# A Partial Label Metric Learning Algorithm for Class Imbalanced Data

**Wenpeng Liu** liuwenpeng@dlnu.edu.cn
**Li Wang** 514957103@qq.com
**Jie Chen** tobbchenjie@gmail.com
**Yu Zhou** yuzhou829@sina.com
**Ruirui Zheng** zrr@dlnu.edu.cn
**Jianjun He** Corresponding author: jianjunhe@live.com
*College of Information and Communication Engineering, Dalian Minzu University, Dalian 116600, Liaoning, China*

**Editors:** Vineeth N Balasubramanian and Ivor Tsang

## Abstract

The performance of machine learning algorithms depends on the distance metric, in addition to the model and loss function, etc. The partial label metric learning technique can improve the accuracy of partial label learning algorithms by using training data to learn a better distance metric, which has gradually attracted the attention of scholars in recent years. The essence of partial label learning is mainly to deal with multi-class classification problems, while class imbalance is a common phenomenon in these problems. The class imbalanced problem affects the prediction accuracy of minority class samples, but the current partial label metric learning algorithms rarely consider the problem. In this paper, we propose two partial label metric learning algorithms (PL-CCML-SFN and PL-CCML-LDD) that can solve the class imbalanced problem. The basic idea is to add a regularization term to the objective function of the PL-CCML model, which can induce each class to be uniformly distributed in the new metric space and thus play the role of balancing each class. The experimental results show that these two algorithms, compared with the existing partial label metric learning algorithms, have improved the overall performance on the class imbalanced data.

**Keywords:** partial label learning, class imbalanced data, metric learning.

## 1. Introduction

In recent years, with the rapid development of the AI industry, AI applications promoted by machine learning have achieved close to or even surpassed human performance in numerous practical tasks. This is not only due to the progress of research on algorithms and computing power by technology giants and research institutions, but also benefits from the large-scale data sets available for training various models. The data in these tasks are often required to be accurately labeled, but in real life, constrained by the external environment, problem characteristics, actual cost and physical resources, we often can only access to the data with weakly supervised label information. Therefore, learning from weakly supervised data becomes an important aspect of machine learning research.

Partial label learning is one of the effective framework to solve classification problems with weakly supervised data, and it is also known as soft label learning (Côme et al., 2009), superset label learning (Liu and Dietterich, 2014), and ambiguous label learning (Hüllermeier and Beringer, 2006). In the partial label learning framework, each input object corresponds to a set of candidate labels, of which only one label is the true label of the input object. The class imbalanced problem refers to the imbalanced distribution of samples among classes, where the number of samples in some classes is much larger than the number of samples in other classes, and we call the classes with a less number of samples as minority class samples and the classes with a larger number of samples as majority class samples. Class imbalance is a common phenomenon in multi-class classification problems such as disease detection, criminal behavior analysis, and consumer behavior learning. In multi-class classification problems, minority class samples often carry important information and should be focused on (Ye et al., 2009), because their misclassification is often more costly than majority class samples. As a multi-class classification framework, partial label learning also faces the class imbalanced problem. However, the existing partial label learning algorithms, especially the partial label metric learning algorithms, rarely consider the impact of class imbalance on their performance. Therefore, in this paper, we propose two partial label metric learning algorithms that can solve the class imbalanced problem. The main contributions include:

- We proposed two partial label metric learning algorithms (termed as PL-CCML-SFN and PL-CCML-LDD) that can learn the metric matrix by drawing on the ideas of the collapsing classes model and the neighborhood component analysis(NCA) model, and construct objective functions containing regularization terms that can induce a uniform distribution of each class in the new metric space.

- Experimental results on four real-world data sets widely used in the field of partial label learning show that our proposed PL-CCML-SFN and PL-CCML-LDD algorithms can effectively improve the overall performance on the class imbalanced data.

## 2. Related work

Partial label learning(PLL) plays an important role in solving complex real-world problems and has gained wide application in computer vision (Zeng et al., 2013), Internet (Luo and Orabona, 2010), ecological informatics (Zhang et al., 2016) and other fields. In 2002, Jin and Ghahramani (2002) formalized the partial label learning framework as a new machine learning framework and proposed a discriminative approach. Propelled by their work, it began to attract more and more attentions. In (Hüllermeier and Beringer, 2006), three traditional classification models including nearest neighbor classification, decision tree, and rule induction were extended to the PLL framework. Côme et al. (2009) developed a PLL algorithm by using maximum likelihood estimation strategy and belief function theory. Luo and Orabona (2010) introduced a large margin formulation to solve the PLL problem. Cour et al. (2011) developed a support vector machines based PLL algorithms. Liu and Dietterich (2012) presented a conditional multinomial mixture model for PLL problem. Chen et al. (2014) proposed a PLL algorithm based on dictionary learning. Zhou et al. (2017) proposed a Gaussian process model based PLL algorithm. Gong et al. (2017) developed a

regularization approach for instance-based PLL model. Tang and Zhang (2017) proposed a PLL algorithm based on boosting technique. Wang et al. (2019) proposed an adaptive graph guided disambiguation model. Feng and An (2019) developed a PLL algorithm by using semantic difference maximization. Lyu et al. (2020) proposed a self-paced PLL algorithm. Yao et al. (2020a,b) proposed two PLL algorithms by deep learning technologies. The existing partial label learning research mainly focuses on how to disambiguate candidate labels to establish more effective learning algorithms based on various models.

