.9037(.0102)	0.9340(.0019)	0.9309(.0040)
Emotion6	**0.7961(.0045)**	0.7094(.0087)	0.4232(.0184)	0.6119(.0098)	0.6090(.0085)	0.4103(.0111)	0.5029(.0121)
Fbp5500	0.9445(.0013)	0.9469(.0015)	0.7402(.0433)	0.9482(.0019)	**0.9485(.0018)**	0.7850(.0047)	0.8869(.0028)
Flickr	**0.8315(.0029)**	0.8097(.0020)	0.7486(.0114)	0.8304(.0024)	0.8307(.0024)	0.7537(.0089)	0.7431(.0034)
RAF_ML	0.8831(.0016)	0.8720(.0036)	**0.9153(.0048)**	0.8733(.0021)	0.8828(.0013)	0.7807(.0055)	0.6867(.0061)
SCUTFBP	**0.8138(.0054)**	0.6575(.0085)	0.6435(.0060)	0.5663(.0104)	0.5909(.0087)	0.6447(.0078)	0.6531(.0053)

Table 4: *Cosine*↑ (the higher the better) results in the **random missing scenario** (at 50% missing rate). Values in parentheses are standard deviations. The best result is in bold and the second best result is underlined.

may be zero, which makes the KL-divergence meaningless.

Two incomplete scenarios. We conduct experiments in two different incomplete scenarios. The first one is the non-random missing scenario, i.e., 50% of the smallest degrees are missing. For fair comparisons, we also consider the second scenario, i.e., 50% of the degrees are uniformly random missing, just as in the compared InLDL work [Xu and Zhou, 2017]. Each method is run for five times with five random data partitions, and for each partitions, 80% of the data are used for training, and the remaining 20% are used for testing. In each trial, all methods are run with exactly the same missing and partitioned dataset. Finally, both the mean and the standard deviation of the results are reported [1].

4.2 Results of the Non-Random Missing Scenario

In this subsection, we report the results of different methods in the non-random missing scenario (50% of the smallest degrees are missing) on ten real datasets. Due to the page limitations, here we only list the results of the *Cosine* ↑ metric, and results of other metrics can be found in the Appendix.

From Table 3 we can discover that our WInLDL ranks first on eight datasets, and the average rank is 1.5. Overall, we win 55 times out of 60 comparisons, with a 91.67% rate to win. Besides, we also conduct the Nemenyi test [Nemenyi, 1963; Demšar, 2006] as the statistic significance test, due to the page limitations, details can be found in the Appendix.

4.3 Results of the Random Missing Scenario

For fair comparisons, we also report the results of different methods in the random missing scenario (50% of the degrees are uniformly random missing) as in the compared methods.

Table 4 has shown that our WInLDL ranks first on eight datasets and second on one dataset, and the average rank is 1.4. Overall, we win 57 times out of 60 comparisons, with a 95% rate to win.

From Tables 3 and 4, we can conclude that WInLDL achieves better performance in most cases, which verifies its effectiveness in addressing the InLDL problem in both random missing and non-random missing scenarios. The reason may be attributed to the weighting scheme adopted by WInLDL, which can better solve the issue that small degrees are easily overlooked, whereas other methods either only focus on mining the correlations between labels or merely adopt an entropy maximization strategy to learn the label distribution, neither of them can well address the above issue, thus leading to the suboptimal performance.

4.4 Different Weighting Schemes

To verify the effectiveness of imposing large weights on the small degrees, we conduct experiments on five different weighted schemes and list the results in Table 5. The formal definitions are: (1) InLDL-U: $\mathbf{Q}_{ij} = 1$, if $(i,j) \in \Omega$, and $\mathbf{Q}_{ij} = 0$, if $(i,j) \in U$; (2) InLDL-I: $\mathbf{Q}_{ij} = \widetilde{\mathbf{D}}_{ij}$, if $(i,j) \in \Omega$, and $\mathbf{Q}_{ij} = \mathbf{D}_{U_{ij}}$, if $(i,j) \in U$; (3) InLDL-II:

