Local Discriminative Distance Metrics Ensemble Learning

Yang Mu[a], Wei Ding[a], Dacheng Tao[b]

[a]*University of Massachusetts Boston,MA,02125, US*
[b]*University of Technology, Sydney, NSW, 2007, AU*

Abstract

The ultimate goal of distance metric learning is to incorporate abundant discriminative information to keep all data samples in the same class close and those from different classes separated. Local distance metric methods can preserve discriminative information by considering the neighborhood influence. In this paper, we propose a new local discriminative distance metrics (LDDM) algorithm to learn multiple distance metrics from each training sample (a focal sample) and in the vicinity of that focal sample (focal vicinity), to optimize local compactness and local separatiblity. Those locally learned distance metrics are used to build local classifiers which are aligned in a probabilistic framework via ensemble learning. Theoretical analysis proves the convergence rate bound, the generalization bound of the local distance metrics and the final ensemble classifier. We extensively evaluate LDDM using synthetic datasets and large benchmark UCI datasets.

Keywords: Local learning, distance metrics learning.

1. Introduction

Distance metric learning plays a crucial role in metric-related pattern recognition tasks including K-means, K-Nearest Neighbors, and kernel-based algorithms such as SVMs [19, 4, 5, 23, 25]. The learning task falls into two categories: unsupervised and supervised distance metric learning. In supervised distance metric learning [21], the ultimate goal is to incorporate the abundant discriminative information in distance metric learning to keep all the data samples in the same class close and those from different classes separated. Zhang *et al.* have shown that a distance metric incorporating discriminative information from labeled data usually outperforms the standard Euclidean distance in classification tasks [24].

Supervised distance metric learning can be further divided into global and local distance metric learning. The first step is to learn a global distance metric from training data to satisfy all pairwise constraints simultaneously [26, 20]. The most representative work is the Xing's algorithm [20], which learns a distance metric on a global scale that minimizes the distance between data pairs according to the equivalence constraints while separating data pairs from each other according to the inequivalence constraints. If data classes exhibit multimodal distributions, equivalence or inequivalence constraints from different data distributions may conflict with each other. Therefore, it is difficult to satisfy all the constraints on a global level. Local distance metric learning is introduced to cope with this problem by considering the locality of data distribution [15, 18, 9]. These local algorithms only consider neighboring pairwise constraints and avoid adopting those conflicting constraints.

All aforementioned approaches are all trying to learn a single metric on all data samples. The deficiencies of learning a single metric include: 1) a single metric is likely inappropriate for all training samples; 2) a single local metric may be easily influenced by noisy samples; 3) a single global metric cannot deal with the multimodal distribution problem. It is recommended to learn multiple metrics to describe different localities of training samples [10, 18, 6].

In this paper, we propose a multiple distance metric approach, the Local Discriminative Distance Metrics (LDDM) algorithm, from a new perspective. We learn a set of local discriminative distance metrics from each training sample (denoted as a focal sample), and in the vicinity of that focal sample (denoted as the focal vicinity), to effectively optimize local compactness and local separatiblity. Those locally learned distance metrics are used to build local classifiers which are aligned in a probabilistic framework via ensemble learning. The LDDM algorithm makes up the deficiency of the existing multiple distance metric methods and differs from them in the following aspects: 1) where DANN [10] uses the optimization process of LDA, LDDM does not need to calculate the inverse of a matrix and hence avoids the small sample size problem; 2) unlike DANN and ADAMENN [6], LDDM does not have the adaptive iterative process, and guarantees a closed form solution; 3) once the training model is learned, the test computation complexity is $O(n)$ for LDDM, while DANN and ADAMENN have the same computation complexity in training and test process (i.e., DANN has computation complexity of $O(nd^3)$, where n is the dataset size and d is the feature dimension); 4) mLMNN [18] requires disjoint clusters on training samples to train multiple distance metrics, while LDDM does not require clusters for training samples.

The proposed LDDM method consists of three key components.

1) **Focal vicinity extraction**. For each training sample, we extract a focal vicinity to learn a local discriminative distance metric to ensure all the similar/dissimilar samples fall into/exclude from the vicinity of each focal sample. This focal vicinity consists of the focal sample, its same class neighborhood, and its dissimilar class neighborhood.

2) **Local distance metrics learning**. The LDDM algorithm divides the training space into a set of focal vicinities and learns a local optimized distance metric at each focal vicinity to keep training samples either close to or distant from a focal sample.

3) **Local classifier ensemble**. We utilize classifier ensemble learning to build upon locally learned distance metrics for the final prediction. To overcome the over-fitting problem, the base classifiers of the ensemble are aligned in a probabilistic framework to form an adjustable model according to each test sample to significantly reduce the influence of noise samples.

We theoretically analyze the correctness of the proposed LDDM method and explain why multiple-distance-metrics approaches should perform superiorly to single-distance-metric approaches while dealing with noisy datasets. We define a new concept called local-domain-based VC-dimension and prove the convergence rate bound for a local distance metric, the risk bound of each local distance metric, and the risk bound of the ensemble local classifiers. We extensively evaluate the LDDM algorithm with experiments on synthetic datasets and the real-world UCI datasets.

The rest of the paper is organized as follows: related work is discussed in Section 2. Section 3 explains how to learn local discriminative distance metrics. The ensemble methods are discussed in detail in Section 4, and theoretical analysis is provided in Section 5. Experimental studies are discussed in Section 6. Section 7 concludes the paper.

2. Related Works

In general, supervised distance metrics can be categorized into global and local approaches. Local approaches are further classified as single-metric and multiple-metric approaches. Our proposed LDDM method is in the family of multiple local distance metrics. Figure 1 briefly illustrates the categories of the state-of-the-art distance metric learning methods and their relationships to the proposed LDDM algorithm.

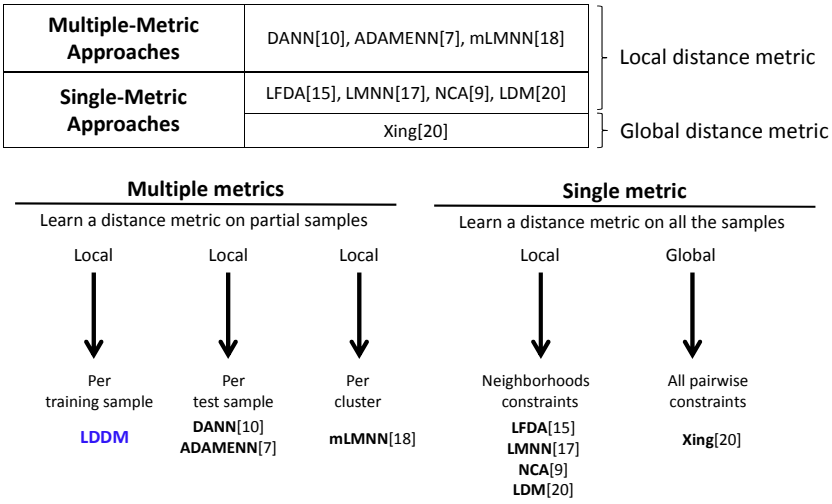

Figure 1: The categories of the state-of-the-art distance metric methods and their relationships to the proposed LDDM method.

The global approaches try to keep all the samples in the same class close together while separate those from different classes. Xing's algorithm [20] is the most representative global method which optimizes these equivalence and inequivalence constraints simultaneously using convex optimization. The advantage of using global approaches is that it may easily capture the distributions of different classes if all samples in the same class obey the same distribution, however, global approaches may fail to learn the appropriate distance metrics if data exhibits a multimodal distribution.

Local approaches use neighborhood information to deal with the multimodal distribution problem. Local Fisher Discriminant Analysis (LFDA) [15] assigns higher weights to the neighborhood pairwise constraints according to locality information. Large Margin Nearest Neighbor (LMNN) [18] utilizes neighborhood constraints to learn a distance metric with a large margin for inequivalence constraints. Yang *et al*. proposed a probability approach [22] to optimize local pairwise constraints. Goldberger *et al*. [9] utilize a stochastic variant of KNN classification to compute the expected leave-one-out classification error. All neighborhood-based single local distance metric methods discussed above may mistakenly consider neighboring noise samples and easily overlook the entire structure of the training samples.

If one single distance metric is used to describe the whole training space, the tradeoff between the learning system and the number of samples may limit the

learning performance [16]. Many methods adopt multiple distance metrics instead of a single distance metric. Weinberger *et al*. [18] proposed the multiple Large Margin Nearest Neighbor metrics (mLMNN) method to cluster training samples and then apply different distance metrics to measure different clusters. Discriminant Adaptive Nearest Neighbor classification (DANN) [10] performs like Liner Discriminant Analysis (LDA) in each local distance metric for each test sample. Domeniconi *et al*. proposed the local ADAptive MEtric Nearest-Neighbor algorithm (ADAMENN) [6], which learns a local distance metric for each test sample to have neighborhoods elongating along less relevant feature dimensions and constricting along most influential ones. Our proposed LDDM method also belongs to this category. LDDM differs from all above by learning a distance metric on each training sample and reducing noise samples' influence with an ensemble approach. Another relevant work is that Frome *et al.* [8] proposed a patched-based distance on image classification which also trains an SVM classifier on each training sample. This method is specifically designed for different image feature types and differs from all the other distance metric based approaches described in Figure 1. Global and local distance metric approaches have been successfully employed in Learning Vector Quantization. Schneider *et al*. successfully learn adaptive distance metrics between test samples and prototypes in LVQ [13, 14] to achieve good performance.

The proposed LDDM method introduces the concept of the local learning framework [2] [16] into distance metric learning. LDDM creates an adjustable model, using a set of locally learned distance metrics, instead of one single metric, to best estimate the vicinity of each focal sample. LDDM uses neighborhood information to extract discriminative information and all local distance metrics are aligned probabilistically for the final prediction to make up for the deficiency of single local distance metric approaches [3].

3. Learning Discriminative Local Metrics

We propose a new local discriminative distance metric method over a focal vicinity of the training space by maximizing local discriminative information for each training sample.

