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

Label Distribution Learning by Exploiting Label Correlations

Xiuyi Jia,[1] Weiwei Li,[2*] Junyu Liu,[1] Yu Zhang[3]
[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
[2]College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing, China
[3]School of Information Science and Engineering, East China University of Science and Technology, Shanghai, China

Abstract

Label distribution learning (LDL) is a newly arisen machine learning method that has been increasingly studied in recent years. In theory, LDL can be seen as a generalization of multi-label learning. Previous studies have shown that LDL is an effective approach to solve the label ambiguity problem. However, the dramatic increase in the number of possible label sets brings a challenge in performance to LDL. In this paper, we propose a novel label distribution learning algorithm to address the above issue. The key idea is to exploit correlations between different labels. We encode the label correlation into a distance to measure the similarity of any two labels. Moreover, we construct a distance-mapping function from the label set to the parameter matrix. Experimental results on eight real label distributed data sets demonstrate that the proposed algorithm performs remarkably better than both the state-of-the-art LDL methods and multi-label learning methods.

Introduction

How to overcome the label ambiguity problem has become a hot research topic in the fields of machine learning and data mining. To solve this issue, the most popular used methods are either based on single-label learning or multi-label learning (Zhang and Zhang 2010). The former assigns a single label while the latter assigns multiple labels for each instance. A large number of studies have indicated that multi-label learning is an effective and widely used learning paradigm. However, there are still some problems which are hard to solve by using multi-label learning. For example, an expression usually contains several different emotional components. In some scenarios, the learning tasks require us to know not only what emotions are contained in an expression but also the importance of these emotional components (Zhou, Xue, and Geng 2015). To solve this kind of learning problem with label ambiguity, a new machine learning paradigm called label distribution learning (LDL) (Geng, Smith-Miles, and Zhou 2010; Geng and Ji 2014) has arisen and increasingly studied in the past few years.

A growing number of studies (Geng and Ling 2017; Hou et al. 2017) had tried to solve the problem of label ambiguity after the label distribution learning was presented. Geng et al. (2010) proposed a method based on IIS-LDL by transforming the original single label data set to the label distribution data set to solve the problem of age estimation. Through this strategy, one face image can contribute to both the learnings of its chronological age and adjacent ages. By using neural network based on conditional probability, the conditional probability neural network (CPNN) algorithm (Geng, Yin, and Zhou 2013) was proposed to further improve the accuracy of age estimatio. The literarue (Geng and Ji 2014) expanded the IIS-LDL algorithm by using the quasi-Newton optimization algorithm, which formed the BFGS-LDL algorithm. In order to solve the problem of shortage and imbalance of training data, Geng et al. (2015) proposed a method based on LDL for effective population estimation in public video surveillance. Yang et al. (2015) developed a deeply label distribution learning (DLDL) algorithm by combining the LDL with deep learning to deal with the apparent age estimation problem.

The methodology of LDL generally consists of three parts: objective function, output model and optimization algorithm. The classical LDL algorithm (Geng, Smith-Miles, and Zhou 2010) adopted Kullback-Leibler (KL) divergence as the objective function, the maximum entropy model as the output model and the BFGS algorithm for model optimization. After that, Geng et al. (2015) proposed the LDSVR algorithm based on the MSVR algorithm (Tuia et al. 2011) and label distribution learning to solve the problem that the output of MSVR algorithm may be negative. Based on a linear model, Xing et al. (2016) extended the maximum entropy model of traditional LDL to a more general LDL model family. The above-mentioned algorithms provided improved performance to deal with the problem of label ambiguity in the field of facial expression recognition, crowd counting and age estimation. However, none of them took the potentially important correlation between labels into account. Investigating the correlation between labels may provide additional information especially when the training data of some labels are limited. The existing literature on solving the label ambiguity problem exploited the label correlation mainly in two different ways: (1) estimates the prior knowledge of relationship between each two labels. For instance, Zhou et al. (2016) solved the problem of text emotions classification using the emotional wheel (Paul 1992) to obtain

*Corresponding author: Weiwei Li (liweiwei@nuaa.edu.cn).

a prior knowledge. (2) builds different models to compute the correlation between each two labels. Huang et al. (2012) proposed the MAHR approach based on the hypothesis that the labels may provide complementary information to each other if they are related. Accordingly this method, which is able to automatically discover and exploit the label relationship for multi-label learning. Huang and Zhou (2012) further developed the ML-LOC approach to more accurately reveal the local label correlation locally instead of globally for improved multi-label learning. Zhang and Yeung (2013) used the covariance matrix to simulate the label correlation as a regularization term to solve the multi-label classification problem. Unfortunately, most of the above algorithms were focused on the multi-label learning but rarely studied in the LDL framework. Based on this, we propose a novel label distribution learning method by exploiting the label correlation (LDLLC) in this paper. In LDLLC, the label correlation is encoded into a distance to measure the similarity of any two labels. For facilitating the learning process, we substitute the distance between any two labels in label set to the distance between the corresponding columns in the parameter matrix. Based on six different evaluation criteria, we validate the algorithm proposed in this paper. The experimental results demonstrate that the proposed LDLLC algorithm achieves better prediction performance than other existing LDL algorithms. Considering that LDL can be seen as an extension of multi-label learning, we also compare LDLLC algorithm with the common multi-label learning algorithms based on five different evaluation criteria. The experimental results can also demonstrate the effectiveness of our algorithm.

