PAPER

Partial Label Metric Learning Based on Statistical Inference

Tian XIE[†a)], **Hongchang CHEN**[††], **Tuosiyu MING**[††], **Jianpeng ZHANG**[††b)], **Chao GAO**[††], **Shaomei LI**[††], ***Nonmembers***, ***and*** **Yuehang DING**[††], ***Member***

SUMMARY In partial label data, the ground-truth label of a training example is concealed in a set of candidate labels associated with the instance. As the ground-truth label is inaccessible, it is difficult to train the classifier via the label information. Consequently, manifold structure information is adopted, which is under the assumption that neighbor/similar instances in the feature space have similar labels in the label space. However, the real-world data may not fully satisfy this assumption. In this paper, a partial label metric learning method based on likelihood-ratio test is proposed to make partial label data satisfy the manifold assumption. Moreover, the proposed method needs no objective function and treats the data pairs asymmetrically. The experimental results on several real-world PLL datasets indicate that the proposed method outperforms the existing partial label metric learning methods in terms of classification accuracy and disambiguation accuracy while costs less time.

key words: *partial label learning, metric learning, statistical inference, likelihood-ratio test*

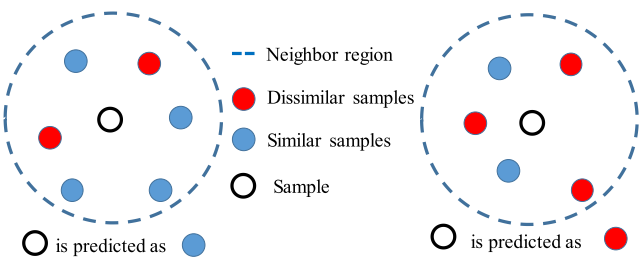
1. Introduction

Since strong supervision information is difficult to obtain due to the high cost of data labeling process, the demand of combining machine learning techniques and weak supervision arises in many real world scenarios. Partial label data is a kind of weakly supervised data in which the ground-truth label of each training example is hidden in a set of candidate labels [1]. The main purpose of partial label learning (PLL) is to train a multi-class classifier with partial label data. [2]

Formally speaking, suppose $\mathcal{X} \in \mathbb{R}^d$ is the d-dimensional feature space and $\mathcal{Y} = \{1, 2, \cdots, Q\}$ is the label space consisting of q class of labels, then the goal of PLL is to learn a multi-class classifier $f: \mathcal{X} \to \mathcal{Y}$ from partial label train set $D = \{(x_i, S_i) \mid 1 \le i \le n\}$. In the training set, $x_i \in \mathcal{X}$ is the feature vector of the instance and $S_i \in \mathcal{Y}$ is the candidate label of x_i. Particularly, the ground-truth label y_i of x_i is hidden in S_i and the learn algorithm is not capable of accessing it directly.

Apparently, since the ground-truth label in the training set is not accessible, it is difficult to learn from partial label data by using the label information directly. Thus, the manifold structure information among the training data is combined with the label information to train the PLL classifier by the state-of-the-art PLL algorithms like PL-KNN [2], IPAL [3] and PL-LEAF [4]. The manifold structure information used in these algorithms is obtained by the Euclidean distance under the manifold assumption. In detail, the manifold assumption assumes that the data in a local domain should have similar properties, which means that nearby instances in feature space should have the same label in the label space. Thus, these PLL algorithms predict the label of instance according to the label information of the nearby instance. However, the manifold assumption may not be satisfied by some real-world data, inevitably reducing the performance of PLL algorithms. For example, in Fig. 1, as the PLL algorithm predicts the label of instance via k-nearest neighborhood principle, the label of the instance may be predicted wrong, if the ground-truth label of an instance is different with that of its neighbors'.

To solve this problem, a simple idea is to map the feature vector of the instance to a new feature space in which the training data will have a new manifold structure and satisfy the manifold assumption as much as possible. In supervised learning, there are many methods to map the data to a new feature space such as isometric mapping [5], locally linear embedding [6], and metric learning [7]. However, the isometric mapping and locally linear embedding map the instance to a new feature space according to its neighbors in the feature space. Therefore, the original manifold structure of the data is preserved, which cannot solve the problem



(a) Right prediction (b) Wrong prediction

Fig. 1 An example of partial label data which do not satisfy the manifold assumption. (a) When the number of similar samples is more than that of dissimilar samples, the sample is predicted rightly. (b) When the number of similar samples is less than that of dissimilar samples, the sample is predicted wrong.