A generic classification task can be stated as follows: given a set of d-dimensional training samples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$ and their associated labels $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n]$, we need to estimate a model to classify unknown test samples.
The distance metric A is in the form of

$$d_A(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\|_A = \sqrt{(\mathbf{a} - \mathbf{b})^T A (\mathbf{a} - \mathbf{b})}, \quad (1)$$

where A is positive semi-definite, and parameterizes a family of distances [20]. Technically, it allows pseudometrics, such that $d_A(\mathbf{a}, \mathbf{b}) = 0$ does not imply $\mathbf{a} = \mathbf{b}$. Replacing A with $\mathbf{W}^T\mathbf{W}$ in Equation (1) based on the Cholesky decomposition, we get:

$$\begin{aligned} d_A(\mathbf{a}, \mathbf{b}) =& \sqrt{(\mathbf{a}-\mathbf{b})^T\mathbf{W}\mathbf{W}^T(\mathbf{a}-\mathbf{b})} \\ =& \left\|\mathbf{W}^T(\mathbf{a}-\mathbf{b})\right\|. \end{aligned} \tag{2}$$

What we want to learn is the projection matrix W in Equation (2) to make *a focal sample* $\mathbf{x}_i$ close to the training samples sharing the same class label and far away from the training samples having different labels. To better keep the local discriminative information, we construct a focal vicinity instead of using all of the training data. For the focal sample $\mathbf{x}_i$, *the focal vicinity* $\mathbf{X}_i$ contains k_1 nearest neighbors in the same class and k_2 nearest neighbors in the different class, formally, $\mathbf{X}_i = [\mathbf{x}_i, \mathbf{x}_{i_{\widetilde{1}}}, \cdots, \mathbf{x}_{i_{\widetilde{k_1}}}, \mathbf{x}_{i_1}, \cdots, \mathbf{x}_{i_{k_2}}]$, where $\widetilde{k_1}$ represents the index for samples from the same-class of the focal sample.

In a focal vicinity, the distances between the focal sample and neighborhood samples in the same class should be as small as possible. Simultaneously, the distances between the focal sample and neighborhood samples in a different class are expected to be as large as possible. We define the objective function to satisfy such a criterion:

$$\arg\min_{\mathbf{A}_i} \left(\sum_{p=1}^{k_1} d^2_{\mathbf{A}_i}(\mathbf{x}_i, \mathbf{x}_{i_{\widetilde{p}}}) - \beta \sum_{q=1}^{k_2} d^2_{\mathbf{A}_i}(\mathbf{x}_i, \mathbf{x}_{i_q}) \right), \tag{3}$$

where β is a multiplicative factor to balance the influence of equivalence constraints and inequivalence constraints with respect to the focal sample. By defining the coefficient vector

$$w_i = \left[\overbrace{1, \cdots, 1}^{k_1} \quad \overbrace{-\beta, \cdots, -\beta}^{k_2} \right] \tag{4}$$

According to Equation (4) and the construction of $\mathbf{X}_i$, equation (3) is reduced to:

$$\begin{aligned}
&\arg\min_{\mathbf{A}_i}\left(\sum_{j=1}^{k_1+k_2} d^2_{\mathbf{A}_i}(\mathbf{X}_i\{1\},\mathbf{X}_i\{j+1\})(w_i)_j\right) \\
=&\arg\min_{\mathbf{W}_i}\left(\sum_{j=1}^{k_1+k_2} \|\mathbf{W}_i\left(\mathbf{X}_i\{1\}-\mathbf{X}_i\{j+1\}\right)\|_2^2(w_i)_j\right) \\
=&\arg\min_{\mathbf{W}_i} tr\left(\mathbf{W}_i^T\mathbf{X}_i\mathbf{L}_i\mathbf{X}_i^T\mathbf{W}_i\right),
\end{aligned} \tag{5}$$

where $\mathbf{X}_i\{j\}$ is the j^{th} column in the focal vicinity matrix $\mathbf{X}_i$, $\mathbf{A}_i$ is decided by $\mathbf{W}_i$ based on Equation (2) and $\mathbf{L}_i \in \mathbb{R}^{(k_1+k_2+1)\times(k_1+k_2+1)}$ is given by

$$\mathbf{L}_i = \begin{bmatrix} \sum_{j=1}^{k_1+k_2}(w_i)_j & -w_i^T \\ -w_i & diag(w_i) \end{bmatrix}. \tag{6}$$

To make the projection matrix $\mathbf{W}_i$ learned from the focal vicinity $\mathbf{X}_i$ linear and orthogonal, we impose $\mathbf{W}_i^T\mathbf{W}_i = \mathbf{I}_d$, where $\mathbf{I}_d$ is a $d \times d$ identity matrix. Equation (5) is then deformed to:

$$\min\ tr\left(\mathbf{W}_i^T\mathbf{X}_i\mathbf{L}_i\mathbf{X}_i^T\mathbf{W}_i\right)\ s.t. \mathbf{W}_i^T\mathbf{W}_i = \mathbf{I}_d. \tag{7}$$

Solutions of Equation (7) can be obtained with the standard eigen-decomposition:

$$\mathbf{X}_i\mathbf{L}_i\mathbf{X}_i^T\mathbf{u} = \lambda\mathbf{u}. \tag{8}$$

Let the column vectors $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_d$ be the solution of Equation (8), ordered according to eigenvalues $\lambda_1 < \lambda_2 < \cdots < \lambda_d$. The optimal projection matrix $\mathbf{W}_i$ is then given by: $\mathbf{W}_i = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}'_d]$, where $d' < d$. Once $\mathbf{W}_i$ is calculated, the local discriminative distance metric $\mathbf{A}_i$ with respect to focal sample $\mathbf{x}_i$ can be calculated using Equation (2).

4. A Probabilistic Approach for Classifiers Ensemble

Given an unknown test sample $\mathbf{x}_j$, let o be the class label of focal sample $\mathbf{x}_i$, the number of possible classes is N_o, the probability of $\mathbf{x}_j$ belonging to the class o, $Pr_i(o|\mathbf{x}_j)$, using the local distance metric $\mathbf{A}_i$ of the i^{th} focal sample $\mathbf{x}_i$ is

$$Pr_i(o|\mathbf{x}_j) = \begin{cases} \frac{\sum_{k=1}^{n}\{\theta(\mathbf{x}_k\in\mathbf{V}(\mathbf{x}_i))\theta(y_k=o)\}}{\sum_{k=1}^{n}\theta(\mathbf{x}_k\in\mathbf{V}(\mathbf{x}_i))} & \text{if } \mathbf{x}_j \in \mathbf{V}_K(\mathbf{x}_i) \\ \frac{1}{N_o} & \text{otherwise} \end{cases} \tag{9}$$

where $\mathbf{V}_K(\mathbf{x}_i)$ is the local vicinity of training sample $\mathbf{x}_i$ which contains K nearest neighbors of $\mathbf{x}_i$ with respect to the learned local distance metric $\mathbf{A}_i$. $\theta(\cdot)$ is an indicator function that returns 1 when the input argument is true, and 0 otherwise. $\theta(\mathbf{x}_j \in \mathbf{V}_K(\mathbf{x}_i)) = 1$ indicates $\mathbf{x}_j$ is among K nearest neighbors of $\mathbf{x}_i$ with respect to $\mathbf{A}_i$, which is calculated in Equation (8). Otherwise, the focal sample $\mathbf{x}_i$ has no influence on the unknown test sample $\mathbf{x}_j$. $\mathbf{V}(\mathbf{x}_i)$ defines a circular clique whose center is the focal sample $\mathbf{x}_i$. The radius r is the distance between the focal sample and the test sample $\mathbf{x}_j$ under the learned local distance metric $\mathbf{A}_i$. Probability $Pr_i(o|\mathbf{x}_j)$ is calculated as purity of circular clique $\mathbf{V}(\mathbf{x}_i)$. Please notice that we propose a new prediction method in Equation (9) instead of the traditional KNN rules because of our objective function defined in Equation (3). We expect vicinity of the focal sample to contain as many similar samples as possible. In this case, if a test sample is not in the K nearest neighbors of the focal sample, it is expected not to be similar to the focal sample. The metric is expected to *pull* the samples with the same/different label as the focal sample $\mathbf{x}_i$ closer to/away from $\mathbf{x}_i$. Note that if the test sample $\mathbf{x}_j$ is the closest sample to $\mathbf{x}_i$ in $\mathbf{V}_K(\mathbf{x}_i)$, the probability is 1 for the test sample $\mathbf{x}_j$ to be assigned as the same class label as $\mathbf{x}_i$.

As illustrated in Figure 2, because the clique of the red circle $\mathbf{V}(\mathbf{x}_i)$ contains a focal sample, four red circles and one blue square, probability for the test sample belonging to the red circle class is $\frac{5}{6}$.

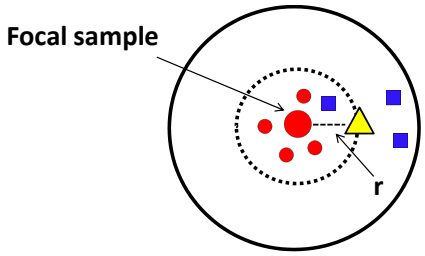

Figure 2: Local distance metric prediction. Red circles and blue squares belong to two classes. The yellow triangle is an unknown test sample. The red circle in the center is the focal sample $\mathbf{x}_i$. Figure illustrates the local distance metric space $\mathbf{A_i}$ learned from the focal sample and its vicinity. The solid-line circle is $V_K(\mathbf{x}_i)$ and the dashed-line circle represents $V(\mathbf{x}_i)$. The probability for the yellow triangle belonging to the red circle class is the number of red circles in $V(\mathbf{x}_i)$ divided by total number of training samples in $V(\mathbf{x}_i)$.