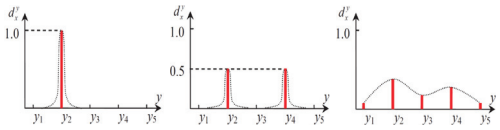
Label distribution learning

Both single-label learning and multi-label learning can be viewed as a special case of label distribution learning. In single-label learning, a single label is assigned to an instance and the possible output is either 0 (incorrectly labeled) or 1 (correctly label). In multi-label learning, each training instance is associated with a relevant label set. From single-label learning to multi-label learning, the size of the output space of the learning process becomes increasingly larger. Specifically, for a problem with n different labels, there are n possible outputs for single-label learning, and $2^n - 1$ possible outputs for multi-label learning. Fig. 1 shows the decision regions of three learning paradigms for a learning problem with two labels. There are two possible labels (red and yellow) in single-label learning while three possible labels (red, yellow and orange) in multi-label learning. More importantly, there are infinitely possible labels in label distribution learning. Each point in Fig. 1 represents a decision region for a label distribution, which is an extension of the representation of discrete values to continuous values. The output of label distribution learning is no longer a relevant label set, but a label distribution.

LDL is an extension of single-label learning and multi-label learning. Fig. 2 gives an label distribution example for single-label learning, multi-label learning, and the general case, respectively. For single-label learning (a), the output can be converted into a label distribution output by setting

(a) Single-label learning (b) Multi-label learning (c) Label distribution learning

Figure 1: The decision regions of three learning paradigms for a learning problem with two labels.



(a) Single-label learning (b) Multi-label learning (c) Label distribution learning

Figure 2: Three ways to label an instance.

the description degree of the correctly label of the instance to 1 while the other is set to 0. For multi-label learning (b), assuming a correct instance has two relevant labels, each of the two labels by default describes 50% of the instance respectively while the other is set to 0. Finally, LDL (c) represents a general case of label distribution, which illustrated that label distribution is more general than both single-label learning and multi-label learning, and thus can provide more flexibility in the learning process and has more usage scenarios.

The definition of LDL contains mainly three aspects: firstly, each training instance is explicitly associated with a label distribution, rather than a simple label or a relevant label set; secondly, each label in the label distribution corresponds to a real value and the overall label distribution will be mainly investigated; thirdly, the performance evaluation measures of previous learning algorithms with numerical label indicators are still those commonly used for single-label learning (e.g., classification accuracy, error rate, etc.) or multi-label learning (e.g., hamming loss, one-error, etc.). On the other hand, the performance of label distribution learning should be evaluated by the similarity or distance between the predicted label distribution and the real label distribution, which will be further discussed in the following section.

Label Distribution Learning Exploiting Label Correlations

Let $\mathcal{X} = \mathcal{R}^m$ denote the input space and $\mathcal{Y} = \{y_1, y_2, \cdots, y_c\}$ is the complete set of labels. The goal of LDL is to learn a mapping function from the instances to the label distributions. 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 $D_i = \{d_{x_i}^{y_1}, d_{x_i}^{y_2}, \cdots, d_{x_i}^{y_c}\}$ is the label distribution associated with x_i. We use the description degree d_x^y to

represent the degree to which label y describes the instance x. Without loss of generality, assume that $d_x^y \in [0,1]$, and the label set is complete, i.e., using all the labels in the set can always fully describe the instance. Then, $\sum_y d_x^y = 1$. d_x^y can be represented by the form of conditional probability, i.e., $d_x^y = p(y|x)$. Now, the goal of LDL is to learn a conditional probability mass function $p(y|x)$ from S, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Suppose $p(y|x)$ is a parametric model $p(y|x;\theta)$, where θ is the parameter vector. Given the training set S, the goal of LDL is to find the θ that can generate a distribution similar to D_i given the instance x_i. Many different criteria can be used to measure the distance between two distributions (Cha 2007), such as Squared χ^2, Jefferys divergence, Euclidean, Kullback-Leibler (KL) divergence and so on. Here we use Kullback-Leibler divergence defined by

$$D_J(Q_a \| Q_b) = \sum_j Q_a^j \ln \frac{Q_a^j}{Q_b^j}. \tag{1}$$

Where Q_a^j and Q_b^j are the j-th elements of the two distributions Q_a and Q_b, respectively. The above formula calculates the sum of all the distances between the description degrees of the same label, i.e., the superscripts of Q_a and Q_b are the same (i.e. j). One potential problem of the definition in Eq. (1) is that the relationship between different labels is not taken into account. In fact, some labels often appear together while some often conflict to each other. Therefore, the label correlations can be used as a new item in the objective function.

