Label Distribution Learning with Label Correlations via Low-Rank Approximation

Tingting Ren[1] , **Xiuyi Jia**[1,2,3*] , **Weiwei Li**[4] and **Shu Zhao**[5]
[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, China
[2]Jiangsu Key Laboratory of Big Data Security & Intelligent Processing, Nanjing University of Posts and Telecommunications, China
[3]State Key Laboratory for Novel Software Technology, Nanjing University, China
[4]College of Astronautics, Nanjing University of Aeronautics and Astronautics, China
[5]School of Computer Science and Technology, Anhui University, China

Abstract

Label distribution learning (LDL) can be viewed as the generalization of multi-label learning. This novel paradigm focuses on the relative importance of different labels to a particular instance. Most previous LDL methods either ignore the correlation among labels, or only exploit the label correlations in a global way. In this paper, we utilize both the global and local relevance among labels to provide more information for training model and propose a novel label distribution learning algorithm. In particular, a label correlation matrix based on low-rank approximation is applied to capture the global label correlations. In addition, the label correlation among local samples are adopted to modify the label correlation matrix. The experimental results on real-world data sets show that the proposed algorithm outperforms state-of-the-art LDL methods.

1 Introduction

Most machine learning algorithms can be viewed as building a mapping from the sample space to the label space. There are two main traditional integrated paradigms for establishing a mapping: single-label learning (SLL) and multi-label learning (MLL) [Tsoumakas *et al.*, 2007]. The former assumes that each instance is associated with a single label, while the latter considers that each instance may have a variety of semantic meanings and be related to a set of labels simultaneously. However, both SLL and MLL can only solve the issue of "*what describes the instance*"; and cannot determine the relative importance of each label to a particular instance. Sometimes, we are more concerned about the intensity of each label, to determine "*how to describe the instance*". To solve this problem, a novel learning paradigm named label distribution learning (LDL) was proposed [Geng *et al.*, 2010]. Different from MLL to output a set of labels, LDL learns a label distribution. Each component in the distribution represents the relevance of the corresponding label to the instance.

*Corresponding Author: Xiuyi Jia (jiaxy@njust.edu.cn).

Since the label distribution can be viewed as a type of label relations, the correlation among labels should not be ignored in the learning process. For example, in image annotations, assuming that *sky* and *cloud* are strongly related, an image is more likely to be correlated with *cloud*, when it is strongly correlated with *sky*. In view of this, some researchers have considered label correlations in LDL. Zhou et al. [2015] captured the label correlation by using Pearson's correlation coefficient for facial emotion recognition. Based on the Plutchik's theory, Zhou et al. [2016] explored the label correlation for text emotion distribution learning. Moreover, the label correlation was encoded into a distance to measure the similarity of any two labels [Jia *et al.*, 2018]. For incomplete LDL problem, a low-rank structure was employed to capture the label correlations [Xu and Zhou, 2017]. However, all above works exploited the label correlations in a global way. In reality, label correlations are usually appeared in a local way. For example, in text annotations, the relevance between *Puma* and *animal* appears in journals of ecology and environment with a high probability, while *Puma* is usually related to *brand* in fashion magazines.

To solve above problems, we try to learn the global and the local label correlations simultaneously. Furthermore, the global correlation is captured through a low-rank approximation and the local correlation is concerned by clustering, in which the samples with similar discrimination will be clustered into one group. Then, we can predict the label distributions for unknown instances based on the high-order label correlations encoded in one correlation matrix.

Our work was inspired by ML-LRC [Xu *et al.*, 2014]. There are two major differences between our method and ML-LRC: (1) ML-LRC only considers the global label correlation. However, as we mentioned, some correlations are only shared by a part of instances. In view of this, in addition to introducing a correlation matrix to utilize the global label correlation, we also try to learn the local label correlations simultaneously. (2) ML-LRC is mainly concerned about the traditional multi-label learning to learn a set of related labels for each instance. Our work tries to propose a novel algorithm in label distribution learning, which places more emphasis on how relevant each label is to a particular instance, rather than focusing on what label is relevant to a specific instance only.

The main contribution of this paper is three folds:

- We propose a novel method to deal with LDL problems by considering global correlations and local correlations simultaneously.
- One correlation matrix is introduced to encode the high-order correlation, in which the global correlation is utilized by low-rank approximation while the local label correlation is exploited by clustering.
- We assume that if the distance between two label distributions is small enough, they should be strongly related to each other.

2 Related Works

The LDL framework can be viewed as the generalization of MLL. It might provide more information than traditional paradigms on model learning. In recent years, LDL has been successfully applied in many applications. This novel learning paradigm was usually exploited to estimate facial age [Gao *et al.*, 2018; Geng *et al.*, 2010; He *et al.*, 2017]. A multivariate label distribution algorithm was used to estimate the head pose [Geng and Xia, 2014]. Considering that crowd images with adjacent class labels contain similar features, the crowd counting problem can be transformed into an LDL problem [Zhang *et al.*, 2015]. Furthermore, the LDL paradigm was applied to identify multiple emotions and their intensities from texts [Zhou *et al.*, 2016]. A deep label distribution learning algorithm by combining LDL with deep learning was proposed to handle the problem of apparent age estimation [Gao *et al.*, 2017].

