Metric Learning from Probabilistic Labels

Mengdi Huai
State University of New York at Buffalo, NY, USA
mengdihu@buffalo.edu

Chenglin Miao
State University of New York at Buffalo, NY, USA
cmiao@buffalo.edu

Yaliang Li
Tencent Medical AI Lab
CA, USA
yaliangli@tencent.com

Qiuling Suo
State University of New York at Buffalo, NY, USA
qiulings@buffalo.edu

Lu Su
State University of New York at Buffalo, NY, USA
lusu@buffalo.edu

Aidong Zhang
State University of New York at Buffalo, NY, USA
azhang@buffalo.edu

ABSTRACT

Metric learning aims to learn a good distance metric that can capture the relationships among instances, and its importance has long been recognized in many fields. In the traditional settings of metric learning, an implicit assumption is that the associated labels of the instances are deterministic. However, in many real-world applications, the associated labels come naturally with probabilities instead of deterministic values. Thus, the existing metric learning methods cannot work well in these applications. To tackle this challenge, in this paper, we study how to effectively learn the distance metric from datasets that contain probabilistic information, and then propose two novel metric learning mechanisms for two types of probabilistic labels, i.e., the instance-wise probabilistic label and the group-wise probabilistic label. Compared with the existing metric learning methods, our proposed mechanisms are capable of learning distance metrics directly from the probabilistic labels with high accuracy. We also theoretically analyze the two proposed mechanisms and provide theoretical bounds on the sample complexity for both of them. Additionally, extensive experiments based on real-world datasets are conducted to verify the desirable properties of the proposed mechanisms.

CCS CONCEPTS

• **Information systems** → **Data mining**; • **Computing methodologies** → *Machine learning*;

KEYWORDS

Metric learning; distance measure; probabilistic labels

ACM Reference Format:
Mengdi Huai, Chenglin Miao, Yaliang Li, Qiuling Suo, Lu Su, and Aidong Zhang. 2018. Metric Learning from Probabilistic Labels. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3219976

KDD '18, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3219976

1 INTRODUCTION

Calculating distances among data instances is an important basis for many data mining and machine learning algorithms, and the performance of such algorithms highly depends on the choice of distance metric. Although some simple metrics, such as Euclidean distance, can be used to measure the similarity between instances, they cannot capture the statistical regularities in the data, which largely degrades the performance of the algorithms [27]. To address this challenge, the task of metric learning has been widely studied [1, 2, 5, 8, 13, 14, 20, 21, 25, 26, 28, 30, 31], and the importance of metric learning has long been recognized in many fields. The distance (or similarity) metric produced by metric learning techniques is capable of capturing the important relationships among instances.

In the traditional settings of metric learning, each instance used for training is usually associated with an attribute set denoting its *features* and a target attribute called *label*. An implicit assumption in these settings is that the associated labels of the instances are deterministic (see Figure 1a). However, in many real-world applications, the associated labels in a training dataset come naturally with *probabilities* due to various reasons, such as uncertainty [16] or privacy issues [9], and the *probabilistic labels* usually exist in the following two forms:

Instance-wise probabilistic label. As shown in Figure 1b, instead of being associated with a deterministic label (e.g., positive or negative in the binary case), *each instance* in the training dataset comes with a probabilistic label, which represents the probability that the instance has a particular deterministic label. This type of probabilistic label is very common in many real-world applications. For example, in crowdsourcing applications [17, 24, 32], a data requester usually outsources the labeling task for each instance to a large crowd of labelers in order to obtain reliable labels at a low cost, then the proportion of the labelers who give a particular label can be treated as the probability that the instance has this particular label. In the medical diagnosis applications, since a physician routinely encounters diagnostic uncertainty in practice, she/he may report a probability that a patient suffers from a disease after the medical examination [16].

Group-wise probabilistic label. Figure 1c shows the dataset associated with group-wise probabilistic labels. The training dataset here consists of several groups of instances, and *each group* is associated with a probabilistic label, which represents the proportion of the instances in this group that have a particular deterministic label [9]. In this case, the label information for each instance is unknown,

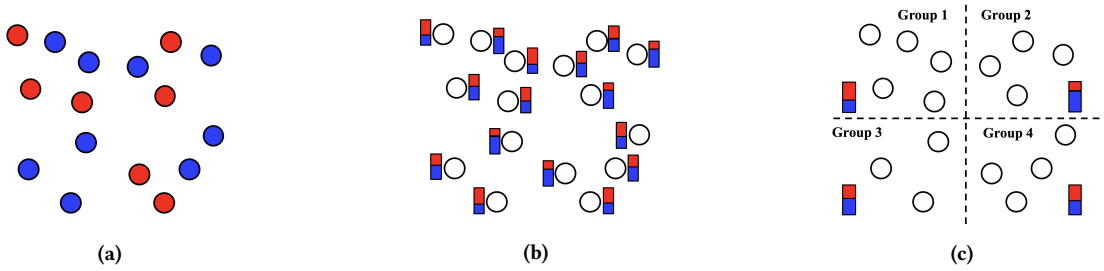


Figure 1: The datasets with different label information. (a) Each instance is associated with a deterministic label. (b) Each instance is associated with a probabilistic label. (c) Each group is associated with a probabilistic label.

and we can only learn the distance metric from the group-wise probabilities. This type of probabilistic label has many interesting applications in real world. For example, in the application of analyzing the outcomes of political elections [18, 22], it is important for the observers of politics to analyze the connections among different voters based on the variables such as age, income or education. However, the voting result of each voter is usually confidential and cannot be revealed to the public. What the observers can know is the proportion of the votes per party in each electoral district. Another example comes from the application of epidemic analysis, where it is usually difficult to know whether a resident living in a district suffers from a disease, but the proportion of the residents who suffer from the disease in this district can be easily obtained.

Despite the prevalence of the probabilistic labels in real-world applications, the existing work on metric learning cannot well address the learning problems with such probabilistic information. In order to deal with the instance-wise probabilistic label, the existing metric learning methods need to transform the associated probability value of each instance to a deterministic label based on a predefined threshold. However, since the probabilistic dataset is usually more informative, many useful information may be lost during the transformation process [16]. Additionally, it is usually difficult to determine an accurate threshold in practice [17]. As for the group-wise probabilistic label, to the best of our knowledge, there is no existing work which can deal with such probabilistic information. Note that the basic assumption behind metric learning is that the distance between similar instances should be smaller than the distance between dissimilar instances [26, 30]. To achieve the goal, the metric is usually trained under sets of pairwise or triplet constraints. However, based on the group-wise probabilistic labels, the pairwise or triplet constraints can not be constructed, which makes the learning task more challenging.

To tackle the above challenges, in this paper, we propose two novel and effective metric learning mechanisms to deal with the aforementioned two types of probabilistic labels, respectively. More specifically, we first design a novel instance-level metric learning mechanism (**InML**), based on which the distance metric can be directly learned from the instance-wise probabilities. In this mechanism, we first construct distance constraints based on the relative comparison relationships that are derived through ranking the instance-wise probabilities, and then we formulate the metric learning process as an optimization problem according to the large margin framework with the hinge loss. To learn a distance metric directly from the group-wise probabilities, we propose a novel group-level metric learning mechanism (**GrML**). In this mechanism, the proportion of the similar instance pairs in each group is first calculated based on the associated group-wise probability, and then we model the latent unknown pairwise similarity labels with the calculated proportions of the similar instance pairs in a maximum likelihood estimation framework, based on which the distance metric can be derived.