Distance metric learning is a way to automatically learn the distance metric to meet specific requirements using the information provided by training samples. According to different learning methods, distance metric learning algorithms can be classified into unsupervised, supervised and semi-supervised. The idea of unsupervised distance metric learning algorithm is to map the original data set into a low-dimensional subspace by means of dimensionality reduction, so as to obtain a low-dimensional representation of the original data set. For example, the ATM (Xie et al., 2018) algorithm, the PCA (Bar-Hillel et al., 2003) algorithm, and the LPP (He and Niyogi, 2004) algorithm. The main idea of the supervised distance metric learning algorithm is to use the sample information of the training set to obtain a metric matrix that effectively reflects the spatial relationship of the samples by optimizing the objective function, such as the LMNN (Weinberger and Saul, 2009) algorithm, the LDA (Fukunada, 1990) algorithm, the NCA (Goldberger et al., 2004) algorithm, etc. Semi-supervised distance metric learning algorithms are usually combination of supervised and unsupervised distance metric learning algorithms to solve the problem with a small number of known samples, including the LRML (Hoi et al., 2010) algorithm, etc. Inspired by the excellent performance of distance metric learning technology under the traditional learning framework, Zhou and Gu (2018) proposed the first partial label metric learning algorithm (PL-GMML) based on geometric mean model, which really could improve the accuracy of the existing partial label learning algorithms. Drawing on the idea of PL-GMML method, Xu et al. (2020) proposed a partial label metric learning algorithm (PL-CCML) based on collapsing classes model. Since it is difficult to precisely determine whether a pair of samples belong to the same class, at present, there are few studies on partial label metric learning.

Performing high-accuracy classification using class imbalanced data has been a challenge for a long time, and there have been considerable number of scholars discussing novel methods to address the problem. The methods can be roughly categorized into four branches. The first branch is the cost-sensitive algorithms that address the problem by using imbalance-sensitive target functions, and assigning special loss functions explicitly or implicitly. For example, Khan et al. (2017) proposed a cost-sensitive deep neural network which can automatically learn robust feature representations for both the majority and minority classes. The second branch is the one-class learning methods that solve the label-imbalanced problem by learning the representation of the majority or minority data. For example, Luo et al. (2018) proposed a novel divergence-encouraging autoencoder (DEA) to explicitly learn features from both of the two classes and have designed an imbalanced data classification algorithm based on the proposed autoencoder. The third branch is the ensemble methods that contain the dynamic ensemble of classifiers. For example, Krawczyk et al. (2018) used the dynamic ensemble of one-class classifiers to train the model with regard to multiple classes; and Brun et al. (2018) proposed adjusting the ensemble based on

the difficulty of classification. The final branch is the re-sampling methods that generate balanced data by under-sampling the majority class, or over-sampling the minority class. For example, Aridas et al. (2019) proposed an under-sampling approach which leverages the usage of a Naive Bayes classifier, in order to select the most informative instances from the available training set, based on a random initial selection. Existing class imbalanced research is mainly conducted for the problem of classifier construction, while the class imbalanced problem in metric learning is rarely considered. Recently, several works (Xie et al., 2018) investigated orthogonality promoting regularization, which encourages the projection vectors in metric learning to be close to being orthogonal. They found that this can make the algorithm perform equally well on samples belonging to minority and majority classes. Therefore, this provides a new strategy for us to solve the class imbalanced problem in metric learning.

The above works show that it is important to explore algorithms to effectively solve the class imbalanced problem in partial label metric learning.

## 3. Proposed method

The basic idea of the collapsing classes model based partial label metric learning algorithm PL-CCML (Xu et al., 2020) is first to take each training sample and its neighbor with shared candidate labels as a similar pair, while each training sample and its neighbor without shared candidate labels as a dissimilar pair, then two probability distributions are defined based on the distance and label similarity of these pairs respectively, finally, the metric matrix is obtained via minimizing the KL divergence of these two probability distributions. The PL-CCML algorithm can achieve good results on the class balanced partial label learning problem, but the results on the class imbalanced problem are not good. In view of this, based on the PL-CCML algorithm, this paper proposes two partial label metric learning algorithms that can deal with class imbalanced problem. The basic idea is to add a regularization item to the objective function of the PL-CCML model that can promote the uniform distribution of classes in the new metric space, thereby balancing each class. Specifically, referring to the idea of literature (Xie et al., 2018), we achieve this goal by making the projection vectors in the new metric space be close to orthogonal. Since two regularization terms named SFN (Squared Frobenius Norm) and LDD (Log Determinant Divergence), two partial label metric learning algorithms named PL-CCML-SFN and PL-CCML-LDD are proposed, respectively. The framework of the proposed algorithms is shown in Fig 1.

### 3.1. Objective function

Suppose $S = \{(x_i, Y_i) \mid i = 1, 2, \cdots, n\}$ is the training set, where $x_i$ denotes the feature vector of the $i$-th training sample, and $Y_i$ is the set of candidate class labels of $x_i$. Similar to traditional metric learning, partial label metric learning also uses the training set to obtain a metric matrix $M$ so that the distance (1) meets certain requirements.

$$d\left(x, x' \mid M\right) = \sqrt{(x - x')^{\mathrm{T}} M \left(x - x'\right)}, \tag{1}$$

where $x, x' \in R^d$ denotes the feature vector of the two samples and $M$ is a symmetric positive definite matrix.