Manuscript received June 27, 2019.
Manuscript revised November 4, 2019.
Manuscript publicized March 5, 2020.

[†]The author is with Information Engineering University, Zhengzhou, Henan, China.
[††]The authors are with China National Digital Switching System Engineering & Technological R&D Center, Zhengzhou, Henan, China.
a) E-mail: xietianxt@foxmail.com
b) E-mail: zjp@ndsc.com.cn (Corresponding author)
DOI: 10.1587/transinf.2019EDP7182

that the data do not satisfy the manifold assumption. But, in supervised learning, metric learning is proposed to train a metric under which the train data will have a new manifold structure and satisfy the manifold assumption better. [8] Consequently, we consider using metric learning to map the data to the new feature space.

However, the traditional metric learning algorithm cannot be applied to the PLL problem because the true label of the example is unknown and the learning algorithm is unable to learn without ground-truth labels. Therefore, we consider generalizing traditional metric learning method to PLL problem, and propose a metric learning algorithm that is suitable for PLL. In practice, there is few related research on this field except that Zhou et al. proposed a partial label metric learning algorithm named PL-GMML [9], which builds sample pairs and learns the metric by minimizing the objective function. However, it is time consuming to calculate the objective function and it treats the similar and dissimilar pairs symmetrically, which will have an adverse effect on the result because of the impact of ambiguous label information.

Inspired by the KISS metric learning algorithm [8], in this paper, a pairwise statistical inference metric learning algorithm is proposed, which calculates a distance metric matrix during the training process. For reason that Euclidean distance and some other traditional distance measurements do not have adjustable parameters, we choose Mahalanobis distance as the distance metric. Then, we employ the likelihood-ratio test method to learn the metric for partial label data and propose a novel metric learning method named PMSI, i.e., *Partial-label Metric-learning based on Statistical Inference*. By using the statistical inference method, PMSI is capable of training the metric matrix M without objective function. Besides, during the statistical process, PMSI treats the similar and dissimilar pairs asymmetrically through regarding them as two independent Gaussian distributions. Finally, by utilizing the mapping matrix L obtained through Cholesky decomposition $M = LL^T$, the partial label data will be mapped into a new feature space which should satisfy the manifold assumption well.

To sum up, our contribution can be summarized as follows: 1) We propose a metric learning algorithm suitable for partial label data to make the data satisfy the manifold assumption well; 2) The proposed algorithm utilizes statistical inference method and needs no objective function so that it will be more efficient.

Experiment on several real-world PLL datasets showed that PMSI method is capable of improving the disambiguation accuracy and classification accuracy of PLL algorithms as a frontend. Moreover, PMSI outperforms the existing partial label metric learning method PL-GMML, and saves at least 47.3% of the training time.

2. The Proposed Method

2.1 Problem Statement

Let $D = \{(x_i, S_i) \mid 1 \leq i \leq n\}$ be the partial label training set, in which $x_i = (x_{i1}, x_{i2}, \cdots, x_{id})^T$, $x_i \in \mathbb{R}^d$ is the d-dimensional feature vector of the i-th instance and $S_i \in \mathcal{Y}$ is the candidate label set of x_i.

The main purpose of distance metric learning is to learn a Mahalanobis distance functions:

$$d_M^2(x_i, x_j) = (x_i - x_j)^T M (x_i - x_j) \tag{1}$$

from the training data pairs (x_i, x_j), which contains similar and dissimilar data pairs. Besides, $M \succcurlyeq 0$ is a positive semi-definite matrix with m^2 parameters, which can be adjusted during training process.

However, in partial label learning, we cannot get the similar and dissimilar training data pairs directly for reason that the ground-truth label of instance is inaccessible. As a result, the traditional metric learning algorithms cannot be applied to PLL problems. Nonetheless, considering that if two instances are from the same class, they must have shared label in their candidate label set. Therefore, to get the data pairs, we can measure the similarity between partial label instances by using the Jaccard index y_{ij} of their candidate label sets:

$$y_{ij} = \frac{|S_i \cap S_j|}{|S_i \cup S_j|} \tag{2}$$

$y_{ij} > 0$ denotes x_i have shared candidate labels with x_j, i.e., (x_i, x_j) is a similar pair and $y_{ij} = 0$ otherwise.