Existing LDL algorithms can be divided into three categories: problem transformation, algorithm adaptation and specialized algorithms. Problem transformation based algorithms transform the LDL problem into traditional problems and then use existing learners to solve it, such as PT-SVM and PT-Bayes [Geng, 2016]. The main idea of algorithm adjustment is to adapt traditional algorithms to fit for LDL paradigm, such as AA-BP [Geng, 2016] and logistic boosting regression method [Xing *et al.*, 2016]. In addition, there are some specialized algorithms designed for LDL can directly model the relative importance of each label to the particular instance [Geng *et al.*, 2014; Geng *et al.*, 2013; Zhao and Zhou, 2018].

To construct a specialized LDL algorithm, we need to consider three parts: the objective function, the output model and the optimization method. Difference between the ground-truth distribution and the predicted distribution can be measured by some distances which can be viewed as objective functions, such as Kullback-Leibler Divergence [Geng *et al.*, 2014; Huo *et al.*, 2016; Jia *et al.*, 2018] and Jeffery divergence [Zhou *et al.*, 2015]. The LDL algorithms mainly use the maximum entropy model as the output model [Geng, 2016; Zheng *et al.*, 2018]. To solve the optimization problem, many methods have been applied to LDL. For example, the improved iterative scaling method can be used as the optimization method [Geng *et al.*, 2010], so does the limited-memory quasi-Newton method (L-BFGS) [Zhou *et al.*, 2016].

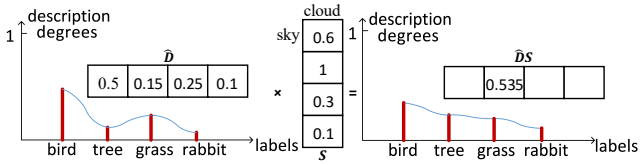


Figure 1: Illustration of global label correlations. $\hat{D}S$ is the reconstructed label distribution.

3 The LDL-LCLR Algorithm

3.1 Framework

The main goal of LDL is to learn a mapping from the input space $X = [x_1; x_2; \cdots; x_n] \in R^{n \times d}$ to the label distribution over a finite set of labels $Y = \{y_1, y_2, \cdots, y_l\}$, where n is the number of instances and l is the number of labels. Given a training set $T = \{(x_1, D_1), (x_2, D_2), \cdots, (x_n, D_n)\}$, where $D_i = [d_{x_i}^{y_1}, d_{x_i}^{y_2}, \cdots, d_{x_i}^{y_l}]$ is the label distribution associated with x_i, we use the description degrees d_x^y, which satisfies $d_x^y \in [0, 1]$ and $\sum_y d_x^y = 1$, to indicate the importance of label y to instance x, where $x \in X$ and $y \in Y$. It is noticed that d_x^y is not the probability that y correctly labels x, but the proportion that y accounts for in a full description of x. The LDL approach can learn a conditional probability mass function $p(y|x)$ from T and use the learned model to predict the label distributions for unseen instances.

Without loss of generality, we choose the maximum entropy model as the output model. Then the component of the predicted label distribution $\hat{d}_{x_i}^{y_j}$, which is the description degree of the j-th label y to the i-th instance x, can be expressed as follows:

$$\hat{d}_{x_i}^{y_j} = p(y_j|x_i; w) = \frac{1}{\mathcal{Z}_i} \exp(\sum_r w_{jr} x_{ir}), \tag{1}$$

where w is the model parameter that needs to be learned and $\mathcal{Z}_i = \sum_j \exp(\sum_r w_{jr} x_{ir})$ is a normalization term to satisfy that the sum of all description degrees of an instance equals 1. The performance of Kullback-Leibler (KL) divergence is the most stable in the field of LDL according to comparison experiments [Zhao and Geng, 2017]. Therefore, we adopt the KL divergence as the basic objective function:

$$\min_W \sum_i D_{KL}(D_i||\hat{D}_i) + \lambda_1 \|W\|_F^2, \tag{2}$$

where $D_{KL}(D_i||\hat{D}_i) = D_i \ln \frac{D_i}{\hat{D}_i}$ is the KL divergence to measure the distance between the predicted label distribution $\hat{D}_i = \{\hat{d}_{x_i}^{y_1}, \hat{d}_{x_i}^{y_2}, \cdots, \hat{d}_{x_i}^{y_l}\}$ and the ground-truth label distribution D_i. The second term aims to prevent the model from over-fitting. λ_1 is the balance factor.

Inspired by the work in [Xu *et al.*, 2014], we adopt a correlation matrix S to model the dependencies among different labels. A simple example is shown in Figure 1, we can obtain that the relative importance of "*cloud*" is changed from 0.15 to 0.535 (It needs to be normalized) for the high correlation between "*cloud*" and "*sky*" shown in matrix S, i.e., the

reconstructed distribution matrix $\hat{D}S$ can provide more information and assign a new relevance of a label to a particular instance based on the global label correlations. To exploit the dependence among labels, Eq. (2) can be rewritten as Eq. (3):

$$\begin{aligned}&\min_{W,E,S}\sum_i D_{KL}(D_i||\hat{D}_i)+\lambda_1\|W\|_F^2+\lambda_2\|E\|_{2,1}+\lambda_3\|S\|_*\\&s.t.\quad D=\hat{D}S+E,\end{aligned}\tag{3}$$

where λ_2 and λ_3 are balance factors. The regularization term on E controls the difference between the reconstructed distribution $\hat{D}S$ and the ground-truth distribution D. Since labels are related to each other globally, one can be represented by a linear combination of other labels, which implies a low-rank structure of S. Considering that the low-rank function is difficult to optimize, we use the nuclear norm $\|\cdot\|_*$ as a convex approximation of the low-rank function.