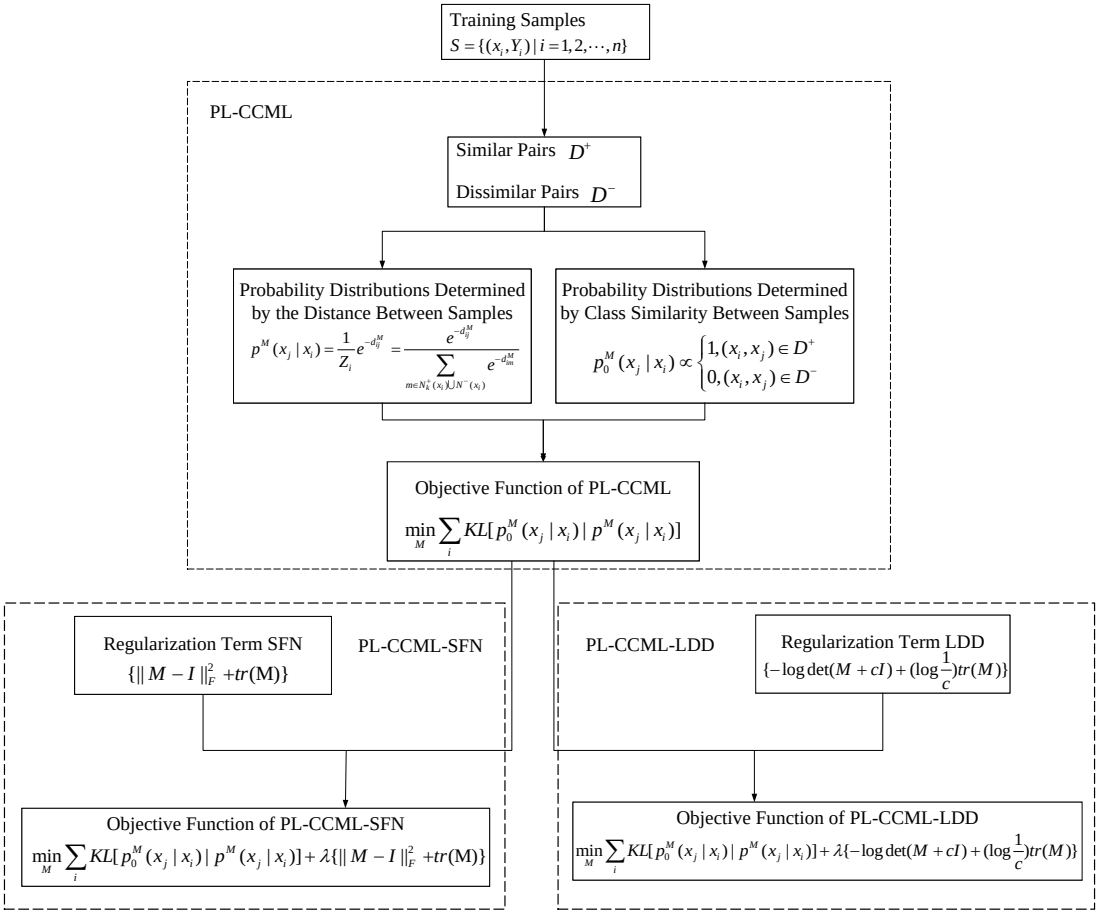


Figure 1: The Framework of PL-CCML-SFN and PL-CCML-LDD algorithms.

Usually, it is required that the distance between samples of the same class in the new metric space is as small as possible, and the distance between samples of different classes is as large as possible. Therefore, the main strategy of existing metric learning algorithms is to learn the metric matrix by constructing a set of similar pairs and a set of dissimilar pairs. However, in partial label learning framework, since each training sample corresponds to a set of candidate labels, it is difficult to determine whether two samples belong to the same class. To address this problem, PL-CCML algorithm proposed a following procedure to obtain similar pairs and dissimilar pairs: for each training sample $x_i$, $X^+(x_i)=\{x_j\mid j=1,2,\cdots,n,j\neq i,Y_i\cap Y_j\neq\varnothing\}$ denotes the set of training samples whose candidate labels intersects with that of $x_i$, $X^-(x_i)=\{x_j\mid j=1,2,\cdots,n,Y_i\cap Y_j=\varnothing\}$ denotes the set of training samples whose candidate labels does not intersect with that of $x_i$, $N^-(x_i)=\left\{j\mid x_j\in X^-(x_i),\|x_i-x_j\|<\max_{i_a\in N_k^+(x_i)}\|x_{i_a}-x_i\|\right\}$ denotes the index set of samples in $X^-(x_i)$ whose distance to $x_i$ is smaller than the distance between $x_i$ and the k-th nearest neighbor of $x_i$ in $X^+(x_i)$, and $N_k^+(x_i)=\{j\mid x_j$ is the k-nearest neighbor of $x_i$ in $X^+(x_i)\}$ denotes the index set of k samples with the smallest distance to $x_i$ in set $X^+(x_i)$. The illus-

trations of the index sets $N_k^+(x_i)$ and $N^-(x_i)$ are shown in Fig 2. According to the above definitions, the set of similar pairs $D^+$ and the set of dissimilar pairs $D^-$ can be defined as:

$$\begin{cases} D^+ = \left\{(x_i, x_j) \mid i = 1, 2, \cdots, n, j \in N_k^+(x_i)\right\} \\ D^- = \left\{(x_i, x_j) \mid i = 1, 2, \cdots, n, j \in N^-(x_i)\right\}. \end{cases} \tag{2}$$

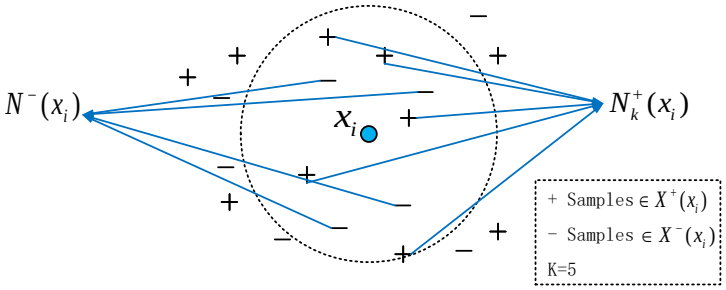


Figure 2: Explanation of the index sets $N_k^+(x_i)$ and $N^-(x_i)$.

