An Efficient Algorithm for Local Distance Metric Learning

Liu Yang and **Rong Jin**
Michigan State University
Dept. of Computer Science & Engineering
East Lansing, MI 48824
{yangliu1, rongjin}@cse.msu.edu

Rahul Sukthankar
Intel Research & Carnegie Mellon
4720 Forbes Avenue, Suite 410
Pittsburgh, PA 15213
rahuls@cs.cmu.edu

Yi Liu
Michigan State University
Dept. of Computer Science & Engineering
East Lansing, MI 48824
liuyi3@cse.msu.edu

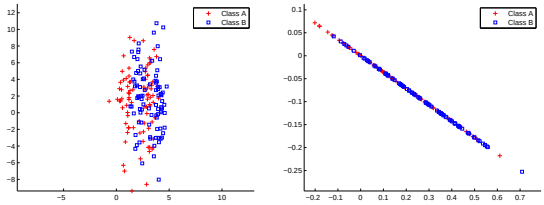
Abstract

Learning application-specific distance metrics from labeled data is critical for both statistical classification and information retrieval. Most of the earlier work in this area has focused on finding metrics that simultaneously optimize compactness and separability in a *global* sense. Specifically, such distance metrics attempt to keep all of the data points in each class close together while ensuring that data points from different classes are separated. However, particularly when classes exhibit multimodal data distributions, these goals conflict and thus cannot be simultaneously satisfied. This paper proposes a Local Distance Metric (LDM) that aims to optimize local compactness and local separability. We present an efficient algorithm that employs eigenvector analysis and bound optimization to learn the LDM from training data in a probabilistic framework. We demonstrate that LDM achieves significant improvements in both classification and retrieval accuracy compared to global distance learning and kernel-based KNN.

Introduction

Distance metric learning has played a significant role in both statistical classification and information retrieval. For instance, previous studies (Goldberger *et al.* 2005; Weinberger, Blitzer, & Saul 2006) have shown that appropriate distance metrics can significantly improve the classification accuracy of the K Nearest Neighbor (KNN) algorithm. In multimedia information retrieval, several papers (He *et al.* 2003; 2004; Muller, Pun, & Squire 2004) have shown that appropriate distance metrics, learned either from labeled or unlabeled data, usually result in substantial improvements in retrieval accuracy compared to the standard Euclidean distance. Most of the work in distance metrics learning can be organized into the following two categories:

- *Unsupervised distance metric learning*, or manifold learning. The main idea is to learn a underlying low-dimensional manifold where geometric relationships (e.g., distance) between most of the observed data points are preserved. Popular algorithms in this category include ISOMAP (Tenenbaum, de Silva, & Langford 2000), Local Linear Embedding (Saul & Roweis 2003), and the Laplacian Eigenmap (Belkin & Niyogi 2003).



(a) (b)

Figure 1: Multimodal data distributions prevent global distance metrics from simultaneously satisfying constraints on within-class compactness and between-class separability. (a) shows the original data distribution. (b) shows the data distribution adjusted by a global distance metric.

- *Supervised/semi-supervised distance metric learning*. Most approaches in this category attempt to learn metrics that keep data points within the same classes close, while separating data points from different classes. Examples include (Hastie & Tibshirani 1996; Domeniconi & Gunopulos 2002; Xing *et al.* 2003; Zhang, Tang, & Kwok 2005; Goldberger *et al.* 2005; Weinberger, Blitzer, & Saul 2006; Shalev-Shwartz, Singer, & Ng 2004).

This paper focuses on learning distance metrics in a supervised setting. Most of previous work in this area attempts to learn *global distance metrics* that keep all of the data points in each class close together, while ensuring that those from different classes remain separated. However, particularly when classes exhibit multimodal data distributions, these goals conflict and cannot be simultaneously satisfied. Figure 1(a) illustrates this point with a simple two-class example where the labeled data belong either to class A or class B (denoted by the positive signs and squares, respectively). Note that each class has two distinct modes, and one mode of each class is sandwiched between the two modes of the other class. Given such a layout, any global metric that attempts to bring the two modes of class A closer together will inadvertently separate the two modes of class B; conversely, bringing the two modes of class B together results in a separation of class A. Figure 1(b) is the distribution of Figure 1(a) adjusted by a global distance metric. Clearly, we see that bringing the modes of each class closer together collapses

the entire data set into a straight line where the data from the two classes is mixed together. It is not surprising that such a global distance metric leads to a significant degradation in classification accuracy for the K Nearest Neighbor (KNN) from 71.0% down to 55.5%. In order to solve the above problem, this paper presents a novel probabilistic framework that learns a *local distance metric* (LDM); rather than satisfying all of the pair-wise constraints, our algorithm focuses on the "local" pairs:

- bringing pairs from the same *mode* of a class closer;
- separating nearby pairs from different classes.

Imposing the notion of locality on the data pairs enables us to generate distance metrics that accommodate multiple modes for each class. Learning a local distance metric for the dataset shown in Figure 1(a) significantly increases the KNN classification accuracy from 71.0% up to 76.5%.

The key challenge in local distance metric learning is the chicken-and-egg dilemma: on one hand, to learn a local distance metric, we need to identify the local pairwise constraints; on the other hand, to identify the local pairwise constraints, we need to know the appropriate local distance metric. To resolve this problem, we propose a probabilistic framework for local distance metric learning that explicitly addresses the chicken-and-egg problem through an EM-like algorithm. Furthermore, this paper presents an algorithm based on eigenvector analysis and bound optimization to efficiently learn such local distance metrics.