Nevertheless, some label correlations are only shared by partial samples. Thus, we try to consider the local relevance among labels simultaneously. The k-means method is introduced to cluster samples into different groups. In one group, we assume that if the distance between two label distributions is small enough, they should be strongly related to each other. Then, the framework of LDL that exploits both the global and local correlation among labels can be formulated as:

$$\begin{aligned}&\min_{W,E,S}\sum_i D_{KL}(D_i||\hat{D}_i)+\lambda_1\|W\|_F^2+\lambda_2\|E\|_{2,1}+\lambda_3\|S\|_*\\&\qquad-\frac{1}{2}\lambda_4\sum_{v=1}^{k}\sum_{m=1}^{l}\sum_{n=1}^{l}S_{m,n}\|D^v_{\cdot m}-D^v_{\cdot n}\|_2^2\\&s.t.\quad D=\hat{D}S+E,\end{aligned}\tag{4}$$

where λ_4 is the balance factor and k is the number of clusters. $S_{m,n}$ is the correlation between the m-th label $D_{\cdot m}$ and the n-th label $D_{\cdot n}$. It is worth mentioning that for $S_{m,n}$, we want to find a larger correlation value for two correlated labels with a smaller distance, thus the last term in the objective function is a maximization problem.

3.2 Optimization

Since the optimization problem in Eq. (4) is convex, it can be optimized globally. In this section, we exploit the ADMM (Alternating Direction Method of Multipliers) [Boyd *et al.*, 2011], which can convert the multi-parameter problem into multiple single-parameter problems, to obtain the model parameter W.

Here, we introduce an auxiliary variable Z to make the objective function separable for the two non-smooth regularization terms in Eq. (4):

$$\begin{aligned}&\min_{W,E,S,Z}\sum_i D_{KL}(D_i||\hat{D}_i)+\lambda_1\|W\|_F^2+\lambda_2\|E\|_{2,1}\\&\qquad+\lambda_3\|Z\|_*-\frac{1}{2}\lambda_4\sum_{v=1}^{k}\sum_{m=1}^{l}\sum_{n=1}^{l}S_{m,n}\|D^v_{\cdot m}-D^v_{\cdot n}\|_2^2\\&s.t.\quad D=\hat{D}S+E, S-Z=0.\end{aligned}\tag{5}$$

Then, the Lagrange multiplier method is used to transform the constrained problem into an unconstrained problem:

$$\begin{aligned}&\min_{W,E,S,Z}\sum_i D_{KL}(D_i||\hat{D}_i)+\lambda_1\|W\|_F^2+\lambda_2\|E\|_{2,1}\\&\qquad+\lambda_3\|Z\|_*-\frac{1}{2}\lambda_4\sum_{v=1}^{k}\sum_{m=1}^{l}\sum_{n=1}^{l}S_{m,n}\|D^v_{\cdot m}-D^v_{\cdot n}\|_2^2\\&\qquad+\frac{\rho}{2}\|D-\hat{D}S-E\|_F^2+\frac{\rho}{2}\|S-Z\|_F^2\\&\qquad+\langle\Gamma_1,D-\hat{D}S-E\rangle+\langle\Gamma_2,S-Z\rangle,\end{aligned}\tag{6}$$

where Γ_1 and Γ_2 are the Lagrange multipliers, ρ is the penalty parameter and $\langle\cdot,\cdot\rangle$ is the Frobenius dot-product.

Eq. (6) can be solved by ADMM. Each iteration of ADMM involves updating one variable, with the other variables fixed to their most recent values. The following gives the updating rules of each parameter during one round of iteration.

To solve for W, Eq. (6) can be reduced to the following alternative methods,

$$\begin{aligned}W=\arg\min_W&\sum_i D_i\ln D_i-\sum_i D_i\ln\frac{\exp(\sum_r w_{jr}x_{ir})}{\sum_j\exp(\sum_r w_{jr}x_{ir})}\\&+\langle\Gamma_1,D-\hat{D}S-E\rangle+\frac{\rho}{2}\|D-\hat{D}S-E\|_F^2+\lambda_1\|W\|_F^2.\end{aligned}\tag{7}$$

In the same way, S can be solved by optimizing the following sub-problem,

$$\begin{aligned}S=\arg\min_S&<\Gamma_1,D-\hat{D}S-E>+\langle\Gamma_2,S-Z\rangle\\&+\frac{\rho}{2}\|D-\hat{D}S-E\|_F^2+\frac{\rho}{2}\|S-Z\|_F^2\\&-\frac{1}{2}\lambda_4\sum_{v=1}^{k}\sum_{m=1}^{l}\sum_{n=1}^{l}S_{m,n}\|D^v_{\cdot m}-D^v_{\cdot n}\|_2^2.\end{aligned}\tag{8}$$

Both Eq. (7) and Eq. (8) can be solved by the limited-memory quasi-Newton method effectively [Yuan, 1991]. The basic idea is to avoid explicit calculation of the inverse Hessian matrix, which is required in the Newton method. For the optimization of Eq. (7) and Eq. (8), the computation of L-BFGS is mainly related to the first-order gradient, which can be obtained by

$$\begin{aligned}\nabla W=&X^T(\hat{D}-D)+2\lambda_1W-X^T<\hat{D}-\hat{D}^2,\Gamma_1>S^T\\&-\rho X^T<\hat{D}-\hat{D}^2,D-\hat{D}S-E>S^T,\end{aligned}\tag{9}$$

$$\begin{aligned}\nabla S=&-\Gamma_1^T\hat{D}+\Gamma_2^T+\rho(S-Z)-\rho\hat{D}^T(D-\hat{D}S-E)\\&-\frac{1}{2}\lambda_4\sum_{v=1}^{k}\sum_{m=1}^{l}\sum_{n=1}^{l}\|D^v_{\cdot m}-D^v_{\cdot n}\|_2^2.\end{aligned}\tag{10}$$