Inspired by the idea of the NCA model (Goldberger et al., 2004), for each training sample $x_i$, in the metric space determined by the metric matrix $M$, the probability $P^M(x_j \mid x_i)$ that samples $x_j$ $\left(j \in N_k^+(x_i) \cup N^-(x_i)\right)$ and $x_i$ are of the same class can be defined as:

$$p^M(x_j \mid x_i) = \frac{1}{Z_i} e^{-d_{ij}^M} = \frac{e^{-d_{ij}^M}}{\sum_{m \in N_k^+(x_i) \cup N^-(x_i)} e^{-d_{im}^M}}, \tag{3}$$

where $d_{ij}^M = d^2(x_i, x_j \mid M) = (x_i - x_j)^T M (x_i - x_j)$, it can be seen that the smaller the distance between $x_j$ and $x_i$, the higher the probability that they are of the same class, and the larger the distance between $x_j$ and $x_i$, the lower the probability that they are of the same class. The ideal form of the probability that samples $x_j$ and $x_i$ are of the same class is as follows:

$$p_0^M(x_j \mid x_i) \propto \begin{cases} 1, (x_i, x_j) \in D^+ \\ 0, (x_i, x_j) \in D^-. \end{cases} \tag{4}$$

In order to find a metric matrix $M$ so that similar paris are as close as possible and dissimilar paris are as far as possible, the PL-CCML algorithm proposed the following objective function:

$$\min_M f(M) = \min_M \sum_i KL\left[p_0^M(x_j \mid x_i) \mid p^M(x_j \mid x_i)\right], \tag{5}$$

where $KL[\cdot \mid \cdot]$ denotes the KL divergence between two probability distributions.

The above objective function can better deal with the training data with class balanced problem, but the performance will be significantly reduced on the class imbalanced training

set. Moreover, the ambiguity of training samples makes traditional class imbalanced learning techniques such as cost-sensitive and re-sampling methods no longer practical here. Recently, several works (Xie et al., 2018) investigated orthogonality promoting regularization, which encourages the projection vectors in metric learning to be close to being orthogonal. They found that this can make the algorithm perform equally well on samples belonging to minority and majority classes. Therefore, this provides a new strategy for us to solve the class imbalanced problem in metric learning. Inspired by Xie et al. (2018), we use the regularizer SFN and the regularizer LDD as regularization terms to make the projection vectors orthogonal to each other and indirectly make the uniform distribution of classes in the new metric space, thus effectively reducing the impact of class imbalance.

According to Xie et al. (2018), the regularizer SFN and the regularizer LDD on the metric matrix $M$ can be defined as

$$\hat{\Omega}_{\text{sfn}}\left(M\right) = \|M - I\|_F^2 + \text{tr(M)}, \tag{6}$$

and

$$\hat{\Omega}_{\text{ldd}}\left(M\right) = -\log\det(M + cI) + \left(\log\frac{1}{c}\right)\text{tr}(M). \tag{7}$$

Therefore, the objective functions of the proposed PL-CCML-SFN and PL-CCML-LDD algorithms are defined as

$$\min_M \sum_i KL\left[p_0^M\left(x_j \mid x_i\right) \mid p^M\left(x_j \mid x_i\right)\right] + \lambda\left\{\|M - I\|_F^2 + \text{tr(M)}\right\} \triangleq \min_M f_{SFN}(M), \tag{8}$$

and

$$\begin{aligned}&\min_M \sum_i KL\left[p_0^M\left(x_j \mid x_i\right) \mid p^M\left(x_j \mid x_i\right)\right] + \lambda\left\{-\log\det(M + cI) + \left(\log\frac{1}{c}\right)\text{tr}(M)\right\} \\ &\triangleq \min_M f_{LDD}(M).\end{aligned} \tag{9}$$

In the case of discrete random variables, the KL divergence can be defined as:

$$KL[P(x) \mid Q(x)] = \sum_{x \in X} P(x)\log\frac{P(x)}{Q(x)}, \tag{10}$$

where $P(x)$, $Q(x)$ are the two probability distributions on the random variable $X$. Based on formula (10), the objective function (5) can be written as:

$$
\begin{aligned}
&\min_M f(M)\\
=&\min_M \sum_i KL\left[p_0^M\left(x_j \mid x_i\right) \mid p^M\left(x_j \mid x_i\right)\right]\\
=&\min_M \sum_i \sum_{j\in N_k^+(x_i)\cup N^-(x_i)} p_0^M\left(x_j \mid x_i\right)\log\frac{p_0^M\left(x_j \mid x_i\right)}{p^M\left(x_j \mid x_i\right)}\\
=&\min_M \sum_i \sum_{j\in N_k^+(x_i)} p_0^M\left(x_j \mid x_i\right)\log\frac{p_0^M\left(x_j \mid x_i\right)}{p^M\left(x_j \mid x_i\right)} + \sum_i \sum_{j\in N^-(x_i)} p_0^M\left(x_j \mid x_i\right)\log\frac{p_0^M\left(x_j \mid x_i\right)}{p^M\left(x_j \mid x_i\right)}\\
=&\min_M \sum_i \sum_{j\in N_k^+(x_i)} p_0^M\left(x_j \mid x_i\right)\log\frac{p_0^M\left(x_j \mid x_i\right)}{p^M\left(x_j \mid x_i\right)}\\
=&\min_M \sum_i \sum_{j\in N_k^+(x_i)} \log\frac{1}{p^M\left(x_j \mid x_i\right)}\\
=&\min_M \sum_i \sum_{j\in N_k^+(x_i)} \log\frac{\sum_{m\in N_k^+(x_i)\cup N^-(x_i)} e^{-d_{im}^M}}{e^{-d_{ij}^M}}\\
=&\min_M \left\{ k\sum_i \log \sum_{m\in N_k^+(x_i)\cup N^-(x_i)} e^{-(x_i-x_m)^{\mathrm{T}}M(x_i-x_m)} + \sum_i \sum_{j\in N_k^+(x_i)} (x_i-x_j)^{\mathrm{T}} M\left(x_i-x_j\right) \right\}.
\end{aligned}
\tag{11}
$$