[1] Code is available at https://github.com/EverFAITH/WInLDL

Datasets	WInLDL	InLDL-U	InLDL-I	InLDL-II	InLDL-Rand
Gene	**0.8356(.0044)**	0.8350(.0044)	0.7894 (.0055)	0.8351(.0044)	0.8349(.0044)
Movie	**0.9351(.0013)**	0.9349(.0014)	0.9110 (.0078)	0.9196 (.0008)	0.9161 (.0010)
Scene	**0.7418(.0085)**	0.7343(.0059)	0.5548 (.0113)	0.7136 (.0065)	0.7065 (.0078)
SBU3DFE	0.9417(.0021)	**0.9426(.0017)**	0.9327 (.0024)	0.9364(.0017)	0.9303 (.0022)
SJAFFE	0.9517(.0043)	**0.9555(.0043)**	0.9459 (.0065)	0.9160 (.0046)	0.9086 (.0083)
Emotion6	**0.7961(.0045)**	0.7929 (.0048)	0.7305 (.0080)	0.7786 (.0022)	0.7619 (.0037)
Fbp5500	**0.9445(.0013)**	0.9436 (.0014)	0.9396 (.0008)	0.9192 (.0018)	0.8977 (.0026)
Flickr	0.8315(.0029)	**0.8321(.0028)**	0.8107 (.0037)	0.8134 (.0027)	0.7880 (.0030)
RAF_ML	**0.8831(.0016)**	0.8798 (.0020)	0.8661 (.0038)	0.8682 (.0025)	0.8460 (.0017)
SCUTFBP	**0.8138(.0054)**	0.8137 (.0039)	0.8009 (.0087)	0.7952 (.0105)	0.7791 (.0108)

Table 5: *Cosine*↑ (the higher the better) results for five different weighting schemes. Values in parentheses are standard deviations. The best result is in bold and the second best result is underlined.

$\mathbf{Q}_{ij} = 2^{\widetilde{\mathbf{D}}_{ij}}$, if $(i,j) \in \Omega$, and $\mathbf{Q}_{ij} = a^{\mathbf{D}_{U_{ij}}}$, if $(i,j) \in U$, $a = 1 + iter/maxIter$; (4) InLDL-Rand: $\mathbf{Q} = \mathbf{Ra}$, where $\mathbf{Ra}$ is a random matrix whose entries are uniformly distribute in $(0,1)$, for WInLDL, referring to Eq. (7). First, the results show that WInLDL consistently performs better than InLDL-I, InLDL-II, and InLDL-Rand, where InLDL-I and InLDL-II impose large weights on large degrees, and InLDL-Rand adopts random weighting. These comparisons demonstrate that imposing large weights on the small degrees is effective. Besides, WInLDL wins 7 times out of 10 comparisons with InLDL-U. Note that, some LDL datasets are transformed from multi-label datasets, thus the ground truth of some degrees may be 0, and in the setting of InLDL, the missing degrees are also set to 0. In such a situation, WInLDL may put too much emphasis on the degrees whose ground truth is 0, while InLDL-U happens to treat these missing degrees as 0, which may explain why WInLDL is inferior to InLDL-U on some datasets.

4.5 Running Time Comparisons

In this subsection, we compare the running time of the different methods and report the total time for ten datasets. All the methods are running on a Linux server with an Intel Xeon(R) W-2255 3.70GHz CPU and 64GB memory. The running time of our WInLDL is 10.95 seconds, which is orders of magnitude faster than most of the compared methods and verifies the efficiency of WInLDL.

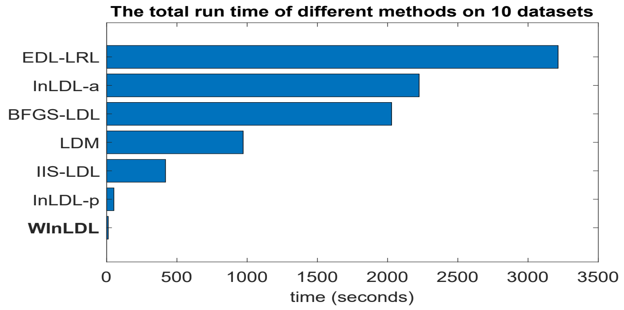
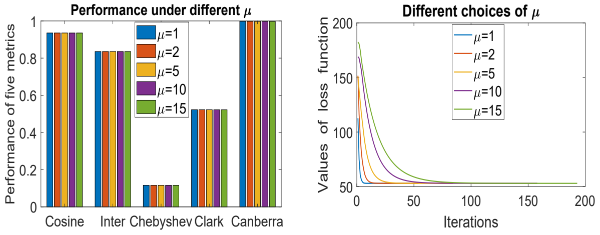


Figure 3: The total runtime of different methods on ten real datasets, where WInLDL is our method.

