The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Label Distribution Learning by Exploiting Sample Correlations Locally

Xiang Zheng, Xiuyi Jia*
School of Computer Science and Engineering
Nanjing University of
Science and Technology
Nanjing 210014, China

Weiwei Li
College of Astronautics
Nanjing University of
Aeronautics and Astronautics
Nanjing 210016, China

Abstract

Label distribution learning (LDL) is a novel multi-label learning paradigm proposed in recent years for solving label ambiguity. Existing approaches typically exploit label correlations globally to improve the effectiveness of label distribution learning, by assuming that the label correlations are shared by all instances. However, different instances may share different label correlations, and few correlations are globally applicable in real-world applications. In this paper, we propose a new label distribution learning algorithm by exploiting sample correlations locally (LDL-SCL). To encode the influence of local samples, we design a local correlation vector for each instance based on the clustered local samples. Then we predict the label distribution for an unseen instance based on the original features and the local correlation vector simultaneously. Experimental results demonstrate that LDL-SCL can effectively deal with the label distribution problems and perform remarkably better than the state-of-the-art LDL methods.

Introduction

Learning with ambiguity is a hot topic in machine learning and data mining areas. There are mainly two sophisticated paradigms for solving label ambiguity at present, namely single-label learning and multi-label learning (Tsoumakas, Katakis, and Taniar 2007) respectively. In single-label learning framework, an instance is associated with a single class label, whereas in multi-label learning, an instance may have multiple class labels simultaneously. A great deal of research (Read, Pfahringer, and Holmes 2008; Read et al. 2011) have shown that multi-label learning is an effective and widely applied paradigm, which can be seen as an extension of single-label learning. However, multi-label learning can only model the ambiguity of "*what describes the instance*" , i.e., it usually outputs a set of labels which are associated with the instance. Sometimes we need to deal with the more general ambiguity of "*how to describe the instance*". For example, in some cases, we need not only know which labels are associated with an instance, but also the extent to which each label describes the instance. To solve such problems, Geng (2010) proposed label distribution learning (LDL), which is a further extension of multi-label learning. Different from traditional multi-label learning to output a label set, LDL outputs a label distribution, and each component in the distribution represents the degree of description of the corresponding label to the instance which is called the *description degree*.

After LDL has been put forward, more and more researchers tried to solve label ambiguity problems by using LDL and achieved good results. To name a few, Geng et al. (2010) proposed the IIS-LLD algorithm and applied it into the facial age estimation problem, in which the original single-label dataset was transformed into a label distribution dataset. In their further works, to improve the efficiency of IIS-LLD, they proposed conditional probability neural network (CPNN) algorithm (Geng, Yin, and Zhou 2013) and BFGS-LLD algorithm (Geng and Ji 2014). Yang et al. (2015) attempted to combine LDL with deep learning and proposed the deep label distribution learning (DLDL) algorithm.

However, the above research does not take into account the information of label correlations, i.e., one label may be related to another one under certain conditions. For example, when a movie has a high description degree on label *action*, it is more likely to have a higher description degree on label *adventure* than label *romance*. By considering the label correlations, LDL can exploit more data information to improve the performance. Certain studies used label correlations in different ways, e.g., in (Zhou et al. 2016), the relations of emotions were captured based on the Plutchiks wheel of emotions (Plutchik 1980); Zhou el al. (2015) considered the label correlations by seeking Pearson's correlation coefficients between two labels. Although above algorithms attempted to exploit label correlations, they usually considered label correlations in a global way by assuming that the correlations were shared by all instances. However, in most real-world applications, label correlations are usually appeared in a local way, where the correlations may be shared by only a subset of instances instead of all instances. For example, as shown in Figure 1, we assume *flower* and *butterfly* have a correlation. In image (b), *butterfly* is less prominent and thus could be difficult to predict; in this case, the correlation between *flower* and *butterfly* can be helpful in learning label *butterfly* since the label *flower* is relatively easier to predict in this image. However, for image (c), with

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

(a) *flower*, *butterfly*, *water*, *tree*

(b) *flower*, *grass*, *cloud*, *butterfly*

(c) *flower*, *cabin*, *tree*, *grass*

Figure 1: Illustration of non-global label correlations.

the *flower* clearly presented, this correlation turns to be misleading since *butterfly* is usually not appeared in a *cabin*. Therefore, exploiting such correlations at the global level will enforce unnecessary or even misleading constraints on some samples, and may hurt the performance by predicting some irrelevant labels.

To solve the above problem, we need to consider the label correlations at a local level. In this paper, we propose a label distribution learning algorithm by exploiting sample correlations locally (LDL-SCL). We assume that the instances can be separated into different clusters, and the instances in each cluster should be as similar as possible, thus the difference of their label distributions is as small as possible, i.e. the instances in each cluster should share the same label correlations. To reflect the influence of local samples, inspired by (Huang and Zhou 2012), we construct a local correlation vector as the additional features for each instance. Each item in the local correlation vector represents the impact of each cluster (local sample) on this instance. With the local correlation vector combined original features, we formulate LDL into a new optimization problem. Besides, we develop an alternating minimization method based on gradient descent for the optimization. Experiments on eight datasets show that our proposed algorithm exhibits a promising performance when considering local sample correlations.

The main contributions of this study can be summarized as follows: 1) local sample correlations are encoded as a local correlation vector; 2) a novel objective function and a novel output model of LDL are proposed. The rest of the paper is organized as follows. First, we discuss existing works related to our proposed approach. Second, we introduce the details of the proposed LDL-SCL algorithm. Finally, we report the experimental results and conclude the paper.

Related Works

There are three strategies for LDL at present. The first strategy is problem transformation, i.e., transforming the LDL problem into existing learning paradigms, such as traditional multi-label learning. The second strategy is algorithm adaptation, i.e., extending existing learning algorithms to deal with label distributions. The third strategy is to design specialized algorithms according to the characteristics of LDL. Different from the indirect strategies of problem transformation and algorithm adaptation, the specialized algorithms directly match the LDL problem. The related research (Geng 2016) have shown that the third strategy is more effective than other two strategies.