According to (8), (9) and (11), the objective functions of the PL-CCML-SFN and PL-CCML-LDD algorithms can be written as:

$$
\begin{aligned}
\min_M f_{SFN}(M) = \min_M \Bigg\{ & k\sum_i \log \sum_{m\in N_k^+(x_i)\cup N^-(x_i)} e^{-(x_i-x_m)^{\mathrm{T}}M(x_i-x_m)} + \\
& \sum_i \sum_{j\in N_k^+(x_i)} (x_i-x_j)^{\mathrm{T}} M\left(x_i-x_j\right) + \lambda\left\{\|M-I\|_F^2 + \mathrm{tr(M)}\right\} \Bigg\},
\end{aligned}
\tag{12}
$$

and

$$
\begin{aligned}
\min_M f_{LDD}(M) = \min_M \Bigg\{ & k\sum_i \log \sum_{m\in N_k^+(x_i)\cup N^-(x_i)} e^{-(x_i-x_m)^{\mathrm{T}}M(x_i-x_m)} + \\
& \sum_i \sum_{j\in N_k^+(x_i)} (x_i-x_j)^{\mathrm{T}} M\left(x_i-x_j\right) + \lambda\left\{-\log\det(M+cI) + \left(\log\frac{1}{c}\right)\mathrm{tr}(M)\right\} \Bigg\}.
\end{aligned}
\tag{13}
$$

### 3.2. Model solving

In this paper, the gradient descent method is used to solve the objective functions, and the iteration rules are as follows:

$$M_{t+1} = M_t - lr_{t+1} \nabla f\left(M_t\right), \tag{14}$$

where $t$ is the number of iterations, $\nabla f\left(M_t\right)$ is the gradient of $f(M)$ at point $M_t$, $lr_{t+1}$ is the step length of the t+1th step, and $lr_{t+1}$ updated by the following formula:

$$lr_{t+1} = \rho \times \frac{1}{1 + \text{decay} \times t}, \tag{15}$$

where $\rho$ is the learning rate, *decay* is the attenuation rate. By deriving the objective function (12) and (13), $\nabla f(M)$ can be obtained as follows:

$$\begin{aligned} \nabla f_{SFN}(M) = & -k \sum_i \frac{\sum_{m \in N_k^+(x_i) \cup N^-(x_i)} e^{-(x_i - x_m)^{\mathrm{T}} M (x_i - x_m)} \left(x_i - x_m\right) \left(x_i - x_m\right)^{\mathrm{T}}}{\sum_{m \in N_k^+(x_i) \cup N^-(x_i)} e^{-(x_i - x_m)^{\mathrm{T}} M (x_i - x_m)}} + \\ & \sum_i \sum_{j \in N_k^+(x_i)} \left(x_i - x_j\right) \left(x_i - x_j\right)^{\mathrm{T}} + \lambda (2M - I), \end{aligned} \tag{16}$$

and

$$\begin{aligned} \nabla f_{LDD}(M) = & -k \sum_i \frac{\sum_{m \in N_k^+(x_i) \cup N^-(x_i)} e^{-(x_i - x_m)^{\mathrm{T}} M (x_i - x_m)} \left(x_i - x_m\right) \left(x_i - x_m\right)^{\mathrm{T}}}{\sum_{m \in N_k^+(x_i) \cup N^-(x_i)} e^{-(x_i - x_m)^{\mathrm{T}} M (x_i - x_m)}} + \\ & \sum_i \sum_{j \in N_k^+(x_i)} \left(x_i - x_j\right) \left(x_i - x_j\right)^{\mathrm{T}} + \lambda \left\{ I \log \frac{1}{c} - (M + cI)^{-1} \right\}. \end{aligned} \tag{17}$$

### 3.3. Algorithm implementation

The detailed flowchart of the proposed algorithms is shown in Algorithm 1. In the flowchart, we first initialize the parameters, and construct similar pairs $D^+$ and dissimilar pairs $D^-$, then, for each iteration, we need to calculate the gradients $\nabla f_{SFN}(M)$ and $\nabla f_{LDD}(M)$, update the step length $lr_{t+1}$ and the metric matrix $M_{t+1}$, finally, calculate the eigenvalues $\{\lambda_l\}$ and eigenvectors $\{u_l\}$ of the metric matrix $M_{t+1}$, and update the metric matrix $M_{t+1}$ again. After the iteration, we get the new metric matrix $M_{t+1}$. It can be seen from the flowchart that the computational cost of the proposed algorithm is mainly dominated by step 2, 3 and 6 of the fowchart, it takes about $O\left(nd^2\right)$ operations in step 3 and $O\left(d^3\right)$ operations in step 6, thus, the total computational cost is $O\left(n^2 d\right)$ when $n > d$, otherwise it is $O\left(d^3\right)$, where $n$ is the number of samples and $d$ is the dimension of feature vector. Because it mainly needs to store the training set $S$, sample pairs $D^+$ and $D^-$, gradients $\nabla f_{SFN}(M)$ and $\nabla f_{LDD}(M)$ and metric matrix $M_{t+1}$, the store requirement is $O\left(nd\right)$ when $n > d$, otherwise it is $O\left(d^2\right)$.