We can obtain a set of locally learned classifiers described in different data space, using the local classifier defined in Equation (9) under each local distance metric. This approach makes these local classifiers independent of each other to facilitate the alignment operation. Each obtained local distance metric best mea-

sures the vicinity of the focal sample and places the same class samples close to the focal sample and the different-class samples far away from the focal sample. To make the training model adjustable according to different test samples, we add a weight coefficient ϕ when combining n local prediction $Pr_i(o|\mathbf{x}_j)$ for a given test sample $\mathbf{x}_j$. Weight ϕ is decided by the distance between the test sample and focal sample. A final prediction is made by aligning n outputs in a probabilistic framework. The alignment process is formally defined as

$$Pr(o|\mathbf{x}_j) = \frac{1}{n}\sum_{i=1}^{n} \phi_i Pr_i(o|\mathbf{x}_j), \tag{10}$$

where n is the number of classifiers and $Pr_i(o|\mathbf{x}_j)$ is the probability of sample $\mathbf{x}_j$ belonging to class o predicted by the i^{th} local classifier. To simplify this process, we give all the training samples equal weights by letting $\phi_i = 1$. This makes the ensemble process behave as an equal weight voting. The class label with the highest probability is the final label of test sample.

An overall summary of our local discriminative distance metrics (LDDM) method is described in Algorithm 1. In training procedure, we need to calculate $\mathbf{W}_i$ by decomposing a $(k_1 + k_2 + 1) \times (k_1 + k_2 + 1)$ matrix $\mathbf{X}_i\mathbf{L}_i\mathbf{X}_i^T$ in Equation (8) for each focal sample $\mathbf{x}_i$ which has time complexity $O(n(k_1 + k_2 + 1)^3)$. When testing an unknown sample, it is linear time $O(n)$ to the training set size, since all the local distance metrics were already obtained in the training phase. The test time complexity only depends on Equation (9) and Equation (10) which just ensemble the results of n training samples using pre-calculated local distance metrics. Note that the projection for all the training samples to the distance metric space can be conducted in the training phase. Despite the high training cost, we can parallelize the proposed model to make it scalable for large-scale problems. Local classifiers could also be learned offline in advance.

5. Theoretical Analysis

We now theoretically prove the stability and efficiency of the proposed LDDM method by analyzing the convergence rate of the local discriminative distance metric and generalization bound of the local metrics and classifiers ensemble.

We assume that all the samples and their labels can be represented by an unknown distribution $F(\mathbf{x}, \mathbf{y})$, defined by pairs $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^d \times \mathbb{R}^1$. The pair $(\mathbf{x}, \mathbf{y})$ is denoted as $\mathbf{z}$ for short. Model $\mathbf{x} \to f(\mathbf{x}, \alpha)$ of the output $\mathbf{y}$ is controlled by a parameter $\alpha \in \Lambda$. $f(\mathbf{x}, \alpha)$ refers to the local classifier defined in Equation (9)

Algorithm 1 LDDM: a multiple distance metrics approach for classification

Training procedure

1: **for** each training sample $\mathbf{x}_i$ **do**
2: Get the focal vicinity $\mathbf{X}_i$ for $\mathbf{x}_i$
3: Build the discriminative matrix $\mathbf{L}_i$ using Equation (6)
4: solve the projection matrix $\mathbf{W}_i$ by Equation (8)
5: **end for**

Test procedure

1: **for** each test sample $\mathbf{x}_j$ **do**
2: Calculate the probability for $\mathbf{x}_j$ belonging to class o when using the training sample $\mathbf{x}_i$ as the focal sample, $Pr_i(o|\mathbf{x}_j)$ by Equation (9)
3: Ensemble all the predictions by different training samples according to Equation (10)
4: **end for**

for LDDM. The $0-1$ loss function $Q(\mathbf{y}, f(\mathbf{x}, \alpha))$ (or $Q(\mathbf{z}, \alpha)$ for short) measures the quality of estimation by $f(\mathbf{x}, \alpha)$ for output $\mathbf{y} \in \{-1, +1\}$. The global risk function is defined as

$$R(\alpha) = \int Q(\mathbf{z}, \alpha) dF(\mathbf{z}) \tag{11}$$

over all functions $\{f(\mathbf{x}, \alpha), \alpha \in \Lambda\}$, and samples $\{\mathbf{z}_i\}_{i=1}^n$ are independently drawn from the unknown distribution $F(\mathbf{z})$.
The empirical risk function with respect to the training samples $\{\mathbf{z}_i\}_{i=1}^n$ is

$$R_{emp}(\alpha) = \frac{1}{n} \sum_{i=1}^{n} Q(\mathbf{z}_i, \alpha). \tag{12}$$

In local algorithms, the local risk function $R(\alpha, \mathbf{x}_0)$ depends on the focal sample $\mathbf{x}_0$ and the vicinity of $\mathbf{x}_0$. The nonnegative locality function $D(\mathbf{x}, \mathbf{x}_0, A)$, which embodies vicinity information of the focal sample, is defined as

$$D(\mathbf{x}, \mathbf{x}_0, A) = \begin{cases} 1 & \text{if } \|\mathbf{x} - \mathbf{x}_0\|_A \leq r \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

where A is the distance metric obtained by letting $\mathbf{x}_0$ be the focal sample and r is the soft threshold of the locality function, which is defined by the distance

between the focal sample and the the test sample, and illustrated in Figure 2 for LDDM, where K is number of neighbors to be considered into the vicinity. The norm of locality function is defined as

$$\|D(\mathbf{x}_0, A)\| = \int D(\mathbf{x}, \mathbf{x}_0, A) dF(\mathbf{z}). \tag{14}$$

Based on the definition of the locality function, samples and labels can be represented by a new distribution $F(\mathbf{z}, A)$ corresponding to local distance metric A. The distribution is defined as

$$\int_A dF(\mathbf{z}, A) = \int_A \frac{D(\mathbf{x}, \mathbf{x}_0, A)}{\|D(\mathbf{x}_0, A)\|} dF(\mathbf{z}). \tag{15}$$

The local distance metric-based unnormalized local risk function is defined as:

$$\mathcal{R}(\alpha, A, \mathbf{x}_0) = \int Q(\mathbf{z}, \alpha) D(\mathbf{x}, \mathbf{x}_0, A) dF(\mathbf{z}), \tag{16}$$

and the local empirical risk function is based on the summation over all focal samples, which is defined as:

$$\mathcal{R}_{emp}(\alpha, A, \mathbf{x}_0) = \frac{1}{n} \sum_{i=1}^{n} Q(\mathbf{z}_i, \alpha) D(\mathbf{x}_i, \mathbf{x}_0, A). \tag{17}$$

Next, we give the bound on the convergence rate of a local classifier, risk bound of one local classifier and risk bound of the ensemble of a set of local classifiers.

5.1. Convergence Rate of Local Classifier

In this paper, we define concept of local domain-based VC-dimension, which is a VC-dimension of a set of functions under a local vicinity. Convergence rate bound of the global risk function only depends on the number of training samples and the VC-dimension that measures the complexity and the expressive power of the set of loss functions $\{Q(\mathbf{z}, \alpha), \alpha \in \Lambda\}$.

In the existing distance metric learning methods, all the VC-dimension and loss functions are under the same distance metric. Thus these distance metric methods obey the bound in the following theorem [16].

Theorem 5.1. *Let $\{Q(\mathbf{z}, \alpha), \alpha \in \Lambda\}$ be a set of nonnegative real functions with VC-dimension h. Then the following bound holds*

$$P\left\{ \sup_{\alpha \in \Lambda} \frac{R(\alpha) - R_{emp}(\alpha)}{\sqrt{\int Q^2(\mathbf{z}, \alpha) dF(\mathbf{z})}} > \epsilon a(\epsilon) \right\} < 12 \left(\frac{2ne}{h} \right)^h \exp\left\{ -\frac{\epsilon^2 n}{4} \right\}, \tag{18}$$

where

$$a(\epsilon) = \sqrt{1 - \frac{1}{2}\ln\epsilon}.$$

Theorem 5.1 shows the bound for the test error $\mathcal{R}_{emp}(\alpha)$. The left part is a probability corresponding to the difference between training error $\mathcal{R}(\alpha)$ and test error $\mathcal{R}_{emp}(\alpha)$. The probability approaches 0 when the test error and the training error have an acceptable difference This probability has been proved to be converged to 0 when there are enough training samples [16]. For our local discriminative distance metrics algorithm, the loss functions are different according to the focal samples since they obey their own local distance metrics obtained from the focal samples. To obtain the convergence rate of a local classifier, we assume that the loss function with the local distance metric satisfy the following mild condition:

$$\sup_{\alpha,A} \frac{\sqrt{\int Q^2(\mathbf{z},\alpha)dF(\mathbf{z},A)}}{\int Q(\mathbf{z},\alpha)dF(\mathbf{z},A)} < \tau. \tag{19}$$

It means that the probability that $\sup_\alpha Q(\mathbf{z},\alpha)$ exceeds some value will decrease quickly with the value increasing. Value τ determines how fast it decreases. We can get the following theorem for convergence rate of local risk function which is bounded in the term of local domain-based VC-dimension h^*.