Accordingly, the best vector parameter θ^* is determined as follows

$$\begin{aligned}\theta^* &= \arg\min_\theta \sum_i D_J(D_i||\overline{D_i}) + \lambda_1\|\theta\|_F^2 \\ &\quad + \lambda_2 \sum_i \sum_j sgn(\rho_{\theta_i,\theta_j}) Dis(\theta_i,\theta_j) \\ &= \arg\min_\theta \sum_i \sum_j (d_{x_i}^{y_j} \ln \frac{d_{x_i}^{y_j}}{p(y_j|x_i;\theta)}) + \lambda_1\|\theta\|_F^2 \\ &\quad + \lambda_2 \sum_i \sum_j sgn(\rho_{\theta_i,\theta_j}) Dis(\theta_i,\theta_j),\end{aligned} \tag{2}$$

where D_i is the ground truth label distribution of the i-th instance and the $\overline{D_i}$ is the predicted one by $p(y|x_i;\theta)$, $\rho_{\theta_i,\theta_j} = \frac{\sum_k (\theta_{ik}-\overline{\theta_{ik}})(\theta_{jk}-\overline{\theta_{jk}})}{\sqrt{\sum_k (\theta_{ik}-\overline{\theta_{ik}})^2}\sqrt{\sum_k (\theta_{jk}-\overline{\theta_{jk}})^2}}$ denotes the Pearson's correlation coefficients between the i-th instance and the j-th instance, $sgn(x)$ denotes the sign function, and $Dis(\theta_i,\theta_j)$ denotes the distance between the i-th label and the j-th label. Taking Euclidean for example, $Dis(\theta_i,\theta_j) = \sqrt{\sum_k (\theta_{ik}-\theta_{jk})^2}$. The second term is a regularizer to prevent the model from overfitting. The third item considers the correlation between different labels and replaces the distance between two labels in the label set by the distance between the corresponding columns in the parameter vector θ. λ_1 and λ_2 are the balance factors.

As for $p(y|x;\theta)$, similar to previous works (Geng, Wang, and Xia 2014; Geng 2016), we assume it follows a maximum entropy model (Berger, Pietra, and Pietra 1996), i.e.,

$$p(y_k|x_i;\theta) = \frac{1}{Z_i} \exp(\sum_r \theta_{kr} x_i^r), \tag{3}$$

where $Z_i = \sum_k \exp(\sum_r \theta_{kr} x_i^r)$ is the normalization factor, x_i^r is the r-th feature of x_i, and θ_{kr} is an element in θ. Substituting Eq.(3) into Eq.(2) yields the objective function of θ:

$$\begin{aligned}T(\theta) = &\sum_i \sum_j d_{x_i}^{y_j} \ln d_{x_i}^{y_j} - \sum_i \sum_j d_{x_i}^{y_j} \sum_r \theta_{kr} x_i^r \\ &+ \sum_i \ln \sum_k \exp(\sum_r \theta_{kr} x_i^r) + \lambda_1\|\theta\|_F^2 \\ &+ \lambda_2 \sum_i \sum_j sgn(\rho_{\theta_i,\theta_j}) \sqrt{\sum_k (\theta_{ik}-\theta_{jk})^2}.\end{aligned} \tag{4}$$

The minimization of the function $T(\theta)$ can be effectively solved by the limited-memory quasi-Newton method (L-BFGS) (Yuan 1991). The basic idea of L-BFGS is to avoid explicit calculation of the inverse Hessian matrix used in the Newton method. L-BFGS approximates the inverse Hessian matrix with an iteratively updated matrix instead of storing the full matrix. Consider the second order Taylor series of $T(\theta)$ at the current estimate of the parameter vector θ:

$$T(\theta^{(l+1)}) \approx T(\theta^{(l)}) + \nabla T(\theta^{(l)})^T \triangle + \frac{1}{2} \triangle^T H(\theta^{(l)}) \triangle, \tag{5}$$

where $\triangle = \theta^{(l+1)} - \theta^{(l)}$ is the update step, $\nabla T(\theta^{(l)})$ and $H(\theta^{(l)})$ are the gradient and Hessian matrix of $T(\theta^{(l+1)})$ at $\theta^{(l)}$, respectively. The minimizer of Eq.(5) is

$$\triangle^{(l)} = -H^{-1}(\theta^{(l)}) \nabla T(\theta^{(l)}). \tag{6}$$

The line search Newton method uses $\triangle^{(l)}$ as the search direction $p^{(l)} = \triangle^{(l)}$ and updates model parameters by

