Learning Distance Metrics from Probabilistic Information

MENGDI HUAI, University of Virginia
CHENGLIN MIAO, State University of New York at Buffalo
YALIANG LI, Alibaba Group
QIULING SUO and LU SU, State University of New York at Buffalo
AIDONG ZHANG, University of Virginia

The goal of metric learning is to learn a good distance metric that can capture the relationships among instances, and its importance has long been recognized in many fields. An implicit assumption in the traditional settings of metric learning 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, which makes it difficult for the existing metric-learning methods to work well in these applications. To address this challenge, in this article, we study how to effectively learn the distance metric from datasets that contain probabilistic information, and then propose several 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 proposed mechanisms and conduct extensive experiments on real-world datasets to verify the desirable properties of these mechanisms.

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

Additional Key Words and Phrases: Metric learning, distance measure, probabilistic labels

ACM Reference format:
Mengdi Huai, Chenglin Miao, Yaliang Li, Qiuling Suo, Lu Su, and Aidong Zhang. 2020. Learning Distance Metrics from Probabilistic Information. *ACM Trans. Knowl. Discov. Data* 14, 5, Article 53 (July 2020), 33 pages.
https://doi.org/10.1145/3364320

1 INTRODUCTION

The problem of measuring the distance between the instance pairs is of fundamental importance in many data mining and machine learning algorithms, and the performance of such algorithms relies

This work is supported in part by the US National Science Foundation under grants IIS-1218393, IIS-1514204, IIS-1924928, IIS-1938167, and OAC-1934600. 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.
Authors' addresses: M. Huai and A. Zhang, Department of Computer Science, University of Virginia, 85 Engineer's Way, Charlottesville, VA 22904, USA; emails: {mh6ck, aidong}@virginia.edu; C. Miao, Q. Suo, and L. Su, 338 Davis Hall, Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260-2500, USA; emails: {cmiao, qiulings, lusu}@buffalo.edu; Y. Li, Alibaba Group, 500 108th Ave NE, Suite 800, Bellevue, WA 98004, USA; email: yaliangl.ub@gmail.com.

1556-4681/2020/07-ART53 $15.00
https://doi.org/10.1145/3364320

heavily on the choice of the distance metric. Although some simple metrics, such as Euclidean distance, can be used to measure the similarity degree between the given instances, they usually fail to capture the statistical regularities in the data, and thus the performance of the algorithms is largely degraded [45]. To address this challenge, the task of metric learning, whose goal is to learn a good distance metric that can well capture the important relationships among instances, has been widely studied [1, 2, 6, 14, 19, 21, 31, 33, 41, 44, 47, 51, 52], and the importance of distance metric learning has long been recognized in many fields.

In the traditional settings of distance metric learning, each instance used for training is usually associated with an attribute set denoting its *features* and a target attribute called *label*. In these settings, an implicit assumption is that the associated labels of the instances are deterministic (see Figure 1(a)). However, in many real-world applications, the associated labels in a training dataset come naturally with *probabilities* due to various reasons, such as uncertainty [24] or privacy issues [15, 40], and the *probabilistic labels* usually exist in the following two forms:

Instance-wise probabilistic label. Instead of being associated with a deterministic label (e.g., positive or negative in the binary case), *each instance* in the training dataset may come with a probabilistic label. As shown in Figure 1(b), this probabilistic label represents the probability that the instance has a particular deterministic label. Such instance-wise probabilistic label is very common in many real-world applications. For example, in crowdsourcing applications [25, 38, 54], the labeling task for each instance is usually outsourced to a large crowd of labelers by a data requester 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 [24, 35, 36].

Group-wise probabilistic label. In Figure 1(c), we show the dataset associated with group-wise probabilistic labels. The training dataset here consists of several disjoint 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 [15]. In this case, the label information for each instance is unknown, and the distance metric can only be learned 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 [26, 34], 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 usually cannot be revealed to the public because it is confidential. 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 instance-wise and group-wise probabilistic labels in real-world applications, the existing metric-learning methods 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 [24]. Additionally, determining an accurate threshold is usually very difficult in practice [25]. 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

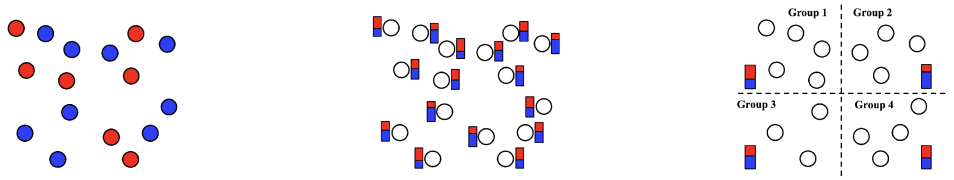


Fig. 1. The datasets with different label information. (a) Each instance is associated with a deterministic label. (b) Each instance is associated with a probabilistic label. (c) Each group is associated with a probabilistic label.

learning is that the distance between similar instances should be smaller than the distance between dissimilar instances [44, 48, 49, 51]. To achieve the goal, the metric is usually trained under sets of pairwise or triplet constraints. However, the pairwise or triplet constraints can not be constructed according to the group-wise probabilistic labels, which makes the learning task more challenging.

To address the above challenges, in this article, we first design a novel instance-level metric learning mechanism (InML), based on which a single (or global) 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. Additionally, considering that in some cases the global distance metric learned from the given training dataset fails to capture the local differences of the input feature space [13, 22, 27, 32, 42], we extend InML and design an effective instance-level local metric learning mechanism (InLoML), which can learn a set of instance specific distance metrics from the instance-wise probabilities. 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. Furthermore, to well capture the local discriminative information between different groups, we also extend GrML and design an effective group-level local metric learning mechanism (GrLoML), which can learn a set of group specific distance metrics from the group-wise probabilities.

In summary, the main contributions of this article are:

- —In order to address the metric-learning problems with the instance-wise probabilistic labels, we propose a novel InML, which can fully utilize the probabilistic information so that the learned distance metric can be more accurate.
- —To well capture the local differences of the input feature space, we extend InML and design an effective InLoML, based on which we can directly learn a set of instance specific distance metrics from the instance-wise probabilities.
- —For the scenarios where the training datasets are associated with group-wise probabilistic labels, we propose a GrML, based on which the distance metric can be directly learned from the group-wise probabilities with high accuracy.
- —We also extend GrML and propose an effective GrLoML, which can learn a set of group specific distance metrics that can well capture the local differences between different discriminative groups.
- —Both theoretical analysis and extensive experiments on real-world datasets demonstrate the advantages of the proposed distance metric learning mechanisms.

2 PROBLEM SETTING

In this section, we describe the problem setting of our proposed metric-learning mechanisms. Suppose there is a set of instances $\mathcal{X} = \{x_i \in \mathbb{R}^u\}_{i=1}^N$, where x_i is a u-dimensional feature vector. The goal of metric learning is to learn the following Mahalanobis distance metric:

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

which can effectively measure the similarity degree 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 given training instances $\mathcal{X}$. 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 dataset usually come with probabilities instead of deterministic values. In this article, we consider the following two different probabilistic cases:

- For the case where the associated probabilistic labels are *instance-wise*, 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.
- For the case where the associated probabilistic labels are *group-wise*, 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 article 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 first present the proposed InML in Section 3.1, and provide the theoretical analysis for InML in Section 3.2. In Section 3.3, we extend InML and describe the details of the proposed InLoML.

3.1 Learning Framework of InML

In the case where each instance $x_i \in \mathcal{X}$ 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 dataset associated with the probabilistic labels is usually more informative, some useful information may be lost during the transformation from probabilistic labels to deterministic labels, and this will degrade the performance of the learned distance metrics. Additionally, it is usually difficult to determine an accurate threshold in reality. To address the above challenges, 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., $\{c_i\}_{i=1}^N$). In order to achieve this goal, we first construct the distance constraints based on the relative comparison relationships that are derived through ranking the instance-wise probabilities (i.e., $\{c_i\}_{i=1}^N$), and then we design an optimization function based on the large margin framework to enforce the relative comparison of the constructed constraints.

Distance Constraint Construction. In this article, we assume without loss of generality that the associated probabilities $\{c_1, c_2, \ldots, c_N\}$ are sorted in decreasing order, i.e., $c_1 > c_2 > \cdots > c_{N-1} > c_N$. We first construct the following partially ordered triplet set:

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

It is obvious that for each triplet $(x_i, x_j, x_k) \in \mathcal{R}$, we have $c_j > c_k$ due to $j < k$. Considering the relationships among c_i, c_j, and c_k, we can divide this partially ordered 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 $(x_i, x_j, x_k) \in \mathcal{R}_1$, the associated probabilities satisfy $c_i > c_j > c_k$ due to $i < j < k$. 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_k should not be smaller than that between x_i and x_j (i.e., $d(x_i, x_j) \le d(x_i, x_k)$).
- $\mathcal{R}_2 = \{(x_i, x_j, x_k), 1 \le j < k < i \le N\}$. For each triplet (x_i, x_j, x_k) in this subset (i.e., $\mathcal{R}_2$), the associated probabilities satisfy that $c_j > c_k > c_i$. 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 (i.e., $\mathcal{R}_3$), the distance between x_i and x_k should not be smaller than that between x_i and x_j (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 (i.e., $\mathcal{R}_4$), 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 (i.e., $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}_3$, and $\mathcal{R}_4$), there is a distance constraint that is constructed according to the relative comparison relationships among the associated probabilities. When conducting metric learning from the instance set $\mathcal{X} = \{x_i\}_{i=1}^N$, we should make sure that these constructed distance constraints are satisfied. In the following, 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. For each triplet $(x_i, x_j, x_k) \in \mathcal{R}$, we first reformulate the above constructed distance constraints as:

$$\begin{cases} d(x_i, x_j) + g \le d(x_i, x_k) & \text{if } (x_i, x_j, x_k) \in \mathcal{R}_1 \\ d(x_i, x_k) + g \le d(x_i, x_j) & \text{if } (x_i, x_j, x_k) \in \mathcal{R}_2, \\ d(x_i, x_j) + g \le d(x_i, x_k) & \text{if } (x_i, x_j, x_k) \in \mathcal{R}_3 \\ d(x_i, x_k) + g \le d(x_i, x_j) & \text{if } (x_i, x_j, x_k) \in \mathcal{R}_4, \end{cases} \tag{3}$$

where $d(x_i, x_j) = (x_i - x_j)^T W (x_i - x_j)$, and g is a parameter used to regularize the gap (or margin) between $d(x_i, x_j)$ and $d(x_i, x_k)$. In this article, we use the unit margin (i.e., $g = 1$). To monitor the constructed inequality constraints in Equation (3), we then propose to minimize the following loss function:

$$\begin{aligned} \min_W \quad & \sum_{(x_i, x_j, x_k) \in \mathcal{R}_1} \max\{0, d(x_i, x_j) - d(x_i, x_k) + g\} \\ & + \sum_{(x_i, x_j, x_k) \in \mathcal{R}_2} \max\{0, d(x_i, x_k) - d(x_i, x_j) + g\} \\ & + \sum_{(x_i, x_j, x_k) \in \mathcal{R}_3} \max\{0, d(x_i, x_j) - d(x_i, x_k) + g\} \\ & + \sum_{(x_i, x_j, x_k) \in \mathcal{R}_4} \max\{0, d(x_i, x_k) - d(x_i, x_j) + g\}, \end{aligned} \tag{4}$$

where W is a positive semidefinite matrix. The operator $\max\{0, \cdot\}$ in Equation (4) denotes the hinge loss function, which is used to penalize the triplets that violate the constructed inequality constrains in Equation (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. Given that there exist some triplets violating the above constructed inequality constraints, we relax these constrains by incorporating nonnegative slack variables to monitor these margin violations. Suppose $\mathcal{R}_1' = \mathcal{R}_1 \cup \mathcal{R}_3$ and $\mathcal{R}_2' = \mathcal{R}_2 \cup \mathcal{R}_4$. Then we proceed to formulate the metric learning process as the following optimization problem:

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

where $\|W\|_*$ represents the nuclear norm to promote low-rankness, and α is the regularization parameter. In the above optimization problem, ξ^1_{ijk}'s and ξ^2_{ijk}'s are the introduced slack variables that allow the large margin inequality in Equation (3) to violate the margin. The above optimization problem is solved based on the sub-gradient descent method, and then we can derive the distance function $d(x_i, x_j) = (x_i - x_j)^T W (x_i - x_j)$.