In summary, the main contributions of this paper are:

- In order to address the metric learning problems with the instance-wise probabilistic labels, we propose a novel instance-level metric learning mechanism (InML) which can fully utilize the probabilistic information so that the learned metric can be more accurate.
- For the scenarios where the training datasets are associated with group-wise probabilistic labels, we design a group-level metric learning mechanism (GrML) based on which the distance metric can be directly learned from the group-wise probabilities with high accuracy.
- Both theoretical analysis and extensive experiments on real-world datasets demonstrate the advantages of the proposed mechanisms.

2 PROBLEM SETTING

Suppose there is a set of instances $\mathcal{X} = \{x_i\}_{i=1}^N$, where $x_i \in \mathbb{R}^u$ is a u-dimensional feature vector. The goal of metric learning is to learn a distance metric

$$d(x_i, x_j) = (x_i - x_j)^T W (x_i - x_j) = (x_i - x_j)^T M^T M (x_i - x_j), \quad (1)$$

which can effectively measure the similarity between any two inputs (instances) x_i and x_j. Here, $d(x_i, x_j)$ is parameterized by a positive semidefinite matrix W, which can be decomposed as $W = M^T M$, and W (or $M \in \mathbb{R}^{u \times u}$) is the parameter that needs to be learned from the training dataset. If the deterministic label of each instance, which belongs to one of two possible categories (e.g., positive or negative), is provided, the metric can be easily learned in a supervised manner according to the existing metric learning methods. However, in many real-world applications, the associated labels in the training datasets come with probabilities instead of deterministic values. Here we consider the following two cases:

- If the case is for the *instance-wise probabilistic label*, we assume that each instance $x_i \in \mathcal{X}$ is associated with a probabilistic label $c_i \in [0, 1]$, which represents the probability that x_i belongs to the positive category.
- If the case is for the *group-wise probabilistic label*, we assume that the dataset $\mathcal{X}$ consists of K disjoint subsets (groups), i.e., $\mathcal{X} = \cup\{\mathcal{X}_k\}_{k=1}^K$, and each group $\mathcal{X}_k$ is associated with a probability $\pi_k \in [0, 1]$, which represents the proportion of instances that belong to the positive category in this group.

Our goal in this paper is to learn the optimal distance function $d(x_i, x_j)$ which is parameterized by $W = M^T M$ from the probabilistic labels provided in the above two cases, respectively.

3 METRIC LEARNING FROM INSTANCE-WISE PROBABILISTIC LABELS

In this section, we present the proposed instance-level metric learning mechanism (i.e., InML) which can learn the distance metrics from the instance-wise probabilistic labels (i.e., $C = \{c_i\}_{i=1}^N$). The details of the proposed mechanism are described in Section 3.1, and the theoretical analysis is provided in Section 3.2.

3.1 Learning Framework

In the case where each instance is associated with a probabilistic label (i.e., c_i) instead of a deterministic label, a straightforward way to learn the distance metric is to assign each instance a deterministic label based on a predefined threshold over the probabilities and then conduct the existing metric learning methods. However, since the probabilistic dataset is usually more informative, some useful information may be lost during the transformation from probabilities to deterministic labels, and this will degrade the accuracy of the learned results. Additionally, it is usually difficult to determine an accurate threshold in reality. To address this challenge, we propose to learn the distance metric directly from the instance set $\mathcal{X} = \{x_i\}_{i=1}^N$ and its associated probabilistic labels (i.e., $\{c_i\}_{i=1}^N$). To achieve the goal, we first construct the distance constraints based on the relative comparison relationships that are derived through ranking probabilities, and then we design an optimization function to enforce the relative comparison of the constructed constraints.

Distance Constraint Construction. Without loss of generality, in this paper we assume that $c_1 > c_2 > \cdots > c_{N-1} > c_N$. We first construct a partially ordered triplet set

$$\mathcal{R} = \{(x_i, x_j, x_k), 1 \le i \ne j \ne k \le N, j < k\}. \tag{2}$$

For each triplet (x_i, x_j, x_k), it is obvious that $c_j > c_k$ due to $j < k$. Considering the relationships among c_i, c_j and c_k, we can divide the triplet set $\mathcal{R}$ into the following four subsets ($\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3 \cup \mathcal{R}_4$):

- $\mathcal{R}_1 = \{(x_i, x_j, x_k), 1 \le i < j < k \le N\}$. For each triplet (x_i, x_j, x_k) in $\mathcal{R}_1$, since $i < j < k$, the associated probabilities satisfy $c_i > c_j > c_k$. That is to say x_i is more similar to x_j than to x_k. Then we can know the distance between x_i and x_j should not be larger than that between x_i and x_k (i.e., $d(x_i, x_j) \le d(x_i, x_k)$).
- $\mathcal{R}_2 = \{(x_i, x_j, x_k), 1 \le j < k < i \le N\}$. In this subset, the associated probabilities for each triplet (x_i, x_j, x_k) satisfy $c_j > c_k > c_i$. Then the distance between x_i and x_k should not be larger than that between x_i and x_j (i.e., $d(x_i, x_k) \le d(x_i, x_j)$).
- $\mathcal{R}_3 = \{(x_i, x_j, x_k), 1 \le j < i < k \le N, c_j > c_i > (c_j + c_k)/2\}$. For each triplet (x_i, x_j, x_k) in this subset, the distance between x_i and x_j should not be larger than that between x_i and x_k (i.e., $d(x_i, x_j) \le d(x_i, x_k)$).
- $\mathcal{R}_4 = \{(x_i, x_j, x_k), 1 \le j < i < k \le N, (c_j + c_k)/2 > c_i > c_k\}$. For each triplet (x_i, x_j, x_k) in this subset, the distance between x_i and x_k should not be larger than that between x_i and x_j (i.e., $d(x_i, x_k) \le d(x_i, x_j)$).

As we can see, for each triplet in the above subsets, there is a distance constraint that is constructed according to the relative comparison relationships among the associated probabilities. When we conduct metric learning from the instance set $\mathcal{X} = \{x_i\}_{i=1}^N$, these distance constraints should be satisfied. Next, we discuss how to learn the distance metric based on these constructed constraints.

Optimization Formulation. In our proposed mechanism, we formulate the metric learning process as an optimization problem based on the large margin framework with the hinge loss. Suppose $\mathcal{R}_1' = \mathcal{R}_1 \cup \mathcal{R}_3$ and $\mathcal{R}_2' = \mathcal{R}_2 \cup \mathcal{R}_4$. For each triplet $(x_i, x_j, x_k) \in \mathcal{R}$, we first reformulate the above constructed distance constraints as

$$\begin{cases} d(x_i, x_j) + g \le d(x_i, x_k) & \text{if } (x_i, x_j, x_k) \in \mathcal{R}_1' \\ d(x_i, x_k) + g \le d(x_i, x_j) & \text{if } (x_i, x_j, x_k) \in \mathcal{R}_2', \end{cases} \tag{3}$$

where $d(x_i, x_j) = (x_i - x_j)^T W (x_i - x_j)$, and g is a parameter that regularizes the gap (or margin) between $d(x_i, x_j)$ and $d(x_i, x_k)$. In this paper, we choose a unit margin. To monitor the inequality constraints in Eqn. (3), we then propose to minimize the following hinge loss function

$$\begin{aligned} \min_W \quad & \sum_{(x_i, x_j, x_k) \in \mathcal{R}_1'} \max\{0, d(x_i, x_j) - d(x_i, x_k) + g\} \\ & + \sum_{(x_i, x_j, x_k) \in \mathcal{R}_2'} \max\{0, d(x_i, x_k) - d(x_i, x_j) + g\} + \alpha \|W\|_*, \end{aligned} \tag{4}$$

where $\|W\|_*$ represents the nuclear norm to promote low-rankness, and α is the regularization parameter. The operator $\max\{0, \cdot\}$ in Eqn. (4) denotes the hinge loss function, which penalizes the triplets that violate the inequality constrains in Eqn. (3). Note that if the inequality does hold, then its hinge loss has a negative argument and makes no contribution to the overall loss function. Since there exist triplets violating the above constraints, we relax these constrains by incorporating nonnegative slack variables to monitor these margin violations. Then we formulate the metric learning process as the following optimization problem.