Similarly, E and Z can be obtained by solving the problems as follows:

$$\begin{aligned}E=\arg\min_E&\lambda_2\|E\|_{2,1}+<\Gamma_1,D-\hat{D}S-E>\\&+\frac{\rho}{2}\|D-\hat{D}S-E\|_F^2\end{aligned}\tag{11}$$

Algorithm 1: The LDL-LCLR Framework

Initialization: W, S, E, Z, Γ_1, Γ_2, ρ, λ_1, λ_2, λ_3, λ_4
While stopping criterion is not satisfied **Do**
  update W by solving (7) using L-BFGS
  update S by solving (8) using L-BFGS
  update E by solving (11)
  update Z by solvign (12)
  update Γ_1, Γ_2 by (13) and (14)
End While

$$Z = \arg\min_{Z} \lambda_3 \|Z\|_* + < \Gamma_2, S - Z > + \frac{\rho}{2}\|S - Z\|_F^2. \tag{12}$$

Both Eq. (11) and Eq. (12) have a closed solution [Liu *et al.*, 2010; Cai *et al.*, 2010]. The multipliers Γ_1 and Γ_2 can be updated directly by

$$\Gamma_1 = \Gamma_1 + \rho(D - \hat{D}S - E), \tag{13}$$

$$\Gamma_2 = \Gamma_2 + \rho(S - Z). \tag{14}$$

The overall procedure of Label Distribution Learning with Label Correlations via Low-Rank approximation (LDL-LCLR) is summarized in Algorithm 1.

4 Experiments

In this part, we will evaluate the proposed method on 15 real-world data sets with seven state-of-the-art LDL approaches over six different measurements. These 15 data sets [Geng, 2016] cover the fields of natural scene recognition, biological information classification and emotional analysis, among others. Six different measures [Geng, 2016] are used to evaluate the performances of the LDL algorithms, i.e., *Sørensen*, *Squared-chord*, *Kullback-Leibler*, *Chebyshev*, *Intersection* and *Cosine*.

4.1 Experimental Setting

The proposed LDL-LCLR algorithm is compared with seven LDL algorithms: PT-SVM, AA-kNN, BFGS-LLD [Geng, 2016], IIS-LLD [Geng *et al.*, 2010], CPNN [Geng *et al.*, 2013], LDLLC [Jia *et al.*, 2018] and LDL-SCL [Zheng *et al.*, 2018]. These seven algorithms can be divided into three groups. The first group is the Problem Transformation (PT, e.g., PT-SVM), methods in this group transform the LDL problem into traditional problem and then use existing learners to solve it. The second group is the Algorithm Adjustment (AA, e.g., AA-kNN), the main idea of AA is to adapt traditional algorithms to fit for LDL paradigm. The others are specialized algorithms designed for LDL, which can directly model the relative importance of each label to the particular instance. IIS-LLD aims to minimize the Kullback-Leibler divergence between the ground truth and the predicting distribution by using improved iterative scaling method (IIS). Since IIS often performs worse than several other optimization algorithms [Malouf, 2002], an improved method BFGS-LLD is proposed to optimize the target function through the quasi-Newton method BFGS. The main assumption made in both IIS-LLD and BFGS-LLD is that the relationship between the feature space and the label distribution space is consistent with the maximum entropy model. CPNN removes this assumption by using a three layer neural network to approximate the relationship. LDLLC and LDL-SCL are two LDL algorithms considering the label correlation. LDLLC exploits the Pearson's correlation to capture global label relevance while LDL-SCL utilizes the local label correlation to obtain more potential supervised information. All the codes are shared by original authors, and we use the suggested parameters reported in corresponding literature, except that we tune the regularization parameters from $10^{\{-4,-3,-2,-1,0,1,2,3\}}$ for LDLLC and LDL-SCL using ten-fold cross-validation. The number of cluster is set to 6 in LDL-SCL.

In LDL-LCLR, the parameters λ_1, λ_2, λ_3, λ_4 and k are set to 0.0001, 0.001, 0.001, 0.001 and 4, respectively. Besides, ρ is simply set as 1. We use k-means to cluster samples. The details of parameter selections are shown in the parameter analysis section. The maximum iteration is set to be 100. Besides, S and Z are initialized by the identity matrix. The initialization of other variables is all-zero.