$$\theta^{(l+1)} = \theta^{(l)} + \alpha^{(l)} p^{(l)}, \tag{7}$$

where the step length $\alpha^{(l)}$ is obtained from a line search procedure to satisfy the strong Wolfe conditions (Nocedal and Wright 2006):

$$T(\theta^{(l)} + \alpha^{(l)} p^{(l)}) \leqslant T(\theta^{(l)}) + c_1 \alpha^{(l)} \nabla T(\theta^{(l)})^T p^{(l)}, \tag{8}$$

$$|\nabla T(\theta^{(l)} + \alpha^{(l)} p^{(l)})| \leqslant c_2 |\nabla T(\theta^{(l)})^T p^{(l)}|, \tag{9}$$

where $0 < c_1 < c_2 < 1$. The idea of L-BFGS is to avoid explicit calculation of $H^{-1}(\theta^{(l)})$ by approximating it with an iteratively updated matrix B, i.e.

$$\begin{aligned}B^{(l+1)} = &(I - \rho^{(l)} s^{(l)} (u^{(l)})^T) B^{(l)} (I - \rho^{(l)} u^{(l)} (s^{(l)})^T) \\ &+ \rho^{(l)} s^{(l)} (s^{(l)})^T,\end{aligned} \tag{10}$$

where

$$s^{(l)} = \theta^{(l+1)} - \theta^l, \tag{11}$$

$$u^{(l)} = \nabla T(\theta^{(l+1)}) - \nabla T(\theta^{(l)}), \tag{12}$$

$$\rho^{(l)} = \frac{1}{s^{(l)} u^{(l)}}. \tag{13}$$

Algorithm 1: L-BFGS based LDLLC

Input: The training set $S = \{X, D\}$ and the convergence criterion ξ.
Output: $p(y|x;\theta)$.

Initialize the model parameter vector $\theta^{(0)}$;
Initialize the inverse Hessian approximation $B^{(0)}$;
Compute $\nabla T(\theta^{(0)})$ by Eq.(14);
$l \leftarrow 0$
repeat
  Compute search direction $\rho^{(l)} \leftarrow -B^{(l)}\nabla T(\theta^{(l)})$;
  Compute the step length α^l by a line search procedure to satisfy Eq. (8) and (9);
  $\theta^{(l+1)} \leftarrow \theta^{(l)} + \alpha^{(l)}p^{(l)}$;
  Compute $\nabla T(\theta^{(l+1)})$ by Eq. (14);
  $s^{(l)} \leftarrow \theta^{(l+1)} - \theta^l$
  $u^{(l)} \leftarrow \nabla T(\theta^{(l+1)}) - \nabla T(\theta^{(l)})$
  $\rho^{(l)} \leftarrow \frac{1}{s^{(l)}u^{(l)}}$
  $B^{(l+1)} \leftarrow (I - \rho^{(l)}s^{(l)}(u^{(l)})^T)B^{(l)}(I - \rho^{(l)}u^{(l)}(s^{(l)})^T) + \rho^{(l)}s^{(l)}(s^{(l)})^T$
  $l \leftarrow l + 1$
until $\|\nabla T(\theta^{(l+1)}) < \xi\|$;
$p(y_k|x_i;\theta) \leftarrow \frac{1}{Z_i}\exp(\sum_r \theta_{kr}x_i^r)$.

As for the optimization of the objective function $T(\theta)$, the computation of L-BFGS is mainly related to the first-order gradient of $T'(\theta)$, which can be achieved by

$$\frac{T(\theta)}{\theta_{kr}} = \begin{cases} \sum_i \frac{\exp(\sum_r \theta_{kr}x_i^r)x_i^r}{\sum_k \exp(\sum_r \theta_{kr}x_i^r)} - \sum_i d_{x_i}^{y_i}x_i^r \\ + \lambda_2 \sum_j \frac{\theta_{kr} - \theta_{jr}}{\sqrt{\sum_r(\theta_{kr} - \theta_{jr})^2}} \\ + 2\lambda_1\theta_{kr} \quad if\ \rho(\theta_k, \theta_j) \geq 0, \\ \sum_i \frac{\exp(\sum_r \theta_{kr}x_i^r)x_i^r}{\sum_k \exp(\sum_r \theta_{kr}x_i^r)} - \sum_i d_{x_i}^{y_i}x_i^r \\ - \lambda_2 \sum_j \frac{\theta_{kr} - \theta_{jr}}{\sqrt{\sum_r(\theta_{kr} - \theta_{jr})^2}} \\ + 2\lambda_1\theta_{kr} \quad if\ \rho(\theta_k, \theta_j) < 0. \end{cases} \tag{14}$$

The L-BFGS-based LDLLC algorithm is described in Algorithm 1.