$$\begin{aligned} \min_{W, \{\xi_{ijk}^1\}, \{\xi_{ijk}^2\}} \quad & \sum_{(x_i, x_j, x_k) \in \mathcal{R}_1'} \frac{1}{|\mathcal{R}_1'|} \xi_{ijk}^1 + \sum_{(x_i, x_j, x_k) \in \mathcal{R}_2'} \frac{1}{|\mathcal{R}_2'|} \xi_{ijk}^2 \\ & + \alpha \|W\|_* \\ \text{s.t.} \quad & \forall (x_i, x_j, x_k) \in \mathcal{R}_1' : \max\{0, d(x_i, x_j) - d(x_i, x_k) + g\} \le \xi_{ijk}^1, \\ & \forall (x_i, x_j, x_k) \in \mathcal{R}_2' : \max\{0, d(x_i, x_k) - d(x_i, x_j) + g\} \le \xi_{ijk}^2, \\ & \forall (x_i, x_j, x_k) \in \mathcal{R}_1' : \xi_{ijk}^1 \ge 0, \\ & \forall (x_i, x_j, x_k) \in \mathcal{R}_2' : \xi_{ijk}^2 \ge 0, \end{aligned} \tag{5}$$

where ξ_{ijk}^1's, ξ_{ijk}^2's are the introduced slack variables that allow the large margin inequality in Eqn. (3) to violate the margin. Then,

we solve the optimization problem via the sub-gradient descent method. Finally, we can derive the distance function $d(x_i, x_j) = (x_i - x_j)^T W(x_i - x_j)$.

Discussion. In our proposed mechanism, we construct the distance constraints by comparing $d(x_i, x_j)$ with $d(x_i, x_k)$ for each triplet. In fact, the constraints can also be derived from the comparison relationship between $d(x_k, x_j)$ and $d(x_k, x_i)$. Additionally, when there are some instances whose associated probabilities are close (or equal) to each other, we can incorporate the binning method into our proposed mechanism and divide the instance sequence (i.e., $x_1, x_2, ..., x_N$) into disjoint bins. Then only the constrains for the triplets whose instances are in different bins are enforced. In this way, the constraint complexity can be reduced and the proposed mechanism will be robust to noise inherent in the probabilities.

3.2 Theoretical Analysis

In this section, we theoretically analyze the error bound that is generated by the proposed mechanism (InML). Suppose $R(W)$ is an unbiased estimator of the true risk and W^* is the true risk minimizer which is estimated from $R(W)$ (i.e., $W^* = \arg\min_W R(W)$). Let $\hat{W}$ be the distance metric learned based on the optimization problem described in Eqn. (5). Then we have the following theorem.

Theorem 3.1. *Let N denote the number of the instances in the training dataset (i.e., $|\mathcal{X}|$) and r denote the rank of $\hat{W}$. Assume that $\|\mathcal{X}\mathcal{X}^T\| = O(N/u)$ and $\max_i(x_i^T \hat{W} x_i) = O(r \log N)$. Then, with the probability at least $1-\delta$, where $\delta \in (0,1)$, we have the following error bound:*

$$R(\hat{W}) - R(W^*) = O(\sqrt{\frac{ru(\log u + \log^2 N \log(2/\delta))}{N^3 - 3N^2 + 2N}}), \tag{6}$$

where u is the dimension of the feature vector.

Theorem 3.1 can be proved based on Rademacher analysis [19]. Due to the space limit, we here omit the proof of the above theorem. According to this theorem, it is easy to verify that the error bound generated by the proposed mechanism is $O(\sqrt{\log^2 N/(N^3 - 3N^2 + 2N)})$, where $N > 3$. Since $\log N < \sqrt[3]{N}$, we can get that the above generated error bound is smaller than the existing best-known bound $O(\sqrt{1/N})$ that is derived from the datasets with binary class labels [3]. That is to say our mechanism can learn a good metric with a smaller number of instances than the existing metric learning methods.

4 METRIC LEARNING FROM GROUP-WISE PROBABILISTIC LABELS

As described in Section 2, in the case where the probabilistic label is group-wise, we only have access to the group-wise probabilities (i.e., $\{\pi_k\}_{k=1}^K$) instead of the instance-wise label information, which makes it more difficult to learn an accurate metric. To address this challenge, we propose a novel and effective learning mechanism (i.e., GrML) which can learn the distance metric directly from the group-wise probabilities. We first formulate the learning framework as an optimization problem and discuss how to effectively solve this problem in Section 4.1 and Section 4.2, respectively. Then we conduct theoretical analysis for the proposed mechanism in Section 4.3.

4.1 Learning Framework

Suppose the instance set $\mathcal{X}$ consists of K disjoint groups, i.e., $\mathcal{X} = \cup\{\mathcal{X}_k\}_{k=1}^K$, where $\mathcal{X}_k = \{x_i^k\}_{i=1}^{|\mathcal{X}_k|}$ is the k-th group and x_i^k represents the i-th instance in group $\mathcal{X}_k$. For each instance pair (x_i^k, x_j^k) in group $\mathcal{X}_k$, we assume that there is a label $y_{ij}^k \in \{1, -1\}$ that denotes whether the two instances are similar (i.e., have the same class label) or not. If x_i^k and x_j^k are similar, y_{ij}^k is equal to 1, otherwise it is equal to -1. Here we associate each group $\mathcal{X}_k$ with another probability $\hat{\pi}_k$, which represents the proportion of the instance pairs whose similarity labels (i.e., y_{ij}^k) are equal to 1 in group $\mathcal{X}_k$. Then $\hat{\pi}_k$ can be derived as

$$\hat{\pi}_k = 1 - \frac{2|\mathcal{X}_k|\pi_k(1-\pi_k)}{|\mathcal{X}_k|-1}. \tag{7}$$

Since π_k is a known probability value for group $\mathcal{X}_k$. $\hat{\pi}_k$ can be treated as a constant during the training process of the group-wise metric learning model.