Discussion. In the proposed instance-level mechanism, the distance constraint for each triplet (x_i, x_j, x_k) is constructed by comparing $d(x_i, x_j)$ with $d(x_i, x_k)$. 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, for the case where the associated probabilities of some instances are close (or equal) to each other, we can incorporate the binning method into the proposed mechanism and divide the instance sequence (i.e., $x_1, x_2, \ldots, x_N$) into multiple nonoverlapping bins according to their probabilistic labels. Then, only the constrains for the constructed triplets whose instances are in different bins are enforced. Then we can reformulate the original proposed optimization problem (i.e., Equation (5)). In this way, the constraint complexity of the original optimization problem can be reduced and the proposed mechanism will be robust to noise inherent in the probabilities.

3.2 Theoretical Analysis for InML

In this section, we provide theoretical analysis for the error bound generated by the proposed InML. Suppose X is the input instance space and C is the probability set. Let $Z = X \times C$ and $z_i = (x_i, c_i) \in Z$ mean that $x_i \in X$ and $c_i \in C$. We use $R(W)$ to denote the unbiased estimator of the true risk that is derived by taking expectation with respect to all possible values, i.e., $R(W) = \mathbb{E}_{z_i, z_j, z_k \sim \mu} \ell(d, z_i, z_j, z_k)$, where μ is the unknown probability distribution over Z and d denotes the distance metric function. $\ell(d, z_i, z_j, z_k)$ is the defined loss function with respect to the triplet $v = (z_i, z_j, z_k)$. Specifically, $\ell(d, z_i, z_j, z_k) = \max\{0, y_{ijk}(d(x_i, x_j) - d(x_i, x_k)) + g\}$, where $y_{ijk} = 1$ if $c_i > c_j > c_k$ or $c_j > c_i > (c_j + c_k)/2$, otherwise $y_{ijk} = -1$. Let W^* denote the true risk minimizer that is estimated from $R(W)$ (i.e., $W^* = \text{argmin}_W R(W)$), and $\hat{W}$ denote the distance metric learned based on the optimization problem described in Equation (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:*

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

where u is the dimension of the feature vector.

Proof. The proof is derived based on Rademacher analysis [28]. Firstly, based on the bounded difference inequality, with probability $1-\delta$, we can derive the following:

$$\begin{aligned} R(\hat{W}) - R(W^*) &= R(\hat{W}) - \hat{R}(\hat{W}) + \hat{R}(\hat{W}) - \hat{R}(W^*) + \hat{R}(W^*) - R(W^*) \\ &\leq 2\sup|\hat{R}(W) - R(W)|. \end{aligned}$$

The observation that $R(\hat{W}) - R(W^*) \leq 2\sup|\hat{R}(W) - R(W)|$ allows us to go from working with $R(\hat{W}) - R(W^*)$ to $\sup|\hat{R}(W) - R(W)|$. Based on that the loss function ℓ is Lipschitz and the corresponding Lipschtiz constant of ℓ is equal to 1, we can derive that $\Phi = \sup|\ell(d, z_i, z_j, z_k) - \ell(d, z_{i'}, z_{j'}, z_{k'})| \leq \max_i(x_i^T\hat{W}x_i) = O(r\log N)$. Then, applying McDiamid's inequality [20] to the above term $2\sup|\hat{R}(W) - R(W)|$, we can derive:

$$\begin{aligned} R(\hat{W}) - R(W^*) &\leq 2\sup|\hat{R}(W) - R(W)| \\ &\leq 2\mathbb{E}[\sup|\hat{R}(W) - R(W)|] + \sqrt{\frac{2\Phi^2\log(2/\delta)}{|\mathcal{R}|}}. \end{aligned}$$

where $|\mathcal{R}| = |\mathcal{R}_1' \cup \mathcal{R}_2'| = N(N-1)(N-2)/2$. By using the standard symmetrization and contraction lemmas, we can introduce the independent identically distributed Rademacher random variables for all $v \in \mathcal{R}$ so that:

$$\begin{aligned} &2\mathbb{E}[\sup|\hat{R}(W) - R(W)|] \\ &\leq \mathbb{E}\left[\frac{2}{|\mathcal{R}|}\sup\left\|\sum_{v\in\mathcal{R}}\epsilon_v\left(x_ix_k^T + x_kx_i^T - x_ix_j^T - x_jx_i^T + x_jx_j^T - x_kx_k^T\right)\right\| \cdot \|W\|_*\right] \\ &\leq 2\mathbb{E}\left[\frac{2\Psi}{|\mathcal{R}|}\sup\left\|\sum_{v\in\mathcal{R}}\epsilon_v\left(x_ix_k^T + x_kx_i^T - x_ix_j^T - x_jx_i^T + x_jx_j^T - x_kx_k^T\right)\right\|\right], \end{aligned}$$

where $v = (z_i, z_j, z_k)$, and $\{\epsilon_v\}_{v=1}^{|\mathcal{R}|}$ denote the introduced independent identically distributed Rademacher random variables [28]. For each Rademacher random variable ϵ_v, it has a 50% chance of being +1 and a 50% chance of being −1. The second inequality is derived based on that $\|W\|_* \leq \Psi = u\sqrt{r}$. By using the matrix Bernstein bound, Theorem 6.6.1 in [39], we can obtain

$$\mathbb{E}[\sup|\hat{R}(W) - R(W)|] \leq \frac{2\Psi}{|\mathcal{R}|}\left[\sqrt{140\frac{\|\mathcal{X}\mathcal{X}^T\|}{N}|\mathcal{R}|\log u} + 2\log u\right],$$

where $\|\mathcal{X}\mathcal{X}^T\| = O(N/u)$. Thus, by combining the above results, we can derive

$$\begin{aligned} R(\hat{W}) - R(W^*) &\leq 2\sup|\hat{R}(W) - R(W)| \\ &\leq 2\mathbb{E}[\sup|\hat{R}(W) - R(W)|] + \sqrt{\frac{2\Phi^2\log(2/\delta)}{|\mathcal{R}|}}. \\ &\leq \frac{4\Psi}{|\mathcal{R}|}\left[\sqrt{140\frac{\|\mathcal{X}\mathcal{X}^T\|}{N}|\mathcal{R}|\log u} + 2\log u\right] + \sqrt{\frac{2\Phi^2\log(2/\delta)}{|\mathcal{R}|}} \\ &= O\left(\sqrt{\frac{ru(\log u + \log^2 N\log(2/\delta))}{N^3 - 3N^2 + 2N}}\right). \end{aligned}$$

So far, we finish the proof. □

According to this theorem, we can easily verify that the error bound generated by the proposed mechanism is $O(\sqrt{\log^2 N/(N^3 - 3N^2 + 2N)})$, where $N > 3$. Cosidering that $\log^2 N < (N^2 - 3N + 2)$ when $N > 3$, we can get that the above generated error bound is tighter than the existing best-known bound $O(\sqrt{1/N})$ that is derived from the datasets with binary class labels [4]. That is to say our mechanism can learn a good metric with a smaller number of instances than the existing metric-learning methods.

3.3 Local Metric Learning from Instance-wise Probabilistic Labels

The proposed InML aims to learn a single (or global) Mahalanobis distance metric from the instance-level probabilities, which keeps all of the instances in the same class close together and ensures that those from different classes remain separated. However, in many real-world applications, learning a global distance metric usually suffers a limitation: it only makes use of a single linear metric to compute the distances among all the instance pairs, and fails to take into account the local differences of the input feature space [13, 22, 27, 32, 42]. Thus, learning a global distance metric may not fit well with the distance over the data manifold. To address this problem, in this section, we extend InML and propose an effective InLoML, based on which we can learn a local distance metric for each instance from the probabilistic labels. The proposed InLoML can increase the expressive power of the standard global metric learning by learning a number of local metrics and well capture the local differences of the input feature space. In the following, we provide the details of the proposed InML. We first construct a set of local distance constraints, and then design an optimization function to enforce the relative comparison of the constructed local constraints.

Local Constraint Construction. Suppose W_i is the local distance metric that needs to be learned for instance x_i. Then, the distance between x_i and x_j can be defined as:

$$d_{W_i}(x_i, x_j) = (x_i - x_j)^T W_i (x_i - x_j). \tag{7}$$

Here we use the notation $d_{W_i}(x_i, x_j)$ to highlight the parameterization of the Mahalanobis distance by W_i. Note that the local distance metric W_i of the i-th instance x_i is usually different from that of the j-th instance x_j. Similar to the constraint construction in InML, we can divide the partially ordered triplet set $\mathcal{R} = \{(x_i, x_j, x_k), 1 \leq i \neq j \neq k \leq N, j < k\}$ into the following four subsets:

- $\Re_1 = \{(x_i, x_j, x_k), 1 \leq j < i < k \leq N, c_j > c_i > (c_j + c_k)/2\}$. 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., $d_{W_i}(x_i, x_j) \leq d_{W_i}(x_i, x_k)$).

- $\Re_2 = \{(x_i, x_j, x_k), 1 \le j < k < i \le N\}$. 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., $d_{W_i}(x_i, x_k) \le d_{W_i}(x_i, x_j)$).
- $\Re_3 = \{(x_i, x_j, x_k), 1 \le i < j < k \le N\}$. For each triplet (x_i, x_j, x_k) in $\Re_3$, 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., $d_{W_i}(x_i, x_j) \le d_{W_i}(x_i, x_k)$).
- $\Re_4 = \{(x_i, x_j, x_k), 1 \le j < i < k \le N, (c_j + c_k)/2 > c_i > c_k\}$. 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., $d_{W_i}(x_i, x_k) \le d_{W_i}(x_i, x_j)$).

The above distance constraints should be satisfied when we conduct metric learning to learn the local distance metric W_i. Next, we discuss how to learn W_i based on these constraints.