Generally speaking, a specialized algorithm consists of three parts, i.e., an output model, an objective function and an optimization method. For the output model, the specialized LDL algorithms mainly use the maximum entropy model (Berger, Pietra, and Pietra 1996), such as the IIS-LLD algorithm and the BFGS-LLD algorithm (Geng 2016). For the objective function, it usually chooses some functions that can measure the similarity between two distributions. Many divergence functions are adopted in existing research, such as the Kullback-Leibler (KL) divergence in the IIS-ALDL algorithm (Geng, Wang, and Xia 2014), the balanced divergence function defined in the EDL algorithm (Zhou et al. 2016), Jeffery divergence (Geng and Xia 2014), and weighted Jeffery divergence (Zhou, Xue, and Geng 2015). To solve the optimization problem, many methods were also applied in LDL. IIS-LLD algorithm (Geng 2016) used improved iterative scaling (IIS) method as the optimization method, while BFGS-LLD (Geng 2016) used BFGS and EDL (Zhou et al. 2016) used L-BFGS.

Similarly, our proposed LDL-SCL algorithm is also constructed from above three aspects. In this paper, the output model and the objective function are improved by taking into account the influence of local sample and an alternating solution is utilized for the optimization.

The LDL-SCL Approach

The Framework

Let $\mathcal{X} = \mathcal{R}^q$ denote the q-dimensional input space and $\mathcal{Y} = \{y_1, y_2, \cdots, y_L\}$ denote the finite set of labels, where L is the number of labels. Given a training set $S = \{(x_1, D_1), (x_2, D_2), \cdots, (x_n, D_n)\}$, where $x_i \in \mathcal{X}$ is an instance and $D_i = \{d_i^1, d_i^2, \cdots, d_i^L\}$ is the label distribution associated with x_i. Such d_i^j is called the description degree of y_j to x_i and satisfies $\sum_{j=1}^{L} d_i^j = 1$. The goal of LDL is to learn a mapping function $f : \mathcal{X} \to D$ which can predict the label distributions for unseen instances.

As discussed in the introduction, we implement local sample correlations by adding additional features to each instance. To be specific, we construct a local correlation vector c_i for each instance x_i to reflect the influence of local samples, and expand the original feature representation with the vector. With the previous discussion, it is obvious that the length of the local correlation vector c is equal to the number of clusters. Generally, similar instances share the similar label correlations, which means similar instances will have similar c. Notice that we measure the similarity between instances in the label space instead of the feature space, because (1) instances with similar label distributions usually share similar label correlations; (2) the label space is usually much smaller than the feature space, resulting in reducing the running time. Considering both original features and local sample correlations simultaneously, we obtain the following LDL output model based on the maximum entropy model:

$$p(y_l|x_i;\theta,w,c) = \frac{1}{Z_i}\exp(\sum_k \theta_{l,k}x_i^k + \sum_t w_{l,t}c_i^t), \quad (1)$$

where $Z_i = \sum_l \exp(\sum_k \theta_{l,k} x_i^k + \sum_t w_{l,t} c_i^t)$ is a normalization term to satisfy the sum of all label description degrees of an instance equals 1. To simplify the formula, we use $p(y_l|x_i)$ to represent $p(y_l|x_i;\theta,w,c)$. The first term in the exponent part illustrates the information of original features while the second term illustrates the information of additional features, i.e., the local sample correlations. θ is the original feature coefficient matrix, in which $\theta_{l,k}$ is the element with respect to the k-th original feature and the l-th label. x_i^k is the k-th original feature of instance x_i. w is the additional feature coefficient matrix, in which $w_{l,t}$ is the element with repect to the t-th additional feature and the l-th label. c_i^t is the t-th element of local correlation vector c_i. Aiming to incorporate global discrimination fitting and local correlation influence into a unified framework, we optimize both θ, w, c by minimizing the following objective function:

$$\min_{\theta,w,c} V(\theta,w,c) + \lambda_1\Omega_1(\theta) + \lambda_2\Omega_2(w) + \lambda_3 Z(\theta,w,c). \quad (2)$$

V is the loss function defined on the training data. Ω_1 and Ω_2 are two penalty terms to control the complexity of the output model. Z is a penalty term to enhance the local characteristic of the correlation vector and λ_1, λ_2 and λ_3 are regularization parameters.

With the previous discussion, the goal of LDL is to make the predicted distribution and the true distribution as similar as possible, therefore the loss function should be a function which can measure the similarity of two distributions. Cha (2007) analyzed many functions that measure the similarity between two distributions, such as K-L divergence, Jeffery divergence, K divergence, etc. Here, we use K-L divergence as the loss function defined by

$$D_J(Q_a\|Q_b) = \sum_j Q_a^j \ln \frac{Q_a^j}{Q_b^j}, \quad (3)$$

where Q_a^j and Q_b^j are the j-th element of the two distributions Q_a and Q_b, respectively. Specifically, in this paper, the expression for V based on K-L divergence is defined as follows:

$$V(\theta,w,c) = \sum_{i=1}^{n}\sum_{l=1}^{L}(d_i^l \ln(\frac{d_i^l}{p(y_l|x_i)})), \quad (4)$$

where d_i^l and $p(y_l|x_i)$ denote the true description degree and the predicted description degree of label y_l to instance x_i, respectively. For the second and the third terms of Eq. (2), we utilize the F-norm of matrix to implement as follows:

$$\Omega_1(\theta) = \|\theta\|_F^2, \quad (5)$$

$$\Omega_2(w) = \|w\|_F^2. \quad (6)$$

The fourth term of Eq. (2) is utilized to enhance the local characteristic of the correlation vector. We assume that the training data can be divided into m clusters $\{G_1, G_2, \cdots, G_m\}$. Each cluster denotes a local sample, in which all instances share a same label correlation. For easy implementation, we use k-means (Kanungo et al. 2002) as the clustering method. In addition, the Euclidean distance of a cluster center and an instance is used to measure the correlation between the local sample and the instance. We use the following formula to obtain the label distribution of a cluster center:

$$p_j = \frac{1}{|G_j|}\sum_{x_k \in G_j} D_k, \quad (7)$$

where $|G_j|$ is the number of instances in the local sample G_j. Then, given an instance x_i, we define c_i^j to measure the local influence of G_j on x_i. The more similar D_i and p_j, the more likely that x_i shares the same correlation with instances in G_j, suggesting that the smaller the value of c_i^j, the smaller the impact of the local sample G_j on instance x_i. We construct a local correlation vector $c_i = [c_i^1, c_i^2, \cdots, c_i^m]$ for each instance x_i, and define the penalty term on the vector as follows:

$$Z(\theta,w,c) = \sum_{i=1}^{n}\sum_{j=1}^{m} c_i^j \|p(y|x_i) - p_j\|_2^2, \quad (8)$$

where $p(y|x_i)$ is the predicted label distribution of instance x_i. By substituting Eqs. (4), (5), (6) and (8) into Eq. (2), we can obtain the following optimization problem:

$$\begin{aligned}\min_{\theta,w,c} &\sum_{i=1}^{n}\sum_{l=1}^{L}(d_i^l \ln(\frac{d_i^l}{p(y_l|x_i)})) + \lambda_1\|\theta\|_F^2 + \lambda_2\|w\|_F^2 \\ &+ \lambda_3\sum_{i=1}^{n}\sum_{j=1}^{m} c_i^j\|p(y|x_i) - p_j\|_2^2 \quad s.t. \quad c_i^j \geq 0,\end{aligned} \quad (9)$$

c_i^j measures the local influence of G_j on x_i, thus, it is constrained to be not less than 0.

Learning by Alternating Minimization

Eq. (9) is an optimization problem with inequality constraints, therefore we utilize an interior point method to convert it into an unconstrained problem:

$$\begin{aligned}\min_{\theta,w,c} &\sum_{i=1}^{n}\sum_{l=1}^{L}(d_i^l \ln(\frac{d_i^l}{p(y_l|x_i)})) + \lambda_1\|\theta\|_F^2 + \lambda_2\|w\|_F^2 \\ &+ \lambda_3\sum_{i=1}^{n}\sum_{j=1}^{m} c_i^j\|p(y|x_i) - p_j\|_2^2 + \mu\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{1}{c_i^j},\end{aligned} \quad (10)$$

where μ is the penalty factor and $\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{1}{c_i^j}$ is the barrier function.

Eq. (10) can be solved by alternating minimization, i.e., optimizing one of the variables with the others fixed in each iteration. Specifically, we update one of the variables in $\{\theta, w, c\}$ with gradient descent, and leave the others fixed.

When we fix w and c to solve θ, Eq. (10) can be reduced to

$$\begin{aligned}\min_{\theta} &\sum_{i=1}^{n}\sum_{l=1}^{L}(d_i^l \ln(\frac{d_i^l}{p(y_l|x_i)})) + \lambda_1\|\theta\|_F^2 \\ &+ \lambda_3\sum_{i=1}^{n}\sum_{j=1}^{m} c_i^j\|p(y|x_i) - p_j\|_2^2.\end{aligned} \quad (11)$$

We solve it with gradient decent and the gradient of the objective w.r.t. θ is

$$\begin{aligned}\nabla\theta_{l,k} =& -\sum_{i=1}^{n} d_i^l x_i^k + \sum_{i=1}^{n} \frac{\exp(x_i.\theta_{.l} + c_i.w_{.l})x_i^k}{\sum_{l'}\exp(x_i.\theta_{.l'} + c_i.w_{.l'})} \\ &+ 2\lambda_3 \sum_{i=1}^{n}\sum_{j=1}^{m} c_i^j (p(y_l|x_i) - p_{j,l}) \frac{\partial p(y_l|x_i)}{\partial \theta_{l,k}} \\ &+ 2\lambda_1 \theta_{l,k},\end{aligned} \tag{12}$$

where $\frac{\partial p(y_l|x_i)}{\partial \theta_{l,k}} = \frac{\exp(x_i.\theta_{.l}+c_i.w_{.l})x_i^k \sum_{l'\neq l}\exp(x_i.\theta_{.l'}+c_i.w_{.l'})}{(\sum_{l'}\exp(x_i.\theta_{.l'}+c_i.w_{.l'}))^2}$, $x_i.$ and $c_i.$ are row vectors, $\theta_{.l}$ and $w_{.l}$ are column vectors, and $p_{j,l}$ is the l-th element of cluster center p_j.

When we fix θ and c to solve w, Eq. (10) can be reduced to

$$\begin{aligned}\min_{w} \sum_{i=1}^{n}\sum_{l=1}^{L}(d_i^l \ln(\frac{d_i^l}{p(y_l|x_i)})) + \lambda_2\|w\|_F^2 \\ + \lambda_3 \sum_{i=1}^{n}\sum_{j=1}^{m} c_i^j \|p(y|x_i) - p_j\|_2^2.\end{aligned} \tag{13}$$

Again, we use gradient descent and the gradient of the objective w.r.t. w is

$$\begin{aligned}\nabla w_{l,t} =& -\sum_{i=1}^{n} d_i^l c_i^t + \sum_{i=1}^{n} \frac{\exp(x_i.\theta_{.l} + c_i.w_{.l})c_i^t}{\sum_{l'}\exp(x_i.\theta_{.l'} + c_i.w_{.l'})} \\ &+ 2\lambda_3 \sum_{i=1}^{n}\sum_{j=1}^{m} c_i^j (p(y_l|x_i) - p_{j,l}) \frac{\partial p(y_l|x_i)}{\partial w_{l,t}} \\ &+ 2\lambda_2 w_{l,t},\end{aligned} \tag{14}$$

where $\frac{\partial p(y_l|x_i)}{\partial w_{l,t}} = \frac{\exp(x_i.\theta_{.l}+c_i.w_{.l})c_i^t \sum_{l'\neq l}\exp(x_i.\theta_{.l'}+c_i.w_{.l'})}{(\sum_{l'}\exp(x_i.\theta_{.l'}+c_i.w_{.l'}))^2}$.