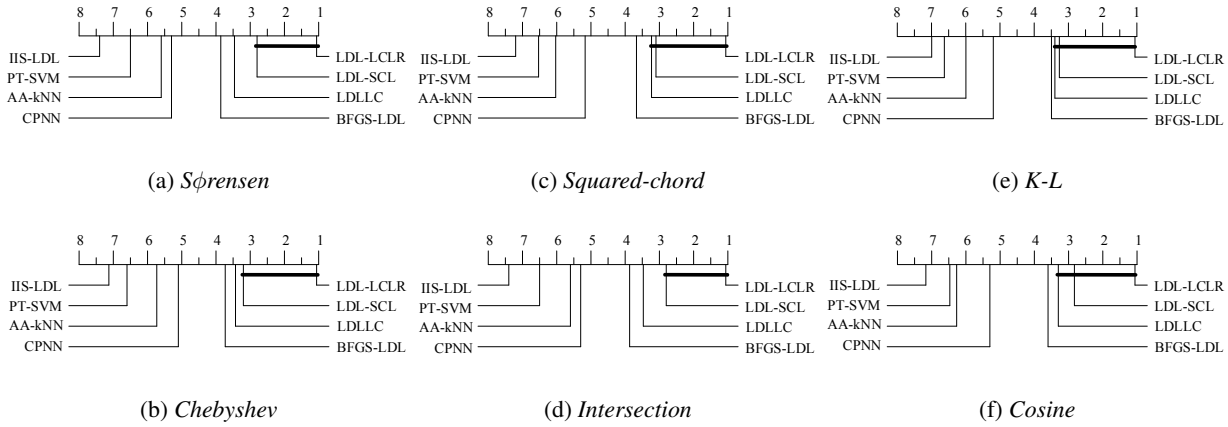
Evaluation metric	F_F	critical value
Sørensen	36.2341	
Squared-chord	35.8563	
K-L	33.5090	
Chebyshev	28.8609	2.1044
Intersection	36.8712	
Cosine	42.8402	

Table 1: Friedman statistics F_F in terms of each evaluation metric and the critical value at 0.05 significance level (# comparing algorithms $k = 8$, # data sets $N = 15$).

4.2 Results and Discussion

Due to page limitation, we only provide detailed results of two measures, which are shown in Table 2 and Table 3. The best performance among the comparing algorithms on each measure is marked in bold. On each data set, ten times ten-fold cross-validation is conducted and the mean value and standard deviation of each evaluation criterion is recorded.

To perform comparative analysis in more well-founded ways, Friedman test is further examined which is a favorable statistical test for comparisons of more than two algorithms over multiple data sets [Demšar, 2006]. Table 1 summarizes the Friedman statistics F_F and the corresponding critical value on each measure. As shown in Table 1, for each evaluation criterion, the null hypothesis of indistinguishable performance at 0.05 significance level among the comparing algorithms is clearly rejected. Consequently, Bonferroni-Dunn test [Demšar, 2006] at 0.05 significance level is employed to test whether our proposed method LDL-LCLR achieves competitive performance against the comparing algorithms, where LDL-LCLR is considered as the control algorithm. The performance between two algorithms is significantly different if their average ranks over all data sets differ by at least one critical difference (CD). Figure 2 shows the CD diagrams



(a) *Sørensen* (c) *Squared-chord* (e) *K-L*

(b) *Chebyshev* (d) *Intersection* (f) *Cosine*

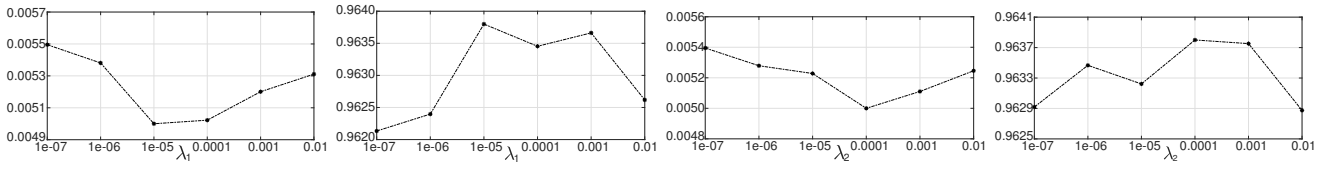
Figure 2: CD diagrams of the comparing algorithms under each evaluation criterion (CD=2.406 at 0.05 significance level).

Data	LDL-LCLR	IIS-LDL	BFGS-LDL	AA-kNN	LDLLC	CPNN	LDL-SCL	PT-SVM
Yeast-alpha	**.0362±.0018**(1)	.0577±.0010(8)	.0392±.0018(4)	.0412±.0006(7)	.0382±.0009(3)	.0402±.0007(6)	.0376±.0008(2)	.0398±.0007(5)
Yeast-cdc	**.0420±.0021**(1)	.0602±.0006(8)	.0459±.0019(5)	.0473±.0007(7)	.0427±.0010(3)	.0465±.0012(6)	.0422±.0010(2)	.0445±.0011(4)
Yeast-cold	**.0560±.0029**(1)	.0708±.0012(8)	.0596±.0014(4)	.0643±.0023(6)	.0585±.0016(2)	.0603±.0023(5)	.0591±.0012(3)	.0662±.0030(7)
Yeast-diau	**.0563±.0044**(1)	.0759±.0014(8)	.0599±.0011(4)	.0632±.0013(5)	.0594±.0018(2)	.0634±.0014(6)	.0596±.0014(3)	.0758±.0034(7)
Yeast-dtt	**.0382±.0030**(1)	.0575±.0035(8)	.0426±.0021(4.5)	.0455±.0014(7)	.0408±.0011(2)	.0426±.0019(4.5)	.0416±.0016(3)	.0446±.0019(6)
Yeast-elu	**.0402±.0019**(1)	.0602±.0007(8)	.0419±.0010(4)	.0455±.0005(7)	.0415±.0009(2)	.0425±.0009(5)	.0417±.0005(3)	.0438±.0008(6)
Yeast-heat	**.0579±.0028**(1)	.0765±.0013(8)	.0608±.0022(4)	.0634±.0012(7)	.0596±.0019(2)	.0615±.0009(5.5)	.0602±.0014(3)	.0615±.0010(5.5)
Yeast-spo	**.0811±.0061**(1)	.0962±.0022(8)	.0854±.0018(4)	.0921±.0026(6)	.0846±.0030(3)	.0859±.0025(5)	.0836±.0016(2)	.0922±.0035(7)
Yeast-spo5	**.0876±.0051**(1)	.0988±.0031(8)	.0927±.0018(5)	.0963±.0026(7)	.0911±.0035(2)	.0915±.0031(3)	.0918±.0032(4)	.0960±.0034(6)
Yeast-spoem	**.0745±.0092**(1)	.0925±.0048(7)	.0880±.0033(3.5)	.0902±.0026(6)	.0854±.0026(2)	.0880±.0028(3.5)	.0882±.0038(5)	.0942±.0075(8)
s-JAFFE	.1186±.0159(2)	.1535±.0086(7)	.1305±.0048(4)	.1291±.0056(3)	.1501±.0099(5)	.1556±.0082(8)	**.1148±.0086**(1)	.1532±.0070(6)
SBU_3DFE	**.1405±.0069**(1)	.1607±.0031(7)	.1585±.0018(4)	.1554±.0024(3)	.1593±.0052(5)	.1601±.0028(6)	.1421±.0035(2)	.1626±.0035(8)
Natural Scene	**.4434±.0312**(1)	.5397±.0039(7)	.4757±.0056(3)	.4706±.0008(2)	.5354±.0076(6)	.5174±.0097(5)	.5000±.0064(4)	.6496±.0175(8)
Movie	**.1578±.0077**(1)	.2014±.0035(7)	.1639±.0033(2)	.1773±.0048(4)	.1849±.0058(5)	.1877±.0067(6)	.1740±.0033(3)	.3054±.0178(8)
Human Gene	**.1935±.0115**(1)	.2176±.0039(4)	.2160±.0017(3)	.2557±.0037(7)	.2726±.0038(8)	.2178±.0025(5)	.2089±.0027(2)	.2179±.0065(6)