Optimization Formulation. The metric-learning process of InLoML is also formulated as an optimization problem based on the large margin framework with the hinge loss. For each constructed triplet $(x_i, x_j, x_k) \in \mathcal{R}$, we reformulate the above constructed distance constraints as:

$$\begin{cases} d_{W_i}(x_i, x_j) + g \le d_{W_i}(x_i, x_k) & \text{if } (x_i, x_j, x_k) \in \Re_1 \\ d_{W_i}(x_i, x_k) + g \le d_{W_i}(x_i, x_j) & \text{if } (x_i, x_j, x_k) \in \Re_2, \\ d_{W_i}(x_i, x_j) + g \le d_{W_i}(x_i, x_k) & \text{if } (x_i, x_j, x_k) \in \Re_3, \\ d_{W_i}(x_i, x_k) + g \le d_{W_i}(x_i, x_j) & \text{if } (x_i, x_j, x_k) \in \Re_4, \end{cases} \tag{8}$$

where g is used to regularize the gap (or margin) between $d_{W_i}(x_i, x_j)$ and $d_{W_i}(x_i, x_k)$. Here we still choose a unit margin. Then, we propose to minimize the following loss function based on the hinge loss framework:

$$\begin{aligned} \min_{\{W_i\}_{i=1}^N} & \sum_{(x_i, x_j, x_k) \in \Re_1} \max\{0, d_{W_i}(x_i, x_j) - d_{W_i}(x_i, x_k) + g\} \\ & + \sum_{(x_i, x_j, x_k) \in \Re_2} \max\{0, d_{W_i}(x_i, x_k) - d_{W_i}(x_i, x_j) + g\} \\ & + \sum_{(x_i, x_j, x_k) \in \Re_3} \max\{0, d_{W_i}(x_i, x_j) - d_{W_i}(x_i, x_k) + g\} \\ & + \sum_{(x_i, x_j, x_k) \in \Re_4} \max\{0, d_{W_i}(x_i, x_k) - d_{W_i}(x_i, x_j) + g\}. \end{aligned} \tag{9}$$

The hinge loss function $\max\{0, \cdot\}$ is used to penalize the triplets that violate the inequality constrains in Equation (8). If the inequality does hold, its hinge loss makes no contribution to the overall loss function. In the above optimization problem, we need to derive the local distance metric for each instance. Thus, the number of parameters that need to be optimized has a complexity of $O(Nu^2)$, where N denotes the number of samples and u denotes the feature dimension. However, learning on the order of $O(Nu^2)$ parameters is computationally expensive. To address this challenge, we propose the following mechanism to learn the local metrics by adopting the premise that positive semi-definite matrices have non-negative eigenvalues and orthogonal eigenvectors.

Decomposition. Note that the local distance metric W_i for the i-th instance x_i is positive semidefinite. We assume that the matrix is of rank S. Then, based on the rank-one decomposition theorem for positive semidefinite matrices, there exist S square matrices $U_1, U_2, \ldots, U_S$, and each of them is of size u and rank one such that:

$$W_i = \lambda_1^i U_1 + \lambda_2^i U_2 + \cdots + \lambda_s^i U_s + \cdots + \lambda_S^i U_S, \tag{10}$$

where $\lambda_s^i \geq 0$, S is a constant in the optimization procedure, and U_s is a rank-one matrix of size $u \times u$. U_s is computed as $U_s = \Lambda_s \Lambda_s^T$, where $\Lambda_s \in \mathbb{R}^u$ is a u-dimensional vector. This is called a rank-one decomposition of W_i of length S. Then, we can rewrite Equation (10) as follows:

$$W_i = \sum_{s=1}^{S} \lambda_s^i \Lambda_s (\Lambda_s)^T, \tag{11}$$

where $(\Lambda_s)^T$ denotes the transpose of Λ_s, and the basis elements $\{\Lambda_s \in \mathbb{R}^u\}_{s=1}^S$ can be generated by using the Fisher discriminant analysis technique. To further reduce the number of parameters that need to be learned, for the i-th instance x_i in the training set $\mathcal{X}$, we proceed to rewrite its local distance metric W_i (a positive semidefinite matrix of size $u \times u$) defined in Equation (11) as follows:

$$W_i = \sum_{s=1}^{S} \left(\sigma_s^T \mathfrak{x}_i + e_s\right)^2 \Lambda_s (\Lambda_s)^T, \tag{12}$$

where $e_s \in \mathbb{R}$, $\mathfrak{x}_i \in \mathbb{R}^{u'}$ is an embedding of $x_i \in \mathbb{R}^u$, and $\sigma_s \in \mathbb{R}^{u'}$ is a column vector. Here, $\mathfrak{x}_i$ is derived by using the radius basis function (RBF) kernel principal component analysis (PCA), and the corresponding bandwidth is set to the median of the Euclidean distances of the instance pairs in the training dataset. For each instance pair, its Euclidean distance is calculated as the square root of the sum of squared differences between corresponding features of the two instances. And u' is treated as a constant in the optimization process. The two sets of parameters $\{\sigma_s \in \mathbb{R}^{u'}\}_{s=1}^S$ and $\{e_s \in \mathbb{R}\}_{s=1}^S$ are the parameters we need to optimize. Thus, for each instance $x_i \in \mathcal{X}$, its local distance function defined in Equation (7) can be rewritten as:

$$d_{W_i}(x_i, x_j) = (x_i - x_j)^T \sum_{s=1}^{S} \left(\sigma_s^T \mathfrak{x}_i + e_s\right)^2 \Lambda_s (\Lambda_s)^T (x_i - x_j). \tag{13}$$

Based on Equations (9) and (13), we can further derive the following objective function:

$$\begin{aligned}
\min_{\{\sigma_s\}_{s=1}^S, \{e_s\}_{s=1}^S} \quad & \sum_{(x_i, x_j, x_k) \in \mathfrak{R}_1} \max\Bigg\{0, (x_i - x_j)^T \sum_{s=1}^{S} \left(\sigma_s^T \mathfrak{x}_i + e_s\right)^2 \Lambda_s \Lambda_s^T (x_i - x_j) \\
& \qquad -(x_i - x_k)^T \sum_{s=1}^{S} \left(\sigma_s^T \mathfrak{x}_i + e_s\right)^2 \Lambda_s \Lambda_s^T (x_i - x_k) + g\Bigg\} \\
& + \sum_{(x_i, x_j, x_k) \in \mathfrak{R}_2} \max\Bigg\{0, (x_i - x_k)^T \sum_{s=1}^{S} \left(\sigma_s^T \mathfrak{x}_i + e_s\right)^2 \Lambda_s \Lambda_s^T (x_i - x_k) \\
& \qquad -(x_i - x_j)^T \sum_{s=1}^{S} \left(\sigma_s^T \mathfrak{x}_i + e_s\right)^2 \Lambda_s \Lambda_s^T (x_i - x_j) + g\Bigg\} \\
& + \sum_{(x_i, x_j, x_k) \in \mathfrak{R}_3} \max\Bigg\{0, (x_i - x_j)^T \sum_{s=1}^{S} \left(\sigma_s^T \mathfrak{x}_i + e_s\right)^2 \Lambda_s \Lambda_s^T (x_i - x_j) \\
& \qquad -(x_i - x_k)^T \sum_{s=1}^{S} \left(\sigma_s^T \mathfrak{x}_i + e_s\right)^2 \Lambda_s \Lambda_s^T (x_i - x_k) + g\Bigg\} \\
& + \sum_{(x_i, x_j, x_k) \in \mathfrak{R}_4} \max\Bigg\{0, (x_i - x_k)^T \sum_{s=1}^{S} \left(\sigma_s^T \mathfrak{x}_i + e_s\right)^2 \Lambda_s \Lambda_s^T (x_i - x_k) \\
& \qquad -(x_i - x_j)^T \sum_{s=1}^{S} \left(\sigma_s^T \mathfrak{x}_i + e_s\right)^2 \Lambda_s \Lambda_s^T (x_i - x_j) + g\Bigg\} \\
& + \Upsilon \left\| \begin{pmatrix} \sigma \\ e \end{pmatrix} \right\|_{2,1},
\end{aligned} \tag{14}$$

where Υ is the regularization parameter, and $\sigma = [\sigma_1, \sigma_2, \ldots, \sigma_S]$ is a matrix of size $u' \times S$. $\{\sigma_s \in \mathbb{R}^{u'}\}_{s=1}^{S}$ and $\{e_s \in \mathbb{R}\}_{s=1}^{S}$ are the parameters we need to optimize. We can efficiently solve this optimization problem by using stochastic composite optimization [7, 46]. From the above objective function (i.e., Equation (14)), we can see that for the training instances $\{x_i \in \mathbb{R}^u\}_{i=1}^{N}$, their local distance metrics can be learned simultaneously, which makes them comparable and help to alleviate the overfitting problem. In this optimization problem (i.e., Equation (14)), the number of parameters needed to be learned grows at the order of $S(u' + 1)$. If we change either the value of S or the value of u', the computation cost of the proposed instance-level local mechanism will be affected. Additionally, the number of parameters needed to be learned is independent of both the feature dimension u and the training set size N, which means that the proposed instance-level local metric-learning mechanism can be very efficient in real-world applications.

Discussion. In the proposed instance-level local metric-learning mechanism, we decompose each local distance metric based on the rank-one decomposition theorem. This method also allows us to reduce the computational cost of InML proposed in the previous section. To achieve the goal, we first rewrite the global distance function defined in Equation (1) as:

$$\begin{aligned} d(x_i, x_j) &= (x_i - x_j)^T W (x_i - x_j) \\ &= (x_i - x_j)^T V \cdot \mathrm{diag}(\delta) \cdot V^T (x_i - x_j), \end{aligned} \tag{15}$$

where $V \in \mathbb{R}^{u \times S}$, $\delta = [\delta_1, \delta_2, \ldots, \delta_S]$ and $W = V \cdot \mathrm{diag}(\delta) \cdot V^T$. Let the s-th column of the matrix V be $v_s \in \mathbb{R}^u$, which means that $V = [v_1, v_2, \ldots, v_S]$. Then, the global distance metric W can be rewritten as $W = \sum_{s=1}^{S} \delta_s v_s (v_s)^T$. Based on this, we can rewrite the optimization problem of InML defined in Equation (4) as follows:

$$\begin{aligned} \min_{\{\delta_s\}_{s=1}^{S}} & \sum_{(x_i, x_j, x_k) \in \mathcal{R}_1} \max \left\{ 0, (x_i - x_j)^T \sum_{s=1}^{S} \delta_s v_s v_s^T (x_i - x_j) \right. \\ & \qquad \left. - (x_i - x_k)^T \sum_{s=1}^{S} \delta_s v_s v_s^T (x_i - x_k) + g \right\} \\ & + \sum_{(x_i, x_j, x_k) \in \mathcal{R}_2} \max \left\{ 0, (x_i - x_k)^T \sum_{s=1}^{S} \delta_s v_s v_s^T (x_i - x_k) \right. \\ & \qquad \left. - (x_i - x_j)^T \sum_{s=1}^{S} \delta_s v_s v_s^T (x_i - x_j) + g \right\} \\ & + \sum_{(x_i, x_j, x_k) \in \mathcal{R}_3} \max \left\{ 0, (x_i - x_j)^T \sum_{s=1}^{S} \delta_s v_s v_s^T (x_i - x_j) \right. \\ & \qquad \left. - (x_i - x_k)^T \sum_{s=1}^{S} \delta_s v_s v_s^T (x_i - x_k) + g \right\} \\ & + \sum_{(x_i, x_j, x_k) \in \mathcal{R}_4} \max \left\{ 0, (x_i - x_k)^T \sum_{s=1}^{S} \delta_s v_s v_s^T (x_i - x_k) \right. \\ & \qquad \left. - (x_i - x_j)^T \sum_{s=1}^{S} \delta_s v_s v_s^T (x_i - x_j) + g \right\}, \end{aligned} \tag{16}$$

where $\{\delta_s \geq 0\}_{s=1}^{S}$ are the parameters we need to learn. In this way, the number of parameters needed to be learned is S, and we also do not need to perform projections onto the positive semidefinite cone. The optimization problem defined in Equation (16) can be solved by using the Regularized Dual Averaging method [46], which offers fast convergence and levels of sparsity in the solution.

4 METRIC LEARNING FROM GROUP-WISE PROBABILISTIC LABELS

In this section, we present how to effectively learn the Mahalanobis distance metrics from the group-wise probabilities (i.e., $\{\pi_k \in [0,1]\}_{k=1}^{K}$). We first formulate the learning framework of the proposed GrML as an optimization problem and discuss how to effectively solve this problem in Sections 4.1 and 4.2, respectively. Then, we conduct theoretical analysis for GrML in Section 4.3. The local metric learning mechanism from group-wise probabilities (i.e., GrLoML) is discussed in Section 4.4.