4.6 Impacts Study of μ

In this subsection, we conduct experiments to confirm that the parameter μ in the ADMM algorithm does not change the performance of our method. From Figure 4(a) we find that all the five metrics remain the same regardless of variations in μ, and from Figure 4(b) we can see that μ only affects the convergence rate. In our WInLDL method, we fix μ at 2, eliminating the need for tuning. Consequently, μ does not impose any additional burden.



(a) Performance. (b) Convergence.

Figure 4: The performance of all five metrics and the convergence rates under different μ.

5 Conclusion

We propose an efficient, effective, and easy-to-implement method called WInLDL without any explicit regularization by properly utilizing the label distribution prior. More importantly, we present two upper bounds between the expected risk and weighted empirical risk, which explicitly depend on the weighted matrix. Furthermore, the generalization error can be bounded by the weighted matrix, implying that such weighting plays an implicit regularization role and may explain why WInLDL still works without any explicit regularization. Besides, by conducting extensive experiments, we have verified that WInLDL achieves better performance in both random missing and non-random missing scenarios. Finally, a promising research in future work would be the nonlinearization of the proposed method and its corresponding theoretical study, potentially through its kernelized version, which may overcome the limitations of the linear model and uncover the complex non-linear relationships.

Acknowledgments

This work is supported by NSFC under Grant No. 62376126.

References

[Bartlett and Mendelson, 2002] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[Boyd *et al.*, 2011] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[Chen *et al.*, 2020] Shikai Chen, Jianfeng Wang, Yuedong Chen, Zhongchao Shi, Xin Geng, and Yong Rui. Label distribution learning on auxiliary label space graphs for facial expression recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13984–13993, 2020.

[Condat, 2016] Laurent Condat. Fast projection onto the simplex and the l 1 ball. *Mathematical Programming*, 158(1-2):575–585, 2016.

[Della Pietra *et al.*, 1997] Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. Inducing features of random fields. *IEEE transactions on pattern analysis and machine intelligence*, 19(4):380–393, 1997.

[Demšar, 2006] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7:1–30, 2006.

[Fletcher, 2013] Roger Fletcher. *Practical methods of optimization.* John Wiley & Sons, 2013.

[Foster and Rakhlin, 2019] Dylan J Foster and Alexander Rakhlin. ℓ_∞ vector contraction for rademacher complexity. *arXiv preprint arXiv:1911.06468*, 6, 2019.

[Gao *et al.*, 2018] Bin-Bin Gao, Hong-Yu Zhou, Jianxin Wu, and Xin Geng. Age estimation using expectation of label distribution learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 712–718, 2018.

[Geng *et al.*, 2021] Xin Geng, Renyi Zheng, Jiaqi Lv, and Yu Zhang. Multilabel ranking with inconsistent rankers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5211–5224, 2021.

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

[Hou *et al.*, 2017] Peng Hou, Xin Geng, Zeng-Wei Huo, and Jia-Qi Lv. Semi-supervised adaptive label distribution learning for facial age estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[Jia *et al.*, 2019a] Xiuyi Jia, Tingting Ren, Lei Chen, Jun Wang, Jihua Zhu, and Xianzhong Long. Weakly supervised label distribution learning based on transductive matrix completion with sample correlations. *Pattern Recognition Letters*, 125:453–462, 2019.

[Jia *et al.*, 2019b] Xiuyi Jia, Xiang Zheng, Weiwei Li, Changqing Zhang, and Zechao Li. Facial emotion distribution learning by exploiting low-rank label correlations locally. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9841–9850, 2019.

[Jia *et al.*, 2021] Xiuyi Jia, Tao Wen, Weiping Ding, Huaxiong Li, and Weiwei Li. Semi-supervised label distribution learning via projection graph embedding. *Information Sciences*, 581:840–855, 2021.

[Koltchinskii, 2001] Vladimir Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5):1902–1914, 2001.