Table 2: *Sørensen* (the lower the better) results on all the data. The value is measured by 10 times 10-fold cross validation shown in mean±std(rank) form. The best results on each row are highlighted.

Data	LDL-LCLR	IIS-LDL	BFGS-LDL	AA-kNN	LDLLC	CPNN	LDL-SCL	PT-SVM
Yeast-alpha	**.9638±.0018**(1)	.9423±.0010(8)	.9608±.0018(4)	.9588±.0006(7)	.9618±.0009(3)	.9598±.0007(6)	9620±.0008(2)	.9602±.0007(5)
Yeast-cdc	**.9580±.0021**(1)	.9398±.0006(8)	.9541±.0019(5)	.9527±.0007(7)	.9573±.0010(3)	.9535±.0012(6)	.9578±.0010(2)	.9555±.0011(4)
Yeast-cold	**.9440±.0029**(1)	.9292±.0012(8)	.9404±.0014(4)	.9357±.0023(6)	.9415±.0016(2)	.9397±.0023(5)	.9409±.0012(3)	.9338±.0030(7)
Yeast-diau	**.9437±.0044**(1)	.9241±.0014(8)	.9401±.0011(4)	.9368±.0013(5)	.9406±.0018(2)	.9366±.0014(6)	.9404±.0014(3)	.9242±.0034(7)
Yeast-dtt	**.9618±.0030**(1)	.9425±.0035(8)	.9574±.0021(4.5)	.9545±.0014(7)	.9592±.0011(2)	.9574±.0019(4.5)	.9584±.0016(3)	.9554±.0019(6)
Yeast-elu	**.9598±.0019**(1)	.9398±.0007(8)	.9581±.0010(4)	.9545±.0005(7)	.9585±.0009(2)	.9575±.0009(5)	.9583±.0006(3)	.9562±.0008(6)
Yeast-heat	**.9421±.0028**(1)	.9235±.0013(8)	.9392±.0022(4)	.9366±.0012(7)	.9404±.0019(2)	.9385±.0009(5.5)	.9398±.0014(3)	.9385±.0010(5.5)
Yeast-spo	**.9189±.0061**(1)	.9038±.0022(8)	.9146±.0018(4)	.9079±.0026(6)	.9154±.0030(3)	.9141±.0025(5)	.9164±.0016(2)	.9078±.0035(7)
Yeast-spo5	**.9124±.0051**(1)	.9012±.0031(8)	.9073±.0018(5)	.9037±.0026(7)	.9089±.0035(2)	.9085±.0031(3)	.9082±.0032(4)	.9040±.0034(6)
Yeast-spoem	**.9255±.0092**(1)	.9075±.0048(7)	.9120±.0033(3.5)	.9098±.0026(6)	.9146±.0026(2)	.9120±.0028(3.5)	.9118±.0038(5)	.9058±.0075(8)
s-JAFFE	.8814±.0159(2)	.8465±.0086(7)	.8695±.0048(4)	.8709±.0056(3)	.8499±.0099(5)	.8444±.0082(8)	**.8851±.0086**(1)	.8468±.0070(6)
SBU_3DFE	**.8595±.0069**(1)	.8393±.0031(7)	.8415±.0018(4)	.8446±.0024(3)	.8407±.0052(5)	.8399±.0028(6)	.8579±.0036(2)	.8374±.0035(8)
Natural Scene	**.5566±.0312**(1)	.4603±.0039(7)	.5243±.0056(3)	.5294±.0008(2)	.4646±.0076(6)	.4826±.0097(5)	.5004±.0064(4)	.3504±.0175(8)
Movie	**.8422±.0077**(1)	.7986±.0035(7)	.8361±.0033(2)	.8227±.0048(4)	.8151±.0058(5)	.8123±.0067(6)	.8260±.0033(3)	.6946±.0178(8)
Human Gene	**.8065±.0115**(1)	.7824±.0039(4)	.7840±.0017(3)	.7443±.0037(7)	.7274±.0038(8)	.7822±.0025(5)	.7911±.0023(2)	.7810±.0100(6)

Table 3: *Intersection* (the higher the better) results on all the data. The value is measured by 10 times 10-fold cross validation shown in mean±std(rank) form. The best results on each row are highlighted.