4.1 Learning Framework of GrML

In the case where the probabilistic label is group-wise, we have no information about the instance-wise labels, and we only have access to the group-wise probabilities (i.e., $\{\pi_k \in [0,1]\}_{k=1}^{K}$). This makes it more difficult to learn an accurate distance 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.

Suppose the training instance set $\mathcal{X}$ consists of K disjoint groups, i.e., $\mathcal{X} = \cup\{\mathcal{X}_k\}_{k=1}^{K}$, where $\mathcal{X}_k = \{x_i^k \in \mathbb{R}^u\}_{i=1}^{|\mathcal{X}_k|}$ 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 the k-th 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. We also associate each disjoint 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 the k-th group $\mathcal{X}_k$. Then $\hat{\pi}_k$ can be derived as:

$$\begin{aligned}\hat{\pi}_k &= \frac{\sum_{i<j} \mathbb{1}[y_{ij}^k]}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)/2} \\ &= \frac{|\mathcal{X}_k|(|\mathcal{X}_k|-1)/2 - |\mathcal{X}_k|\pi_k|\mathcal{X}_k|(1-\pi_k)}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)/2} \\ &= 1 - \frac{2|\mathcal{X}_k|\pi_k(1-\pi_k)}{|\mathcal{X}_k|-1}, \end{aligned} \tag{17}$$

where $\mathbb{1}[y_{ij}^k]$ is the indicator function that outputs 1 if $y_{ij}^k > 0$ and 0 otherwise. $|\mathcal{X}_k|(|\mathcal{X}_k|-1)/2$ is the number of all possible instance pairs in the k-th group $\mathcal{X}_k$, and $|\mathcal{X}_k|\pi_k|\mathcal{X}_k|(1-\pi_k)$ denotes the number of the dissimilar instance pairs whose similarity labels are equal to -1 in $\mathcal{X}_k$. During the training process of the proposed group-level metric-learning mechanism, $\hat{\pi}_k$ can be treated as a constant because π_k is a known probability value for the k-th group $\mathcal{X}_k$.

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 the positive semidefinite matrix W as $M^T M$ [10, 52]. Note that the matrix $M \in \mathbb{R}^{u \times u}$ is not required to be positive semidefinite. In this way, projections onto the positive semidefinite cone are not needed, and we can also reduce the number of parameters we need to learn by imposing a low-rank constraint on

the matrix M. In order to learn the distance metric, we propose to adopt maximum likelihood estimation. Specifically, we choose the parameter M that makes the likelihood of having the obtained instance pairs maximum, and the likelihood function with respect to the unknown parameter M is defined as the product of the similarity probabilities of all possible instance pairs. For each instance pair (x_i^k, x_j^k), its similarity probability with respect to the parameter M is modeled as follows:

$$\Pr\left(y_{ij}^k \middle| x_i^k, x_j^k; M, b\right) = \frac{1}{1 + \exp\left(-y_{ij}^k\left(d\left(x_i^k, x_j^k\right) - b\right)\right)}, \tag{18}$$

where $d(x_i^k, x_j^k) = (x_i^k - x_j^k)^T M^T M (x_i^k - x_j^k)$ and b is the bias that 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., $y_{ij}^k = -1$). In this article, we set b as 1. Since the log function is a monotonic increasing function, maximizing the likelihood function is equivalent to maximizing the log likelihood, and also to minimizing the negative log likelihood. Then we can formulate the following optimization problem.

$$\begin{aligned}
\min_{I,M} \quad & \sum_{k=1}^{K} \sum_{i<j} \frac{2\log(1 + \exp(-y_{ij}^k((x_i^k - x_j^k)^T M^T M (x_i^k - x_j^k) - b)))}{|\mathcal{X}_k|(|\mathcal{X}_k| - 1)} + \frac{\|M\|_F^2}{2}, \\
\text{s.t.} \quad & \sum_{i<j} \frac{y_{ij}^1}{|\mathcal{X}_1|(|\mathcal{X}_1| - 1)} + \frac{1}{2} = \hat{\pi}_1, \\
& \sum_{i<j} \frac{y_{ij}^2}{|\mathcal{X}_2|(|\mathcal{X}_2| - 1)} + \frac{1}{2} = \hat{\pi}_2, \\
& \vdots \\
& \sum_{i<j} \frac{y_{ij}^K}{|\mathcal{X}_K|(|\mathcal{X}_K| - 1)} + \frac{1}{2} = \hat{\pi}_K,
\end{aligned} \tag{19}$$

where $I = \{y_{ij}^k | i < j, k = 1, \ldots, K\}$. The objective function in this optimization problem contains the following two terms: the first term denotes the complete data log-likelihood and it is derived from the negative log likelihood of the instance pairs. The second term denotes the Frobenius-norm regularization. The constraints in the above optimization problem are used to enforce that for each group, the estimated proportion of the similar instance pairs (i.e., $\sum_{i<j} \frac{y_{ij}^k}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)} + \frac{1}{2}$) is equal to the pre-defined similarity proportion $\hat{\pi}_k$.

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 Equation (19). However, it is difficult to solve this optimization problem due to the categorical property of y_{ij}^k. To address this 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 technique [5] 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 Equation (19) can be rewritten as:

$$\begin{aligned}\min_{P,M}\ & \sum_{k=1}^{K}\sum_{i<j}\frac{2p_{ij}^k}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}\log\Big(1+\exp\Big(-\Big(\big(x_i^k-x_j^k\big)^T M^T M\big(x_i^k-x_j^k\big)-b\Big)\Big)\Big)\\ &+\sum_{k=1}^{K}\sum_{i<j}\frac{2(1-p_{ij}^k)}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}\log\Big(1+\exp\Big(\Big(\big(x_i^k-x_j^k\big)^T M^T M\big(x_i^k-x_j^k\big)-b\Big)\Big)\Big)+\frac{1}{2}\|M\|_F^2,\\ \text{s.t.}\ & \sum_{i<j}\frac{2p_{ij}^1}{|\mathcal{X}_1|(|\mathcal{X}_1|-1)}=\hat{\pi}_1,\\ & \sum_{i<j}\frac{2p_{ij}^2}{|\mathcal{X}_2|(|\mathcal{X}_2|-1)}=\hat{\pi}_2,\\ & \vdots\\ & \sum_{i<j}\frac{2p_{ij}^K}{|\mathcal{X}_K|(|\mathcal{X}_K|-1)}=\hat{\pi}_K,\end{aligned}\tag{20}$$

where $P=\{p_{ij}^k|i<j,k=1,2,\ldots,K\}$. To mitigate local minima, an entropy term [5] for the distributions defined by p_{ij}^k is also added to the above objective function. Finally, the following optimization problem is formulated to learn the distance metric.

$$\begin{aligned}\min_{P,M}\ \mathcal{L}(P,M)=&\sum_{k=1}^{K}\sum_{i<j}\frac{2p_{ij}^k}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}\log\Big(1+\exp\Big(-\Big(\big(x_i^k-x_j^k\big)^T M^T M\big(x_i^k-x_j^k\big)-b\Big)\Big)\Big)\\ &+\sum_{k=1}^{K}\sum_{i<j}\frac{2(1-p_{ij}^k)}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}\log\Big(1+\exp\Big(\Big(\big(x_i^k-x_j^k\big)^T M^T M(x_i^k-x_j^k)-b\Big)\Big)\Big)\\ &+\sum_{k=1}^{K}\frac{2T}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}\sum_{i<j}\Big(p_{ij}^k\log p_{ij}^k+\big(1-p_{ij}^k\big)\log\big(1-p_{ij}^k\big)\Big)+\frac{1}{2}\|M\|_F^2,\\ \text{s.t.}\ & \sum_{i<j}\frac{2p_{ij}^1}{|\mathcal{X}_1|(|\mathcal{X}_1|-1)}=\hat{\pi}_1,\\ & \sum_{i<j}\frac{2p_{ij}^2}{|\mathcal{X}_2|(|\mathcal{X}_2|-1)}=\hat{\pi}_2,\\ & \vdots\\ & \sum_{i<j}\frac{2p_{ij}^K}{|\mathcal{X}_K|(|\mathcal{X}_K|-1)}=\hat{\pi}_K,\end{aligned}\tag{21}$$

where T is a penalty parameter.

4.2 Optimization for GrML

In this section, we discuss how to solve the optimization problem described in Equation (21). 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:

$$\min_{M} \ \mathcal{L}_1(M) = \sum_{k=1}^{K} \sum_{i<j} \frac{2p_{ij}^k}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)} \log\left(1+\exp\left(-\left(\left(x_i^k-x_j^k\right)^T M^T M\left(x_i^k-x_j^k\right)-b\right)\right)\right)$$

$$+\sum_{k=1}^{K}\sum_{i<j}\frac{2(1-p_{ij}^k)\log(1+\exp(((x_i^k-x_j^k)^T M^T M(x_i^k-x_j^k)-b)))}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}+\frac{\|M\|_F^2}{2}. \tag{22}$$

Here we adopt gradient descent method to update M. Specifically, in the t-th iteration, we update the parameter M as follows:

$$M^t = M^{t-1} - \theta\frac{\partial \mathcal{L}_1}{\partial M} \tag{23}$$

where $\frac{\partial \mathcal{L}_1}{\partial M}$ denotes the derivative of $\mathcal{L}_1(M)$ with respect to $M \in \mathbb{R}^{u\times u}$, and θ denotes the learning rate. The derivative $\frac{\partial \mathcal{L}_1}{\partial M}$ is calculated as:

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

$$+\sum_{k=1}^{K}\sum_{i<j}\frac{2(1-p_{ij}^k)}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}\frac{2M(x_i^k-x_j^k)^T(x_i^k-x_j^k)}{1+\exp(-((x_i^k-x_j^k)^T M^T M(x_i^k-x_j^k)-b))}+M, \tag{24}$$

where $M \in \mathbb{R}^{u\times u}$, $x_i^k \in \mathbb{R}^u$, $p_{ij}^k \in [0,1]$, and $|\mathcal{X}_k|$ is the number of instances in the k-th group $\mathcal{X}_k$.