As shown in Eq. (1), if two data points are close to each other, they should have a low values of $d_M^2(x_i, x_j)$. Besides, the purpose of our work is to get a distance metric matrix under which the similar data will be close to each other. Thus, if x_i and x_j are from the same class, they should have a low value of $d_M^2(x_i, x_j)$. Suppose H_0 denotes that x_i and x_j are from different classes. Accordingly, H_1 denotes x_i and x_j are from the same class. Then, the distance between x_i and x_j are can be inferenced by the likelihood-ratio test of H_0 and H_1:

$$\delta(x_i, x_j) = \log\left(\frac{p(x_i, x_j \mid H_0)}{p(x_i, x_j \mid H_1)}\right) \tag{3}$$

The possibility of H_1 decreases with the value of $\delta(x_i, x_j)$.

To rule out the effect of the actual position of the instance in the feature space, we can calculate $\delta(x_i, x_j)$ by the difference of x_i and x_j: $x_{ij} = x_i - x_j$. Then the likelihood ratio Eq. (3) can be rewrite as:

$$\delta(x_i, x_j) = log\left(\frac{p(x_{ij} \mid H_0)}{p(x_{ij} \mid H_1)}\right) \tag{4}$$

Suppose that $p(x_{ij} \mid H_0)$ and $p(x_{ij} \mid H_1)$ obey the

Gaussian distribution $\mathcal{N}(0, \Sigma_{y_{ij}=0})$ and $\mathcal{N}(0, \Sigma_{y_{ij}>0})$ respectively, $\Sigma_{y_{ij}=0}$ and $\Sigma_{y_{ij}>0}$ are the corresponding covariance matrixes, which can be obtain according to statistics. Then, we can re-write Eq. (4) as:

$$\delta(x_{ij}) = log\left(\frac{\frac{1}{\sqrt{2\pi|\Sigma_{y_{ij}=0}|}}\exp\left(-\frac{1}{2}x_{ij}^T\Sigma_{y_{ij}>0}^{-1}x_{ij}\right)}{\frac{1}{\sqrt{2\pi|\Sigma_{y_{ij}>0}|}}\exp\left(-\frac{1}{2}x_{ij}^T\Sigma_{y_{ij}=0}^{-1}x_{ij}\right)}\right) \tag{5}$$

However, as there are many false labels in the candidate label set, two samples in a similar pair could also belong to different classes. Therefore, we should give each similar pair a confidence ratio to identify the possibility of that they belong to the same class. Considering that in the feature space, the instances close to each other are more likely to come from the same class. Accordingly, the confident ratio of that the samples in a similar pair are from the same class can be measured through the weight variable w_{ij}:

$$w_{ij} = 1 - \frac{d_{ij}}{\sum_{a\in N_{y_{ij}>0}(x_i)} d_{ia}} \tag{6}$$

where $N_{y_{ij}>0}(x_i)$ denotes the index of the k-nearest neighbors of x_i in the similar data set $\{j \,|\, 1 \le j \ne i \le n,\ y_{ij} > 0\}$, and d_{ij} is the distance between x_i and x_j. The value of w_{ij} in Eq. (6) will increase with the possibility of that they belong to the same class.

As PLL is a multi-class classification problem, the quantity of the dissimilar pairs is much larger than that of similar pairs. Therefore, in order to reduce the computation, we only consider the dissimilar pairs which are closer than the k-th nearest neighbor of x_i. Accordingly, the remaining dissimilar pairs can be expressed as $N_{y_{ij}=0}(x_i) = \{j \,|\, 1 < j \ne i \le n,\ y_{ij} = 0,\ d_{ij} < \max_{a\in N_{y_{ij}>0}(x_i)} d_{ia}\}$. Considering that in the feature space, the instances far from each other are more possible to come from the different classes. Then, the weight variable w_{ij} for the dissimilar pairs can be calculated:

$$w_{ij} = \frac{d_{ij}}{\sum_{a\in N_{y_{ij}=0}(x_i)} d_{ia}} \tag{7}$$

The value of w_{ij} in Eq. (7) will increase with the possibility of that they belong to the different classes.