When we fix θ and w to solve c, Eq. (10) can be reduced to

$$\begin{aligned}\min_{c} \sum_{i=1}^{n}\sum_{l=1}^{L}(d_i^l \ln(\frac{d_i^l}{p(y_l|x_i)})) + \mu\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{1}{c_i^j} \\ + \lambda_3 \sum_{i=1}^{n}\sum_{j=1}^{m} c_i^j \|p(y|x_i) - p_j\|_2^2.\end{aligned} \tag{15}$$

Similarly, we use gradient descent and the gradient of the objective w.r.t. c is

$$\begin{aligned}\nabla c_i^j =& -\sum_{l=1}^{L} d_i^l w_{l,j} + \frac{\sum_l(\exp(x_i.\theta_{.l} + c_i.w_{.l})w_{l,j})}{\sum_l \exp(x_i.\theta_{.l} + c_i.w_{.l})} \\ &+ 2\lambda_3 c_i^j \sum_{l}(p(y_l|x_i) - p_{j,l}) \frac{\partial p(y_l|x_i)}{\partial c_i^j} \\ &+ \lambda_3 \|p(y|x_i) - p_j\|_2^2 - \mu\frac{1}{{c_i^j}^2},\end{aligned} \tag{16}$$

where $\frac{\partial p(y_l|x_i)}{\partial c_i^j} = -\frac{\exp(x_i\cdot\theta_{.l}+c_i.w_{.l})\sum_{l'\neq l}\exp((x_i.\theta_{.l}+c_i.w_{.l'})w_{l',j})}{(\sum_{l'}\exp(x_i.\theta_{.l}+c_i.w_{.l'}))^2}$.

Algorithm 1: The LDL-SCL algorithm

Input: training set $\{X, D\}$, parameters λ_1, λ_2, λ_3 and m.
Output: the label distribution D_t.

```
Train:
initialize θ, w, c and μ;
repeat
    update θ according to Eq. (12);
    update w according to Eq. (14);
    update c according to Eq. (16);
    μ ← βμ, 0 < β < 1;
until convergence or maximum number of iterations;
for j = 1 to m do
    train a linear regression model R_j;
end
Test:
for j = 1 to m do
    predict c_t^j using R_j;
end
return the label distribution D_t according to Eq. (1).
```

Table 1: Number of labels in eight datasets.

Dateset	alpha	cdc	elu	diau	heat	spo	cold	dtt
#Samples	2465	2465	2465	2465	2465	2465	2465	2465
#Features	24	24	24	24	24	24	24	24
#Labels	18	15	14	7	6	6	4	4

Notice that given a test instance x_t, its local correlation vector c_t is unknown. We thus train m regression models using the original features and the local correlation matrix of training instances. Then, in testing phase, the local correlation vector of the test instance can be obtained from the outputs of regression models.

The pseudo code of LDL-SCL is presented in Algorithm 1. The coefficient matrices θ and w are initialized with all elements equal 1. The local correlation matrix c is initialized with the result of k-means clustering on the label space. In detail, c_i^k is assigned 1 if x_i is in the k-th cluster and 0 otherwise. Then, we use an alternating solution to update the variables θ, w and c. Since Eq. (9) is a convex function, it will converge to a global minimum. After that, m regression models are trained with the original features as inputs and the local correlation matrix as outputs. Given a test instance x_t, the local correlation vector $c_t = [c_t^1, c_t^2, \cdots, c_t^m]$ is obtained with the m regression models. Notice that the regression models can be trained with a low computational cost because usual the cluster number m is not large. Finally, the label distribution D_t is obtained by Eq. (1).

Experiments

Datasets

The datasets used in the experiments were collected from biological experiments on the budding yeast Saccharomyces cerevisiae (Eisen et al. 1998). There are 2465 yeast genes in

Table 2: Evaluation measures for LDL algorithms.

	Name	Formula
Distance	Euclidean↓	$Dis_1 = \sqrt{\sum_{j=1}^{c}(P_j - Q_j)^2}$
	Sϕrensen↓	$Dis_2 = \frac{\sum_{j=1}^{c}\lvert P_j - Q_j\rvert}{\sum_{j=1}^{c}\lvert P_j + Q_j\rvert}$
	Squared χ^2 ↓	$Dis_3 = \sum_{j=1}^{c}\frac{(P_j - Q_j)^2}{P_j + Q_j}$
	Kullback-Leibler(KL)↓	$Dis_4 = \sum_{j=1}^{c} P_j \ln \frac{P_j}{Q_j}$
Similarity	Intersection↑	$Sim_1 = \sum_{j=1}^{c}\min(P_j, Q_j)$
	Fidelity↑	$Sim_2 = \sum_{j=1}^{c}\sqrt{P_j Q_j}$

total, each of that is represented by an associated phylogenetic profile vector of length 24. The labels correspond to the discrete time points in different biological experiments. The gene expression level (after normalization) at each time point provides a natural measure of the description degree of the corresponding label. There are ten datasets in total in this series, and we just choose the datasets with the number of labels greater than or equal to 4 since the datasets with fewer labels are lack of the information of label correlations. The details of the eight datasets are summarized in Table 1.

Evaluation Measures

In this paper, six measures are chosen as the evaluation measures for the LDL algorithms. The names and formulas of the six measures are presented in Table 2, where P_j and Q_j are the j-th element of the ground-truth label distribution and the predicted distribution, respectively. For the former four distance measures, "↓" indicates "the smaller the better". For the latter two similarity measures, "↑" indicates "the larger the better".