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

$$\begin{aligned}\mathcal{L}_2(P) &= \mathcal{L}(P,M) - \sum_{k=1}^{K}\lambda_k\left(\sum_{i<j}\frac{2p_{ij}^k}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}-\hat{\pi}_k\right)\\ &= \sum_{k=1}^{K}\sum_{i<j}\frac{2p_{ij}^k}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}\log\left(1+\exp\left(-\left(\left(x_i^k-x_j^k\right)^T M^T M(x_i^k-x_j^k)-b\right)\right)\right)\\ &\quad+\sum_{k=1}^{K}\sum_{i<j}\frac{2(1-p_{ij}^k)}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}\log\left(1+\exp\left(\left(\left(x_i^k-x_j^k\right)^T M^T M(x_i^k-x_j^k)-b\right)\right)\right)\\ &\quad+\sum_{k=1}^{K}\frac{2T}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}\sum_{i<j}\left(p_{ij}^k\log p_{ij}^k+\left(1-p_{ij}^k\right)\log\left(1-p_{ij}^k\right)\right)+\frac{1}{2}\|M\|_F^2\\ &\quad-\sum_{k=1}^{K}\lambda_k\left(\sum_{i<j}\frac{2p_{ij}^k}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}-\hat{\pi}_k\right).\end{aligned} \tag{25}$$

The partial derivative of $\mathcal{L}_2(P)$ with respect to the variable p_{ij}^k is computed as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}_2}{\partial p_{ij}^k} =& \frac{2}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)} \log\Big(1+\exp\Big(-\Big(\Big(x_i^k-x_j^k\Big)^T M^T M\Big(x_i^k-x_j^k\Big)-b\Big)\Big)\Big) \\ &+ \frac{-2}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)} \log\Big(1+\exp\Big(\Big(\Big(x_i^k-x_j^k\Big)^T M^T M\Big(x_i^k-x_j^k\Big)-b\Big)\Big)\Big) \\ &+ \frac{2T}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)}\Big(\log p_{ij}^k - \log\Big(1-p_{ij}^k\Big)\Big) \\ &- \frac{2\lambda_k}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)},\end{aligned} \tag{26}$$

where $|\mathcal{X}_k|$ denotes the number of instances in the k-th group. Let the partial derivative of $\mathcal{L}_2(P)$ with respect to p_{ij}^k be zero (i.e., $\frac{\partial \mathcal{L}_2}{\partial p_{ij}^k}=0$), and we can derive the following:

$$\begin{aligned}&\log\Big(1+\exp\Big(-\Big(\Big(x_i^k-x_j^k\Big)^T M^T M\Big(x_i^k-x_j^k\Big)-b\Big)\Big)\Big) \\ &-\log\Big(1+\exp\Big(\Big(\Big(x_i^k-x_j^k\Big)^T M^T M\Big(x_i^k-x_j^k\Big)-b\Big)\Big)\Big)-\lambda_k = T\Big(\log\Big(1-p_{ij}^k\Big)-\log p_{ij}^k\Big), \\ &\Rightarrow \frac{1}{T}\log\frac{1+\exp(-((x_i^k-x_j^k)^T M^T M(x_i^k-x_j^k)-b))}{1+\exp(((x_i^k-x_j^k)^T M^T M(x_i^k-x_j^k)-b))}-\frac{\lambda_k}{T}=\log\left(\frac{1-p_{ij}^k}{p_{ij}^k}\right), \\ &\Rightarrow \exp\left(\frac{1}{T}\log\frac{1+\exp(-((x_i^k-x_j^k)^T M^T M(x_i^k-x_j^k)-b))}{1+\exp(((x_i^k-x_j^k)^T M^T M(x_i^k-x_j^k)-b))}-\frac{\lambda_k}{T}\right)=\frac{1-p_{ij}^k}{p_{ij}^k}, \\ &\Rightarrow 1+\exp\left(\frac{1}{T}\log\frac{1+\exp(-((x_i^k-x_j^k)^T M^T M(x_i^k-x_j^k)-b))}{1+\exp(((x_i^k-x_j^k)^T M^T M(x_i^k-x_j^k)-b))}-\frac{\lambda_k}{T}\right)=\frac{1}{p_{ij}^k}.\end{aligned}$$

Based on the above, we can further get:

$$p_{ij}^k = \frac{1}{1+\exp\left(\frac{1}{T}\log\frac{1+\exp(-((x_i^k-x_j^k)^T M^T M(x_i^k-x_j^k)-b))}{1+\exp(((x_i^k-x_j^k)^T M^T M(x_i^k-x_j^k)-b))}-\frac{\lambda_k}{T}\right)}. \tag{27}$$

Combining Equation (27) with the constraint in Equation (21), we can get:

$$\sum_{i<j}\frac{2}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)\left(1+\exp\left(\frac{1}{T}\log\frac{1+\exp(-((x_i^k-x_j^k)^T M^T M(x_i^k-x_j^k)-b))}{1+\exp(((x_i^k-x_j^k)^T M^T M(x_i^k-x_j^k)-b))}-\frac{\lambda_k}{T}\right)\right)}=\hat{\pi}_k, \tag{28}$$

where λ_k denotes the Lagrange multiplier, and can be calculated by solving the root finding problem. Finally, the calculated λ_k is plugged into Equation (27) such that p_{ij}^k can be updated.

The above two steps will be iteratively conducted until the convergence criterion is satisfied. In this article, we calculate the KL-divergence of P in two consecutive iterations and set a threshold

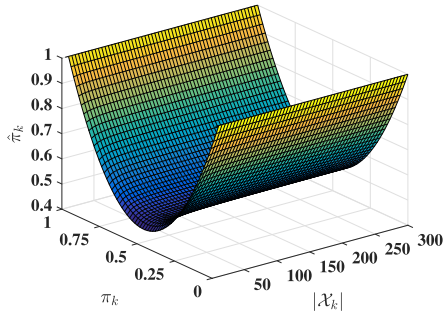


Fig. 2. $\hat{\pi}_k$ w.r.t. π_k and $|X_k|$.