After that, we can obtain the covariance matrix $\Sigma_{y_{ij}>0}$ and $\Sigma_{y_{ij}=0}$ by a statistical method with the weight variable w_{ij}:

$$\Sigma_{y_{ij}>0} = \frac{1}{1 - \sum_{N_{y_{ij}>0}(x_i)} w_{ij}^2} \sum_{N_{y_{ij}>0}(x_i)} w_{ij}x_{ij}x_{ij}^T \tag{8}$$

$$\Sigma_{y_{ij}=0} = \frac{1}{1 - \sum_{N_{y_{ij}=0}(x_i)} w_{ij}^2} \sum_{N_{y_{ij}=0}(x_i)} w_{ij}x_{ij}x_{ij}^T \tag{9}$$

Then, by taking the log, Eq. (5) can be written as:

$$\begin{aligned}\delta(x_{ij}) = &\frac{1}{2}\left(x_{ij}^T\Sigma_{y_{ij}>0}^{-1}x_{ij} - x_{ij}^T\Sigma_{y_{ij}=0}^{-1}x_{ij}\right)\\ &+ \frac{1}{2}\left(\log(|\Sigma_{y_{ij}=0}|) - \log(|\Sigma_{y_{ij}>0}|)\right)\end{aligned} \tag{10}$$

Table 1 The pseudo-code of PMSI

Algorithm 1 PMSI

Input: the partial label train set: D and the number of k-nearest neighbors in $N_{y_{ij}>0}(x_i)$: k

Output: the metric matrix M learned for partial label data

1: Calculate the candidate label set similarity y_{ij} according to Eq.(1)
2: Calculate weight variable w_{ij} according to Eq.(6) and (7)
3: Calculate weighted covariance $\Sigma_{y_{ij>0}}$ and $\Sigma_{y_{ij}=0}$ according to Eq.(8) and (9)
4: Calculate the matrix $\hat{M} = (\Sigma_{y_{ij}>0}^{-1} - \Sigma_{y_{ij}=0}^{-1})$
5: Using eigen-analysis to project $\hat{M}$ onto the PSD cone and get the metric matrix M

As the second term of Eq. (10) only provides an offset, so we simplify Eq. (10) to:

$$\delta(x_{ij}) = x_{ij}^T\left(\Sigma_{y_{ij}>0}^{-1} - \Sigma_{y_{ij}=0}^{-1}\right)x_{ij} \tag{11}$$

As the likelihood-ratio test in Eq. (11) and the Mahalanobis distance functions have the same monotonicity, $\delta(x_{ij})$ can be used to measure the distance between the instance x_i and x_j. Finally, by comparing Eq. (11) with Eq. (1), we can observe that the Mahalanobis distance metric matrix could be described as:

$$\hat{M} = \left(\Sigma_{y_{ij}>0}^{-1} - \Sigma_{y_{ij}=0}^{-1}\right) \tag{12}$$

Consequently, we do not need to solve any objective function to get the metric matrix M as the matrix $\left(\Sigma_{y_{ij}>0}^{-1} - \Sigma_{y_{ij}=0}^{-1}\right)$ is calculate through a statistical method.

Considering that M should be a positive semi-definite matrix, we can use eigen-analysis to project Eq. (12) onto its PSD (positive semi-definite) cone so that to get the metric matrix M.

2.2 Algorithm Description

The complete procedure of the proposed PMSI approach is summarized in Table 1. Given the partial label training set, the similar and dissimilar pairs are generated by calculating y_{ij}. After that the weight variable of the similar and dissimilar pairs are calculated in an asymmetrical method with different formulas. Accordingly, the data distributions of the similar and dissimilar pairs are obtained by statistical process and finally the metric matrix is calculated.

3. Experiment

3.1 Experiment Setup

The main purpose of PMSI is to learn a Mahalanobis distance metric matrix form partial label training set. Besides, under the learned metric, the partial label data will satisfy the manifold assumption well in order to promote the accuracy of PLL algorithms. Accordingly, to evaluate the performance of our distance metric learning algorithm, we do experiments on five real world datasets collected from different application domains. Specifically, *Lost* [10] and *Yahoo!*

Table 2 The detail characteristics of real-world data sets

Data set	Examples	Features	Classes	Candidate labels		
				Min	Max	Average
Lost	1122	108	16	1	3	2.23
MSRCv2	1758	48	23	1	7	3.16
BirdSong	4998	38	13	1	4	2.18
FG-NET	1002	262	78	2	11	7.48
Yahoo! News	22991	163	219	1	5	1.91