[Li and Deng, 2019] Shan Li and Weihong Deng. Blended emotion in-the-wild: Multi-label facial expression recognition using crowdsourced annotations and deep locality feature learning. *International Journal of Computer Vision*, 127(6-7):884–906, 2019.

[Li *et al.*, 2022] Weiwei Li, Jin Chen, Yuqing Lu, and Zhiqiu Huang. Filling missing labels in label distribution learning by exploiting label-specific feature selection. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.

[Liang *et al.*, 2018] Lingyu Liang, Luojun Lin, Lianwen Jin, Duorui Xie, and Mengru Li. Scut-fbp5500: A diverse benchmark dataset for multi-paradigm facial beauty prediction. In *2018 24th International conference on pattern recognition (ICPR)*, pages 1598–1603. IEEE, 2018.

[Lu *et al.*, 2023] Yunan Lu, Weiwei Li, Huaxiong Li, and Xiuyi Jia. Predicting label distribution from tie-allowed multi-label ranking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[Maurer, 2016] Andreas Maurer. A vector-contraction inequality for rademacher complexities. In *Algorithmic Learning Theory: 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings 27*, pages 3–17. Springer, 2016.

[McDiarmid and others, 1989] Colin McDiarmid et al. On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188, 1989.

[Mohri *et al.*, 2018] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[Nemenyi, 1963] Peter Bjorn Nemenyi. *Distribution-free multiple comparisons.* Princeton University, 1963.

[Peng *et al.*, 2015] Kuan-Chuan Peng, Tsuhan Chen, Amir Sadovnik, and Andrew C Gallagher. A mixed bag of emotions: Model, predict, and transfer emotion distributions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 860–868, 2015.

[Qian *et al.*, 2022a] Wenbin Qian, Ping Dong, Shiming Dai, Jintao Huang, and Yinglong Wang. Incomplete label distribution feature selection based on neighborhood-tolerance discrimination index. *Applied Soft Computing*, 130:109693, 2022.

[Qian *et al.*, 2022b] Wenbin Qian, Ping Dong, Yinglong Wang, Shiming Dai, and Jintao Huang. Local rough set-based feature selection for label distribution learning with incomplete labels. *International Journal of Machine Learning and Cybernetics*, 13(8):2345–2364, 2022.

[Rupprecht *et al.*, 2017] Christian Rupprecht, Iro Laina, Robert DiPietro, Maximilian Baust, Federico Tombari, Nassir Navab, and Gregory D Hager. Learning in an uncertain world: Representing ambiguity through multiple hypotheses. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3591–3600, 2017.

[Shalev-Shwartz and Ben-David, 2014] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[Teng and Jia, 2021] Qifa Teng and Xiuyi Jia. Incomplete label distribution learning by exploiting global sample correlation. In *Multimedia Understanding with Less Labeling on Multimedia Understanding with Less Labeling*, pages 9–16. 2021.

[Wang and Carreira-Perpinán, 2013] Weiran Wang and Miguel A Carreira-Perpinán. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *arXiv preprint arXiv:1309.1541*, 2013.

[Wang and Geng, 2023] Jing Wang and Xin Geng. Label distribution learning by exploiting label distribution manifold. *IEEE Transactions on Neural Networks and Learning Systems*, 34(2):839–852, 2023.

[Wang *et al.*, 2022] Jing Wang, Xin Geng, and Hui Xue. Reweighting large margin label distribution learning for classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5445–5459, 2022.

[Xie *et al.*, 2015] Duorui Xie, Lingyu Liang, Lianwen Jin, Jie Xu, and Mengru Li. Scut-fbp: A benchmark dataset for facial beauty perception. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1821–1826. IEEE, 2015.

[Xu and Zhou, 2017] Miao Xu and Zhi-Hua Zhou. Incomplete label distribution learning. In *International Joint Conferences on Artificial Intelligence*, pages 3175–3181, 2017.

[Yang *et al.*, 2017] Jufeng Yang, Ming Sun, and Xiaoxiao Sun. Learning visual sentiment distributions via augmented conditional probability neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[Zhang *et al.*, 2022] Jing Zhang, Hong Tao, Tingjin Luo, and Chenping Hou. Safe incomplete label distribution learning. *Pattern Recognition*, 125:108518, 2022.