**Algorithm 1:** PL-CCML-SFN and PL-CCML-LDD

**Input:** Training sets $S = \{(x_i, Y_i) \mid i = 1, 2, \cdots, n\}$, regularization parameter $\lambda$, $c$, learning rate $\rho$, attenuation rate $decay$, k-nearest neighbor parameter $k$

1. Initialization: $t \leftarrow 0$, $M_t \leftarrow I$.
2. According to (2), construct $D^+$ and $D^-$.
3. According to (16) or (17) , calculate the gradients of $f_{SFN}(M)$ or $f_{LDD}(M)$ at $M_t$.
4. Updating the step length $lr_{t+1}$ according to (15).
5. Updating $M_{t+1}$ according to (14);
6. Calculating the eigenvalues $\{\lambda_l\}$ and the eigenvectors $\{u_l\}$ of $M_{t+1}$, replacing the negative values in $\{\lambda_l\}$ with 0, updating $M_{t+1} \leftarrow \sum_l max(\lambda_l, 0) u_l u_l^T$.
7. If $t$ does not reach the maximum iterations, $t \leftarrow t + 1$, return step **3**.

**Output:** $M_{t+1}$

## 4. Experiments

### 4.1. Experimental settings

In this paper, four real-world data sets, FG-NET, Lost, MSRCv2, and BirdSong, which are widely used in the field of partial label learning, are selected for experiments. The FG-NET (Weinberger and Saul, 2009) data set comes from the face age estimation problem containing 1002 images of 78 people, in which the set of candidate labels for each face image consists of the set of labeled ages and true ages. The Lost (Côme et al., 2008) data set is composed of 1122 face images of 16 people cutting from TV series, in which the set of candidate labels for each face image consists of the names in the associated captions. The MSRCv2 (Hüllermeier and Beringer, 2006) data set comes from the target detection problem containing 1758 segmented image regions from 23 classes, in which the set of candidate labels for each object region consists of the classes of all objects that appear on the same image. The BirdSong (Fukunada, 1990) data set comes from the bird song classification task containing 4998 bird song audios from 13 different bird species, in which the set of candidate labels for each audio consists of the names of all bird species around the recording device. Since this paper is oriented to the processing of class imbalanced data, in order to evaluate the performance of the proposed algorithms, it is necessary to use the partial label data sets with more significant class imbalanced problems. Therefore, we first test the class imbalance of these four data sets. We sort the number of samples by class in the data sets, then take the top 20% class samples with the largest number in each data set as the majority class samples, and the top 20% class samples with the least number as the minority class samples, and find that the average number of the majority class samples in the four data sets is much larger than the average number of the minority class samples, which is consistent with the class imbalanced data characteristics. Some details of these data sets are shown in Table 1.

Table 1: Characteristics of the experimental data sets.

| Data sets | Samples | Labels | Majority class samples | Minority class samples |
|---|---|---|---|---|
| FG-NET | 1002 | 78 | 40 | 2 |
| Lost | 1122 | 16 | 181 | 21 |
| MSRCv2 | 1758 | 23 | 191 | 23 |
| BirdSong | 4998 | 13 | 897 | 78 |

In this paper, we use ten-time five-fold cross validation to make the results more accurate. In order to better test the performance on class imbalanced data, we use the Average Precision, Average Recall, and Average F1-measure as the evaluation metrics.

From the implementation details of the proposed algorithms, it can be seen that the values of parameters such as $\lambda$, $c$, $\rho$, *decay*, $k$ need to be specified. Thus, we first conducted a sensitivity analysis on these parameters, and we found that the algorithms can achieve better results on all the data sets when $\lambda$ and *decay* are set to a fixed value, while the algorithms can also achieve better results when other parameters need to be set to specific values in each data set. Due to page constraints, these related results are not listed here. So the values of $\lambda$ and *decay* will be fixed as 0.000001 and 0.01 in the following experiments, respectively, and other parameters set different values on different data sets. The detailed settings of these parameters are shown in Table 2.

Table 2: Experimental parameter setting for PL-CCML-SFN and PL-CCML-LDD algorithms.

| | | $k$ | $\rho$ | *decay* | $\lambda$ | $c$ |
|---|---|---|---|---|---|---|
| PL-CCML-SFN | FG-NET | 3 | 0.0001 | 0.01 | 1000 | – |
| | Lost | 5 | 0.0001 | 0.01 | 100 | – |
| | MSRCv2 | 9 | 0.0001 | 0.01 | 0.0001 | – |
| | BirdSong | 15 | 0.0001 | 0.01 | 1 | – |
| PL-CCML-LDD | FG-NET | 3 | 0.0001 | 0.01 | 1000 | 1 |
| | Lost | 5 | 0.0001 | 0.01 | 100 | 0.1 |
| | MSRCv2 | 9 | 0.0001 | 0.01 | 0.1 | 1000000 |
| | BirdSong | 15 | 0.0001 | 0.01 | 100000 | 1 |

### 4.2. Experimental results and analysis

In this section, the PL-CCML-SFN and PL-CCML-LDD algorithms are compared with PL-CCML algorithm when they are used as the front-end of PL-kNN algorithm on the four real-world data sets. Table 3 presents the detailed experimental results, where the best result (the larger the better) on each data set is shown in bold face. It can be seen that the algorithms proposed in this paper have greatly improved over the PL-kNN algorithm, and

also outperform the PL-CCML algorithm in almost all evaluation metrics, which is in line with the expected experimental results.