Table 3 Classification accuracy (mean ± std) of each comparing algorithm on the real-world partial label data sets

Algorithms	Accurancy(mean±std)				
	Lost	MSRCv2	FG-NET	BirdSong	Yahoo ! News
PLKNN	41.94±1.15•	41.79±0.20•	3.19±0.47•	55.32±0.26•	48.83±0.78•
PLKNN+PLGMML	52.32±0.78	42.72±3.08•	3.29±0.68•	**67.23±0.86**◦	50.80±0.46•
PLKNN+PMSI	**55.97±0.89**	**44.19±0.47**	**4.39±1.06**	66.07±0.37	**60.04±0.51**
IPAL	64.83±0.74•	53.07±0.35•	5.47±0.21•	58.22±0.37•	59.58±0.64•
IPAL+PLGMML	68.54±2.37•	52.84±2.58•	5.19±1.13•	61.16±0.27•	60.89±0.61•
IPAL+PMSI	**74.54±1.48**	**54.08±0.98**	**6.13±0.88**	**73.18±0.43**	**65.01±0.63**
PL-ECOC	62.92±2.33•	35.72±1.30•	2.32±0.95•	58.23±2.14•	44.53±2.08•
PL-ECOC+PLGMML	67.65±1.49•	43.57±1.66•	2.38±0.76•	73.19±1.35•	47.67±0.71•
PL-ECOC+PMSI	**70.41±1.24**	**43.80±1.32**	**5.45±0.39**	**74.23±1.61**	**53.45±0.82**
PL-LEAF	71.04±5.05•	51.82±2.05◦	7.34±1.32	55.68±1.12•	N/A
PL-LEAF+PLGMML	75.32±3.72•	**51.87±3.09**◦	7.39±0.85	58.24±1.20•	N/A
PL-LEAF+PMSI	**76.39±2.88**	49.49±3.20	**7.59±1.06**	**72.83±1.51**	N/A
PL-AGGD	74.52±4.97•	50.79±3.79◦	7.28±1.90•	53.86±1.01•	N/A
PL-AGGD+PLGMML	76.30±5.39•	**50.85±3.94**◦	**7.88±1.61**◦	56.42±1.22•	N/A
PL-AGGD+PMSI	**77.63±5.21**	48.12±3.51	7.79±1.97	**71.65±0.84**	N/A

•/◦ indicates whether PMSI is statistically superior/inferior to the comparing algorithm on each data set (pairwise t-test at 0.05 significance level)

News [11] are from face annotation problems, *MSRCv2* [12] is from object detection problem, *BirdSong* [13] is from bird song classification problem and *FG-NET* [14] is from human face age estimation problem. The detail characteristics of the real-world datasets are listed in Table 2.

Moreover, we combine our method with five state-of-the-art partial label learning algorithms to measure the performance:

- PL-KNN [2]: a k-nearest neighbor based partial label learning algorithm, constructs a similarity graph by k-nearest neighbor method and uses weighted voting to predict the label.
- IPAL [3]: a graph based partial label learning algorithm, regards the candidate labels equally and predicts the label by using label propagation algorithm.
- PL-LEAF [4]: a graph based partial label learning algorithm, calculates the confidence of each candidate label during the training phase. The algorithm learns the predictive model by carrying out regularized multi-output regression with confident variables.
- PL-ECOC [15]: a disambiguation-free partial label learning algorithm, represents the labels by binary codes, and builds a group of binary classifiers, using the binary output to predict the label of instance.
- PL-AGGD [16]: an adaptive graph guided partial label learning algorithm, which performs label disambiguation and predictive model training simultaneously by using adaptive graph.

Besides, the existing distance metric learning method for partial label data named PL-GMML [9], which uses geometric mean metric method to train the metric matrix M for partial label data, is used as comparison.

3.2 Experiment Results

Table 3 reports the mean classification accuracy and the standard deviation of each state-of-the-art partial label learning algorithm when it is (or not) combining with partial label distance metric learning method. For clarity, in Table 3, the best one among the three results of each PLL algorithm is marked in boldface. Two-sample t-test at 0.05 significance level is employed based on the ten-fold cross-validation while •/◦ indicates whether PMSI is statistically superior/inferior to the comparing algorithm on each data set.