LDL can be viewed as an extension of single-label learning and multi-label learning. This paper compares the LDL algorithm with the common multi-label learning algorithm. In order to implement this comparison, labels in the predicted distribution need to be divided into two sets, i.e, the relevant and irrelevant sets. For this purpose, an extra virtual label y_0 is added into the label set, i.e., the extended label set $\mathcal{Y}' = \mathcal{Y} \cup y_0 = \{y_0, y_1, y_2, \cdots, y_c\}$. Using the new extended label set in the training process, the optimal parameter vector θ^* is learned. As y_0 is the label that distinguishes the relevant and irrelevant labels directly, it is initialized as the threshold used in multi-label learning. Given an instance x', its label distribution is predicted by $p(y|x';\theta^*)$. The intensity value of y_0 splits the predicted distribution into two sets. The labels with the intensity value higher than y_0's are regarded as the relevant labels, and the rest labels are regarded as irrelevant ones. Therefore, LDLLC in fact implements the function of multi-label learning without the need of setting the threshold manually.

Experiments

Datasets

The datasets used in the experiments were collected from five biological experiments on the budding yeast Saccharomyces cerevisiae. There are 2465 yeast genes in total, each of which is represented by an associated phylogenetic profile vector of length 24. The labels correspond to the discrete time points in different biological experiments, respectively. The gene expression level (after normalization) at each time point provides a natural measure of the description degree of the corresponding label. There are 10 data sets in the series, and we just choose 8 of them with number of labels greater than or equal to 4 since the datasets with less labels lack information of label correlations. The details of the eight datasets are summarized in Table 1.

Table 1: Statistics of the 8 datasets used in the experiments.

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

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_i and Q_i are the i-th element of the true label distribution and the predicted distribution, respectively. For the four distance measures, "↓" indicates "the smaller the better". For the two similarity measures, "↑" indicates "the larger the better".

As LDL can provide both the relevant labels and their description degrees, multi-label learning can be seen as a special case of LDL because it only gives the labels as outputs. Several typical evaluation criteria used in multi-label learning mainly contain Hamming loss, Ranking loss, OneError, Coverage, Average Precision (Zhang and Zhou 2014), which are summarized in Table 3. They can also be adopted to evaluate the ability of LDL for distinguishing relevant labels from irrelevant ones.

Hamming loss indicates how many times a label pair is misclassified, i.e., a label not belonging to the example is predicted or a label belonging to the example is not predicted. Ranking loss measures the fraction of reversely ordered label pairs, i.e. an irrelevant label is ranked higher than a relevant label. One-error measures the fraction of examples whose top-ranked label is not in the relevant label set. Coverage measures how many steps are needed to move down the ranked label list so as to cover all the relevant labels of the

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	$Sim_2 = \sum_{j=1}^{c}\sqrt{P_j Q_j}$

Table 3: Evaluation measures for MLL algorithms.

Name	Formula
Hamming Loss	$hloss(h) = \frac{1}{P}\sum_{i=1}^{P}\lvert h(x_i) \triangle Y_i\rvert$
One Error	$one - error(f) = \frac{1}{P}\sum_{i=1}^{P}[\arg\max_{y\in Y} f(x_i, y)] \notin Y_i$
Coverage	$Coverage(f) = \frac{1}{P}\sum_{i=1}^{P}\max_{y\in Y_i} rank_f(x_i, y) - 1$
Rank Loss	$rloss(f) = \frac{1}{P}\sum_{i=1}^{P}\frac{1}{\lvert Y_i\rvert\lvert\overline{Y}_i\rvert}\cdot\lvert R\rvert, where$ $R = (y', y'')\lvert f(x_i, y') \leq f(x_i, y'') \in Y_i \times \overline{Y}_i$
Average Precision	$Average(f) = \frac{1}{P}\sum_{i=1}^{P}\frac{1}{Y_i}\sum_{y\in Y_i}\frac{\lvert P_i\rvert}{rank_f(x_i, y)}, where$ $P_i = y'\lvert rank_f(x_i, y') \leq rank_f(x_i, y), (y)' \in Y_i$

example. Average precision evaluates the average fraction of the relevant labels ranked higher than a particular label . For each evaluation metric, "↓" indicates "the smaller the better" while "↑" indicates "the larger the better".

Experiments setup

To demonstrate effectiveness of our proposed LDLLC algorithm, we carried out extensive experimental comparisons. LDLLC is first compared with seven existing label distribution learning methods, 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 et al. 2016). The parameter settings of algorithms are summarized 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 k in AA-kNN is set to 5. The number of hidden-layer neurons for AA-BP is set to 60. For BFGS-LLD, the parameters in Eqs. (8) and (9) are set to: $c_1 = 10^{-4}$ and $c_2 = 0.9$. For LDLLC, the parameters are set to: $\lambda_1 = 0.1$ and $\lambda_2 = 0.01$.