Theorem 5.2. *Let the vicinity of x_0 be under the local distance metric A and the set of loss functions $\{Q(\mathbf{z},\alpha)D(\mathbf{x},\mathbf{x}_0,A),\alpha\in\Lambda\}$ have the local domain based VC-dimension h^*. Then the following bound holds:*

$$P\left\{\sup_{\alpha\in\Lambda}\frac{\mathcal{R}(\alpha,A,\mathbf{x}_0)-\mathcal{R}_{emp}(\alpha,A,\mathbf{x}_0)}{\mathcal{R}(\alpha,A,\mathbf{x}_0)} > \frac{\tau\epsilon a(\epsilon)}{\sqrt{\|D(\mathbf{x}_0,A)\|}}\right\} < 12\left(\frac{2Ke}{h^*}\right)^{h^*}\exp\left\{-\frac{\epsilon^2 K}{4}\right\} \tag{20}$$

where

$$a(\epsilon) = \sqrt{1 - \frac{1}{2}\ln\epsilon}.$$

Proof Theorem 5.1 implies the following inequality:

$$P\left\{\sup_{\alpha\in\Lambda}\frac{\mathcal{R}(\alpha,A,\mathbf{x}_0)-\mathcal{R}_{emp}(\alpha,A,\mathbf{x}_0)}{\sqrt{\int Q^2(\mathbf{z},\alpha)D^2(\mathbf{x},\mathbf{x}_0,A)dF(\mathbf{z})}} > \epsilon a(\epsilon)\right\} < 12\left(\frac{2Ke}{h^*}\right)^{h^*}\exp\left\{-\frac{\epsilon^2 K}{4}\right\}, \tag{21}$$

where K is the size of the focal vicinity defined in Equation (9). According to Equation (13) and Equation (15), we have

$$\begin{aligned}&\sqrt{\int Q^2(\mathbf{z},\alpha)D^2(\mathbf{x},\mathbf{x}_0,A)dF(\mathbf{z})}\\ \leq&\sqrt{\int Q^2(\mathbf{z},\alpha)\left\|D(\mathbf{x}_0,A)\right\|dF(\mathbf{z},A)}.\end{aligned} \tag{22}$$

According to Equation (15), (16) and (19), we have

$$\sqrt{\int Q^2(\mathbf{z},\alpha)dF(\mathbf{z},A)} < \tau\int Q(\mathbf{z},\alpha)dF(\mathbf{z},A) = \tau\frac{\mathcal{R}(\alpha,A,\mathbf{x}_0)}{\|D(\mathbf{x}_0,A)\|}. \tag{23}$$

According to Equation (22) and (23), we have

$$\sqrt{\int Q^2(\mathbf{z},\alpha)D^2(\mathbf{z},\mathbf{x}_0,A)dF(\mathbf{z})} < \tau\frac{\mathcal{R}(\alpha,A,\mathbf{x}_0)}{\sqrt{\|D(\mathbf{x}_0,A)\|}}. \tag{24}$$

The inequality Equation (20) can be obtained from Equations (21) and (24) immediately. This completes the proof.

Theorem 5.1 gives the convergence rate for the approaches based on a single distance metric in Equation 2. Theorem 5.2 gives the convergence rate using the local domain based VC-dimension for a single distance metric $\mathbf{A}_i$ in Equation 3. In the following theorem, we show the risk bound of a local classifier according to Theorem 5.2.

5.2. Bound of Local Classifiers

For local classifiers learned on local distance metrics according to Equation 9, we have the following theorem.

Theorem 5.3. *Let the distance metric of the vicinity of $\mathbf{x}_0$ be A. The set of loss functions $\{Q(\mathbf{z},\alpha)D(\mathbf{x},\mathbf{x}_0,A),\alpha\in\Lambda\}$ have the local domain-based VC-dimension h^*. The following inequality holds for all $\alpha\in\Lambda$ with probability $1-\eta$:*

$$R(\alpha,A,\mathbf{x}_0) \leq \frac{1}{\|D(\mathbf{x}_0,A)\|}\cdot\left[\mathcal{R}_{emp}(\alpha,A,\mathbf{x}_0)+\nu\left(1+\sqrt{1+\frac{4}{\nu}\mathcal{R}_{emp}(\alpha,A,\mathbf{x}_0)}\right)\right] \tag{25}$$

where

$$\nu = 2\frac{(h^*)\left\{\ln[2K/(h^*)]+1\right\} - \ln\frac{\eta}{24}}{K}$$

Proof In Equation (20), let $\eta/2$ denote the right-side.
By solving the equation

$$12\left(\frac{2Ke}{h^*}\right)^{h^*}\exp\{-\frac{\epsilon^2 K}{4}\} = \eta/2 \tag{26}$$

and replacing the result into Equation (20), we obtain the following inequality with probability $1-\eta/2$.

$$\mathcal{R}(\alpha, A, \mathbf{x}_0) \leq \mathcal{R}_{emp}(\alpha, A, \mathbf{x}_0) + \nu\left(1+\sqrt{1+\frac{4}{\nu}\mathcal{R}_{emp}(\alpha, A, \mathbf{x}_0)}\right), \tag{27}$$

where

$$\nu = 2\frac{(h^*)\left\{\ln[2n/(h^*)]+1\right\} - \ln\frac{\eta}{24}}{n}.$$

By defining the normalized empirical risk for the vicinity of $\mathbf{x}_0$

$$R(\alpha, A, \mathbf{x}_0) = \int Q(\mathbf{z}, \alpha)\frac{D(\mathbf{x}, \mathbf{x}_0, A)}{\|D(\mathbf{x}_0, A)\|}dF(\mathbf{z}),$$

we can get Equation (25) by dividing both sides of inequality Equation (27) by $\|D(\mathbf{x}_0, A)\|$. This completes the proof.

5.3. Bound of Classifiers Ensemble

We now further explain the generalization bound of the classifiers ensemble method discussed in Section 4. Since every training sample will be treated as a focal sample in turn, n samples drawn from the unknown distribution $F(\mathbf{x}, \mathbf{y})$ can generate n local distance metrics. For each unknown test sample $\mathbf{x}$, the base classifier $f_i(\mathbf{x}, A_i) \in \mathcal{H}$ can be obtained by Equation (9), where A_i is the local distance metric learned by focal sample $(\mathbf{x}_i, y_i)$, which embodies local discriminative information and the size of $\mathcal{H}$ is n. According to the alignment procedure in Equation (10), we define the final classifier after ensemble as

$$g(\mathbf{x}) = \text{sign}(\sum_{i=1}^{n} f_i(\mathbf{x}, A_i)). \tag{28}$$

In Equation (28), $g(\cdot)$ gives a wrong prediction on the sample $(\mathbf{x}, \mathbf{y})$ only if $\mathbf{y}g(\mathbf{x}) \leq 0$. $f_i(\mathbf{x}, A_i)$ is the i^{th} element of $f(x, A)$. The margin function is given by $\mathbf{y}g(\mathbf{x})$. Equation (28) is fundamentally a majority vote on all base classifiers. [12] has shown a bound which applies to all majority-vote classifier. Inspired by this, we show the following theorem which states that the generalization error of the ensembled classifier can be bounded in terms of the number of training samples with the margin below a threshold θ and in the capacity of base classifier space $\mathcal{H}$.

Theorem 5.4. *Let S be a set of n samples independently drawn from the distribution $F(\mathbf{x}, \mathbf{y})$ over $X \times \{-1, +1\}$. Assume that the base-classifier space $\mathcal{H}$ is finite, and let $\sigma > 0$. Then with probability at least $1 - \sigma$ over the random choice of the training set S, every weighted average function $g(\cdot)$ satisfies the following bound for all $\theta > 0$:*

$$P_F(\mathbf{y}g(\mathbf{x}) \leq 0) \leq P_S(\mathbf{y}g(\mathbf{x}) \leq \theta) + O\left(\frac{1}{\sqrt{n}}\left(\frac{\log n \log |\mathcal{H}|}{\theta^2} + \log(1/\sigma)\right)^{1/2}\right). \tag{29}$$

For detailed proof please refer to Theorem 1 in [12].

6. Experiments

We compare the proposed LDDM against other state-of-the-art distance metric learning algorithms, including the representative single global distance metric approach: Xing's method (Xing) [20], two latest single local distance metric approaches: Local Fisher Discriminate Analysis (LFDA)[15] and Large Margin Nearest Neighbor-based distance metric (LMNN)[18], the state-of-the-art multiple local distance metrics approaches: multiple Large Margin Nearest Neighbor metrics (mLMNN)[17], the Discriminant Adaptive Nearest Neighbor (DANN) and the five adaptive iteration i-DANN [10], the local ADAptive MEtric Nearest-Neighbor algorithm (ADAMENN) and the five adaptive iteration i-ADAMENN [7].

We generate a synthetic multimodal dataset to discuss the distance projection problem. In addition, we use a synthetic noisy dataset to discuss the problem of noise tolerance. Furthermore, we evaluate all those algorithms using eleven benchmark UCI datasets[1].

[1] http://archive.ics.uci.edu/ml/

The experiments are designed as follows: we first run each algorithm on training samples to learn a model which could be a distance metric or a set of distance metrics. For DANN we perform PCA to reduce dimension to at least $n - c$ (n is the number of samples, c is the number of classes) to avoid the singularity problem [1]. Finally, each algorithm predicts test samples using KNN classifier. For the parameter settings, all the methods compared in this paper use the parameter settings, as proposed in their original papers[20, 15, 18, 17, 10, 7] and validated by cross validation. We report the optimal dimensions for Xing, LFDA and LMNN in the experimental results.

LDDM sets four adjustable tuning parameters:

* k_1: the number of neighbors in the same class.

* k_2: the number of neighbors in the different class.

* β: multiplicative factor in Equation 3.

* K: the number of neighbors to be considered as the vicinity of focal sample in Equation 9.

Based on empirical observation, we set $k_1 = \max\{floor(0.15n), 3\}$, $k_2 = \max\{floor(0.1n), 2\}$, $\beta = 0.1$ where n is the total number of training samples. $K \in \{floor(0.1n), floor(0.2n), floor(0.3n)\}$ and we determine the value for K by cross-validation. d' for Equation (8) is determined is determined as follows. We sort the nonnegative eigenvalues in descending order, the sum of the first p eigenvalues that exceed 80% of the total sum of all nonnegative eigenvalues are discarded. The eigenvectors corresponding to the rest nonnegative eigenvalues as well as all the negative eigenvalues are preserved for the projection. The number of selected eigenvectors forms d'.

6.1. A Multimodal Dataset

Multimodal data distribution is ubiquitous in real-world data. It happens when samples from the same class do not always share similar distributions. For multimodal distribution dataset, the superiority of the local distance metric over global distance metric is widely admitted [22, 9, 17]. We construct a multimodal data set to evaluate: 1) whether multiple local distance metric algorithms are superior to single distance metric algorithms; 2) the ability to explore discriminative information for different algorithms; 3) the visualization under different projected distance metric space.

Table 1: Best recognition rates(%) of the multimodal synthetic dataset.