Experimental Setting

The LDL-SCL algorithm proposed in this paper is compared with seven state-of-the-art algorithms, i.e., PT-Bayes (Geng and Ji 2014), PT-SVM (Geng, Wang, and Xia 2014), AA-kNN (Geng, Smith-Miles, and Zhou 2010), AA-BP (Geng, Yin, and Zhou 2013), IIS-LLD (Geng, Smith-Miles, and Zhou 2010), BFGS-LLD (Geng, Yin, and Zhou 2013) and EDL (Zhou, Xue, and Geng 2015). The parameter settings of all algorithms are as follows. For PT-Bayes, maximum likelihood estimation is used to estimate the Gaussian class-conditional probability density functions. PT-SVM is implemented as the "C-SVC" type in LIBSVM using the RBF kernel with the parameters $C = 1.0$ and $Gamma = 0.01$. The number of neighbors k in AA-kNN is set to 5. The number of hidden-layer neurons for AA-BP is set to 60. For LDL-SCL, the parameters in Eq. (9) are set to: $\lambda_1 = 0.001$, $\lambda_2 = 0.001$ and $\lambda_3 = 0.001$.

Experimental Results

For each dataset, 10 times 10-fold cross-validation are employed and average results are recorded. The experimental results are shown in Table 3.

The experimental results are presented in the form of "mean±std (rank)". Ranking refers to the prediction effect of all LDL algorithms in metrics. The smaller the value the better the performance. Furthermore, the best performance among the 8 comparing algorithms is shown in boldface. The 8 LDL algorithms include two algorithms obtained from problem transformation (PT-Bayes and PT-SVM), two algorithms obtained from algorithm adaptation (AA-kNN and AA-BP) and four specialized algorithms (IIS-LLD, BFGS-LLD, EDL and LDL-SCL). The LDL-SCL algorithm is our proposed approach in this paper.

By analyzing the experimental results shown in Table 3, in summary, the specialized LDL algorithms generally perform better than those algorithms obtained from problem transformation and algorithm adaptation. Furthermore, the proposed LDL-SCL algorithm has the best performance on datasets *alpha*, *cdc*, *elu*, *heat* and *cold*, while has the sub-optimum performance on datasets *diau*, *spo* and *dtt*. The result demonstrates the effectiveness of LDL-SCL. It is worth mentioning that, on all datasets, the LDL-SCL algorithm performs better than the EDL algorithm which exploits label correlations globally.

Influence of Parameters

In order to examine the robustness of the proposed algorithm, we also analyze the influence of parameters in the experiment, including λ_1, λ_2, λ_3 in Eq. (9) and the number of clusters m. We run LDL-SCL with λ_1 which is set to 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000 and so does with λ_2 and λ_3. In addition, we run LDL-SCL with m varying from 0 to 14 with step size of 2. Due to the page limit, we only present the experimental results on dataset *cold* with 6 evaluation measures. The results are shown in Figure 2 and Figure 3. Notice that, for criteria *Euclidean*, *Sϕrensen*, *Squared* χ^2 and *KL*, the smaller the value, the better the performance; but for criteria *Intersection* and *Fidelity*, the larger the value, the better the performance. As for parameters λ_1, λ_2 and λ_3 shown in Figure 2, we can find that, the performance gets worse when three parameters take large values since the objective function is not dominated by the first term. Besides, we can see that the influence of λ_2 is smaller than the influence of λ_1 and λ_3. As for the number of clusters m shown in Figure 3, it has the worst result when m is equal to 0, which means the influence of local samples is ignored and the result is similar to the results of IIS-LLD and BFGS-LLD. Furthermore, when m is less than 6, the performance of LDL-SCL rises steadily, otherwise, the performance rises slowly and tends to be stable.

Convergency

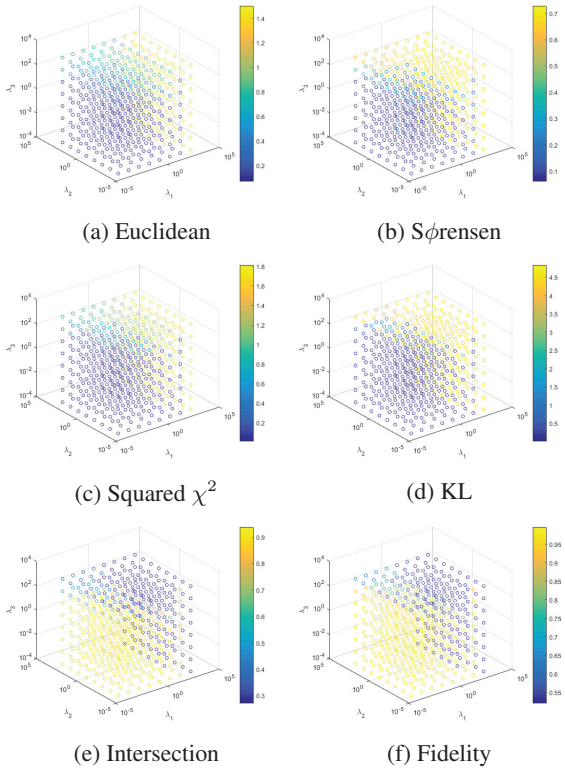
In order to investigate the convergence of the alternating minimization algorithm to solve LDL-SCL model, we plot the value of objective function Eq. (10) on two datasets (*alpha* and *cold*) in Figure 4. It can be observed that the objective function value decreases with respect to iterations, and the value approaches to be a fixed value after a few iterations (about 10 iterations for *alpha* and *cold*).