As listed in Table 3, PMSI is capable of improv-

Table 4 Disambiguation accuracy (mean ± std) of each comparing algorithm on the real-world partial label data sets

Algorithms	Accurancy(mean±std)				
	Lost	MSRCv2	FG-NET	BirdSong	Yahoo ! News
PLKNN	53.67±0.24•	49.05±0.25•	8.49±0.13•	61.58±0.08•	60.15±0.10•
PLKNN+PLGMML	64.86±0.96•	49.96±0.90•	9.23±1.10•	**70.54±0.16**	61.92±0.46•
PLKNN+PMSI	**65.07±0.32**	**50.91±0.18**	**10.25±0.34**	69.84±0.83	**67.48±0.41**
IPAL	76.20±0.16•	**70.72±0.05**◦	15.13±0.32•	76.60±0.13•	82.02±0.17•
IPAL+PLGMML	77.16±2.14•	70.63±0.39◦	14.84±0.92•	78.12±0.33•	82.52±0.19•
IPAL+PMSI	**83.33±0.30**	70.19±0.56	**16.24±0.99**	**83.53±0.22**	**84.56±0.25**
PL-ECOC	69.88±0.81•	37.57±1.98•	5.25±0.79•	59.35±0.82•	45.52±1.79•
PL-ECOC+PLGMML	75.38±0.95•	**47.67±1.45**	5.25±0.59•	74.86±0.60•	48.64±0.23•
PL-ECOC+PMSI	**76.11±1.38**	47.57±0.74	**7.13±0.36**	**77.07±0.47**	**55.10±0.30**
PL-LEAF	78.77±1.93•	58.38±0.73◦	15.12±0.15	55.95±0.22•	N/A
PL-LEAF+PLGMML	80.24±1.64•	**58.66±0.63**◦	14.45±0.67•	58.57±0.15•	N/A
PL-LEAF+PMSI	**82.71±1.36**	55.18±1.03	**15.28±0.54**	**74.80±0.83**	N/A
PL-AGGD	82.78±1.65•	**62.63±1.61**◦	14.37±0.73•	54.02±0.34•	N/A
PL-AGGD+PLGMML	83.71±1.25•	62.40±1.73◦	14.67±1.26•	56.68±0.34•	N/A
PL-AGGD+PMSI	**85.18±1.07**	56.68±1.62	**15.67±0.73**	**73.93±1.19**	N/A

•/◦ indicates whether PMSI is statistically superior/inferior to the comparing algorithm on each data set (pairwise t-test at 0.05 significance level).

ing the performance of PLKNN, IPAL, PL-ECOC and PL-LEAF on all data sets. Besides, the improvement of these partial learning algorithms combining PMSI is superior to that of the state-of the-art partial label metric PL-GMML PLKNN, IPAL, PL-ECOC and PL-LEAF on all data sets for 0.2∼14.59%, except that of PLKNN on *BirdSong* data set and PL-AGGD on MSRCv2 data set. Besides, as we can see in Table 6, PMSI is more efficient than PL-GMML during the metric training phase and is at least 1.89 times faster than PL-GMML on all five data sets. In conclusion, PMSI performs advantageously than PL-GMML and consumes less time.

In addition to the classification performance listed in Table 3, the disambiguation performance, which reflects the capability to predict the ground-truth label of each instance form candidate label set, is also investigated in Table 4.

As listed in Table 4, it is clearly to observe that: 1) PMSI is capable of improving the disambiguation accuracy of all five partial label learning algorithms on *Lost*, *FG-NET*, *BirdSong* and *Yahoo! News* data sets and is capable of improving the performance of most of the PLL algorithms on MSRCv2 datasets. Especially, on *BirdSong* data set, PMSI is capable of improving the disambiguation accuracy of PL-LEAF algorithm by 18.85%, which is 16.33% higher than PL-GMML. 2). Out of the 23 comparative experiments, PMSI has a better disambiguation performance than PL-GMML on 18 comparative experiments.

Table 5 listed the win/tie/loss counts on the classification performance and disambiguation performance of PMSI against the comparing algorithms PL-GMML and the origin PLL algorithms which use Euclidean distance. It is clearly that in most of the cases, PMSI can improve classification performance and disambiguation performance of the PLL algorithm. Besides, PMSI has a higher improvement to the PLL algorithms than the existing partial label metric learning algorithm PL-GMML in most of experiments. Besides, in Table 6, The average training time costed by our method PMSI on the real-world partial label data sets is much less than that of PL-GMML.