	Global metric	Single local metric		Multiple local metrics					
Methods	Xing	LFDA	LMNN	mLMNN	DANN	i-DANN	ADAMENN	i-ADAMENN	LDDM
Multimodal	74.25(50)	85.25(5)	89.75(3)	81.25	67.00	53.37	82.25	82.25	**90.38**(0.1n)

We generate a synthetic dataset which makes samples obey desired multimodal distribution. Positive samples have two different Gaussian distributions. Negative samples also have two different Gaussian distributions, where one is between two positive distributions, and the other stays outside of the positive distribution. The synthetic training data contains 400 positive and 400 negative samples with 50 dimensions. The test data is drawn from the same distribution as the training data and has the same size.

The experimental results are shown in Table 1. Clearly, in general, we have the performance of *the single local distance metric* $>$ *multiple distance metrics* $>$ *global distance metric*. LDDM , which is in the category of multiple distance metrics methods, has the best performance as an exception. To better understand this result, we visualize these distance metric approaches under their projected distance metric space using Equation 2.

Figure 3(a) shows the dataset projection in PCA space which gives us direct illustration of this case. Red circles and blue stars represent the positive samples and negative samples respectively. The green square is a test sample belonging to the negative class (blue stars) but is blind to the learning systems. The green square resides on the boundary between those two classes in the PCA projection space. Because some distance metric algorithms (LDDM, Xing, LFDA and LMNN) can be also regarded as the dimension reduction methods, we visualize them under 2-dimension projection. For other distance metric algorithms (DANN, ADAMENN), we adopt Sammon's Mapping [11] to project the data under 2-dimension space which preserves the Euclidean distance relation under their distance metrics.

Figures 3(b), (c) and (d) depict the visualization of the single metric methods, XING, LFDA and LMNN. Global distance metric method fails to deal with conflicting localized pairwise constraints which are located in different distributions. Thus, Xing's method mixes the positive and negative samples in Figure 3(b) and performs the worst. LFDA locally adopts the discriminative information which makes the samples in the same class closer. LMNN optimizes for large margin which makes the test sample move away from the original boundary. However, this optimization cannot really remove the boundary and make a large margin be-

tween two classes, because some samples that may not be on the boundary in the original space produce a new boundary in the new projected space. This is a known problem of the single-metric algorithms. Optimization may not achieve the desired goal for a multimodal dataset.

Figure 4 depicts the visualization of multiple metrics methods. mLMNN performs clustering and then learns the LMNN distance metric on each cluster. From Figure 4(a), we can find that mLMNN is not very different from LMNN. mLMNN may have a great performance gain only when clustering does a good job. In a close view of Figure 4(b), DANN locally minimizes the within-class distance and maximizes the between-class distance only on the neighborhood of the test sample, however discriminative information might not be sufficient to correctly classify this test sample. ADAMENN in Figure 4(c) has the similar hallmark which learns only on the vicinity of the test sample but fails to include sufficient discriminative information.

Figure 5(a) is a close view of PCA projection in Figure 3(a). The big red circle and the big blue star are the two nearest neighbors of the green square. This is a challenging problem because these two nearest neighbors may not be in the same class as the test sample. For our LDDM method, we learn distance metric on these neighbors respectively as in Figures 5(b) and (c). Our pairwise constraints only take effect on a small area, where they can be optimized much more efficiently. We can clearly see that vicinity of the focal samples does not mix different-class samples in the close view figures. The test sample is in the vicinity of the focal blue star in Figure 5(c) while it is not in the vicinity of the focal red circle in Figure 5(b). We regard that the test sample is in the same class as focal sample only when test sample is in vicinity of the focal sample. In such case, we can evaluate whether the nearest neighbors of test sample in the original space is really similar to test sample under the local discriminative distance metric. Because LDDM has a totally new way to explore discriminative information and use an ensemble for prediction which avoids the deficiency of other multiple metric methods, LDDM achieves a better result compared to other local distance metric methods.

6.2. A Noise Dataset

To compare the tolerance to noise among these algorithms, we build a synthetic dataset with different noise scales in training data. 200 positive and 200 negative samples are drawn from 50 dimensional Gaussian distributions with different mean values. 50 positive and 50 negative test samples are generated by sharing the Gaussian distribution of the same mean value with the training sample

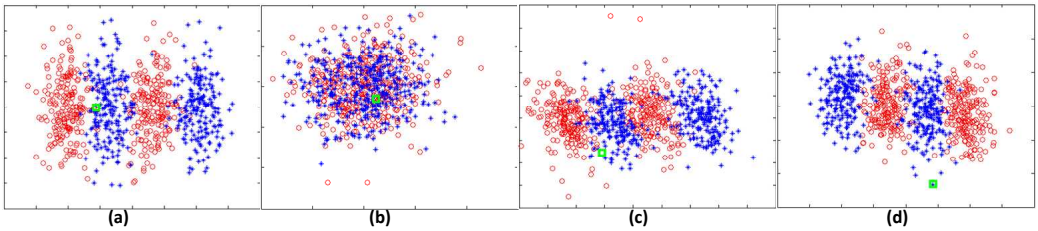

Figure 3: The multimodal data and its two-dimensional projection visualization. (a) The PCA projection space. (b) The Xing's projection space. (c) The LFDA projection space. (d) The LMNN projection space.

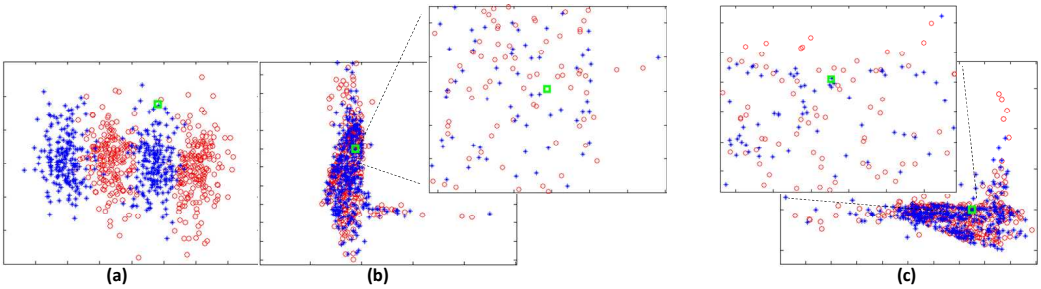

Figure 4: visualization of multiple distance metrics algorithms. (a) The mLMNN projection space under a distance metric learned from clusters of training samples. (b) Sammon's Mapping of the DANN distance metric learned for the test sample and its close view. (c) Sammon's mapping of the ADAMENN distance metric learned for the test sample and its close view.

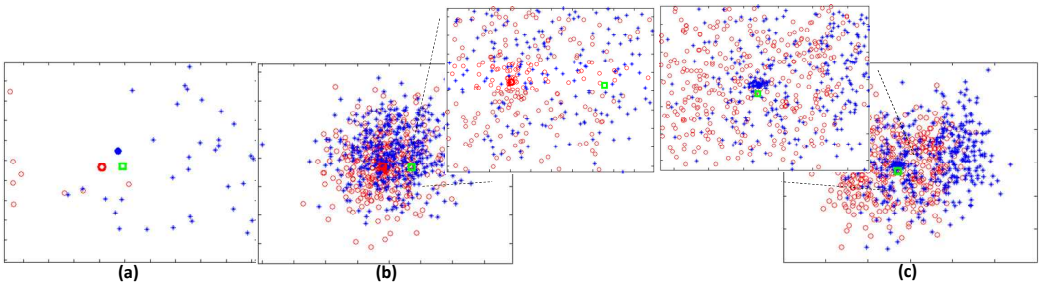

Figure 5: The visualization of LDDM algorithm. (a) A close view of the neighbors of test sample in the PCA space, the big red circle and the big blue star are its two nearest neighbors. (b) The LDDM projection learned from the big red circle. (c) The LDDM projection learned from the big blue star.

but vary a little in variance. Each class contains some noise samples which distribute more like the opposite class. In the experiments, we vary the percentage of noise data from 5% to 25%. We expect that a robust algorithm should be able to deal with the noise data in the training process and find the intrinsic distribution of the training samples. A dataset with 10% noise-level is visualized in Figure 6(a).

This is a specially built dataset and the noise samples are in a well-designed distribution. The samples in two different classes are far away. Ideally, if K in our LDDM is greater than $0.5n$, it is no doubt that LDDM will achieve perfect results. Because the number of noise samples is smaller than $0.5n$, the ensemble process will eliminate noise influence which is similar to apply a K-nearest-neighbor classifier with $K = 0.5n$. However, this is not fair for the comparison. We report our results with $K = 0.1n$ and study the influence of focal vicinity in LDDM.

From the experimental results reported in Table 2, we generally have this conclusion about the performance: *multiple distance metrics* $>$ *global distance metric* $>$ *single local distance metric*. Single local distance algorithms use neighborhoods to learn the discriminative information which might be highly influenced by the noise and inevitably encounters the over-fitting problem. The global method Xing performs better than traditional local methods, because the global pairwise constraints are helpful for capturing the overall distribution. Multiple metrics methods perform better than all the single metric methods. mLMNN optimizes the local pairwise constraints and also takes the global sense for each class into consideration. DANN and ADAMEN which use different distance metrics to model the samples in different areas greatly reduce the influence of noise. Differently, LDDM is very robust to noise which learns the discriminative information from focal vicinity and eliminates the noise information by classifiers ensemble. Even with 25% noise data which also means there are 25% mislabeled training samples, our LDDM method can still reliably find the intrinsic distribution.

To show the influence of focal vicinity to LDDM, we report the relation between classification accuracy and size of focal vicinity in Figure 6(b) using 10% noise level data. The focal vicinity consists of k_1 same-class nearest neighbors and k_2 different-class nearest neighbors. From the figure, we can find that when k_2 becomes larger, classification accuracy will increase significantly. If more different class samples are involved, it will be increasingly easier to capture the principal part of different class samples and ignore noise samples. If many local classifiers can achieve good classification results, then weighted combination of local classifiers will perform better.

We can also find that if the sizes of k_1 or k_2 approach 200, the focal vicinity is equivalent to the whole dataset, which means LDDM is degenerated to the global

Table 2: Best recognition rates(%) on the synthetic data with different noise levels.