(a) *K-L* w.r.t λ_1 (b) *Intersection* w.r.t λ_1 (c) *K-L* w.r.t λ_2 (d) *Intersection* w.r.t λ_2

Figure 3: Influence of λ_1 and λ_2 with 2 measures on Yeast-alpha.

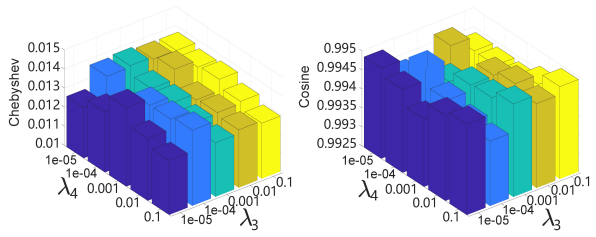


Figure 4: Influence of λ_3 and λ_4 with 2 measures on Yeast-alpha.

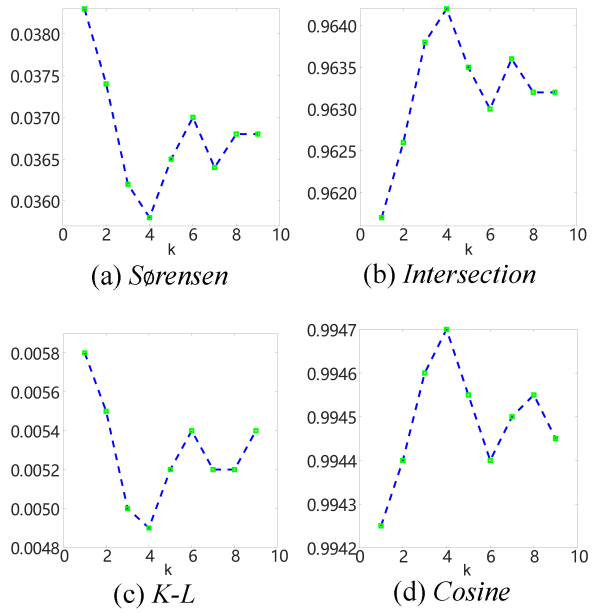


Figure 5: Influence of k with 4 measures on data set Yeast-alpha.

on each evaluation criterion. In each sub-figure, any comparing algorithm whose average rank is within one CD to that of LDL-LCLR is connected. Otherwise, any algorithm not connected with LDL-LCLR is considered to have significantly different performance between them.

Based on these experimental results, the following observations can be made: (1) As shown in Table 2 and Table 3, it can be observed that the proposed method LDL-LCLR is superior to the baselines in general. (2) The correlation-driven methods LDLLC, LDL-SCL and LDL-LCLR generally outperform other algorithms. And the performance of LDL-LCLR is better than LDLLC and LDL-SCL on all data sets except for s-JAFFE. The results are as expected since the proposal simultaneously exploits the global and the local label correlations. (3) The specialized LDL algorithms generally outperform the algorithms that are obtained by problem transformation and algorithm adaptation.

In summary, the proposed LDL-LCLR algorithm achieves a competitive performance against other well-established label distribution algorithms. The results demonstrate the effectiveness of LDL-LCLR.

4.3 Sensitivity Analysis of Parameters

Next, we investigate the sensitivity of the proposed LDL-LCLR method to the parameter setting, including λ_1, λ_2, λ_3, λ_4 in (6) and the number of clusters k. Figure 3 shows the influence of λ_1 and λ_2 with measures *K-L* and *intersection* on data set Yeast-alpha. We can observe that the performance is relatively stable if the parameters λ_1 and λ_2 respectively falls in a certain range (i.e., $\lambda_1 \in [0.00001, 0.001]$, $\lambda_2 \in [0.00001, 0.001]$) and the performance deteriorates when they fall outside of the range. Besides, we can see that the influence of λ_2 is smaller than the influence of λ_1.

Considering that parameters λ_3 and λ_4 determine the global correlation and local correlation in (6), we design a set of experiments to investigate how these two parameters jointly affect the prediction performance of LDL-LCLR. As shown in Figure 4, we can notice that the proposed method is very sensitive to the variations of the parameters λ_3 and λ_4. When $\lambda_3 \in [0.001, 0.1]$, in general, the performance rises gradually with the increase of λ_4, which demonstrates the importance of local correlations among labels in LDL learning process. Moreover, when λ_3 is set to 10^{-5}, the performance of our method deteriorates firstly and then becomes better as λ_4 increases.

In addition, we run LDL-LCLR with k varying from 1 to 9 with step size of 1. We present the experimental results on data set Yeast-alpha with 4 evaluation measures in Figure 5. Notice that, for criteria *Sørensen* and *K-L*, the smaller the value, the better the performance; but for criteria *Intersection* and *Cosine*, the larger the value, the better the performance. First, we can observe that the performance rises rapidly when k is less than 4. After that, the performance of the algorithm fluctuates in a certain range (i.e, *Sørensen* $\in [0.0365, 0.0370]$, *Intersection* $\in [0.9630, 0.9635]$). Therefore, we set $k = 4$ in LDL-LCLR.