Table 3: Comparison results of different label distribution algorithms on the eight datasets (mean±std(rank)).

data	algorithm	Euclidean↓	Sørensen↓	Squared χ^2 ↓	KL↓	Intersection↑	Fidelity↑
alpha	LDL-SCL	**.0229±.0003(1)**	**.0372±.0005(1)**	**.0054±.0001(1)**	**.0054±.0001(1)**	**.9627±.0005(1)**	**.9986±.0001(1)**
	PT-Bayes	.2298±.0124(8)	.3485±.0154(8)	.3879±.0277(8)	.5607±.0710(8)	.6515±.0154(8)	.8777±.0100(8)
	PT-SVM	.0276±.0006(5)	.0445±0009(5)	.0071±.0003(5)	.0071±.0003(5)	.9565±.0009(5)	.9981±.0001(5)
	AA-kNN	.0279±.0006(6)	.0449±.0012(6)	.0073±.0003(6)	.0074±.0004(7)	.9561±.0012(6)	.9980±.0001(6)
	AA-BP	.0871±.0070(7)	.1475±.0131(7)	.1399±.0501(7)	.0673±.0058(6)	.8538±.0117(7)	.9839±.0017(7)
	IIS-LLD	.0269±.0004(4)	.0429±.0012(3)	.0069±.0004(4)	.0069±.0004(4)	.9571±.0012(3)	.9983±.0011(4)
	BFGS-LLD	.0251±.0004(2)	.0408±.0011(2)	.0063±.0008(2)	.0063±.0004(2)	.9574±.0009(2)	.9985±.0011(2)
	EDL	.0260±.0011(3)	.0429±.0022(4)	.0067±.0006(3)	.0068±.0006(3)	.9570±.0022(4)	.9983±.0002(3)
cdc	LDL-SCL	**.0274±.0008(1)**	**.0417±.0012(1)**	**.0068±.0005(1)**	**.0067±.0004(1)**	**.9582±.0013(1)**	**.9983±.0001(1)**
	PT-Baye	.2399±.0103(8)	.3455±.0111(8)	.3853±.0210(8)	.5374±.0503(8)	.6545±.0111(8)	.8778±.0075(8)
	PT-SVM	.0298±.0007(5)	.0458±.0012(5)	.0077±.0004(5)	.0076±.0004(5)	.9554±.0012(5)	.9980±.0001(5.5)
	AA-kNN	.0301±.0009(6)	.0462±.0013(6)	.0080±.0004(6)	.0079±.0004(6)	.9538±.0013(6)	.9980±.0001(5.5)
	AA-BP	.0769±.0081(7)	.1192±.0109(7)	.0842±.0281(7)	.0511±.0121(7)	.8829±.0134(7)	.9879±.0051(7)
	IIS-LLD	.0290±.0010(4)	.0445±.0015(3)	.0073±.0005(4)	.0072±.0005(4)	.9556±.0015(4)	.9982±.0012(4)
	BFGS-LLD	.0284±.0011(3)	.0449±.0016(4)	.0070±.0004(2)	.0070±.0005(2)	.9558±.0016(3)	.9983±.0011(2)
	EDL	.0283±.0006(2)	.0429±.0008(2)	.0072±.0004(3)	.0072±.0004(3)	.9571±.0008(2)	.9982±.0001(3)
elu	LDL-SCL	**.0278±.0005(1)**	**.0412±.0005(1)**	**.0063±.0003(1)**	**.0062±.0003(1)**	**.9587±.0005(1)**	**.9984±.0001(1)**
	PT-Bayes	.2588±.0203(8)	.3558±.0198(8)	.4081±.0408(8)	.6062±.1030(8)	.6442±.0198(8)	.8689±.0156(8)
	PT-SVM	.0293±.0008(3)	.0438±.0012(3)	.0068±.0005(3)	.0068±.0005(3)	.9562±.0012(3)	.9983±.0002(3)
	AA-kNN	.0297±.0010(4)	.0443±.0014(4)	.0071±.0006(5)	.0071±.0006(5)	.9557±.0014(4)	.9982±.0002(4)
	AA-BP	.0733±.0037(7)	.1100±.0048(7)	.0731±.0026(7)	.0481±.0061(7)	.8891±.0064(7)	.9890±.0025(7)
	IIS-LLD	.0307±.0009(5)	.0472±.0014(5)	.0071±.0004(4)	.0071±.0004(4)	.9528±.0015(6)	.9982±.0035(5)
	BFGS-LLD	.0308±.0009(6)	.0475±.0012(6)	.0075±.0004(56)	.0073±.0003(6)	.9552±.0017(5)	.9979±.0009(6)
	EDL	.0289±.0005(2)	.0431±.0008(2)	.0067±.0003(2)	.0067±.0003(2)	.9569±.0007(2)	.9983±.0001(2)
diau	LDL-SCL	.0531±.0008(2)	.0582±.0011(2)	.0125±.0005(2)	.0124±.0005(2)	.9417±.0011(2)	.9968±.0001(2)
	PT-Bayes	.4027±.0183(8)	.4177±.0170(8)	.5280±.0281(8)	.8512±.0772(8)	.5823±.0170(8)	.8230±.0107(8)
	PT-SVM	.0628±.0037(6)	.0686±.0041(6)	.0169±.0018(6)	.0167±.0017(6)	.9314±.0041(6)	.9957±.0004(6)
	AA-KNN	.0567±.0019(4)	.0622±.0022(4)	.0145±.0011(4)	.0145±.0010(4)	.9378±.0022(4)	.9963±.0003(4)
	AA-BP	.0802±.0051(7)	.0863±.0059(7)	.0276±.0013(7)	.0291±.0069(7)	.9142±.0067(7)	.9929±.0031(7)
	IIS-LLD	.0539±.0031(3)	.0593±.0032(3)	.0144±.0014(3)	.0141±.0013(3)	.9407±.0033(3)	.9964±.0036(3)
	BFGS-LLD	**.0444±.0022(1)**	**.0476±.0023(1)**	**.0089±.0008(1)**	**.0083±.0008(1)**	**.9513±.0027(1)**	**.9978±.0031(1)**