Our goal in this section is to learn the distance metric $d(x_i, x_j) = (x_i - x_j)^T M^T M(x_i - x_j)$, which is parameterized by M. Here, we seek an alternative approach by decomposing matrix W as $M^T M$ [31]. To achieve the goal, we adopt maximum likelihood estimation here. That is to say, we need to find a matrix M that can maximize the likelihood of the instance pairs in each group $\mathcal{X}_k$. We model the probability for each instance pair (x_i^k, x_j^k) and the corresponding unknown label y_{ij}^k as

$$\Pr(y_{ij}^k | x_i^k, x_j^k; M, b) = \frac{1}{1 + \exp(-y_{ij}^k(d(x_i, x_j) - b))}, \tag{8}$$

where $y_{ij}^k \in \{-1, 1\}$ and b is the bias, which also works as a threshold. The two instances x_i^k and x_j^k are treated as similar (i.e., $y_{ij}^k = 1$) only when $d(x_i^k, x_j^k)$ is greater than or equal to b, otherwise they are treated as dissimilar (i.e., $y_{ij} = -1$). In this paper, we set b as 1. Then we can formulate the following optimization problem

$$\min_{I,M} \sum_{k=1}^K \sum_{i<j} \frac{2\log(1 + \exp(-y_{ij}^k(d(x_i^k, x_j^k) - b)))}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)} + \frac{\|M\|_F^2}{2},$$
$$\text{s.t.} \quad \sum_{i<j} \frac{y_{ij}^k}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)} + \frac{1}{2} = \hat{\pi}_k, \quad k = 1, 2, ..., K \tag{9}$$

where $I = \{y_{ij}^k | i < j, k = 1, ..., K\}$. The objective function in this optimization problem contains two terms. The first term is derived from the negative log likelihood of the instance pairs and the second term is the Frobenius-norm regularization. Since the elements (i.e., y_{ij}^k's) in set I are not known *a priori*, we also need to estimate them during the optimization process, and the estimated y_{ij}^k's should satisfy the constraint in Eqn. (9). However, the categorical property of y_{ij}^k makes it difficult to solve this optimization problem.

In order to address the above challenge, we relax each y_{ij}^k to a continuous probability-like variable $p_{ij}^k \in [0, 1]$. This idea is inspired from the Deterministic Annealing (DA) technique [4] and the variable p_{ij}^k can be interpreted as probability that y_{ij}^k is equal to 1. Obviously, the probability that $y_{ij}^k = -1$ is $1 - p_{ij}^k$. Then the

optimization problem in Eqn. (9) can be rewritten as

$$\begin{aligned}\min_{P,M}\ & \sum_{k=1}^{K}\sum_{i<j}\frac{2p_{ij}^k}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}\log(1+\exp(-(d(x_i^k,x_j^k)-b))) \\ & +\sum_{k=1}^{K}\sum_{i<j}\frac{2(1-p_{ij}^k)}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}\log(1+\exp((d(x_i^k,x_j^k)-b))) \\ & +\frac{1}{2}\|M\|_F^2, \\ \text{s.t.}\quad & \sum_{i<j}\frac{2p_{ij}^k}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}=\hat{\pi}_k,\quad k=1,2,...,K \end{aligned} \tag{10}$$

where $P=\{p_{ij}^k|i<j,k=1,2,...,K\}$. To mitigate local minima, an entropy term [4] for the distributions defined by p_{ij}^k is also added to the above objective function. Finally, we formulate the metric learning process as the following optimization problem

$$\begin{aligned}\min_{P,M}\ \mathcal{L}(P,M) = & \sum_{k=1}^{K}\sum_{i<j}\frac{2p_{ij}^k}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}\log(1+\exp(-(d(x_i^k,x_j^k)-b))) \\ & +\sum_{k=1}^{K}\sum_{i<j}\frac{2(1-p_{ij}^k)}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}\log(1+\exp((d(x_i^k,x_j^k)-b))) \\ & +\sum_{k=1}^{K}\frac{2T}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}\sum_{i<j}(p_{ij}^k\log p_{ij}^k+(1-p_{ij}^k)\log(1-p_{ij}^k)) \\ & +\frac{1}{2}\|M\|_F^2, \\ \text{s.t.}\quad & \sum_{i<j}\frac{2p_{ij}^k}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}=\hat{\pi}_k,\quad k=1,2,...,K \end{aligned} \tag{11}$$

where T is a penalty parameter.

4.2 Optimization

In this section, we discuss how to solve the optimization problem described in Eqn. (11). The solution we adopted here is a two step iterative procedure.

Step 1: We first fix P, which is estimated in the previous iteration. If it is the first iteration, the elements in P are randomly initialized. Then we solve the following optimization problem

$$\begin{aligned}\min_{M}\ \mathcal{L}_1(M) = & \sum_{k=1}^{K}\sum_{i<j}\frac{2p_{ij}^k}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}\log(1+\exp(-(d(x_i^k,x_j^k)-b))) \\ & +\sum_{k=1}^{K}\sum_{i<j}\frac{2(1-p_{ij}^k)\log(1+\exp((d(x_i^k,x_j^k)-b)))}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}+\frac{\|M\|_F^2}{2}. \end{aligned} \tag{12}$$

Here we adopt gradient descent method to update M, and the gradient is calculated as

$$\begin{aligned}\frac{\partial\mathcal{L}_1}{\partial M} = & \sum_{k=1}^{K}\sum_{i<j}\frac{2p_{ij}^k}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}\frac{2(-M)(x_i^k-x_j^k)^T(x_i^k-x_j^k)}{1+\exp(d(x_i^k,x_j^k)-b)} \\ & +\sum_{k=1}^{K}\sum_{i<j}\frac{2(1-p_{ij}^k)}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}\frac{2M(x_i^k-x_j^k)^T(x_i^k-x_j^k)}{1+\exp(-(d(x_i^k,x_j^k)-b))}+M, \end{aligned} \tag{13}$$

where $d(x_i^k,x_j^k)=(x_i^k-x_j^k)^TM^TM(x_i^k-x_j^k)$.

Step 2: In this step, we fix M that is estimated in step 1, and then update P. Through introducing the Lagrange multipliers $\{\lambda_k\}_{k=1}^K$, we get the Lagrange form of the optimization problem for P:

$$\mathcal{L}_2(P)=\mathcal{L}(P,M)-\sum_{k=1}^{K}\lambda_k(\sum_{i<j}\frac{2p_{ij}^k}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}-\hat{\pi}_k). \tag{14}$$

Let the partial derivative of $\mathcal{L}_2(P)$ with respect to p_{ij}^k be zero, and we can get

$$p_{ij}^k=\frac{1}{1+\exp(\frac{1}{T}\log\frac{1+\exp(-(d(x_i^k,x_j^k)-b))}{1+\exp((d(x_i^k,x_j^k)-b))}-\frac{\lambda_k}{T})}. \tag{15}$$

Combining Eqn. (15) with the constraint in Eqn. (11), we get

$$\sum_{i<j}\frac{2}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)(1+\exp(\frac{1}{T}\log\frac{1+\exp(-(d(x_i^k,x_j^k)-b))}{1+\exp((d(x_i^k,x_j^k)-b))}-\frac{\lambda_k}{T}))}=\hat{\pi}_k, \tag{16}$$

where the the Lagrange multiplier λ_k can be calculated by solving the root finding problem. Finally, the calculated λ_k is plugged into Eqn. (15) such that p_{ij}^k can be updated.

The above two steps will be iteratively conducted until the convergence criterion is satisfied. In this paper, we calculate the KL-divergence of P in two consecutive iterations and set a threshold (e.g., 10^{-6}) of the KL-divergence as the convergence criterion [4]. The optimization procedure is summarized in Algorithm 1.

Algorithm 1 Metric learning from group-wise probabilities

Input: Instance groups $\{\mathcal{X}_k\}_{k=1}^K$ and group-wise probabilities $\{\pi_k\}_{k=1}^K$
Output: The parameter M

```
Calculate π̂_k according to Eqn. (7);
Initialize P = {p_ij^k | i < j, k = 1, 2, ..., K};
repeat
    Update M according to step 1 in Section 4.2;
    Update P according to step 2 in Section 4.2;
until The convergence criterion is satisfied;
return The parameter M.
```

4.3 Theoretical Analysis

As the only available label information, the associated probabilities $\{\pi_k\}_{k=1}^K$ play an important role during the learning process. In this section, we first provide an intuitive understanding about what kinds of π_k's can generate the most informative groups, and then give the sample complexity analysis.