5 Conclusion

Label distribution learning is a generalized and effective method to deal with label ambiguity problems. Many LDL algorithms have concerned the label correlations to improve the performances. However, most of them apply the label correlations either in a global or a local way. In this paper, different from previous works, we propose a novel LDL algorithm that simultaneously exploits the global and the local label correlations. In the proposed LDL-LCLR method, the global correlations are captured through a low-rank approximation and the local correlations are utilized by clustering samples. The experimental results on several data sets show that LDL-LCLR outperforms many state-of-the-art LDL algorithms.

Acknowledgements

This work is jointly supported by the National Key R&D Program of China (No. 2018YFB1003902), Natural Science Foundation of Jiangsu Province (No. BK20170809), the National Natural Science Foundation of China (Nos. 61773208, 61876001, 71671086), and Jiangsu Key Laboratory of Big Data Security and Intelligent Processing (NJUPT).

References

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

[Cai *et al.*, 2010] Jianfeng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):1956–1982, 2010.

[Demšar, 2006] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(Jan):1–30, 2006.

[Gao *et al.*, 2017] Binbin Gao, Chao Xing, Chenwei Xie, Jianxin Wu, and Xin Geng. Deep label distribution learning with label ambiguity. *IEEE Transactions on Image Processing*, 26(6):2825–2838, 2017.

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

[Geng and Xia, 2014] Xin Geng and Yu Xia. Head pose estimation based on multivariate label distribution. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1837–1842, 2014.

[Geng *et al.*, 2010] Xin Geng, Kate Smith-Miles, and Zhi-Hua Zhou. Facial age estimation by learning from label distributions. In *AAAI Conference on Artificial Intelligence*, pages 451–456, 2010.

[Geng *et al.*, 2013] Xin Geng, Chao Yin, and Zhi-Hua Zhou. Facial age estimation by learning from label distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10):2401–2412, 2013.

[Geng *et al.*, 2014] Xin Geng, Qin Wang, and Yu Xia. Facial age estimation by adaptive label distribution learning. In *International Conference on Pattern Recognition*, pages 4465–4470, 2014.

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

[He *et al.*, 2017] Zhouzhou He, Xi Li, Zhongfei Zhang, Fei Wu, Xin Geng, Yaqing Zhang, Ming-Hsuan Yang, and Yueting Zhuang. Data-dependent label distribution learning for age estimation. *IEEE Transactions on Image Processing*, 26(8):3846–3858, 2017.

[Huo *et al.*, 2016] Zengwei Huo, Xu Yang, Chao Xing, Ying Zhou, Peng Hou, Jiaqi Lv, and Xin Geng. Deep age distribution learning for apparent age estimation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 722–729, 2016.

[Jia *et al.*, 2018] Xiuyi Jia, Weiwei Li, Junyu Liu, and Yu Zhang. Label distribution learning by exploiting label correlations. In *AAAI Conference on Artificial Intelligence*, 2018.

[Liu *et al.*, 2010] Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*, pages 663–670, 2010.

[Malouf, 2002] Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Conference on Natural Language Learning*, pages 49–55, 2002.

[Tsoumakas *et al.*, 2007] G Tsoumakas, I Katakis, and D Taniar. Multi-label classification: An overview. *International Journal of Data Warehousing & Mining*, 3(3):1–13, 2007.

[Xing *et al.*, 2016] Chao Xing, Xin Geng, and Hui Xue. Logistic boosting regression for label distribution learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4489–4497, 2016.

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

[Xu *et al.*, 2014] Linli Xu, Zhen Wang, Zefan Shen, Yubo Wang, and Enhong Chen. Learning low-rank label correlations for multi-label classification with missing labels. In *IEEE International Conference on Data Mining*, pages 1067–1072, 2014.

[Yuan, 1991] Ya-Xiang Yuan. A modified bfgs algorithm for unconstrained optimization. *IMA Journal of Numerical Analysis*, 11(3):325–332, 1991.

[Zhang *et al.*, 2015] Zhaoxiang Zhang, Mo Wang, and Xin Geng. Crowd counting in public video surveillance by label distribution learning. *Neurocomputing*, 166:151–163, 2015.

[Zhao and Geng, 2017] Quan Zhao and Xin Geng. Selection of target function in label distribution learning. *Frontiers of Computer Science and Technology*, 5(11):708–719, 2017.

[Zhao and Zhou, 2018] Peng Zhao and Zhi-Hua Zhou. Label distribution learning by optimal transport. In *AAAI Conference on Artificial Intelligence*, pages 4506–4513, 2018.

[Zheng *et al.*, 2018] Xiang Zheng, Xiuyi Jia, and Weiwei Li. Label distribution learning by exploiting sample correlations locally. In *AAAI Conference on Artificial Intelligence*, pages 4556–4563, 2018.

[Zhou *et al.*, 2015] Ying Zhou, Hui Xue, and Xin Geng. Emotion distribution recognition from facial expressions. In *ACM International Conference on Multimedia*, pages 1247–1250, 2015.

[Zhou *et al.*, 2016] Deyu Zhou, Xuan Zhang, Yin Zhou, Quan Zhao, and Xin Geng. Emotion distribution learning from texts. In *Conference on Empirical Methods in Natural Language Processing*, pages 638–647, 2016.