Table 5 The win/tie/loss counts on the classification performance and disambiguation performance of PMSI against the comparing algorithms

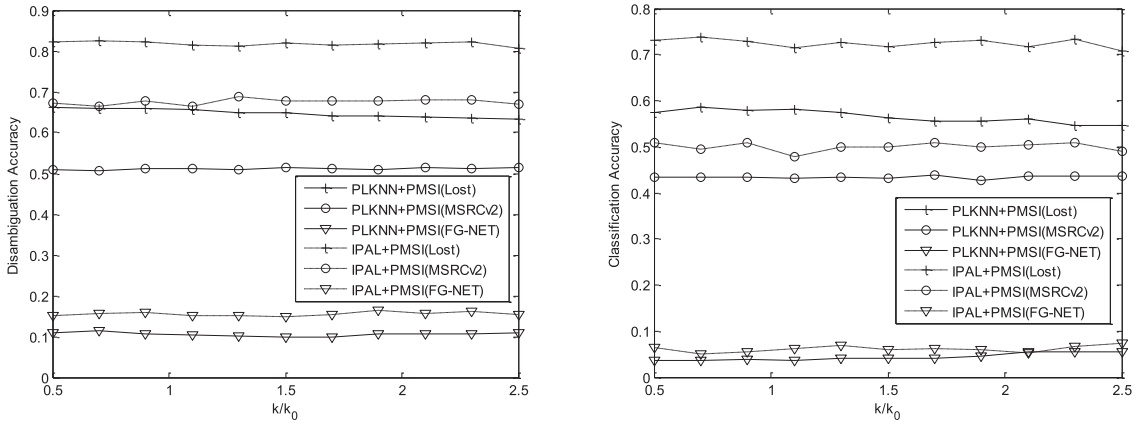
	PMSI against	
	Euclidean	PL-GMML
Disambiguation Accuracy	19/1/3	18/2/3
Classification Accuracy	20/1/2	19/1/3

Table 6 The average training time of our method PMSI and PL-GMML on the real-world partial label data sets

Data set	PL-GMML	PMSI
Lost	0.805s	**0.227s**
MSRCv2	1.091s	**0.196s**
FG-NET	1.199s	**0.571s**
BirdSong	3.509s	**1.379s**
Yahoo ! News	205.787s	**108.351s**

3.3 Parameter Sensitivity Analysis

According to the flowchart, PMSI learns from partial label examples by employing the parameter k, which denotes the number of k-nearest neighbors in $N_{y_{ij}>0}(x_i)$. To investigate the sensitivity of PMSI under parameter k, Fig. 2 illustrates the disambiguation accuracy and classification accuracy of PMSI using different parameter configurations. For the convenience of analysis, 3 data sets are chosen for sensitivity analysis and two PLL algorithms are chosen as backend of PMSI. It is obvious that the disambiguation accuracy and classification accuracy of PL-KNN and IPAL algorithms changes slightly while parameter k varies. Therefore, we



(a) Disambiguation accuracy changes by varying k

(b) Classification accuracy changes by varying k

Fig. 2 Parameter sensitivity analysis for PMSI algorithm on the Lost, MSRCv2 and FG-NET data sets. (a) Disambiguation accuracy of PMSI changes as the ratio k/k_0 increases from 0.5 to 2.5 with step-size 0.2, k is the number of neighbors used in PMSI, k_0 is the number of neighbors used by PLKNN, IPAL. (b) Classification accuracy of PMSI changes as the ratio k/k_0 increases from 0.5 to 2.5 with step-size 0.2, k is the number of neighbors used in PMSI, k_0 is the number of neighbors used by PLKNN, IPAL.

can set $k = k_0$ for convenience.

4. Conclusion

In this paper, a statistical inference based partial label metric learning algorithm PMSI was proposed, which utilizes likelihood-ratio test to obtain the metric matrix M for partial label data. The PMSI method calculates the metric matrix by the statistics distribution of similar and dissimilar sample pairs so that it needs no objective function and is time-saving. Moreover, as the metric matrix M is a semi-definite matrix, it can be decomposed to a mapping matrix L by Cholesky decomposition $M = LL^T$ and maps the data to a new feature space $x' = Lx$ in which the data will satisfy the manifold assumption better. Thus, the PMSI method can be used as a frontend of the state-of–the-art PLL algorithms to improve the performance on disambiguation and classification. Furthermore, the PMSI method compares favorably against the existing partial label metric learning algorithm PL-GMML on disambiguation accuracy and classification accuracy in most cases of the experiments and meanwhile demands at most 53% of the process time. In future work, we will research on add multi modal function to the partial label metric learning algorithms.