Recall that we introduce $\hat{\pi}_k$, i.e., the proportion of the instance pairs whose similarity labels are equal to 1 in group $\mathcal{X}_k$, as the supervision information during the learning process. For each group $\mathcal{X}_k$, the larger (or less) the value of $\hat{\pi}_k$, the more informative the group. When $\hat{\pi}_k$ equals to 0 or 1, group $\mathcal{X}_k$ is the most informative for metric learning because we can know the similarity labels (i.e.,$\{y_{ij}^k\}$'s) of all the instance pairs in this group. In order to analyze the effect of π_k on $\hat{\pi}_k$, we plot the graph of Eqn. (7) in Figure 2, from which we can see $\hat{\pi}_k$ reaches its minimum values (around

0.5) when $\pi_k = 0.5$, and $\hat{\pi}_k$ approaches its maximum values (i.e., 1) when π_k approximates 0 or 1. This means that if π_k approaches 0 or 1, $\mathcal{X}_k$ will be an informative group and provide more information for the metric learning process. Next, we provide the following theorem to show the upper bound of the size of the training dataset that is used for generating an informative group.

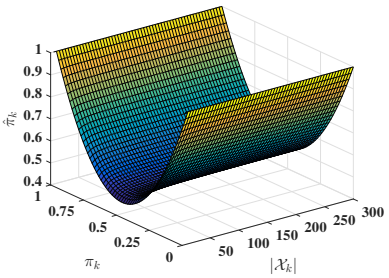


Figure 2: $\hat{\pi}_k$ w.r.t. π_k and $|\mathcal{X}_k|$.

Theorem 4.1. *Suppose that the instance set $\mathcal{X}$ is randomly split into K groups with equal group size m, and $\Gamma \in [0, 1]$ denotes the proportion of the positive instances in $\mathcal{X}$. Let η ($\eta \neq \Gamma$ and $\eta \neq 1 - \Gamma$) be a positive constant that is close to 0. For the k-th group, the probability that $\min\{1 - \pi_k, \pi_k\} \leq \eta$ is $O(e^{-\beta m})$. Thus the number of the instances in set $\mathcal{X}$ is at most $O(me^{\beta m})$, where β is a constant that depends on Γ and η.*

Proof. For random sampling, we assume that the probability that the number of positive instances in $\mathcal{X}_k$ is less than $m\eta$ or more than $m(1-\eta)$ is denoted as $\mathbb{P} = \Pr(\sum_{i=1}^{m} q_i^k \leq m\eta \text{ or } \geq m(1-\eta))$, where $q_i^k \in \{0, 1\}$ is a random variable which indicates whether x_i^k is a positive instance and takes 1 with probability Γ. Based on Bernstein inequality, we have

$$\Pr(\sum_{i=1}^{m} q_i^k \geq m(1-\eta)) \leq \exp(-\frac{3m(1-\eta-\Gamma)^2}{2\Gamma(1-\Gamma) + 2(1-\eta-\Gamma)}) = e^{-\beta_1 m}$$

and

$$\Pr(\sum_{i=1}^{m} q_i^k \leq m\eta) \leq \exp(-\frac{3m(\Gamma-\eta)^2}{2\Gamma(1-\Gamma) + 2(\Gamma-\eta)}) = e^{-\beta_2 m},$$

where $\beta_1 = 3m(1-\eta-\Gamma)^2/(2\Gamma(1-\Gamma) + 2(1-\eta-\Gamma))$ and $\beta_2 = 3m(\Gamma-\eta)^2/(2\Gamma(1-\Gamma)+2(\Gamma-\eta))$. Then, there exists a constant β satisfying $\mathbb{P} = e^{-\beta m}$. Therefore, in order to satisfy $\min\{1-\pi_k, \pi_k\} \leq \eta$, the total number of instances in set $\mathcal{X}$ is $N = m/\mathbb{P}$, i.e. $N = O(me^{\beta m})$. □

The above theorem shows that once the size of set $\mathcal{X}$ (i.e., N) is fixed, the increase of the group size m will lead to the decrease of the probability that $\min\{1-\pi_k, \pi_k\} \leq \eta$. In other words, for a fixed dataset $\mathcal{X}$, when it is divided into subsets with larger group size, the proportion of informative groups becomes smaller, and then the performance of the proposed mechanism is degraded due to the less informative training data.

5 EXPERIMENTS

We conduct experiments on real-world datasets to evaluate the performance of the proposed mechanisms. The experimental setup is first described in Section 5.1. Then we show the experimental results for the instance-level mechanism (InML) and the group-level mechanism (GrML) in Section 5.2 and Section 5.3, respectively.

5.1 Experimental Setup

In this section, we first describe the adopted real-world datasets for the two proposed mechanisms, respectively. Then we introduce the baselines which are compared with the proposed mechanisms.

Datasets for the instance-level mechanism. To verify the advantages of InML, we adopt eight real-world datasets which are grouped into the following three categories:

- *Regression Datasets.* We first adopt three UCI datasets (i.e., Concrete, Housing, and Energy) that are used in the regression task. For each instance in these datasets, we normalize its real-valued output to $[0, 1]$ and take the normalized value as the probability (i.e., c_i) that this instance belongs to the positive category. In order to adapt these datasets to the baseline methods, we also define a threshold based on these probabilities to distinguish the positive and negative categories. For example, in the housing dataset, the real-valued outputs represent the attractiveness of houses to the customers. After normalizing the real-valued outputs, we sort the instances (i.e., houses) by the probability (i.e., c_i) in a descending order. Then we label the top 30% of the instances with positive category (high attractiveness) and the remaining instances with negative category (low attractiveness).
- *Ordinal Classification Datasets.* We also adopt three other real-world datasets[1] (i.e., Cancer, Stock, and Machine) which come with multiple classes and full-order relations among classes. For each dataset, the associated probabilities (i.e., $\{c_i\}_{i=1}^N$) are generated by utilizing the min-max normalization strategy on the ordinal class labels. Additionally, we also define a binary threshold for each dataset according to the meaning of ordinal classes. For example, the Cancer dataset contains six ordinal classes $\{1, 2, 3, 4, 5, 6\}$. After the normalization, the class labels are transformed to the probabilities $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$. Since $\{1, 2\}$ represent benignancy and $\{3, 4, 5, 6\}$ represent the different stages of malignancy, we can set the threshold as 0.3 for the binary label.
- *Crowdsourced Datasets.* Finally, we adopt two crowdsourced datasets, i.e., the movie review dataset and the music genre dataset [17]. The movie review dataset contains 5000 movies and the task of the workers is to judge whether the review of a movie is positive or negative. In the music genre dataset, there are 700 pieces of music and the workers need to judge whether a piece of music is rock (positive) or non-rock (negative). For each instance (a movie or a piece of music), the associated probability (i.e., c_i) is defined as the fraction of the workers who provide positive labels for this instance. Additionally, we set a threshold (0.5 in this paper) over the probabilities to generate the binary label for each instance.

Datasets for the group-level mechanism. As for GrML, we evaluate its performance on three popular datasets: the Ionosphere dataset, the Heart dataset and the Diabetes dataset [29], which are widely used in the settings with group probabilities.

The details of the adopted datasets are described in Table 1.