ALGORITHM 1: Group-level metric learning

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 Equation (17);
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.
```

(e.g., 10^{-6}) of the KL-divergence as the convergence criterion [5]. The optimization procedure for the proposed GrML is summarized in Algorithm 1.

4.3 Theoretical Analysis for GrML

Since the associated probabilities $\{\pi_k\}_{k=1}^K$ are the only available label information, they 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 Equation (17) in Figure 2, which shows that $\hat{\pi}_k$ 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.

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:

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

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

where $\beta_1 = 3(1-\eta-\Gamma)^2/(2\Gamma(1-\Gamma)+2(1-\eta-\Gamma))$ and $\beta_2 = 3(\Gamma-\eta)^2/(2\Gamma(1-\Gamma)+2(\Gamma-\eta))$. 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. $N = O(me^{\beta m})$. □

From the above theorem, we can see 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 $\min\{1-\pi_k, \pi_k\} \leq \eta$. That is to say, 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 GrML is degraded due to the less-informative training data. Additionally, Theorem 4.1 is derived based on the assumption that all groups are of the same size $m = N/K$. When we increase the number of groups K, the value of m decreases and the proportion of informative groups become larger. On the other hand, for a fixed η that is infinitely close to 0, as we increase the value of Γ, the probability that the number of positive instances in $\mathcal{X}_k$ is no larger than $m\eta$ decreases, while the probability that the number of positive instances in $\mathcal{X}_k$ is no less than $m(1-\eta)$ increases.

4.4 Local Metric Learning from Group-wise Probabilistic Labels

The proposed mechanism GrML aims to learn a global distance metric that can be used to measure the distance between all the instance pairs. However, in some cases, GrML cannot well capture the local differences of the input feature space because this mechanism does not take into account the group localities. Thus, instead of learning a global distance metric for targeted tasks, it is more appropriate for each group to claim its own group specific distance metric so that the instances' distances can be measured from its own perspective. To achieve the goal, in this section, we extend GrML and propose a novel GrLoML, which allows us to learn a local distance metric for each disjoint group from the group-wise probabilistic labels (i.e., $\{\pi_k \in [0,1]\}_{k=1}^{K}$).

Instead of measuring instances' distances from the global view, the proposed mechanism GrLoML calculates the distance between $x_i^k \in \mathcal{X}_k$ and $x_j^k \in \mathcal{X}_k$ as:

$$d\left(x_i^k, x_j^k\right) = \left(x_i^k - x_j^k\right)^T (M^k)^T M^k \left(x_i^k - x_j^k\right), \tag{29}$$

where $M^k \in \mathbb{R}^{u \times u}$ is the local matrix for the k-th group $\mathcal{X}_k$. In many real-world applications, these group specific local distance metrics $\{(M^k)^T M^k\}_{k=1}^{K}$ can give us more flexibility. Based on the above definition for the group specific distance metric, we can rewrite the optimization problem of GrML

defined in Equation (21) as:

$$\begin{aligned}\min_{P,\{M^k\}_{k=1}^K} \mathcal{L}_3\left(P,\{M^k\}_{k=1}^K\right) &= \sum_{k=1}^K \sum_{i<j} \frac{2p_{ij}^k}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)} \log\left(1+\exp\left(-\left(x_{ijk}^T (M^k)^T M^k x_{ijk} - b\right)\right)\right) \\ &+ \sum_{k=1}^K \sum_{i<j} \frac{2(1-p_{ij}^k)}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)} \log\left(1+\exp\left(\left(x_{ijk}^T (M^k)^T M^k x_{ijk} - b\right)\right)\right) \\ &+ \sum_{k=1}^K \frac{2T}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)} \sum_{i<j} \left(p_{ij}^k \log p_{ij}^k + \left(1-p_{ij}^k\right)\log\left(1-p_{ij}^k\right)\right) + \frac{1}{2}\sum_{k=1}^K \|M^k\|_F^2, \\ \text{s.t.} \quad & \sum_{i<j} \frac{2p_{ij}^1}{|\mathcal{X}_1|(|\mathcal{X}_1|-1)} = \hat{\pi}_1, \\ & \sum_{i<j} \frac{2p_{ij}^2}{|\mathcal{X}_2|(|\mathcal{X}_2|-1)} = \hat{\pi}_2, \\ & \vdots \\ & \sum_{i<j} \frac{2p_{ij}^K}{|\mathcal{X}_K|(|\mathcal{X}_K|-1)} = \hat{\pi}_K,\end{aligned} \tag{30}$$

where $x_{ijk} = (x_i^k - x_j^k)$, and $(M^k)^T M^k$ denotes the local distance metric for the k-th group $\mathcal{X}_k$.

In the above optimization problem, we have two sets of parameters that need to be learned, i.e., $P = \{p_{ij}^k | i < j, k = 1, 2, \ldots, K\}$ and $\{M^k \in \mathbb{R}^{u\times u}\}_{k=1}^K$. The two sets of parameters in this optimization problem can be learned together by optimizing the developed objective function (i.e., Equation (30)) through a joint procedure. Specifically, we iteratively update the values of the parameters in one set to minimize the objective function while fixing the values of the parameters in another set until convergence. This two-step iterative procedure, referred to as block coordinate descent approach [3], will keep reducing the value of the objective function. To minimize the objective function in Equation (30), we iteratively conduct the following two steps:

Step 1. With an initial estimate of the similarity probabilities $P = \{p_{ij}^k | i < j, k = 1, 2, \ldots, K\}$, we first update the local matrix M^k for the group $\mathcal{X}_k$ by minimizing the following objective function:

$$\begin{aligned}\min_{M^k \in \mathbb{R}^{u\times u}} \quad \mathcal{L}_4(M^k) &= \sum_{k=1}^K \sum_{i<j} \frac{2p_{ij}^k}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)} \log\left(1+\exp\left(-\left(x_{ijk}^T (M^k)^T M^k x_{ijk} - b\right)\right)\right) \\ &+ \sum_{k=1}^K \sum_{i<j} \frac{2(1-p_{ij}^k)\log(1+\exp((x_{ijk}^T (M^k)^T M^k x_{ijk} - b)))}{|\mathcal{X}_k|(|\mathcal{X}_k|-1)} + \frac{\|M^k\|_F^2}{2},\end{aligned} \tag{31}$$

where $x_{ijk} = (x_i^k - x_j^k)$. By deriving the local matrix using the above equation for each group, we can obtain the collection of local matrices $\{M^k \in \mathbb{R}^{u\times u}\}_{k=1}^K$ which minimize $\mathcal{L}_3(P, \{M^k\}_{k=1}^K)$ with fixed $P = \{p_{ij}^k | i < j, k = 1, 2, \ldots, K\}$.

Step 2. In this step, the local matrix of each group is fixed, and we update the similarity probabilities $P = \{p_{ij}^k | i < j, k = 1, 2, \ldots, K\}$ by solving the following optimization problem:

$$\min_{P} \quad \mathcal{L}_5(P) = \mathcal{L}_3\left(P, \{M^k\}_{k=1}^K\right) - \sum_{k=1}^{K} \Delta_k \left(\sum_{i<j} \frac{2p_{ij}^k}{|\mathcal{X}_k|(|\mathcal{X}_k| - 1)} - \hat{\pi}_k \right), \tag{32}$$

where $\{\Delta_k\}_{k=1}^K$ are the introduced Lagrange multipliers. In this step, we fix the local matrices $\{M^k\}_{k=1}^K$ for the groups, and compute the similarity probabilities that jointly minimize the objective function subject to the regularization constraints. Note that we start with an initial estimate of the similarity probabilities and then iteratively conduct the local matrix update step and the similarity probability update step until convergence. The convergence of the proposed optimization solution for GrLoML can be guaranteed according to the proposition on the convergence of block coordinate descent [3].

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 InML and InLoML in Section 5.2 and Section 5.3, respectively. The experimental results for GrML are shown in Section 5.4.

5.1 Experimental Setup

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

Datasets for the proposed instance-level mechanisms. In order to evaluate the performance of the proposed instance-level mechanisms (i.e., InML and InLoML), we adopt the following real-world datasets which are grouped into three categories:

- *Regression Datasets.* We adopt five UCI datasets[1] (i.e., Concrete, Housing, Energy, Airfoil Self-Noise, and Yacht Hydrodynamics) that are used in the regression task. Specifically, the Concrete, Housing, and Energy datasets are used to evaluate the performance of InML, and the Airfoil Self-Noise and Yacht Hydrodynamics datasets are used to evaluate the performance of InLoML. 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[2] (i.e., Cancer, Stock, and Machine) which come with multiple classes and full-order relations among classes. For each dataset, we generate the associated probabilities (i.e., $\{c_i\}_{i=1}^N$) 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

[1] https://archive.ics.uci.edu/ml/datasets.html.
[2] http://www.gagolewski.com/resources/data/ordinal-regression/.

Table 1. The Statistics of the Adopted Datasets

Dataset	#Samples	#Dimensions	Dataset	#Samples	#Dimensions
Concrete	1,030	8	Machine	199	6
Housing	506	13	Movie	5,000	1,199
Energy	768	8	Music	700	123
Airfoil Self-Noise	1,503	6	Ionosphere	351	34
Yacht Hydrodynamics	308	7	Heart	303	23
Cancer	194	32	Diabetes	768	9
Stock	950	9	-	-	-

example, the Cancer dataset contains six ordinal classes $\{1, 2, 3, 4, 5, 6\}$. The class labels are transformed to the probabilities $\{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ after the normalization. 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, two crowdsourced datasets, i.e., the movie review dataset and the music genre dataset [25], are adopted. For the movie review dataset, the task of the workers is to judge whether the review of a movie is positive or negative and it contains 5,000 movies. In the music genre dataset, the workers need to judge whether a piece of music is rock (positive) or non-rock (negative) and there are 700 pieces of music. 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 article) over the probabilities to generate the binary label for each instance.

Datasets for the group-level mechanisms. As for the proposed global-level mechanisms (i.e., GrML and GrLoML), we evaluate their performance on three popular datasets: the Ionosphere dataset, the Heart dataset, and the Diabetes dataset [50], which are widely used in the settings with group probabilities. The details of the adopted datasets are described in Table 1.

Baseline Methods. In this article, we compare the proposed mechanisms with the following state-of-the-art metric-learning methods. Geometric Mean Metric Learning (*GMML*) [51] addresses the task of learning a symmetric positive definite matrix by formulating it as a smooth, strictly convex optimization problem. The formulation can be viewed as an optimization problem on the Riemannian manifold of symmetric positive definite matrices. Information Theoretic Metric Learning (*ITML*) [6] is an information-theoretic approach that aims to learn a Mahalanobis distance function, and the authors formulate the problem as minimizing the differential relative entropy between two multivariate Gaussians under the constraints on the distance function. Large Margin Nearest Neighbor (*LMNN*) [44] aims to learn a Mahanalobis distance metric for k-nearest neighbor classification by semidefinite programming. The distance metric is trained with the goal that the k-nearest neighbors always belong to the same class while instances from different classes are separated by a large margin. *LowRank* [52] is a similarity algorithm based on pairwise constraints, which aims to deal with the data with noise and redundancy. This algorithm is implemented by encoding a low-rank structure to the distance metric-learning process. *R2LML* [14] is a local distance metric-learning method based on a conical combination Mahalanobis metric and pairwise similarities between the data. Its formulation allows for controlling 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.

Table 2. The Accuracy of InML Under Different Training Dataset Sizes

Training size	Methods	Regression datasets			Ordinal datasets			Crowdsourced datasets	
		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

5.2 Experiments for InML

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

Performance comparison. We first evaluate the performance of InML under different training dataset sizes. In this experiment, 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 randomly select some instances from the testing set and add them into the training dataset. Table 2 reports the classification accuracy of InML and that of the baseline methods. Note that in this article, 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 [14, 44, 51]. Specifically, to determine the class label of a given testing instance, we first need to calculate the distances between it and all the instances in the training set according to the learned distance metric learning model. Then we derive its corresponding ranked list of the training instances sorted by the calculated distances. Finally, the class label of this testing instances can be determined by majority voting over its top-$\mathbb{K}$ closest (nearest) training instances. In this article, unless otherwise specified, the value of the target neighbors $\mathbb{K}$ is set as 5. The results in Table 2 show that InML performs much better than the baselines in all cases. This is mainly because InML can extract more information from instance probabilities, while the baselines can only derive limited knowledge using the class labels.