	Global metric	Single local metric		Multiple local metrics					
Dataset	Xing	LFDA	LMNN	mLMNN	DANN	i-DANN	ADAMENN	i-ADAMENN	LDDM
5%	98(34)	90(18)	87(10)	99	99	80	90	92	**100**(0.1n)
10%	92(14)	96(22)	85(23)	93	98	59	97	99	**100**(0.1n)
15%	82(2)	85(22)	75(7)	83	97	64	97	96	**100**(0.1n)
20%	91(7)	91(27)	68(15)	77	88	55	95	95	**100**(0.1n)
25%	79(16)	88(23)	61(5)	72	87	51	86	84	**97**(0.1n)

Table 3: Properties of datasets.

	balance-scale	glass	image	ionosphere	soybean	tic-tac-toe	waveform	iris	wine	wdbc	car
samples	625	214	2310	351	47	958	5000	150	178	569	1728
dimensions	4	9	19	34	35	9	21	4	13	30	6
classes	3	6	7	2	4	2	3	3	3	2	4

distance metric method. When k_1 approaches 200 and k_2 approaches 0, LDDM is equivalent to Xing's method without the negative constraint which is to separate samples in different class. In this case, all the samples will converge to one dot and have the worst result. If k_1 and k_2 approach 200 simultaneously, LDDM achieves nearly 10% error rate as Xing's global method.The performance is relatively stable when the patch is small enough to encode the discriminative information and big enough to form a reliable patch.

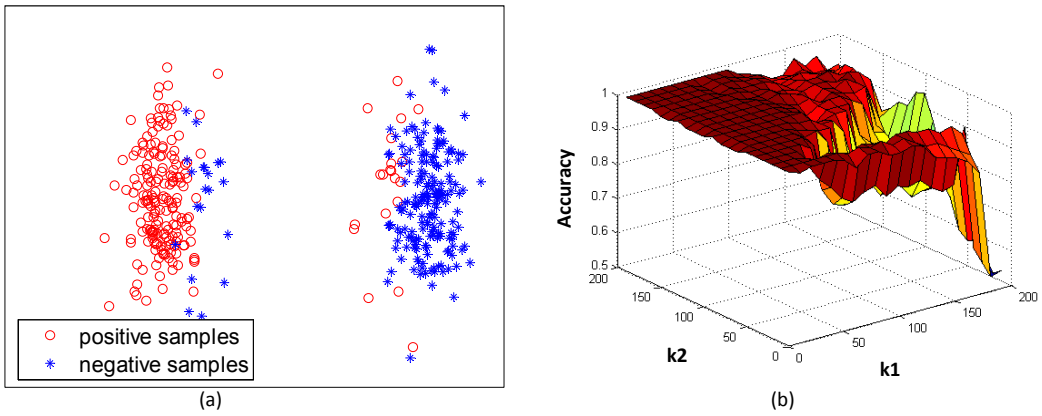

Figure 6: The noise synthetic data. (a) The training data with 10% noise samples in a 2-dimensional PCA space. (b) Classification accuracy vs. sizes of focal vicinities (k_1 and k_2) using 10% noise level data.

Table 4: Recognition rates(%) on UCI datasets by distance metrics approaches.

		Global metric	Single local metric		Multiple local metrics					
		Xing	LFDA	LMNN	mLMNN	DANN	i-DANN	ADAMENN	i-ADAMENN	LDDM
balance-scale	10%	75.48 ± 4.90	85.36 ± 2.74	$\mathbf{86.98 \pm 1.75}$	81.03 ± 2.23	82.54 ± 2.40	82.56 ± 2.74	69.17 ± 3.06	69.50 ± 2.02	85.79 ± 2.72
	80%	77.10 ± 3.79	91.61 ± 4.8	90.48 ± 4.78	78.23 ± 5.00	88.06 ± 2.66	79.68 ± 3.97	80.48 ± 6.29	80.16 ± 5.22	$\mathbf{93.39 \pm 3.44}$
glass	10%	51.19 ± 7.44	44.46 ± 5.20	46.19 ± 7.62	52.08 ± 5.56	46.25 ± 5.05	46.61 ± 4.92	47.86 ± 7.52	46.90 ± 7.07	$\mathbf{53.15 \pm 4.84}$
	80%	69.05 ± 9.59	66.19 ± 9.64	57.14 ± 9.54	67.14 ± 11.54	63.33 ± 8.99	70.00 ± 5.52	$\mathbf{80.48 \pm 6.29}$	80.16 ± 5.22	72.38 ± 7.03
image	10%	86.14 ± 1.42	85.00 ± 2.73	$\mathbf{91.86 \pm 2.00}$	86.78 ± 1.57	88.07 ± 1.72	89.32 ± 1.05	57.77 ± 3.86	44.77 ± 3.97	87.20 ± 1.28
	80%	95.84 ± 1.62	92.90 ± 5.19	97.01 ± 1.01	96.71 ± 1.29	96.28 ± 1.04	$\mathbf{97.45 \pm 1.03}$	22.94 ± 3.90	22.29 ± 3.10	91.52 ± 1.80
ionosphere	10%	87.46 ± 1.89	89.18 ± 0.38	88.79 ± 2.76	89.46 ± 0.74	89.18 ± 0.29	89.04 ± 0.38	89.21 ± 0.63	89.11 ± 0.48	$\mathbf{90.00 \pm 4.10}$
	80%	89.14 ± 4.22	89.14 ± 2.63	87.43 ± 6.89	88.57 ± 4.26	89.43 ± 2.71	86.57 ± 5.56	$\mathbf{90.57 \pm 5.56}$	90.29 ± 3.07	89.14 ± 2.63
soybean	40%	75.50 ± 18.63	74.50 ± 20.61	70.50 ± 25.22	91.00 ± 12.65	96.00 ± 6.58	96.50 ± 5.80	79.00 ± 15.06	83.00 ± 8.23	$\mathbf{97.00 \pm 2.58}$
	80%	85.00 ± 17.48	$\mathbf{100 \pm 0.00}$	88.89 ± 13.18	$\mathbf{100 \pm 0.00}$	$\mathbf{100 \pm 0.00}$	97.50 ± 7.91	92.50 ± 12.08	$\mathbf{100 \pm 0.00}$	$\mathbf{100 \pm 0.00}$
tic-tac-toe	10%	89.71 ± 1.39	96.54 ± 3.58	97.14 ± 2.22	86.04 ± 2.32	91.21 ± 1.99	$\mathbf{97.28 \pm 0.84}$	75.03 ± 4.52	74.95 ± 2.71	80.59 ± 3.50
	80%	97.47 ± 0.89	98.63 ± 1.12	97.79 ± 0.78	$\mathbf{99.89 \pm 0.33}$	98.32 ± 1.02	98.32 ± 1.02	95.26 ± 2.59	94.74 ± 2.58	89.89 ± 3.94
waveform	10%	76.60 ± 0.69	81.97 ± 1.02	77.41 ± 0.76	78.19 ± 0.71	79.59 ± 0.44	72.66 ± 0.35	69.01 ± 1.30	53.58 ± 1.65	$\mathbf{83.70 \pm 0.46}$
	80%	77.84 ± 1.67	81.84 ± 2.32	77.70 ± 2.21	81.32 ± 1.38	81.94 ± 1.21	76.28 ± 2.35	50.10 ± 5.52	54.97 ± 3.07	$\mathbf{84.41 \pm 0.44}$
iris	10%	89.17 ± 8.11	$\mathbf{94.75 \pm 2.42}$	91.67 ± 5.60	86.83 ± 6.81	86.83 ± 7.17	86.75 ± 7.57	87.67 ± 8.51	87.25 ± 8.74	89.33 ± 7.28
	80%	94.00 ± 5.84	96.00 ± 3.44	95.33 ± 5.49	96.00 ± 4.66	94.67 ± 5.26	96.00 ± 4.66	94.67 ± 6.13	94.67 ± 5.26	$\mathbf{96.67 \pm 4.71}$
wine	10%	75.44 ± 7.12	53.68 ± 12.39	67.87 ± 13.26	$\mathbf{90.51 \pm 6.38}$	89.85 ± 8.02	90.37 ± 7.37	66.76 ± 6.38	67.50 ± 6.61	88.09 ± 5.58
	80%	80.59 ± 7.87	72.35 ± 15.7	89.41 ± 10.3	91.76 ± 6.32	91.18 ± 6.93	95.88 ± 3.97	66.47 ± 6.23	86.47 ± 6.82	$\mathbf{96.47 \pm 4.96}$
wdbc	10%	88.33 ± 3.55	88.46 ± 2.44	91.34 ± 1.89	91.27 ± 2.08	90.85 ± 1.28	90.29 ± 1.68	89.33 ± 2.17	89.73 ± 2.49	$\mathbf{92.52 \pm 2.45}$
	80%	89.82 ± 4.84	90.71 ± 3.75	91.61 ± 4.54	93.57 ± 4.39	90.71 ± 3.93	93.93 ± 3.97	92.50 ± 3.35	85.89 ± 4.25	$\mathbf{94.11 \pm 4.13}$
car	10%	77.49 ± 3.22	81.06 ± 2.77	82.44 ± 3.41	80.69 ± 1.67	75.46 ± 2.02	72.72 ± 1.52	73.98 ± 1.11	73.63 ± 0.73	$\mathbf{82.66 \pm 1.38}$
	80%	85.99 ± 4.23	87.44 ± 3.75	$\mathbf{96.10 \pm 1.43}$	94.01 ± 1.78	84.19 ± 2.01	81.45 ± 2.81	72.79 ± 3.93	67.56 ± 3.73	81.45 ± 2.96

6.3. UCI Datasets

We compare the algorithms on eleven benchmark UCI datasets. The properties of datasets are described in Table 3. The average accuracy and standard deviation is calculated by the 10-fold cross validation with 1 fold as validation set. We report the results by varing the training set size: use 1 fold for training (10%) and 8 folds for training (80%). This can help us to better understand the performance for insufficient training data. Soybean dataset is too small, therefore, we adopt 40% and 80% settings.