[1] http://www.gagolewski.com/resources/data/ordinal-regression/

Table 1: The statistics of the adopted datasets.

Dataset	Size	Dimension	Dataset	Size	Dimension
Concrete	1,030	8	Movie	5,000	1,199
Housing	506	13	Music	700	123
Energy	768	8	Ionosphere	351	34
Cancer	194	32	Heart	303	23
Stock	950	9	Diabetes	768	9
Machine	199	6	-	-	-

Baseline Methods. In this paper, we compare the proposed mechanisms with the following state-of-the-art metric learning methods. **GMML** [30] learns the distance metric by formulating an unconstrained smooth and convex optimization problem. **ITML** [5] aims to learn a Mahalanobis distance function, and the authors formulate the problem as minimizing the differential relative entropy between two multivariate Gaussians. **LMNN** [26] learns the distance metric by letting the k-nearest neighbors always belong to the same class while instances from different classes are separated by a large margin. **LowRank** [31] takes into account the sparse feature selection, which is implemented by encoding a low-rank structure to the distance metric learning process. **R2ML** [8] is a local distance metric learning method in which a sparse-inducing matrix norm is introduced to control the rank of the involved mappings. Additionally, **Cosine** and **Euclidean** are also taken as baselines, which adopt cosine similarity and l_2-norm distance to measure the similarity between two instances.

5.2 Experiments for the Instance-level metric learning Mechanism

In this section, we evaluate the performance of InML. The experiments are conducted for 10 times and we report the average results.

Performance comparison. We first compare the accuracy of InML with that of the baseline methods under different training dataset size. Here we consider two cases where the training set size is set as 50 and 100, respectively. For each dataset, we first randomly select half of all instances as the testing set, and then randomly extract the training dataset from the remaining instances. For the case where the training dataset size is set as 100, if the number of the instances used for training is less than 100, we will randomly select some instances from the testing set and add them into the training dataset. The results for the two cases are shown in Table 2. In this paper, the accuracy is calculated based on the instance labels in the testing set and the KNN classifier is adopted to evaluate the performance of the methods [8, 26, 30]. From Table 2, we can see InML performs much better than the baselines in all cases. The reason is that InML can extract more information from instance probabilities, while the baselines can only derive limited knowledge using the class labels.

Convergence. In order to evaluate the convergence of InML, we calculate the objective value in each iteration. Figure 3 reports the evolution of the objective value on the Concrete dataset. The results in this figure show that the objective value gradually converges to 0 with the increase of the iteration number, and this verifies that the convergence of InML can be guaranteed.

Performance on unbalanced datasets. We also evaluate the performance of InML when the datasets are unbalanced, i.e., there

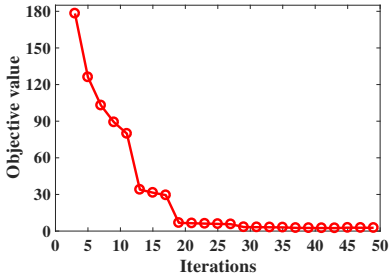


Figure 3: Convergence of InML on the Concrete dataset.

are only a small number of instances that belong to the positive (or negative) category in the dataset. In this experiment, we adopt the regression datasets (i.e., Concrete, Housing and Energy) and set the binary threshold as 10% instead of 30%. That is to say, only 10% of the instances in each dataset are positive. For each dataset, we still randomly select half of all instances as the training dataset and take the remaining instances as the testing set. Then we calculate the G-mean which is used for performance assessment over unbalanced dataset and is defined as the square root of the product of the sensitivity and specificity for each method. The results are shown in Figure 4, from which we can see InML still has the best performance when the datasets are unbalanced. The reason is that the proposed mechanism can extract more information through the ranking-based relative comparisons while the baseline methods can only exploit the binary class labels.

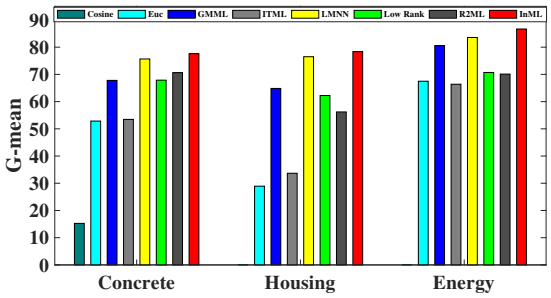


Figure 4: G-mean on unbalanced datasets.

Robustness. In reality, the instance-wise probabilistic labels may be noisy due to various reasons [16]. Thus, it is important to evaluate the robustness of InML when probabilistic labels are perturbed by different levels of noise. In this experiment, we consider three levels of noise: weak noise, moderate noise and strong noise, which are generated from $0.05 * \mathcal{N}(0,1)$, $0.15 * \mathcal{N}(0,1)$ and $0.30 * \mathcal{N}(0,1)$, respectively. Then we add the generated noise to the associated probability for each instance. Please note that the summation would be projected to range $[0,1]$ if it is larger than 1 or less than 0. For each dataset, we randomly select half of all instances as the training dataset and take the remaining instances as the testing set. Table 3 shows the accuracy of all the methods on Concrete, Stock, and Machine datasets. The results in this table show that InML significantly outperforms the baseline methods in all cases. More importantly, compared with the baselines, InML performs more stably when the level of the noise varies, and this

Table 2: The accuracy of the instance-level mechanism (InML) under different training dataset sizes.

		Regression Datasets			Ordinal Datasets			Crowdsourced Datasets	
Training set size	Methods	Concrete	Housing	Energy	Cancer	Stock	Machine	Movie	Music
50	**InML**	**0.8002**	**0.8268**	**0.8969**	**0.6531**	**0.8887**	**0.9233**	**0.6997**	**0.7604**
	Cosine	0.6996	0.7001	0.6905	0.5000	0.5767	0.3200	0.5234	0.6557
	Euc	0.7387	0.7283	0.8468	0.5306	0.8655	0.8717	0.5173	0.7091
	GMML	0.7400	0.7835	0.8831	0.5514	0.8782	0.8733	0.5180	0.7343
	ITML	0.7117	0.7500	0.7719	0.3299	0.6279	0.8132	0.5524	0.7296
	LMNN	0.7713	0.8255	0.8890	0.6474	0.8799	0.8840	0.6767	0.7588
	LowRank	0.6957	0.7746	0.8779	0.5340	0.8739	0.3300	0.5440	0.7091
	R2ML	0.7707	0.7395	0.8368	0.5629	0.8666	0.8900	0.5652	0.6777
100	**InML**	**0.8123**	**0.8596**	**0.9251**	**0.6759**	**0.9139**	**0.9300**	**0.7020**	**0.7868**
	Cosine	0.7031	0.7113	0.7056	0.5510	0.6155	0.3500	0.5352	0.6792
	Euc	0.7542	0.7434	0.8727	0.5680	0.8866	0.8767	0.5290	0.7248
	GMML	0.7471	0.8019	0.9030	0.5710	0.8939	0.8983	0.5358	0.7374
	ITML	0.7335	0.7569	0.8021	0.3544	0.6674	0.8191	0.5673	0.7563
	LMNN	0.7845	0.8425	0.9123	0.6533	0.9007	0.8872	0.6787	0.7781
	LowRank	0.7193	0.7962	0.8983	0.5663	0.8575	0.3500	0.5652	0.7233
	R2ML	0.7774	0.8110	0.8883	0.5714	0.9097	0.8933	0.6020	0.6934

Table 3: The accuracy of the instance-level mechanism (InML) under different noise levels.