We also compare LDLLC with three widely used multi-label learning methods, namely ML-KNN (Zhang and Zhou 2007), Rank-SVM (Andre and Jason 2002) and BP-MLL (Zhang and Zhou 2006). Among compared algorithms, ML-KNN is derived from the traditional k-nearest neighbor (kNN) algorithm. Maximum a posteriori principle is used to determine which label set is related to the given instance. Rank-SVM provides a way of controlling the complexity of the overall learning system while having a small empirical error. The architecture of Rank-SVM is based on linear models of support vector machines. BP-MLL is derived from the famous backpropagation algorithm through employing a novel error function capturing the characteristics of multi-label learning. For multi-label learning methods, the value of k is set to 10 in ML-KNN. Rank-SVM uses the RBF kernel with the cost parameter c equals to 1. For BP-MLL, the number of hidden neurons in neural networks is 20% of the number of features. The learning rate α is set to 0.05 and the maximum times of iterations for training are 100.

Experimental Results

For each data set, we randomly partition the data into the training (80%) and test (20%) sets for classifier calibration and performance evaluation. Such precedure is repeated 10 times, and the average results as well as standard deviations are caculated over the 10 repetitions. Table 4 summarizes the average results of each compared algorithm on these eight data sets in terms of different evaluation metrics. When two or more algorithms obtain the same performance on one data set for a given evaluation metric, the value of rank for these algorithms are assigned with the average result of them. Furthermore, the best performance among the 7 comparing algorithms is shown in boldface.

The experimental results in Table 4 are in accordance with the "average and standard deviation (ranking)" format given here. Ranking refers to the prediction effect of 7 algorithms in metrics. The smaller the value the better the performance.

From Table 4, we find that our proposed LDLLC algorithm is superior to the other seven classical LDL algorithms on four datasets. On data sets Elu, Spo and Dtt, LDLLC algorithm also performs better than other algorithms on several metrics. BFGS-LLD obtains the best result on data set Diau. The PT-Bayes algorithm performs poorly on all eight data sets. It is possible that the Gauss distribution assumption in the PT-Bayes algorithm may not apply to complex mapping models from features to label distributions. Various algorithms often have different rankings on different evaluation metrics which reflects the diversity of the six evaluation measures. Thus, when comparing the prediction effects of two algorithms on a new data set, six kinds of evaluation metrics should be considered comprehensively.

The experimental results of performance comparison between the proposed approach and other multi-label learning baseline methods are summarized in Table 5. The results in Table 5 consistently indicate that the proposed LDLLC algorithm performs better than the multi-label learning algorithms on data sets Alpha, Cdc, Heat, and Dtt. For remaining three data sets, LDLLC also achieves the best results on four evaluation metrics. The multi-label learning can be realized by setting up a virtual label in LDLLC. The description degree of virtual label is used as the threshold to separate all the labels into relevant labels and irrelevant labels. As a sophisticated extension of multi-label learning, our LDLLC method can not only divide relevant labels and irrelevant ones but also get intensity information. This accordingly resulted in improved experimental results.

Table 4: Comparison results of different label distribution algorithms on eight datasets (mean±std(rank)).

data	algorithm	Euclidean↓	Sørensen↓	Squared χ^2 ↓	KL↓	Intersection↑	Fidelity↑
Alpha	LDLLC	**.0227±.0001(1)**	**.0272±.0001(1)**	**.0026±.0001(1)**	**.0054±.0001(1)**	**.9628±.0004(1)**	**.9987±.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	LDLLC	**.0276±.0009(1)**	**.0422±.0013(1)**	**.0035±.0004(1)**	**.0068±.0001(1)**	**.9577±.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	LDLLC	**.0277±.0006(1)**	**.0412±.0006(1)**	.0068±.0004(2)	.0068±.0003(2)	**.9580±.0013(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(7)	.0075±.0004(6)	.0073±.0003(6)	.9552±.0017(5)	.9979±.0009(6)
	EDL	.0289±.0005(2)	.0431±.0008(2)	**.0067±.0003(1)**	**.0067±.0003(1)**	.9569±.0007(2)	.9983±.0001(2)
Diau	LDLLC	.0541±.0022(3)	.0596±.0026(3)	.0132±.0005(2)	.0130±.0010(2)	.9404±.0026(3)	.9962±.0002(4)
	PT-Bayes	.4027±.0183(8)	.4177±.0170(8)	.5280±.0281(8)	.8512±.0772(8)	.5823±.0170(8)	.8230±.0107(8)
	PT-SVM	.0628±.0037(7)	.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(3)
	AA-BP	.0802±.0051(6)	.0863±.0059(7)	.0276±.0013(7)	.0291±.0069(7)	.9142±.0067(7)	.9929±.0031(7)
	IIS-LLD	.0539±.0031(2)	.0593±.0032(2)	.0144±.0014(3)	.0141±.0013(3)	.9407±.0003(2)	.9964±.0036(2)
	BFGS-LLD	**.0444±.0022(1)**	**.0476±.0023(1)**	**.0089±.0008(1)**	**.0083±.0009(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	LDLLC	**.0605±.0015(1)**	**.0610±.0016(1)**	**.0128±.0008(1)**	**.0131±.0006(1)**	**.9389±.0014(1)**	**.9966±.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	LDLLC	**.0806±.0019(1)**	**.0830±.0019(1)**	**.0222±.0007(1)**	.0236±.0013(2)	**.9169±.0019(1)**	.9939±.0003(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(2)	.0229±.0019(2)	**.0226±.0021(1)**	.9168±.0039(2)	**.9951±.0007(1)**
	EDL	.0843±.0029(3)	.0872±.0029(4)	.0268±.0015(4)	.0269±.0016(4)	.9128±.0028(4)	.9932±.0004(4)
Cold	LDLLC	**.0679±.0003(1)**	**.0589±.0019(1)**	**.0120±.0006(1)**	**.0119±.0006(1)**	**.9414±.0019(1)**	**.9967±.0023(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	LDLLC	**.0477±.0016(1)**	.0415±.0013(2)	.0061±.0003(2)	.0061±.0005(2)	**.9585±.0013(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(6)
	AA-kNN	.0512±.0019(4)	.0443±.0017(4)	.0071±.0007(5)	.0070±.0007(5)	.9557±.0017(4)	.9982±.0002(5)
	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(4)	.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(3)	.9560±.0018(3)	.9982±.0003(4)