The experimental results are reported in Table 4. It is difficult to say whether any algorithm can always perform best on various datasets. Based on the table, we generally have *multiple distance metrics* $>$ *single local distance metric* $>$ *global distance metric*. Multiple metrics methods also perform better for the insufficient training data because it adopts multiple metrics to explore more useful information, for example, the multiple extension version mLMNN performs better than the single local metric LMNN in most cases. i-DANN and i-ADAMENN are not always perform better than their one-step DANN and ADAMENN. This is mainly because the converged solution will prune to be proportional to the identity metric which gives equal weights for different features [7]. The proposed LDDM method performs better in the overall comparison which won 12 out of 22 comparisons. LDDM is the most stable algorithm in this comparison.

Table 5: Recognition rates(%) on UCI datasets for the state-of-the-art classifiers and distance metric methods using ensemble framework.

		Classifiers				Ensemble distance metric		
		kNN	Decision tree	Naive Bayes	SVM	Xing	LFDA	LMNN
balance-scale	10%	76.33 ± 3.51	70.46 ± 3.80	0.77 ± 0.04	84.52 ± 2.36	82.60 ± 1.99	86.31 ± 2.41	86.45 ± 2.17
	80%	74.68 ± 4.37	77.42 ± 3.31	0.78 ± 0.03	90.81 ± 3.95	89.52 ± 4.11	89.84 ± 4.17	92.10 ± 4.26
glass	10%	54.29 ± 6.25	45.77 ± 7.29	0.76 ± 0.06	41.37 ± 5.97	49.40 ± 9.86	37.86 ± 6.79	47.38 ± 7.29
	80%	73.33 ± 8.75	66.19 ± 8.23	0.78 ± 0.11	64.29 ± 9.59	68.57 ± 8.16	66.67 ± 7.45	64.76 ± 0.11
image	10%	86.31 ± 0.01	89.03 ± 0.02	0.73 ± 0.02	61.04 ± 5.04	78.14 ± 0.02	14.28 ± 0.00	87.26 ± 0.02
	80%	96.19 ± 0.02	95.28 ± 0.01	0.78 ± 0.02	92.99 ± 1.67	82.64 ± 0.02	14.29 ± 0.02	86.10 ± 0.01
ionosphere	10%	88.86 ± 1.05	87.25 ± 0.02	0.73 ± 0.01	89.36 ± 0.41	86.29 ± 2.92	68.25 ± 6.16	21.21 ± 1.11
	80%	89.14 ± 4.00	88.00 ± 5.84	0.83 ± 0.04	89.14 ± 2.63	89.14 ± 3.24	89.14 ± 2.63	81.43 ± 2.43
soybean	40%	94.50 ± 9.56	65.00 ± 9.72	0.71 ± 0.14	69.00 ± 7.38	51.00 ± 1.90	29.00 ± 1.13	54.50 ± 10.39
	80%	100.00 ± 0.00	92.50 ± 16.87	0.78 ± 0.08	85.00 ± 17.48	52.50 ± 2.99	30.00 ± 2.30	62.50 ± 3.38
tic-tac-toe	10%	89.83 ± 1.51	71.25 ± 2.45	0.76 ± 0.09	74.30 ± 5.67	78.92 ± 0.04	98.21 ± 0.00	97.37 ± 0.01
	80%	100.00 ± 0.00	93.58 ± 2.83	0.77 ± 0.05	99.05 ± 0.92	97.79 ± 0.02	98.32 ± 1.02	96.84 ± 0.02
waveform	10%	76.85 ± 0.63	71.87 ± 1.17	0.77 ± 0.01	85.61 ± 0.53	83.30 ± 0.53	85.84 ± 0.44	83.70 ± 0.56
	80%	77.74 ± 1.82	75.68 ± 1.93	0.75 ± 0.01	87.34 ± 1.63	84.70 ± 2.38	86.72 ± 2.11	84.16 ± 2.16
iris	10%	93.75 ± 2.43	70.92 ± 14.10	0.76 ± 0.05	83.75 ± 12.29	72.17 ± 9.58	71.58 ± 13.94	64.25 ± 10.70
	80%	96.00 ± 5.62	95.33 ± 4.50	0.79 ± 0.03	97.33 ± 3.44	95.33 ± 4.50	95.33 ± 6.32	95.33 ± 7.06
wine	10%	66.18 ± 4.57	72.87 ± 9.98	0.71 ± 0.10	39.49 ± 10.87	65.66 ± 10.53	38.75 ± 7.30	37.28 ± 8.94
	80%	76.47 ± 10.00	94.12 ± 4.80	0.78 ± 0.11	63.53 ± 10.30	88.82 ± 9.38	97.65 ± 3.04	96.47 ± 4.11
wdbc	10%	89.93 ± 1.63	89.33 ± 3.48	0.70 ± 0.01	63.53 ± 1.07	88.12 ± 1.89	84.29 ± 4.97	84.49 ± 5.19
	80%	91.43 ± 3.45	92.68 ± 2.30	0.74 ± 0.03	90.00 ± 3.28	94.82 ± 3.09	96.96 ± 2.39	95.89 ± 2.24
car	10%	78.94 ± 1.54	85.24 ± 1.84	0.74 ± 0.02	77.60 ± 3.00	76.46 ± 1.48	79.54 ± 3.06	79.19 ± 2.28
	80%	83.95 ± 3.29	95.29 ± 1.36	0.78 ± 0.02	91.51 ± 2.56	77.73 ± 3.23	79.36 ± 3.60	78.78 ± 3.02

We further conduct the experiments using the state-of-the-art classifiers: kNN, Decision Tree, Naive Bayes and Support Vector Machine (SVM). We also perform the other distance metric methods using our proposed ensemble local distance metrics framework. The results are reported in Table 5. Some simple classifier achieves stable results in this comparison. For example, kNN achieves the best results in some tests. The ensemble framework cannot guarantee to improve the other distance metric methods, which is because the probability approach defined in Equation (9) may not suit for other distance metrics. In LDDM, the samples in the same class as the focal sample will be pulled towards the focal sample, while this property cannot be obtained by other distance metrics methods.

The corresponding computation time for experiments reported in Tables 4 and 5 is depicted in Tables 6 and 7 respectively. The experiments were performed on a DELL server with an 8 cores Intel Xeon X5675 3.07GHz processor and 100G

Table 6: Computation time on UCI datasets for the representative distance metrics approaches. The numbers in each cell represent training time(s)/test time(s) respectively.

		Global metric	Single local metric		Multiple local metrics					
		Xing	LFDA	LMNN	mLMNN	DANN	i-DANN	ADAMENN	i-ADAMENN	LDDM
balance-scale	10%	0.3558/0.0044	0.0029/0.0038	0.5933/0.0042	0.0941/0.0049	−/0.7233	−/2.7456	−/7.7971	−/20.7603	0.8862/0.0238
	80%	9.6484/0.0056	0.0105/0.0057	1.5798/0.0153	0.1935/0.0138	−/0.2017	−/0.9278	−/3.1324	−/8.1041	1.7260/0.0057
glass	10%	0.1729/0.0005	0.0054/0.0018	0.9264/0.0006	0.3521/0.0022	−/0.1950	−/0.5709	−/7.9225	−/23.3047	0.0847/0.0064
	80%	1.6422/0.0010	0.0026/0.0009	4.6951/0.0010	0.7653/0.0053	−/0.0579	−/0.2586	−/1.2399	−/3.4112	0.1952/0.0012
image	10%	3.2450/0.0437	0.0061/0.1306	1.5325/0.0455	0.9026/0.0351	−/6.4283	−/29.7896	−/330.5465	−/876.0339	4.5026/0.0657
	80%	208.6087/0.1917	0.0418/0.0938	6.1121/0.0776	2.9737/0.1870	−/1.0535	−/4.0571	−/438.7548	−/1309.1023	21.1064/0.0337
ionosphere	10%	0.3073/0.0026	0.0125/0.0023	4.9997/0.0023	0.3815/0.0032	−/0.3327	−/1.1508	−/20.0901	−/8.8713	0.1177/0.0043
	80%	1.9854/0.0184	0.0085/0.0029	6.1501/0.0038	0.6414/0.0074	−/0.1842	−/0.7420	−/3.4221	−/1.8950	0.4573/0.0016
soybean	40%	0.3236/0.0005	0.0038/0.0002	0.9865/0.0004	0.1196/0.0022	−/0.0179	−/0.0551	−/2.3844	−/0.9900	0.0144/0.0006
	80%	0.4297/0.0005	0.0019/0.0003	5.6716/0.0005	0.1425/0.0026	−/0.0060	−/0.0177	−/0.5183	−/0.2364	0.0227/0.0002
tic-tac-toe	10%	0.4584/0.0075	0.0014/0.0062	1.6829/0.0067	0.0950/0.0067	−/1.3941	−/5.6510	−/20.9977	−/43.4990	1.6074/0.0275
	80%	608.0292/0.0453	0.0279/0.0095	6.4564/0.0198	0.1770/0.0282	−/0.3650	−/1.4081	−/9.7731	−/28.9718	4.2377/0.0090
waveform	10%	12.2411/0.2928	0.0104/0.2482	1.0755/0.2812	0.5865/0.0932	−/15.4851	−/65.1324	−/855.5814	−/611.6011	145.4131/0.7518
	80%	829.7202/0.4492	0.4571/0.3928	8.9339/0.4198	6.7461/0.7184	−/2.9219	−/9.2873	−/1680.3573	−/1861.4028	623.8684/0.4934
iris	10%	0.1153/0.0031	0.0011/0.0028	0.5758/0.0031	0.1215/0.0022	−/0.1053	−/0.3295	−/1.2787	−/3.3414	0.0589/0.0049
	80%	0.8489/0.0051	0.0017/0.0031	1.9140/0.0032	0.1008/0.0026	−/0.0284	−/0.1259	−/0.2043	−/0.6039	0.1204/0.0008
wine	10%	0.1679/0.0006	0.0010/0.0004	0.5031/0.0005	0.3585/0.0022	−/0.1197	−/0.3897	−/4.8668	−/13.0849	0.0634/0.0053
	80%	1.5706/0.0011	0.0022/0.0007	4.4151/0.0009	0.5401/0.0033	−/0.0391	−/0.1857	−/0.6620	−/2.0059	0.1366/0.0009
wdbc	10%	0.3483/0.0030	0.0015/0.0019	0.8616/0.0031	0.2970/0.0051	−/0.7537	−/2.6102	−/28.8852	−/14.5917	0.5634/0.0139
	80%	8.6359/0.0049	0.0132/0.0037	0.0579/0.0109	0.8931/0.0191	−/0.3075	−/1.3105	−/8.2812	−/5.5584	1.4089/0.0031
car	10%	1.0257/0.0232	0.0028/0.0224	0.6065/0.0228	0.5038/0.0133	−/3.3769	−/15.2333	−/40.7142	−/109.4065	6.6843/0.0656
	80%	67.7757/0.0367	0.0894/0.0346	0.4174/0.0444	0.3354/0.0777	−/0.6319	−/2.6011	−/49.8702	−/148.4166	23.4052/0.0408

RAM. All algorithms are implemented in MATLAB. Both training and testing time are reported. The DANN, ADAMENN and kNN classifiers are "lazy leaners, hence training time is not applicable for those classifiers. Note that the training time of the LDDM approach in Table 6 and the proposed distance metric methods using ensemble framework in Table 7 is high, but their testing time is rather low, which empirically confirm what has been analyzed in Section 4. It is encouraging to point out that in our experiments the testing time for LDDM is comparable to the state-of-the-art efficient classifiers and its stableness in performance makes it applicable for complex real world applications.