Convergence. Next, we evaluate the convergence of InML through calculating the objective value in each iteration. The evolution of the objective value on the Concrete dataset is reported in Figure 3, from which we can see 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.

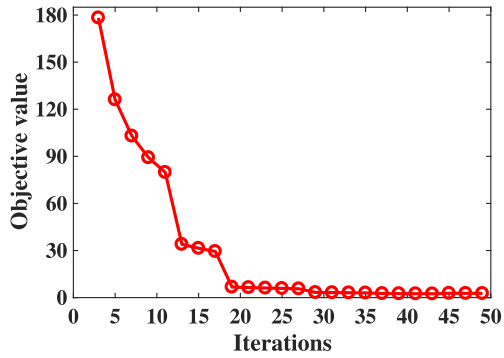


Fig. 3. Convergence of InML on the Concrete dataset.

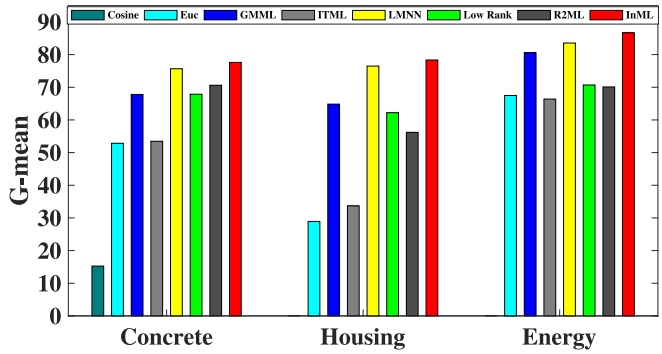


Fig. 4. G-mean on unbalanced datasets.

Performance on unbalanced datasets. Here, we evaluate the performance of the proposed mechanism InML on the datasets which are unbalanced, i.e., there 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. Figure 4 shows the results, 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.

Robustness. In real-world applications, the instance-wise probabilistic labels may be noisy due to various reasons [24]. 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 the following 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

Table 3. The Accuracy of 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

Table 4. The Accuracy of InLoML Under Different Training Sizes

	Airfoil			Yacht Hydrodynamics		
Methods	20	40	60	20	40	60
InLoML	**0.7128**	**0.7251**	**0.7507**	**0.8084**	**0.8852**	**0.8942**
InML	0.7036	0.7136	0.7348	0.8039	0.8710	0.8816
Euc	0.6888	0.6941	0.7014	0.6888	0.6941	0.7114

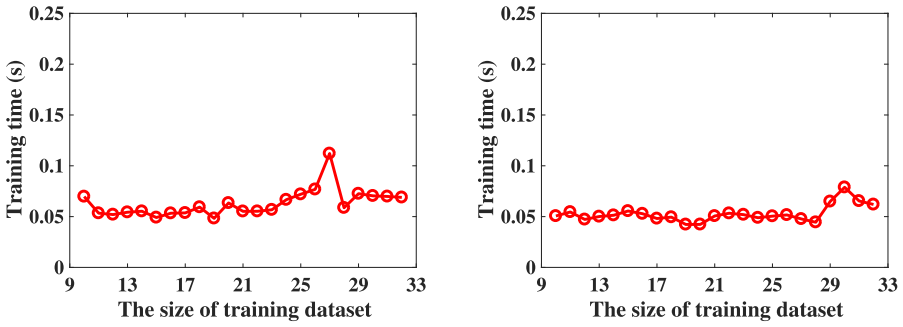
shows the accuracy of all the methods on the Concrete, Stock, and Machine datasets. The results in this table show that InML significantly outperforms the baseline methods in all cases. More importantly, compared with the baselines, InML performs more stably when the level of the noise varies, and this 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 InLoML

In this section, we evaluate the performance of of the proposed instance-level local metric-learning mechanism InLoML on the Airfoil Self-Noise and Yacht Hydrodynamics datasets. The experiments are conducted for 10 times and we report the average results. In the following, unless otherwise stated, the number of the basis elements is set as 10 (i.e., $S = 10$) without loss of generality.

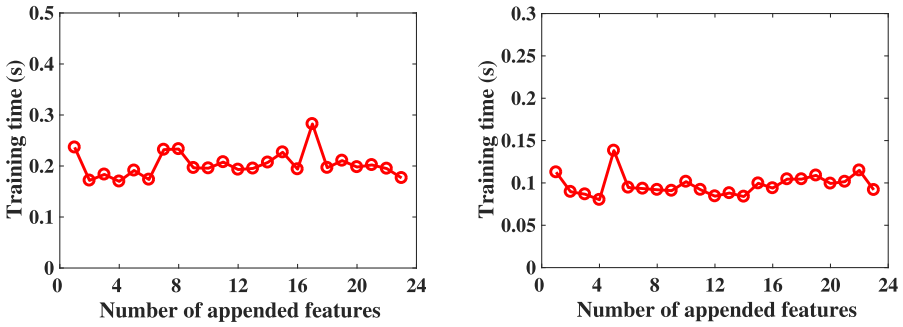
Performance comparison. In this experiment, we take InML and Euclidean as the baseline methods and then compare the accuracy of the proposed InLoML with that of the two baselines. For each dataset, we first randomly select half of all instances as the testing set, and then randomly extract the training dataset from the remaining instances. Here we consider three cases where the number of the training instances is set as 20, 40, and 60, respectively. The regularization parameter Υ is set as $1e-5$. The experimental results are reported in Table 4. From this table, we can see that the proposed mechanism InLoML achieves the best performance in all cases, while InML and Euc have relatively poor performance. The reason is that InLoML learns a set of local metrics instead of a single global distance metric and can well capture the local data differences. Additionally, the experimental results also show that the performance becomes better when the number of instances in the training set increases.

The effect of the training size on the computational cost. In this experiment, we investigate the effect of the training size (i.e., N) on the computational cost of InLoML. Specifically, for a given dataset, we evaluate the training time of InLoML under different training sizes. Figure 5 reports



(a) Airfoil dataset (b) Yacht Hydrodynamics dataset

Fig. 5. The training time of InLoML under different training sizes.

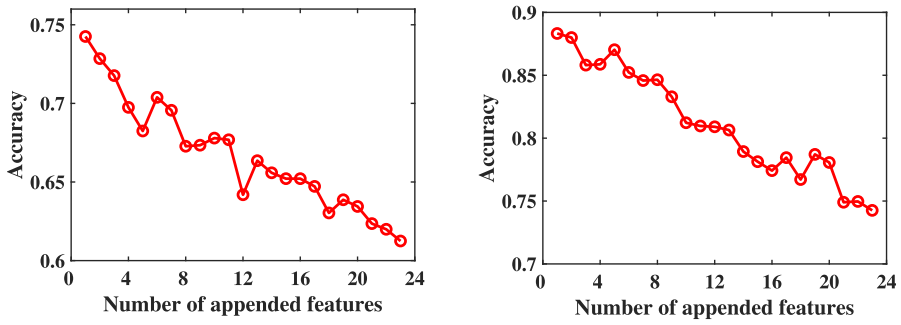


(a) Airfoil dataset (b) Yacht Hydrodynamics dataset

Fig. 6. The training time of InLoML under different number of added feature dimensions.

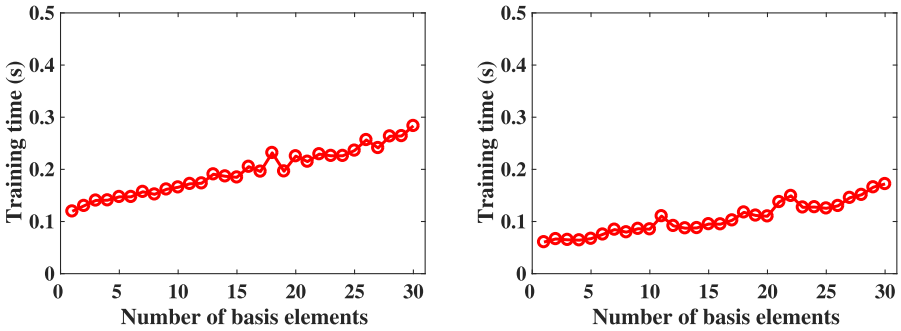
the average results when the training size varies from 10 to 32 on the Airfoil Self-Noise dataset and the Yacht Hydrodynamics dataset. As we can see, the training size exerts negligible influence on the computational cost of InLoML. The results also verify that in our theoretical analysis in Section 3.3, the number of parameters needed to be learned is independent of the training set size N. Thus, in practical applications, we can use a moderately large training set to learn a good learning model with high efficiency.

The effect of the feature dimension. Next, we measure the effect of the feature dimension (i.e., u) on the computational cost and accuracy of the proposed InLoML. In this experiment, the training size is set as 40 (i.e., $N = 40$). For each dataset, we append different numbers of Gaussian noisy features to expend its feature dimensions. More specifically, we vary the number of appended noisy features from 1 to 23. Figure 6 shows the evolution of the training time (in seconds) with respect to the number of appended features. The experimental results in this figure show that the training time remains roughly stable when we increase the number of appended features. This is also in accordance with the complexity analysis in Section 3.3, which shows that the computational cost is independent of the feature dimension (i.e., u). Thus, the proposed InLoML is very efficient to deal with the datasets with large dimensions. Additionally, we report the accuracy results of the proposed InLoML under different input feature dimensions in Figure 7. From this figure, we can observe that as we increase the number of the appended noisy features, the accuracy of the proposed InLoML decreases. The reason is that the appended noisy features can hide the relationship between the targeted metric learning task and the relevant input features.



(a) Airfoil dataset (b) Yacht Hydrodynamics dataset

Fig. 7. The accuracy of InLoML under different number of added feature dimensions.



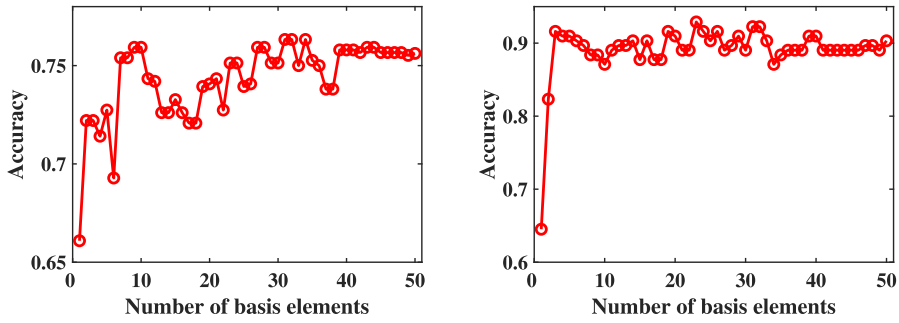
(a) Airfoil dataset (b) Yacht Hydrodynamics dataset

Fig. 8. The training time of InLoML under different number of basis elements.

The effect of the basis elements. In this experiment, we evaluate the effect of the number of the basis elements (i.e., S) on the training time and accuracy of InLoML. In this experiment, we still set the training sizes of the two regression datasets as 40 (i.e., $N = 40$). Figure 8 reports the evolution of the training time of InLoML with respect to the number of basis elements. Here, for each dataset, we vary the number of basis elements from 1 to 30. As it can be observed in this figure, the training time of the proposed InLoML has a linear growth when we increase the number of basis elements. This observation can also be derived from the computational analysis in Section 3.3, which shows that the computational complexity is linearly proportional to the number of basis elements (i.e., S). Figure 9 shows the accuracy of InLoML under different numbers of the basis elements. Here, the number of the basis elements is varied from 1 to 50. From this figure, we can see that with the increase of the number of basis elements, the classification accuracy of the proposed InLoML first increases and then tends to be stable. This also accords with our intuition that with sufficient basis elements, we can derived a well-trained local metric learning model.

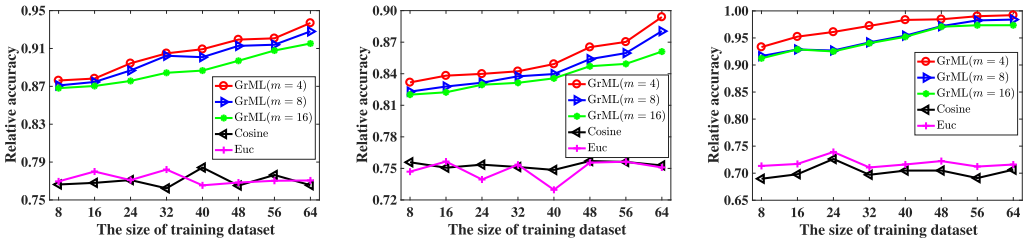
5.4 Experiments for GrML

In this section, we evaluate the performance of GrML on three real-world datasets [50] (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



(a) Airfoil dataset (b) Yacht Hydrodynamics dataset

Fig. 9. The accuracy of InLoML under different number of basis elements.



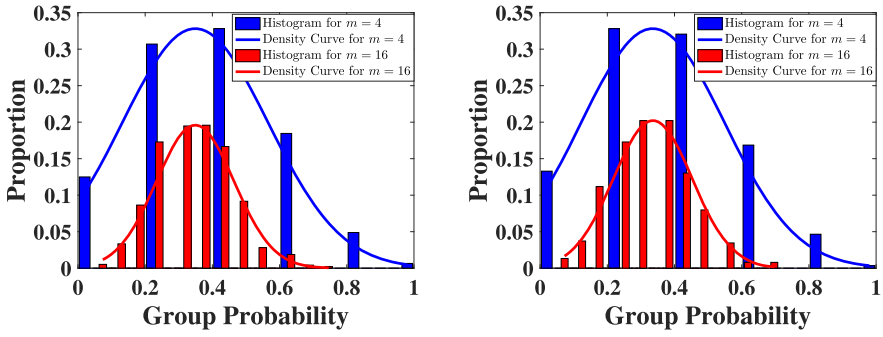
(a) Diabetes dataset (b) Heart dataset (c) Ionosphere dataset

Fig. 10. Relative accuracy of the group-level mechanism w.r.t. the size of training dataset.

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 article) that has full access to the deterministic labels. And the accuracy calculated here is based on the predicted similarity labels of the testing instance pairs, which are calculated by using the learned distance metric.

Performance comparison. We first evaluate the performance of GrML 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. In this experiment, we choose the values of $m = 4, 8$, and 16. Figure 10 reports the relative accuracy of GrML and that of the baseline methods on the three datasets. 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 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 10 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



(a) Diabetes dataset (b) Ionosphere dataset

Fig. 11. The distribution of the group-wise probabilities.

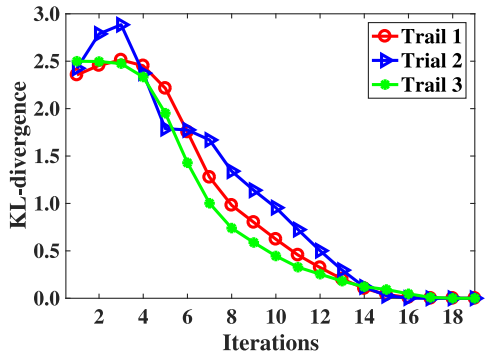


Fig. 12. Convergence of the group-level mechanism on the Ionosphere dataset.

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 11 shows the histograms of the two datasets, and each solid line represents a fit to the exponential distribution. As Figure 11 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 10. From Figure 10, 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.

Convergence. We also 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. The results on the Ionosphere dataset are reported in Figure 12. 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. The results show that 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.

6 RELATED WORK

In the past few years, we have witnessed a great increase in the number of metric-learning works [1, 2, 6, 9, 11–14, 16–19, 21, 22, 27, 31, 32, 37, 41, 42, 44, 47, 51, 52]. Existing metric-learning works can be organized into the following two categories: global metric learning and local metric learning.

Global metric learning. Global metric-learning mechanisms [1, 2, 6, 9, 11, 14, 16–19, 21, 31, 41, 44, 47, 51, 52] aim to learn a single (or global) distance metric that can measure the similarity of different instances from the global view. Traditional supervised global metric-learning works [11, 14, 19, 31, 41, 44, 51, 52] optimize the distance metrics with the assumption that the available training dataset is fully labeled. The authors in [11, 14, 44, 51, 52] address the binary datasets and use the associated binary labels to generate a set of constraints which are then used as the supervised information. [41] proposes a metric-learning method under the scenarios where some class labels in the training dataset are mislabeled. The work in [31] deals with the metric-learning problem where the training dataset has multiple class values. The authors in [19] present an algorithm to learn distance metrics for multi-label problems where each training instance in the training set is associated with a set of class labels. Additionally, there are some other global metric-learning works [1, 2, 6, 21, 47] which address the semi-supervised metric-learning problems. Following the entropy regularization, the authors in [21] present a metric-learning method which maximizes the entropy of the probability on labeled isntances and minimizes it on unlabeled instances. Meanwhile, there exist some semi-supervised global metric-learning works [1, 2, 6, 47] that use a set of pairwise similarity and dissimilarity constraints as the semi-supervised information to address different metric-learning problems. Xing et al. [47] propose a semi-supervised global metric-learning method by using these pairwise similarity and dissimilarity constrains. Following this work, there are several emerging metric-learning works [1, 2, 6] which study the metric-learning problems by exploiting the given relevant constraints. And, there are some other global metric-learning works [9, 16–18] that address the multiple instance problems where the given training dataset is provided as a set of labeled bags. The goal of these global metric-learning works is to learn a distance metric, which pushes bags that do not share any label apart, and makes bags that share a label closer [9, 17].

Local metric learning. Although global metric learning provides an effective way to measure the distances among all the instance pairs, it fails to take into account the local differences of the input feature space. Thus, the learned global distance metric may not fit well the distance over the data manifold. To address this problem, different local metric-learning works [13, 22, 27, 32, 42] have been proposed, which allow the distance metrics to vary across the feature space to capture the semantic distance much better. The authors in [22] aim to take the advantage of information from parametric generative models in the context of metric learning. Specifically, they focus on the bias in the information-theoretical error, and find an appropriate local metric that maximally reduces the bias based upon knowledge from generative models. Wang et al. [42] propose a parametric local metric-learning method where a Mahalanobis distance metric is learned for each training instance. More precisely, they parametrize the distance metric matrix of each instance as a linear combination of basis metric matrices of a small set of anchor points, and this parametrization is derived from an error bound on local metric approximation. [27] proposes a coordinated local metric-learning method, which can learn a set of local Mahalanobis metrics and integrate them in a global representation. The authors in [13] propose a method of learning local similarity-aware deep feature embeddings in an end-to-end manner. The proposed method can adaptively measure the local feature similarity in a heterogeneous space, and the learned local-adaptive similarity metric can be exploited to search for high-quality hard samples in local neighborhood to