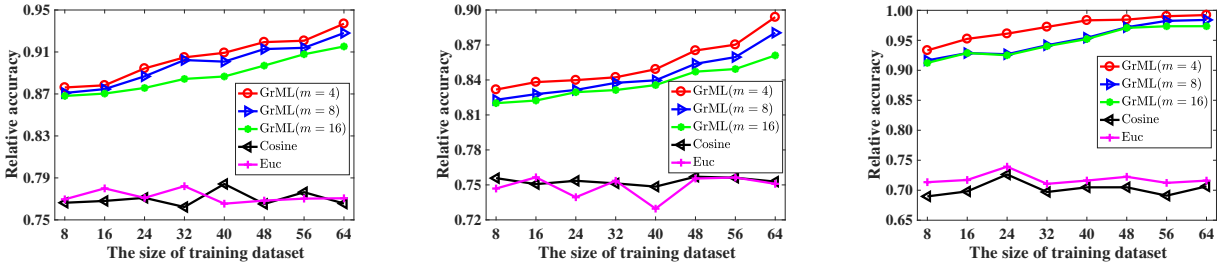
	Weak noise			Moderate noise			Strong noise		
Methods	Concrete	Stock	Machine	Concrete	Stock	Machine	Concrete	Stock	Machine
InML	**0.7926**	**0.8739**	**0.9050**	**0.7917**	**0.8718**	**0.8800**	**0.7915**	**0.8717**	**0.8767**
Cosine	0.6922	0.5756	0.3250	0.6715	0.5745	0.2950	0.6641	0.4636	0.2500
Euc	0.7293	0.7962	0.8400	0.7232	0.7721	0.8250	0.7080	0.7718	0.8017
GMML	0.7345	0.8221	0.8400	0.7322	0.8197	0.8342	0.7025	0.8109	0.8167
ITML	0.7112	0.6081	0.7965	0.7049	0.5960	0.7889	0.7002	0.5463	0.7345
LMNN	0.7688	0.8401	0.8550	0.7568	0.8272	0.8411	0.7479	0.7850	0.8052
LowRank	0.6667	0.7871	0.3300	0.5568	0.7535	0.2950	0.5326	0.7710	0.2517
R2ML	0.7526	0.7920	0.8400	0.7329	0.7917	0.8398	0.7145	0.8116	0.8300

verifies that the proposed mechanism is more robust against the noise. This is mainly because we construct the relative constraints based on the ranking technique, instead of using concrete numerical probabilities which are usually subject to noise in real world.

5.3 Experiments for the Group-level Metric Learning Mechanism

In this section, we evaluate the performance of GrML on three real-world datasets [29] (i.e., Ionosphere, Heart and Diabetes). To generate the probabilistic examples, we randomly split the training dataset into groups of data size m. For each group, the associated probability (i.e., π_k) is the fraction of positive instances in this group, and it can be easily calculated based on the true label information of the datasets. In this experiment, we only take **Cosine** and **Euclidean** as baselines. The reason is that other baselines need to access each instance's label during the learning process and they cannot address the group-wise probability. Additionally, we measure each method's performance by the *relative accuracy*, which is defined as the accuracy of GrML relative to the accuracy that can be achieved by a metric learning method (we use **LMNN** in this paper) that has full access to the deterministic labels. Note that the accuracy calculated here is based on the predicted similarity labels of the instance pairs in the testing set, and these similarity labels are derived based on the learned distance metric.

Performance comparison. We first compare the relative accuracy of GrML with that of the baselines when the training dataset size and the group size vary. Here we consider eight cases where the training dataset size varies from 8 to 64. For each case, we first randomly select half of all instances in each dataset as the testing set, and then extract the training dataset from the remaining instances. Values of $m = 4, 8$ and 16 are chosen in this experiment. The results on the three datasets are shown in Figure 5, from which we can see GrML performs much better than the baselines in all cases, and the advantage of GrML becomes large when the training set size increases. Since **Cosine** and **Euclidean** only adopt cosine similarity and l_2-norm distance to measure the similarity of the instance pairs in the testing set, the performance of the two baselines

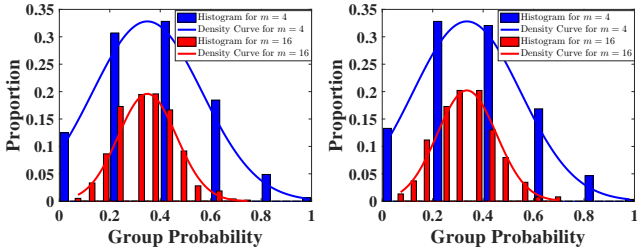


(a) Diabetes dataset (b) Heart dataset (c) Ionosphere dataset

Figure 5: Relative accuracy of the group-level mechanism w.r.t. the size of training dataset.

keeps stable when the training set size varies. Additionally, we can see that GrML achieves very high relative accuracy (the minimum value is lager than 0.8). This means that the performance of GrML is almost equivalent to that of the learning method which has full access to the instance labels. The results in Figure 5 also show that the relative accuracy of GrML decreases when the group size (i.e., m) becomes larger. This is mainly because the groups become less informative when the group size increases, which is consistent with the theoretical analysis in Section 4.3.

Distribution of the group-wise probabilities. Next, we study the effect of the group size (i.e., m) on the distribution of the group probabilities. In this experiment, we adopt the Diabetes dataset and the Ionosphere dataset. We first split each dataset into subsets (or groups) with equal size ($m = 4, 16$), and then compute the associated probability for each group based on the true label information. Then, we use histograms to provide visual displays of the distribution of group probabilities. To construct a histogram, we firstly divide the entire range of group probabilities (i.e., $[0, 1]$) into a series of consecutive and non-overlapping intervals (bins) and then compute the proportion of the groups that fall into each bin, with the sum of the heights equal to 1. Figure 6 shows the histograms of the two datasets, and each solid line represents a fit to the exponential distribution. As Figure 6 shows, the proportion of groups whose group probabilities are closer to 0 and 1 decreases when we increase the group size (i.e., m) from 4 to 16, which means the groups become less informative. This is consistent with the theoretical analysis and the experimental results in Figure 5. From Figure 5, we can also see that GrML can achieve good performance even in the challenging situation where $m = 16$, which means that the proposed mechanism is insensitive to the changes of the group size.



(a) Diabetes dataset (b) Ionosphere dataset

Figure 6: The distribution of the group-wise probabilities.

Convergence. Last but not least, we evaluate the convergence of GrML. In this experiment, the training dataset size and the group size is set as 48 and 8, respectively. Then we calculate the KL-divergence between values of $\{p_{ij}^k\}$ in consecutive iterations. Figure 7 shows the results on the Ionosphere dataset. Here we conduct the experiment for three times (i.e., Trail 1, Trail 2 and Trail 3). Each time the instances in the training dataset are randomly selected. From this figure, we can see the KL-divergences gradually converge to 0 with the increase of the iteration number, and this confirms that the convergence of GrML can be guaranteed.

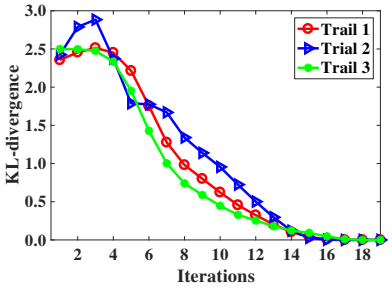


Figure 7: Convergence of the group-level mechanism on the Ionosphere dataset.