Conclusion

As a generalization of multi-label learning and single-label learning, LDL can deal with label ambiguity problems by considering more label informations. To further improve the effectiveness of LDL, we expoited the label correlations and proposed a novel label distribution learning method LDLLC. The experimental results on several real data sets demonstrate that the proposed algorithm is suitable for the label distribution framework and can achieve good performances. In the present study, we calculated the Pearson's correlation coefficients for estimating the correlation between labels. In future work, we will try to find more accurate represention of the correlation between labels to further improve the LDL.

Table 5: Comparison results of different multi-label learning algorithms on eight datasets (mean±std(rank)).

data	algorithm	Hamming loss↓	Ranking loss↓	One Error ↓	Coverage↓	Average Precision↑
Alpha	LDLLC	**.4310±.0098(1)**	**.3947±.0112(1)**	**.3224±.0330(1)**	**15.2268±.1188(1)**	**.6519±.0086(1)**
	ML-KNN	.4380±.0100(2)	.3972±.0115(2)	.3320±.0335(2)	15.3320±.1192(3)	.6498±.0094(2)
	Rank-SVM	.4564±.0170(4)	.4258±.0205(4)	.4691±.1157(4)	15.3967±.1836(4)	.6185±.0236(4)
	BP-MLL	.4382±.0099(3)	.4141±.0131(3)	.3976±.0462(3)	15.3175±.1257(2)	.6333±.0119(3)
Cdc	LDLLC	**.4218±.0101(1)**	**.3968±.0104(1)**	**.3272±.0229(1)**	**12.4837±0.0417(1)**	**.6710±.0110(1)**
	ML-KNN	.4371±.0149(2)	.4220±.0169(3)	.3400±.0296(2)	12.7080±0.2843(3)	.6567±.0241(2)
	Rank-SVM	.4555±.0187(4)	.4313±.0228(4)	.3996±.0993(4)	12.8179±0.1330(4)	.6473±.0268(4)
	BP-MLL	.4429±.0082(3)	.4194±.0099(2)	.3703±.0298(3)	12.4894±0.0549(2)	.6535±.0107(3)
Elu	LDLLC	**.4162±.0179(1)**	**.3811±.0070(1)**	**.3240±.0043(1)**	11.4760±0.0326(3)	**.6845±.0011(1)**
	ML-KNN	.4181±.0161(2)	.3866±.0083(2)	.3248±.0125(2)	**11.3260±0.0682(1)**	.6828±.0081(2)
	Rank-SVM	.4452±.0153(4)	.0438±.0012(2)	.3732±.0480(4)	11.6093±0.2428(4)	.6612±.0100(4)
	BP-MLL	.4231±.0134(3)	.3982±.0139(3)	.3443±.0231(3)	11.4004±0.1080(2)	.6757±.0113(3)
Diau	LDLLC	.3295±.0157(2)	**.2721±.0191(1)**	**.2065±.0280(1)**	**4.2679±0.1221(1)**	**.7880±.0158(1)**
	ML-KNN	**.3293±.0107(1)**	.2787±.0133(2)	.2199±.0255(2)	4.2707±0.0890(2)	.7867±.0097(2)
	Rank-SVM	.3406±.0176(3)	.3066±.0308(3)	.2320±.0571(3)	4.5040±0.1313(3)	.7662±.0288(3)
	BP-MLL	.4533±.0381(4)	.4367±.0818(4)	.4057±.2007(4)	4.9947±0.2741(4)	.6808±.0669(4)
Heat	LDLLC	**.4205±.0096(1)**	**.3914±.0154(1)**	**.3650±.0176(1)**	**3.8293±0.0869(1)**	**.7175±.0098(1)**
	ML-KNN	.4383±.0085(2)	.4078±.0114(2)	.3837±.0239(2)	3.8417±0.0599(2)	.7094±.0081(2)
	Rank-SVM	.4987±.0094(4)	.5163±.0342(4)	.5313±.0338(4)	4.2386±0.1192(4)	.6354±.0190(4)
	BP-MLL	.4593±.0432(3)	.4258±.0263(3)	.3840±.0549(3)	3.8840±0.2109(3)	.7026±.0248(3)
Spo	LDLLC	**.4211±.0198(1)**	**.3981±.0243(1)**	.4098±.0331(2)	**3.4411±0.1106(1)**	**.7280±.0167(1)**
	ML-KNN	.4447±.0325(3)	.4217±.0268(3)	**.4080±.0317(1)**	3.4800±0.1341(2)	.7155±.0141(3)
	Rank-SVM	.5096±.0255(4)	.4934±.0561(4)	.5089±.0858(4)	4.0467±0.2704(4)	.6382±.0427(4)
	BP-MLL	.4250±.0243(2)	.4059±.0220(2)	.4171±.0270(3)	3.5715±0.1167(3)	.7163±.0143(2)
Cold	LDLLC	**.3740±.0393(1)**	**.3450±.0375(1)**	**.2640±.0536(1)**	2.0240±0.1566(3)	**.7821±.0348(1)**
	ML-KNN	.3876±.0168(3)	.3559±.0181(3)	.3171±.0354(3)	1.9411±0.0424(2)	.7682±.0141(3)
	Rank-SVM	.3834±.0103(2)	.3474±.0136(2)	.3069±.0221(2)	**1.9301±0.0391(1)**	.7772±.0085(2)
	BP-MLL	.4593±.0432(4)	.4258±.0263(4)	.3840±.0549(4)	3.8840±0.2109(4)	.7026±.0248(4)
Dtt	LDLLC	**.4192±.0138(1)**	**.3921±.0188(1)**	**.3687±.0245(1)**	**2.0593±0.0611(1)**	**.7535±.0107(1)**
	ML-KNN	.4291±.0145(2)	.3922±.0181(2)	.3736±.0210(2)	2.0598±0.0566(2)	.7511±.0114(2)
	Rank-SVM	.4380±.0359(3)	.4203±.0228(3)	.4520±.0157(3)	2.0880±0.0912(3)	.7252±.0206(3)
	BP-MLL	.4924±.0089(4)	.5055±.0593(4)	.4667±.0754(4)	2.3321±0.1179(4)	.6856±.0365(4)