	EDL	.0597±.0010(5)	.0653±.0010(5)	.0158±.0005(5)	.0155±.0005(5)	.9347±.0010(5)	.9960±.0002(5)
heat	LDL-SCL	**.0592±.0016(1)**	**.0597±.0014(1)**	**.0127±.0009(1)**	**.0126±.0009(1)**	**.9402±.0014(1)**	**.9967±.0003(1)**
	PT-Bayes	.4500±.0231(8)	.4354±.0193(8)	.5450±.0361(8)	.8678±.1198(8)	.5646±.0193(8)	.8180±.0131(8)
	PT-SVM	.0625±.0023(3)	.0627±.0022(2)	.0141±.0010(2.5)	.0141±.0010(2.5)	.9373±.0022(2)	.9964±.0003(2.5)
	AA-kNN	.0624±.0020(2)	.0632±.0018(3)	.0141±.0010(2.5)	.0141±.0010(2.5)	.9368±.0018(3)	.9964±.0003(2.5)
	AA-BP	.0793±.0068(7)	.0822±.0071(7)	.0235±.0047(7)	.0246±.0053(7)	.9198±.0061(7)	.9937±.0028(7)
	IIS-LLD	.0703±.0036(5)	.0692±.0033(5)	.0182±.0016(5)	.0182±.0016(5)	.9309±.0033(5)	.9954±.0042(6)
	BFGS-LLD	.0728±.0031(6)	.0791±.0029(6)	.0188±.0016(6)	.0186±.0015(6)	.9304±.0034(6)	.9961±.0048(5)
	EDL	.0629±.0016(4)	.0633±.0017(4)	.0143±.0008(4)	.0143±.0008(4)	.9366±.0017(4)	.9963±.0003(4)
spo	LDL-SCL	**.0809±.0019(1)**	.0836±.0018(2)	.0243±.0010(2)	.0242±.0010(2)	.9164±.0019(2)	.9938±.0002(2)
	PT-Bayes	.4038±.0162(8)	.4030±.0134(8)	.4972±.0246(8)	.7172±.0840(8)	.5971±..0134(8)	.8342±.0095(8)
	PT-SVM	.0878±.0019(5)	.0893±.0022(5)	.0280±.0015(5)	.0284±.0015(5)	.9107±.0022(5)	.9929±.0004(5)
	AA-kNN	.0879±.0030(6)	.0899±.0024(6)	.0286±.0020(6)	.0286±.0002(6)	.9096±.0034(6)	.9927±.0005(6)
	AA-BP	.0979±.0041(7)	.1012±.0038(7)	.0344±.0038(7)	.0359±.0039(7)	.8982±.0037(7)	.9906±.0010(7)
	IIS-LLD	.0863±.0041(4)	.0861±.0036(3)	.0251±.0036(3)	.0252±.0022(3)	.9139±.0036(3)	.9937±.0005(3)
	BFGS-LLD	.0819±.0045(2)	**.0833±.0038(1)**	**.0229±.0019(1)**	**.0226±.0021(1)**	**.9168±.0039(1)**	**.9951±.0007(1)**
	EDL	.0843±.0029(3)	.0872±.0029(4)	.0268±.0015(4)	.0269±.0016(4)	.9128±.0028(4)	.9932±.0004(4)
cold	LDL-SCL	**.0675±.0003(1)**	**.0585±.0013(1)**	**.0121±.0006(1)**	**.0121±.0006(1)**	**.9414±.0013(1)**	**.9969±.0001(1)**
	PT-Bayes	.5252±.0224(8)	.4479±.0189(8)	.5873±.0352(8)	.9089±.1042(8)	.5521±.0189(8)	.7991±.0134(8)
	PT-SVM	.0753±.0080(4)	.0654±.0069(5)	.0147±.0033(4)	.0146±.0033(4)	.9346±.0069(5)	.9963±.0008(4)
	AA-kNN	.0724±.0027(2)	.0630±.0024(2)	.0136±.0011(2)	.0136±.0011(2)	.9370±.0024(2)	.9966±.0003(3)
	AA-BP	.0838±.0045(7)	.0710±.0027(7)	.0178±.0011(7)	.0163±.0030(7)	.9328±.0029(7)	.9952±.0017(7)
	IIS-LLD	.0767±.0004(5)	.0653±.0034(4)	.0157±.0015(6)	.0155±.0015(6)	.9347±.0034(4)	.9960±.0039(6)
	BFGS-LLD	.0745±.0004(3)	.0641±.0035(3)	.0139±.0013(3)	.0143±.0015(3)	.9348±.0035(3)	.9968±.0036(2)
	EDL	.0771±.0018(6)	.0668±.0016(6)	.0154±.0009(5)	.0153±.0009(5)	.9332±.0016(6)	.9961±.0003(5)
dtt	LDL-SCL	**.0474±.0017(1)**	.0411±.0015(2)	.0059±.0004(2)	.0058±.0005(2)	**.9587±.0015(1)**	.9985±.0002(2)
	PT-Bayes	.4879±.0242(8)	.4156±.0192(8)	.5416±.0438(8)	.9069±.1580(8)	.5844±.0192(8)	.8113±.0186(8)
	PT-SVM	.0516±.0029(5)	.0447±.0024(5)	.0071±.0009(6)	.0071±.0009(6)	.9553±.0024(5)	.9982±.0003(5.5)
	AA-kNN	.0512±.0019(4)	.0443±.0017(4)	.0071±.0007(5)	.0070±.0007(5)	.9557±.0017(4)	.9982±.0002(4)
	AA-BP	.0622±.0032(7)	.0531±.0029(7)	.0097±.0012(7)	.0122±.0037(7)	.9465±.0024(7)	.9969±.0011(7)
	IIS-LLD	.0535±.0023(6)	.0480±.0023(6)	.0068±.0005(3)	.0068±.0005(3)	.9520±.0023(6)	.9983±.0013(3)
	BFGS-LLD	.0495±.0019(2)	**.0409±.0017(1)**	**.0058±.0005(1)**	**.0054±.0004(1)**	.9584±.0023(2)	**.9989±.0010(1)**
	EDL	.0508±.0022(3)	.0440±.0018(3)	.0069±.0007(4)	.0068±.0008(4)	.9560±.0018(3)	.9982±.0003(5.5)