facilitate a more effective deep embedding learning. [32] proposes a new method for sparse compositional local Mahalanobis distance metric learning, and the proposed method learns a set of distance metrics which are composed of local and global components.

However, the above discussed metric-learning works fail to deal with the probabilistic class labels. In practice, learning from such probabilistic information is of great importance [50]. Some works in other fields [8, 15, 23, 24, 26, 29, 30, 50] also consider how to learn models from the probabilistic labels. However, the problem settings in these papers are quite different from ours. The authors in [15] present learning models for the class ratio estimation problem, which takes an unlabeled set of instances as input and predicts the proportions of instances in the set belonging to different classes. [23] aims to learn a supervised classifier when only label proportions for bags of observations are known. The article [24] presents an approach to learn a classifier from group probabilities based on support vector regression and the idea of inverting a classifier calibration process. The authors in [26] propose a method called proportion-SVM, which explicitly models the latent unknown instance labels together with the known group label proportions in a large-margin framework. By introducing pinball loss, [29] presents a method for learning from label proportions, which is built upon a recursive algorithm to alternatively predict the unknown labels and minimize the objective function. [30] proposes an end-to-end learning from label proportions model based on convolutional neural network called IDLLP, which employs the idea of inverting a classifier calibration process to learn a classifier from bag probabilities. [8] focuses on the binary label setting and formalizes a model for learning a hypothesis class by only examples drawn from a distribution and the proportion of them receiving each label, with the goal of finding a hypothesis that matches these statistics on the underlying distribution.

In our preliminary work [10], we study how to effectively learn the distance metric from datasets that contain probabilistic information, and then propose two novel metric-learning mechanisms (InML and GrML). However, the two proposed global metric-learning mechanisms (i.e., InML and GrML) can only learn a global distance metric from the given training dataset and they fail to capture the local differences of the input feature space. In this article, we extend InML and GrML, and propose two local metric-learning mechanisms, based on which we can learn a set of local distance metrics that can well capture the local differences of the input feature space. For the proposed instance-wise global method InML, we conduct experiments to show that it works well not only on the balanced dataset but also on the unbalanced dataset. Although there exist some works that also address the unbalanced dataset [43, 53], their problem settings are quite different from ours. The authors in [43] study the problem of partial label learning and aim to induce a multi-class classifier from training instances where each of them is associated with a set of candidate labels, among which only one is valid. [53] investigates the problem of online active learning and proposes an online adaptive active learning algorithm to handle imbalanced datastream under limited query budgets. However, the goal of our proposed InML is to learn a distance metric function that can effectively calculate the similarity degree of different instance pairs. Additionally, the input of the proposed InML is different from that of the methods in [43, 53]. Specifically, the proposed InML takes the probabilistic information instead of the deterministic class labels as the input, while [43, 53] only utilize the deterministic class labels.

7 CONCLUSIONS

In this article, we first propose an InML, based on which, a global distance metric can be learned directly from the given training dataset with instance-wise probabilistic labels. To well capture the local differences of the input feature space, we then extend InML and propose a novel InLoML, which aims to learn a set of instance specific local metrics from the instance-wise probabilities. For the cases where the datasets are associated with group-wise probabilistic labels, we

first design a GrML, which can learn the global distance metric directly from the group-wise probabilistic labels with high accuracy. Furthermore, we also extend GrML and design an effective GrLoML, based on which we can learn a set of group specific local distance metrics from the given group-wise probabilities. Both theoretical analysis and extensive experiments on real-world datasets are provided to demonstrate the advantages of the proposed metric-learning mechanisms.

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] Dimitri P. Bertsekas. 1999. *Nonlinear Programming*. Athena Scientific, Belmont.

[4] Qiong Cao, Zheng-Chu Guo, and Yiming Ying. 2016. Generalization bounds for metric and similarity learning. *Machine Learning* 102, 1 (2016), 115–132.

[5] 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.

[6] 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.

[7] John Duchi and Yoram Singer. 2009. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research* 10, Dec (2009), 2899–2934.

[8] Benjamin Fish and Lev Reyzin. 2017. On the complexity of learning from label proportions. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 1675–1681.

[9] 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.

[10] Mengdi Huai, Chenglin Miao, Yaliang Li, Qiuling Suo, Lu Su, and Aidong Zhang. 2018. Metric learning from probabilistic labels. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1541–1550.

[11] 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*. 270–278.

[12] Mengdi Huai, Hongfei Xue, Chenglin Miao, Liuyi Yao, Lu Su, Changyou Chen, and Aidong Zhang. 2019. Deep metric learning: The generalization analysis and an adaptive algorithm. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2535–2541.

[13] Chen Huang, Chen Change Loy, and Xiaoou Tang. 2016. Local similarity-aware deep feature embedding. In *Proceedings of the Advances in Neural Information Processing Systems*. 1262–1270.

[14] Yinjie Huang, Cong Li, Michael Georgiopoulos, and Georgios C. Anagnostopoulos. 2013. Reduced-rank local distance metric learning. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 224–239.

[15] 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.

[16] 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.

[17] 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*.

[18] Dewei Li and Yingjie Tian. 2016. Multi-view metric learning for multi-instance image classification. *Arxiv Preprint Arxiv:1610.06671* (2016).

[19] Weiwei Liu and Ivor W. Tsang. 2015. Large margin metric learning for multi-label prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 2800–2806.

[20] Colin McDiarmid. 1989. On the method of bounded differences. *Surveys in Combinatorics* 141, 1 (1989), 148–188.

[21] 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.

[22] Yung-Kyun Noh, Byoung-Tak Zhang, and Daniel D. Lee. 2010. Generative local metric learning for nearest neighbor classification. In *Proceedings of the Advances in Neural Information Processing Systems*. 1822–1830.

[23] Giorgio Patrini, Richard Nock, Paul Rivera, and Tiberio Caetano. 2014. (Almost) no label no cry. In *Proceedings of the Advances in Neural Information Processing Systems*. 190–198.

[24] 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*. 307–315.
[25] Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. 2014. Gaussian process classification and active learning with multiple annotators. In *Proceedings of the International Conference on Machine Learning*. 433–441.
[26] Stefan Rueping. 2010. SVM classifier estimation from group probabilities. In *Proceedings of the 27th International Conference on Machine Learning (ICML'10)*. 911–918.
[27] Shreyas Saxena and Jakob Verbeek. 2015. Coordinated local metric learning. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 127–135.
[28] Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.
[29] Yong Shi, Limeng Cui, Zhensong Chen, and Zhiquan Qi. 2019. Learning from label proportions with pinball loss. *International Journal of Machine Learning and Cybernetics* 10, 1 (2019), 187–205.
[30] Yong Shi, Jiabin Liu, and Zhiquan Qi. 2018. Inverse convolutional neural networks for learning from label proportions. In *Proceedings of the 2018 IEEE/WIC/ACM International Conference on Web Intelligence*. IEEE, 643–646.
[31] Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Proceedings of the Advances in Neural Information Processing Systems*. 1857–1865.
[32] Joseph St. Amand and Jun Huan. 2017. Sparse compositional local metric learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1097–1104.
[33] 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.
[34] Tao Sun, Dan Sheldon, and Brendan O'Connor. 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 '17)*. IEEE, 445–454.
[35] Qiuling Suo, Fenglong Ma, Ye Yuan, Mengdi Huai, Weida Zhong, Jing Gao, and Aidong Zhang. 2018. Deep patient similarity learning for personalized healthcare. *IEEE Transactions on Nanobioscience* 17, 3 (2018), 219–227.
[36] 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'17)*. IEEE, 811–816.
[37] Qiuling Suo, Weida Zhong, Fenglong Ma, Yuan Ye, Mengdi Huai, and Aidong Zhang. 2018. Multi-task sparse metric learning for monitoring patient similarity progression. In *Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM'18)*. IEEE, 477–486.
[38] Tian Tian and Jun Zhu. 2015. Max-margin majority voting for learning from crowds. In *Proceedings of the Advances in Neural Information Processing Systems*. 1621–1629.
[39] Joel A. Tropp et al. 2015. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning* 8, 1–2 (2015), 1–230.
[40] Di Wang, Mengdi Huai, and Jinhui Xu. 2018. Differentially private sparse inverse covariance estimation. In *Proceedings of the 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP'18)*. IEEE, 1139–1143.
[41] 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.
[42] Jun Wang, Alexandros Kalousis, and Adam Woznica. 2012. Parametric local metric learning for nearest neighbor classification. In *Proceedings of the Advances in Neural Information Processing Systems*. 1601–1609.
[43] Jing Wang and Min-Ling Zhang. 2018. Towards mitigating the class-imbalance problem for partial label learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2427–2436.
[44] Kilian Q. Weinberger, John Blitzer, and Lawrence K. Saul. 2006. Distance metric learning for large margin nearest neighbor classification. In *Proceedings of the Advances in Neural Information Processing Systems*. 1473–1480.
[45] 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.
[46] Lin Xiao. 2010. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research* 11, Oct (2010), 2543–2596.
[47] 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 *Proceedings of the Advances in Neural Information Processing Systems*. 521–528.
[48] Liuyi Yao, Sheng Li, Yaliang Li, Mengdi Huai, Jing Gao, and Aidong Zhang. 2018. Representation learning for treatment effect estimation from observational data. In *Proceedings of the Advances in Neural Information Processing Systems*. 2633–2643.

[49] Liuyi Yao, Yaliang Li, Yezheng Li, Hengtong Zhang, Mengdi Huai, Jing Gao, and Aidong Zhang. 2019. DTEC: Distance transformation based early time series classification. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. 486–494.

[50] Felix X. Yu, Dong Liu, Sanjiv Kumar, Tony Jebara, and Shih-Fu Chang. 2013. SVM for learning with label proportions. In *Proceedings of the 30th International Conference on International Conference on Machine Learning*.

[51] Pourya Zadeh, Reshad Hosseini, and Suvrit Sra. 2016. Geometric mean metric learning. In *Proceedings of the International Conference on Machine Learning*. 2464–2471.

[52] 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'16)*. IEEE.

[53] Yifan Zhang, Peilin Zhao, Jiezhang Cao, Wenye Ma, Junzhou Huang, Qingyao Wu, and Mingkui Tan. 2018. Online adaptive asymmetric active learning for budgeted imbalanced data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2768–2777.

[54] Dengyong Zhou, Qiang Liu, John Platt, and Christopher Meek. 2014. Aggregating ordinal labels from crowds by minimax conditional entropy. In *Proceedings of the International Conference on Machine Learning*. 262–270.

Received December 2018; revised July 2019; accepted September 2019