7. Conclusion

In this paper, we present a local discriminative distance metrics (LDDM) learning algorithm for classification under the local learning framework. LDDM trains a set of local discriminative distance metrics according to different training samples and predicts a test sample by classifiers ensemble. LDDM performs well on the multimodal distribution problem and greatly reduces the influence of noise samples. We theoretically prove the convergence rate bound and the risk bound

Table 7: Computation time on UCI datasets for the the state-of-the-art classifiers and representative distance metrics approaches under ensemble framework. The numbers in each cell represent training time(s)/test time(s) respectively.

		Classifiers				Ensemble distance metric		
		kNN	Decision tree	Naive Bayes	SVM	Xing	LFDA	LMNN
balance-scale	10%	−/0.0309	0.0168/0.0063	0.0258/0.2455	0.0021/0.0020	1.5200/0.0200	0.8152/0.0223	1.3328/0.0221
	80%	−/0.0056	0.0359/0.0064	0.0252/0.0768	0.0083/0.0016	12.3733/0.0054	1.6028/0.0102	2.5549/0.0070
glass	10%	−/0.0098	0.0168/0.0077	0.0919/0.2496	0.0022/0.0011	0.8072/0.0111	0.1507/0.0111	0.9137/0.0110
	80%	−/0.0016	0.0272/0.0064	0.1104/0.0778	0.0045/0.0005	2.4737/0.0020	0.2312/0.0020	1.4948/0.0021
image	10%	−/0.1512	0.0295/0.0074	0.2742/0.2232	0.0134/0.0286	34.1132/0.2933	23.1882/0.2921	31.4615/0.2843
	80%	−/0.0465	0.0915/0.0069	0.2665/0.0771	0.4141/0.0207	314.7649/0.1509	78.2493/0.1627	121.2005/0.1861
ionosphere	10%	−/0.0169	0.0153/0.0063	0.1392/0.2429	0.0013/0.0023	11.0466/0.3133	0.2299/0.0080	4.3036/0.0083
	80%	−/0.0032	0.0311/0.0084	0.1369/0.0762	0.0069/0.0018	73.2879/0.0016	1.0480/0.0280	3.1444/0.0018
soybean	40%	−/0.0013	0.0181/0.0063	0.2743/0.2673	0.0015/0.0009	5.7415/0.1771	0.0346/0.0006	3.9483/0.0011
	80%	−/0.0003	0.0175/0.0064	0.2727/0.0758	0.0017/0.0002	0.8831/0.0003	0.0361/0.0012	4.0519/0.0003
tic-tac-toe	10%	−/0.0500	0.0204/0.0056	0.0360/0.2416	0.0032/0.0046	3.3813/0.0272	1.4729/0.0271	3.0261/0.0270
	80%	−/0.0101	0.0385/0.0050	0.0378/0.0948	0.0339/0.0036	556.2084/0.0085	3.2032/0.0088	5.9560/0.0099
waveform	10%	−/0.4666	0.0696/0.0110	0.1281/0.2800	0.0125/0.0814	121.3833/0.6829	104.7261/0.7042	109.3327/0.7150
	80%	−/0.2567	0.4735/0.0075	0.1476/0.0844	0.4664/0.0652	1499.7533/0.4752	545.5000/0.4541	655.7652/0.5579
iris	10%	−/0.0070	0.0128/0.0062	0.0241/0.2516	0.0005/0.0004	8.4097/0.6977	0.0491/0.0046	1.0363/0.0047
	80%	−/0.0011	0.0156/0.0061	0.0245/0.0887	0.0022/0.0013	13.3046/0.6850	0.0692/0.0007	0.9776/0.0009
wine	10%	−/0.0079	0.0134/0.0058	0.0769/0.2479	0.0020/0.0014	0.6282/0.0053	0.0704/0.0053	0.8044/0.0054
	80%	−/0.0013	0.0223/0.0067	0.0792/0.0792	0.0060/0.0011	1.7969/0.0010	0.0970/0.0009	2.6352/0.0011
wdbc	10%	−/0.0272	0.0157/0.0060	0.1201/0.2417	0.0028/0.0040	1.4807/0.0146	0.5788/0.0138	2.4588/0.0140
	80%	−/0.0049	0.0383/0.0050	0.1179/0.0786	0.0257/0.0034	11.0047/0.0037	1.1426/0.0033	4.6641/0.0047
car	10%	−/0.1005	0.0227/0.0064	0.0490/0.2491	0.0034/0.0071	11.5376/0.1088	9.4427/0.1130	10.1548/0.1091
	80%	−/0.0266	0.0420/0.0063	0.0519/0.0809	0.0610/0.0061	211.4799/0.0610	28.0524/0.0485	35.0583/0.0672

of local classifiers using local metrics by introducing a new concept of local domain based VC-dimension. We also prove the risk bound of final classifiers ensemble. We extensively evaluate LDDM using two synthetic datasets and eleven UCI datasets. The experiments show that the proposed method outperforms many state-of-the-art distance metric learning algorithms.

References

[1] P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1997) 711–720.

[2] L. Bottou, V. Vapnik, Local learning algorithms, Neural computation 4 (1992) 888–900.

[3] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 123–140.

[4] C.C. Chang, Generalized iterative relief for supervised distance metric learning, Pattern Recogn. 43 (2010) 2971–2981.

[5] C.C. Chang, A boosting approach for supervised mahalanobis distance metric learning, Pattern Recognition 45 (2012) 844–862.

[6] C. Domeniconi, Adaptive nearest neighbor classification using support vector machines, Proc. NIPS (2002).

[7] C. Domeniconi, D. Gunopulos, Locally adaptive metric nearest-neighbor classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 1281–1285.

[8] A. Frome, Y. Singer, F. Sha, J. Malik, Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification, 2007 IEEE 11th International Conference on Computer Vision (2007) 1–8.

[9] J. Goldberger, S. Roweis, G. Hinton, R. Salakhutdinov, Neighbourhood components analysis, Proc. NIPS (2005).

[10] T. Hastie, R. Tibshirani, Discriminant adaptive nearest neighbor classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (1996) 607–616.

[11] J.W. Sammon, A Nonlinear mapping for data structure analysis, IEEE Transactions on Computers 18 (1969) 401–409.

[12] R.E. Schapire, Y. Freund, P. Bartlett, W.S. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, The Annals of Statistics 26 (1998) 1651–1686.

[13] P. Schneider, M. Biehl, B. Hammer, Distance learning in discriminative vector quantization, Neural Computation 21 (2009) 2942–2969.

[14] P. Schneider, K. Bunte, H. Stiekema, B. Hammer, T. Villmann, M. Biehl, Regularization in matrix relevance learning, IEEE Transactions on Neural Networks 21 (2010) 831–840.

[15] M. Sugiyama, Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis, The Journal of Machine Learning Research 8 (2007) 1027–1061.

[16] V. Vapnik, L. Bottou, Local algorithms for pattern recognition and dependencies estimation, Neural Computation 5 (1993) 893–909.

[17] K.Q. Weinberger, L.K. Saul, Fast solvers and efficient implementations for distance metric learning, Proceedings of the 25th international conference on Machine learning - ICML '08 (2008) 1160–1167.

[18] K.Q. Weinberger, L.K. Saul, Distance Metric Learning for Large Margin Nearest Neighbor Classification, Journal of Machine Learning Research 10 (2009) 207–244.

[19] S. Xiang, F. Nie, C. Zhang, Learning a mahalanobis distance metric for data clustering and classification, Pattern Recogn. 41 (2008) 3600–3612.

[20] E. Xing, A. Ng, M. Jordan, S. Russell, Distance metric learning with application to clustering with side-information, Proc. NIPS (2003).

[21] L. Yang, R. Jin, Distance metric learning: A comprehensive survey, Michigan State Universiy (2006) 1–51.

[22] L. Yang, R. Jin, R. Sukthankar, Y. Liu, An efficient algorithm for local distance metric learning, in: Proceedings of the National Conference on Artificial Intelligence, volume 21, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 543.

[23] Y. Yu, J. Jiang, L. Zhang, Distance metric learning by minimal distance maximization, Pattern Recogn. 44 (2011) 639–649.

[24] T. Zhang, D. Tao, X. Li, J. Yang, Patch Alignment for Dimensionality Reduction, IEEE Transactions on Knowledge and Data Engineering 21 (2009) 1299–1313.

[25] W. Zhang, X. Xue, Z. Sun, H. Lu, Y.F. Guo, Metric learning by discriminant neighborhood embedding, Pattern Recogn. 41 (2008) 2086–2096.

[26] Z. Zhang, J. Kwok, D. Yeung, Parametric distance metric learning with label information, in: Proc. IJCAI.