In order to have a deeper understanding, Tables 4 and 5 respectively list the experimental results of each algorithm on the majority class samples and the minority class samples. It can be seen that the proposed algorithms can not only improve the prediction performance on the minority class samples, but also can improve the prediction performance on the majority class samples.

Table 3: Experimental results of each compared algorithm on the whole data set.

| | | FG-NET | Lost | MSRCv2 | BirdSong |
|---|---|---|---|---|---|
| Average Precision (%) | PL-kNN | 2.76±1.36 | 41.26±5.85 | 35.21±4.75 | 59.22±0.87 |
| | PL-CCML | 3.30±1.06 | 40.06±6.32 | 37.14±3.41 | 59.64±1.46 |
| | PL-CCML-SFN | 4.16±1.04 | **46.81±6.32** | **39.14±4.49** | 59.84±1.64 |
| | PL-CCML-LDD | **4.42±1.59** | 45.76±7.09 | 38.50±5.18 | **60.45±1.45** |
| Average Recall (%) | PL-kNN | 3.10±1.06 | 26.93±4.23 | 27.02±2.43 | 52.52±1.63 |
| | PL-CCML | 3.94±1.51 | 36.17±2.63 | 28.39±2.41 | 52.09±1.34 |
| | PL-CCML-SFN | 4.58±1.56 | 36.44±3.58 | 29.31±2.15 | 52.66±2.24 |
| | PL-CCML-LDD | **4.61±1.39** | **38.63±3.44** | **29.98±2.28** | **53.99±1.39** |
| Average F1-measure (%) | PL-kNN | 5.65±2.26 | 28.53±4.58 | 28.27±3.50 | 53.65±1.74 |
| | PL-CCML | 8.17±2.44 | 38.53±4.24 | 30.12±2.84 | 53.54±1.48 |
| | PL-CCML-SFN | 9.00±2.27 | 38.54±4.52 | **30.98±3.01** | 53.73±1.48 |
| | PL-CCML-LDD | **10.20±2.78** | **40.18±5.30** | 30.77±2.35 | **55.13±1.58** |

Table 4: Experimental results of each compared algorithm on the minority class samples of each data set.

| | | FG-NET | Lost | MSRCv2 | BirdSong |
|---|---|---|---|---|---|
| Average Precision (%) | PL-kNN | **2.4** | 14.6 | 12.0 | 52.8 |
| | PL-CCML | 1.6 | 16.7 | 12.0 | 54.3 |
| | PL-CCML-SFN | 2.3 | 19.0 | **14.9** | 54.3 |
| | PL-CCML-LDD | 1.9 | **27.5** | 13.5 | **55.1** |
| Average Recall (%) | PL-kNN | 2.6 | 5.1 | 7.5 | 30.4 |
| | PL-CCML | 2.8 | 5.4 | 8.6 | 30.1 |
| | PL-CCML-SFN | **3.4** | 6.9 | **9.0** | 31.0 |
| | PL-CCML-LDD | 1.8 | **14.1** | 8.6 | **33.6** |
| Average F1-measure (%) | PL-kNN | 1.8 | 7.8 | 7.5 | 37.9 |
| | PL-CCML | 1.4 | 8.0 | 8.4 | 39.0 |
| | PL-CCML-SFN | **2.3** | 8.6 | **9.3** | 39.0 |
| | PL-CCML-LDD | 1.7 | **17.7** | 8.6 | **40.5** |

Table 5: Experimental results of each compared algorithm on the majority class samples of each data set.

| | | FG-NET | Lost | MSRCv2 | BirdSong |
|---|---|---|---|---|---|
| Average Precision (%) | PL-kNN | 4.8 | 39.8 | 39.2 | 65.6 |
| | PL-CCML | 7.4 | 44.6 | 40.7 | 65.6 |
| | PL-CCML-SFN | **9.8** | **45.2** | 40.7 | 65.6 |
| | PL-CCML-LDD | **9.8** | **45.2** | **40.8** | **66.7** |
| Average Recall (%) | PL-kNN | 4.4 | 45.2 | 54.3 | 69.8 |
| | PL-CCML | 6.5 | 56.9 | 54.9 | 70.0 |
| | PL-CCML-SFN | **8.2** | **57.6** | 55.3 | 70.2 |
| | PL-CCML-LDD | 7.9 | **57.6** | **55.5** | **70.3** |
| Average F1-measure (%) | PL-kNN | 4.0 | 40.0 | 43.2 | 65.3 |
| | PL-CCML | 6.3 | 49.5 | 44.6 | 65.3 |
| | PL-CCML-SFN | **8.1** | **50.7** | 44.8 | 65.3 |
| | PL-CCML-LDD | 7.8 | **50.7** | **44.9** | **65.9** |

## 5. Conclusion

In this paper, we propose two partial label metric learning algorithm termed PL-CCML-SFN and PL-CCML-LDD for class imbalanced data. Experimental results on real-world data sets show that the accuracy of the proposed algorithms outperforms existing partial label metric learning algorithms. In future work, we will explore the use of deep learning techniques in the model and try to evaluate the performance of the algorithms by conducting experiments on wider class imbalanced data sets.

## Acknowledgments

This research was funded by the National Natural Science Foundation of China (62102062) and the Natural Science Foundation of Liaoning Province (2020-MS-134, 2020-MZLH-29).

## References

Christos K Aridas, Stamatis Karlos, Vasileios G Kanas, Nikos Fazakis, and Sotiris B Kotsiantis. Uncertainty based under-sampling for learning naive bayes classifiers under imbalanced data sets. *IEEE Access*, 8:2122–2133, 2019.

Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. Learning distance functions using equivalence relations. In *Proceedings of the 20th International Conference on Machine Learning*, pages 11–18, 2003.

André L Brun, Alceu S Britto Jr, Luiz S Oliveira, Fabricio Enembreck, and Robert Sabourin. A framework for dynamic classifier selection oriented by the classification problem difficulty. *Pattern Recognition*, 76:175–190, 2018.

Yichen Chen, Vishal M Patel, Rama Chellappa, and P Jonathon Phillips. Ambiguously labeled learning using dictionaries. *IEEE Transactions on Information Forensics and Security*, 9(12):2076–2088, 2014.

Etienne Côme, Latifa Oukhellou, Thierry Denœux, and Patrice Aknin. Mixture model estimation with soft labels. In *Soft Methods for Handling Variability and Imprecision*, pages 165–174. 2008.

Etienne Côme, Latifa Oukhellou, Thierry Denoeux, and Patrice Aknin. Learning from partially supervised data using mixture models and belief functions. *Pattern Recognition*, 42(3):334–348, 2009.

Timothee Cour, Ben Sapp, and Ben Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 12(May):1501–1536, 2011.

Lei Feng and Bo An. Partial label learning by semantic difference maximization. International Joint Conference on Artificial Intelligence, 2019.

K Fukunada. Introduction to statistical pattern recognition. *Academic Press Inc., San Diego, CA, USA*, 1990.

Jacob Goldberger, Sam Roweis, Geoffrey Hinton, and Russ Salakhutdinov. Neighbourhood components analysis. *Advances in Neural Information Processing Systems*, 17, 2004.

Chen Gong, Tongliang Liu, Yuanyan Tang, Jian Yang, Jie Yang, and Dacheng Tao. A regularization approach for instance-based superset label learning. *IEEE Transactions on Cybernetics*, 48(3):967–978, 2017.

Xiaofei He and Partha Niyogi. Locality preserving projections. *Advances in Neural Information Processing Systems*, 16(16):153–160, 2004.

Steven CH Hoi, Wei Liu, and Shih-Fu Chang. Semi-supervised distance metric learning for collaborative image retrieval and clustering. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 6(3):1–26, 2010.

Eyke Hüllermeier and Jürgen Beringer. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):419–439, 2006.

Rong Jin and Zoubin Ghahramani. Learning with multiple labels. In *Advances in Neural Information Processing Systems*, 2002.

Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8):3573–3587, 2017.

Bartosz Krawczyk, Mikel Galar, Michał Woźniak, Humberto Bustince, and Francisco Herrera. Dynamic ensemble selection for multi-class classification with one-class classifiers. *Pattern Recognition*, 83:34–51, 2018.

Liping Liu and Thomas G Dietterich. A conditional multinomial mixture model for superset label learning. In *Advances in Neural Information Processing Systems*, pages 548–556, 2012.

Liping Liu and Thomas G Dietterich. Learnability of the superset label learning problem. In *International Conference on Machine Learning*, pages 1629–1637, 2014.

Jie Luo and Francesco Orabona. Learning from candidate labeling sets. Technical report, MIT Press, 2010.

Ruisen Luo, Qian Feng, Chen Wang, Xiaomei Yang, Haiyan Tu, Qin Yu, Shaomin Fei, and Xiaofeng Gong. Feature learning with a divergence-encouraging autoencoder for imbalanced data classification. *IEEE Access*, 6:70197–70211, 2018.

Gengyu Lyu, Songhe Feng, Tao Wang, and Congyan Lang. A self-paced regularization framework for partial-label learning. *IEEE Transactions on Cybernetics*, 2020.

Caizhi Tang and Minling Zhang. Confidence-rated discriminative partial label learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Dengbao Wang, Li Li, and Minling Zhang. Adaptive graph guided disambiguation for partial label learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 83–91, 2019.

Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(2), 2009.

Pengtao Xie, Wei Wu, Yichen Zhu, and Eric Xing. Orthogonality-promoting distance metric learning: Convex relaxation and theoretical analysis. In *International Conference on Machine Learning*, pages 5403–5412, 2018.

Shuang Xu, Min Yang, Yu Zhou, Ruirui Zheng, Wenpeng Liu, and Jianjun He. Partial label metric learning by collapsing classes. *International Journal of Machine Learning and Cybernetics*, 11(11):2453–2460, 2020.

Yao Yao, Jiehui Deng, Xiuhua Chen, Chen Gong, Jianxin Wu, and Jian Yang. Deep discriminative cnn with temporal ensembling for ambiguously-labeled image classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12669–12676, 2020a.

Yao Yao, Chen Gong, Jiehui Deng, and Jian Yang. Network cooperation with progressive disambiguation for partial label learning. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2020b.

Zhifei Ye, Yimin Wen, and Baoliang Lu. A survey of imbalanced pattern classification problems. *CAAI Transactions on Intelligent Systems*, 4(2):148–156, 2009.

Zinan Zeng, Shijie Xiao, Kui Jia, Tsung-Han Chan, Shenghua Gao, Dong Xu, and Yi Ma. Learning by associating ambiguously labeled images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 708–715, 2013.

Minling Zhang, Binbin Zhou, and Xuying Liu. Partial label learning via feature-aware disambiguation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1335–1344, 2016.

Yu Zhou and Hong Gu. Geometric mean metric learning for partial label data. *Neurocomputing*, 275:394–402, 2018.

Yu Zhou, Jianjun He, and Hong Gu. Partial label learning via gaussian processes. *IEEE Transactions on Cybernetics*, 47(12):4443–4450, 2017.