Acknowledgments

This work was supported by National Natural Science Foundation of China under Grant (61601513).

References

[1] Z.H. Zhou, "A brief introduction to weakly supervised learning," Natl. Sci. Rev., vol.5, no.2, pp.48–57, Jan. 2017.

[2] E. Hüllermeier and J. Beringer, "Learning from ambiguously labeled examples," Intell. Data Anal., vol.10, no.5, pp.419–439, Jan. 2006.

[3] M.L. Zhang and F. Yu, "Solving the partial label learning problem: An instance-based approach," Proc. 24th Int. Joint Conf. Artif. Intell., Argentina, pp.4048–4054, 2015.

[4] M.-L. Zhang, B.-B. Zhou, and X.-Y. Liu, "Partial label learning via feature-aware disambiguation," Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. data Min., USA, pp.1335–1344, 2016.

[5] J.B. Tenenbaum, "A global geometric framework for nonlinear dimensionality reduction," Science, vol.290, no.5500, pp.2319–2323, 2000.

[6] S.T. Roweis and L.K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," Science, vol.290, no.5500, pp.2323–2326, 2000.

[7] B. Kulis, "Metric learning: A survey," Found. and Trends in Mach. Learn., vol.5, no.4, pp.287–364, 2012.

[8] K. Martin, M. Hirzer, P. Wohlhart, P.M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," Proc. CVPR, USA, pp.2288–2295, 2012.

[9] Y. Zhou and H. Gu, "Geometric mean metric learning for partial label data," Neurocomputing, vol.275, pp.394–402, Aug. 2018.

[10] T. Cour and B. Sapp, "Learning from partial labels," J. Mach. Learn. Res., vol.12, no.2, pp.1501–1536, 2011.

[11] M. Guillaumin, J. Verbeek, and C. Schmid, "Multiple instance metric learning from automatically labeled bags of faces," Proc. ECCV, Greece, pp.634–647, 2010.

[12] L.L. Liu and T.G. Dietterich, "A conditional multinomial mixture model for superset label learning," Proc. Adv. Neural Inf. Process. Syst., USA, pp.557–565, 2012.

[13] F. Briggs, X.Z. Fern, and R. Raich, "Rank-loss support instance machines for MIML instance annotation," Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. data Min., Beijing, China, pp.534–542, 2012.

[14] G. Panis and A.L. B, "An overview of research activities in facial age estimation using the FG-NET aging database," Proc. ECCV, Switzerland, pp.737–750, 2014.

[15] M. Zhang, F. Yu, and C. Tang, "Disambiguation-free partial label learning," IEEE Trans. Knowl. Data Eng., vol.29, no.10, pp.2155–2167, 2017.

[16] D. Wang, L. Li, and M.-L. Zhang, "Adaptive Graph Guided Disambiguation for Partial Label Learning," Proc. 25th ACM SIGKDD Conf. on Knowl. Discov. and Data Min., Anchorage, AK, pp.83–91, 2019.

Tian Xie received the B.S. in Tsinghua University in 2017. He is currently pursuing the master's degree with the Information Engineering University. His research interests is weakly supervised learning.

Hongchang Chen is currently a professor in the National Digital Switching System Engineering & Technological R&D Center (NDSC), China. His research interests include machine learning and big data analytics.

Tuosiyu Ming received the B.S. in Beijing Institute of Technology in 2016. He is currently pursuing the master's degree with the Information Engineering University. His research interests is natural language processing.

Jianpeng Zhang is an assistant professor in the National Digital Switching System Engineering & Technological R&D Center (NDSC), China. His research interests include data mining, big data analytics and social network analysis.

Chao Gao is currently working as a lecturer at NDSC, Zhengzhou, China. His research interests include multiclass object detection, image categorization, and semantic segmentation.

Shaomei Li is currently working as an associate professor at NDSC. Her current research interests include pattern recognition and computer vision, especially human action analysis, target tracking, and image understanding.

Yuehang Ding received the B.S. in Harbin University of Science and Technology in 2017. She is currently pursuing the master's degree with the Information Engineering University. Her research interests is knowledge graph and natural language processing.