6 RELATED WORK

The past few years have witnessed a great increase in the number of metric learning works [1, 2, 5–8, 10–14, 20, 23, 25, 26, 28, 30, 31]. Traditional supervised metric learning algorithms [7, 8, 13, 20, 23, 25, 26, 30, 31] optimize the similarity metrics with the assumption that a fully labeled training dataset is available. The works in [7, 8, 23, 26, 30, 31] use the binary labels to generate sets of constraints which are then used as the supervised information. The authors in [20] deal with the metric learning problem with multi-class data. [25] presents a metric learning method to address the scenarios where some class labels in the training dataset are mislabeled. [13] proposes a method to learn a distance metric for multi-label problems where each instance in the training dataset is associated with a set of labels. There are also some other works [1, 2, 5, 14, 28] which address the problems of semi-supervised metric learning. [14] proposes a method which maximizes the entropy of the probability on labeled data and minimizes it on unlabeled data following

entropy regularization. Meanwhile, the semi-supervised information for metric learning problems can also be given in terms of a set of pairwise similarity and dissimilarity constraints [1, 2, 5, 28]. A well-known metric learning method with these constrains was proposed by Xing et al. [28]. Following this work, there are several emerging works [1, 2, 5] which study the metric learning problems by exploiting the given relevant constraints. Additionally, there are some works [6, 10–12] that address the multiple instance metric learning problem where the training dataset is provided as a set of labeled bags. They aim to learn a distance metric, which makes bags that share a label closer, and pushes bags that do not share any label apart [6, 11]. However, the above discussed metric learning works fail to deal with the probabilistic class labels.

Learning from such probabilistic information is of great importance [29]. Some works in other fields [9, 15, 16, 18, 29] also consider how to learn models from the probabilistic labels. However, the problem settings in these papers are quite different from ours. For example, the authors in [9] present class ratio models, which take as input an unlabeled set of data and predict the proportions of instances in the set belonging to different classes.

7 CONCLUSIONS

In this paper, we first propose an instance-level metric learning mechanism (InML), based on which the distance metrics can be learned directly from the instance-wise probabilistic labels. Compared with the existing metric learning methods, InML can fully utilize the probabilistic information and learn a more accurate metric. For the cases where the datasets are associated with group-wise probabilistic labels, we design a group-level metric learning mechanism (GrML), which can learn distance metrics directly from the group-wise probabilistic labels with high accuracy. Both theoretical analysis and extensive experiments on real-world datasets are provided to demonstrate the advantages of the proposed metric learning mechanisms.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and helpful suggestions. This work is supported in part by the US National Science Foundation under grants IIS-1218393 and IIS-1514204. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Mahdieh Soleymani Baghshah and Saeed Bagheri Shouraki. 2009. Semi-Supervised Metric Learning Using Pairwise Constraints. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 1217–1222.
[2] Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. 2005. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research* 6, Jun (2005), 937–965.
[3] Qiong Cao, Zheng-Chu Guo, and Yiming Ying. 2016. Generalization bounds for metric and similarity learning. *Machine Learning* 102, 1 (2016), 115–132.
[4] Olivier Chapelle, Vikas Sindhwani, and Sathiya S Keerthi. 2008. Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research* 9, Feb (2008), 203–233.
[5] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. 2007. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*. ACM, 209–216.
[6] Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. 2010. Multiple instance metric learning from automatically labeled bags of faces. In *Proceedings of the European conference on Computer Vision*. Springer, 634–647.
[7] Mengdi Huai, Chenglin Miao, Qiuling Suo, Yaliang Li, Jing Gao, and Aidong Zhang. 2018. Uncorrelated Patient Similarity Learning. In *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 270–278.
[8] Yinjie Huang, Cong Li, Michael Georgiopoulos, and Georgios C Anagnostopoulos. 2013. Reduced-rank local distance metric learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 224–239.
[9] Arun Shankar Iyer, J Saketha Nath, and Sunita Sarawagi. 2016. Privacy-preserving class ratio estimation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 925–934.
[10] Rong Jin, Shijun Wang, and Zhi-Hua Zhou. 2009. Learning a distance metric from multi-instance multi-label data. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*. IEEE, 896–902.
[11] Marc T Law, Yaoliang Yu, Raquel Urtasun, Richard S Zemel, and Eric P Xing. 2017. Efficient multiple instance metric learning using weakly supervised data. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*.
[12] Dewei Li and Yingjie Tian. 2016. Multi-view metric learning for multi-instance image classification. *arXiv preprint arXiv:1610.06671* (2016).
[13] Weiwei Liu and Ivor W Tsang. 2015. Large Margin Metric Learning for Multi-Label Prediction.. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 15. 2800–2806.
[14] Gang Niu, Bo Dai, Makoto Yamada, and Masashi Sugiyama. 2014. Information-theoretic semi-supervised metric learning via entropy regularization. *Neural computation* 26, 8 (2014), 1717–1762.
[15] Giorgio Patrini, Richard Nock, Paul Rivera, and Tiberio Caetano. 2014. (Almost) no label no cry. In *Advances in Neural Information Processing Systems*. 190–198.
[16] Peng Peng, Raymond Chi-Wing Wong, and Philip S Yu. 2014. Learning on probabilistic labels. In *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 307–315.
[17] Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. 2014. Gaussian process classification and active learning with multiple annotators. In *International Conference on Machine Learning*. 433–441.
[18] Stefan Rueping. 2010. SVM classifier estimation from group probabilities. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 911–918.
[19] Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.
[20] Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*. 1857–1865.
[21] Jimeng Sun, Fei Wang, Jianying Hu, and Shahram Edabollahi. 2012. Supervised patient similarity measure of heterogeneous patient records. *ACM SIGKDD Explorations Newsletter* 14, 1 (2012), 16–24.
[22] Tao Sun, Dan Sheldon, and Brendan OâĂŹConnor. 2017. A Probabilistic Approach for Learning with Label Proportions Applied to the US Presidential Election. In *Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 445–454.
[23] Qiuling Suo, Fenglong Ma, Ye Yuan, Mengdi Huai, Weida Zhong, Aidong Zhang, and Jing Gao. 2017. Personalized Disease Prediction Using a CNN-Based Similarity Learning Method. In *Proceedings of the 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*.
[24] Tian Tian and Jun Zhu. 2015. Max-margin majority voting for learning from crowds. In *Advances in Neural Information Processing Systems*. 1621–1629.
[25] Dong Wang and Xiaoyang Tan. 2014. Robust Distance Metric Learning in the Presence of Label Noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 1321–1327.
[26] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. 2006. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*. 1473–1480.
[27] Kilian Q Weinberger and Lawrence K Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research* 10, Feb (2009), 207–244.
[28] Eric P Xing, Michael I Jordan, Stuart J Russell, and Andrew Y Ng. 2003. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*. 521–528.
[29] Felix X Yu, Dong Liu, Sanjiv Kumar, Tony Jebara, and Shih-Fu Chang. 2013. SVM for learning with label proportions. *In Internalization conference on machine learning* (2013).
[30] Pourya Zadeh, Reshad Hosseini, and Suvrit Sra. 2016. Geometric mean metric learning. In *International Conference on Machine Learning*. 2464–2471.
[31] Mengting Zhan, Shilei Cao, Buyue Qian, Shiyu Chang, and Jishang Wei. 2016. Low-rank sparse feature selection for patient similarity learning. In *Proceeding of the 2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE.
[32] Dengyong Zhou, Qiang Liu, John Platt, and Christopher Meek. 2014. Aggregating ordinal labels from crowds by minimax conditional entropy. In *International conference on machine learning*. 262–270.