Conclusion

Label distribution learning is a generalization form of multi-label learning, which can deal with more label ambiguity problems. To improve the effectiveness of the approach, certain existing LDL methods exploited the label correlations in a global way. However, it is more reasonable to exploit label correlations at a local level since different instances may share different label correlations in most real-world applications. In this paper, we derive a local correlation vector for each instance to encode the influence of local samples

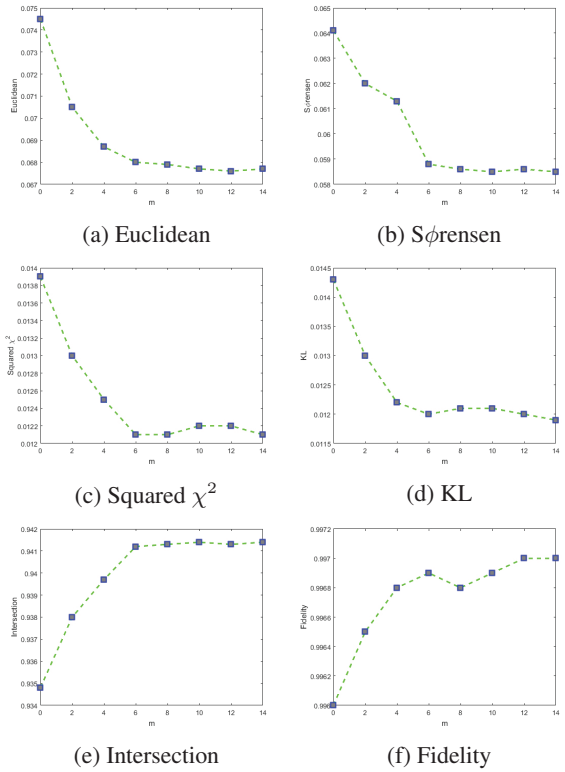


(a) Euclidean (b) Sϕrensen

(c) Squared χ^2 (d) KL

(e) Intersection (f) Fidelity

Figure 2: Influence of λ_1, λ_2 and λ_3 with 6 measures on dataset *cold*.

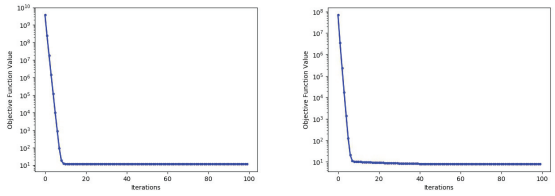


(a) Euclidean (b) Sϕrensen

(c) Squared χ^2 (d) KL

(e) Intersection (f) Fidelity

Figure 3: Influence of m with 6 measures on dataset *cold*.



(a) *alpha* (b) *cold*

Figure 4: Convergence of LDL-SCL on *alpha* and *cold*.

and propose the LDL-SCL algorithm. The experimental results demonstrate that LDL-SCL can effectively deal with the label distribution problems and perform remarkable better than the state-of-the-art LDL methods. In future work, we will try other measures to reflect the influence of local samples and design related optimization methods with faster convergence speed.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61773208, 61403200), and the Natrual Science Foundation of Jiangsu Province (BK20170809).

References

Berger, A. L.; Pietra, V. J. D.; and Pietra, S. A. D. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics* 22(1):39–71.

Cha, S. H. 2007. Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical Models & Methods in Applied Sciences* 1(4):300–307.

Eisen, M. B.; Spellman, P. T.; Brown, P. O.; and Botstein, D. 1998. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America* 95(25):14863–14868.

Geng, X., and Ji, R. 2014. Label distribution learning. In *IEEE International Conference on Data Mining Workshops*, 377–383.

Geng, X., and Xia, Y. 2014. Head pose estimation based on multivariate label distribution. In *Computer Vision and Pattern Recognition*, 1837–1842.

Geng, X.; Smith-Miles, K.; and Zhou, Z. H. 2010. Facial age estimation by learning from label distributions. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 451–456.

Geng, X.; Wang, Q.; and Xia, Y. 2014. Facial age estimation by adaptive label distribution learning. In *IEEE International Conference on Pattern Recognition*, 4465–4470.

Geng, X.; Yin, C.; and Zhou, Z. H. 2013. Facial age estimation by learning from label distributions. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 35(10):2401–2412.

Geng, X. 2016. Label distribution learning. *IEEE Transactions on Knowledge & Data Engineering* 28(7):1734–1748.

Huang, S. J., and Zhou, Z. H. 2012. Multi-label learning by exploiting label correlations locally. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 949–955.

Kanungo, T.; Mount, D. M.; Netanyahu, N. S.; Piatko, C. D.; Silverman, R.; and Wu, A. Y. 2002. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 24(7):881–892.

Plutchik, R. 1980. *Chapter 1 - A General Psychoevolutionary Theory of Emotion*. Elsevier Inc.

Read, J.; Pfahringer, B.; Holmes, G.; and Frank, E. 2011. Classifier chains for multi-label classification. *Machine Learning* 85(3):333–348.

Read, J.; Pfahringer, B.; and Holmes, G. 2008. Multi-label classification using ensembles of pruned sets. In *Eighth IEEE International Conference on Data Mining*, 995–1000.

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

Yang, X.; Gao, B. B.; Xing, C.; and Huo, Z. W. 2015. Deep label distribution learning for apparent age estimation. In *IEEE International Conference on Computer Vision Workshop*, 344–350.

Zhou, D.; Zhang, X.; Zhou, Y.; Zhao, Q.; and Geng, X. 2016. Emotion distribution learning from texts. In *Conference on Empirical Methods in Natural Language Processing*, 638–647.

Zhou, Y.; Xue, H.; and Geng, X. 2015. Emotion distribution recognition from facial expressions. In *ACM International Conference on Multimedia*, 1247–1250.