Acknowledgements

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

References

Andre, E., and Jason, W. 2002. A kernel method for multi-labelled classification. *Advances in Neural Information Processing Systems* 681–687.

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 and Methods in Applied Sciences* 1(4):300–307.

Geng, X., and Hou, P. 2015. Pre-release prediction of crowd opinion on movies by label distribution learning. In *International Conference on Artificial Intelligence*, 3511–3517.

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

Geng, X., and Ling, M. 2017. Soft video parsing by label distribution learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 1331–1337.

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.

Hou, P.; Geng, X.; Hou, Z.; and Lv, J. 2017. Semi-supervised adaptive label distribution learning for facial age estimation. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2015–2021.

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.

Huang, S. J.; Yu, Y.; and Zhou, Z. H. 2012. Multi-label hy-

pothesis reuse. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 525–533.

Nocedal, J., and Wright, S. J. 2006. *Numerical optimization*. Springer.

Paul, E. 1992. An argument for basic emotions. *Cognition & Emotion* 6(3-4):169–200.

Tuia, D.; Verrelst, J.; Alonso, L.; Perez-Cruz, F.; and Camps-Valls, G. 2011. Multioutput support vector regression for remote sensing biophysical parameter estimation. *IEEE Geoscience & Remote Sensing Letters* 8(4):804–808.

Xing, C.; Geng, X.; and Xue, H. 2016. Logistic boosting regression for label distribution learning. In *Computer Vision and Pattern Recognition*, 4489–4497.

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.

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

Zhang, Y., and Yeung, D. Y. 2013. *Multilabel relationship learning*. ACM.

Zhang, M. L., and Zhang, K. 2010. Multi-label learning by exploiting label dependency. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 999–1008.

Zhang, M. L., and Zhou, Z. H. 2006. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge & Data Engineering* 18(10):1338–1351.

Zhang, M. L., and Zhou, Z. H. 2007. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition* 40(7):2038–2048.

Zhang, M., and Zhou, Z. 2014. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge & Data Engineering* 26(8):1819–1837.

Zhang, Z.; Wang, M.; and Geng, X. 2015. *Crowd counting in public video surveillance by label distribution learning*. Elsevier Science Publishers B. V.

Zhou, D. Y.; 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.