In the remaining sections, we first present a brief introduction to the problem of supervised distance metric learning and reviews related work; Then we introduce the probabilistic framework for local distance metric learning and present an efficient algorithm for learning them from data. Finally experimental results are presented on two application domains (image classification and text categorization).

Supervised Distance Metric Learning

Unlike typical supervised learning, where each training example is annotated with its class label, the label information in distance metric learning is usually specified in the form of pairwise constraints on the data: (1) equivalence constraints, which state that the given pair are semantically-similar and should be close together in the learned metric; and (2) inequivalence constraints, which indicate that the given points are semantically-dissimilar and should not be near in the learned metric. Most learning algorithms try to find a distance metric that keeps all the data pairs in the equivalence constraints close while separating those in the inequivalence constraints.

The work of (Xing *et al.* 2003) formulates distance metric learning as a constrained convex programming problem. It learns a global distance metric that minimizes the distance between the data pairs in the equivalence constraints subject to the constraint that the data pairs in the ineqivalence constraints are well separated. This algorithm is further extended to the nonlinear case in (Kwok & Tsang 2003) by the introduction of kernels. In addition to general purpose algorithms for distance metric learning, several papers have presented approaches to learn appropriate distance metrics for the KNN classifier. The approach presented in (Domeniconi & Gunopulos 2002) tries to find feature weights that adapt to individual test examples. Although it attempts to address a similar problem as this paper, their hand-crafted local distance metric is unable to fully exploit the training data. The research most related to this paper is neighborhood component analysis (NCA) (Goldberger *et al.* 2005) and the large margin nearest neighbor classifier (Weinberger, Blitzer, & Saul 2006). Both focus on learning a distance metric from the local neighborhood, which is similar to the motivation of this paper. However, both of these attempt to learn complete distance metrics from the training data, which is computationally expensive and prone to overfitting (Weinberger, Blitzer, & Saul 2006); instead, our proposed method approximates the distance metric using the direct product of the principal eigenvectors extracted from both the labeled and the unlabeled data. Our approach has several important advantages. First, it significantly improves computational efficiency. Second, it enhances the robustness of the learned distance metrics, which is critical for problems such as text classification that involve a large number of features. Third, instead of using general-purpose optimization algorithms, such as greedy ascent or semi-definite programming, we present an efficient algorithm that is specifically targeted to our optimization problem, and is guaranteed to converge to a local optimum. Finally, and most importantly, our approach provides a natural means of effectively exploiting *unlabeled* data, which can further enhance the quality of the learned distance metrics.

Supervised Local Distance Metric Learning

This section introduces the notion of a *local distance metric*, (LDM). As discussed in the introduction, it may be impossible to simultaneously satisfy all of the given constraints in a global sense. We present a probabilistic framework for distance metric learning that focuses on "local" constraints. The essential idea is to first identify a subset of constraints that only involve points that are relatively close together. Then, we can learn a distance metric that satisfies this subset of constraints. To accomplish this, we employ an iterative procedure based on the bound optimization algorithm (Salakhutdinov & Roweis 2003). Specifically, we initialize our algorithm by using the Euclidean metric to identify the initial set of local constraints. Then we alternately iterate between the step of local distance metric learning and the step of refining the subset of local constraints until convergence is reached.

A Probabilistic Framework for LDM

Our probabilistic framework is based on leave-one-out evaluation using the kernel-based KNN classifier. To facilitate our discussion, we first introduce some necessary notation. Let $\mathcal{C} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ be a collection of data points, where n is the number of samples and each $\mathbf{x}_i \in \mathbb{R}^m$ is a vector of m features. Let the set of equivalence constraints and the set of inequivalence constraints denoted by

$$\begin{aligned} \mathcal{S} &= \{(\mathbf{x}_i, \mathbf{x}_j) |\ \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to the same class}\} \\ \mathcal{D} &= \{(\mathbf{x}_i, \mathbf{x}_j) |\ \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to different classes}\} \end{aligned}$$

respectively. Let the distance metric be denoted by matrix $\mathbf{A} \in \mathbf{R}^{m\times m}$, and the distance between two points $\mathbf{x}$ and $\mathbf{y}$ be expressed by $d^2_{\mathbf{A}}(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|^2_{\mathbf{A}} = (\mathbf{x}-\mathbf{y})^\top \mathbf{A}(\mathbf{x}-\mathbf{y})$.

Consider a data point $\mathbf{x}$ that is involved in one of the constraints in the set $\mathcal{S}$ and the set $\mathcal{D}$. Let $\Phi_S(\mathbf{x}) = \{\mathbf{x}_i|(\mathbf{x},\mathbf{x}_i) \in \mathcal{S}\}$ include all of the data points that pair with $\mathbf{x}$ in the equivalence constraints. Similarly, let $\Phi_D(\mathbf{x}) = \{\mathbf{x}_i|(\mathbf{x},\mathbf{x}_i) \in \mathcal{D}\}$ include all of the data points that pair with $\mathbf{x}$ in the inequivalence constraints. Now, according to the kernel-based KNN, the probability of making the right prediction for $\mathbf{x}$, denoted by $\Pr(+|\mathbf{x})$, can be written as

$$\Pr(+|\mathbf{x}) = \frac{\sum\limits_{\mathbf{x}_i\in\Phi_S(\mathbf{x})} f(\mathbf{x},\mathbf{x}_i)}{\sum\limits_{\mathbf{x}_i\in\Phi_S(\mathbf{x})} f(\mathbf{x},\mathbf{x}_i) + \sum\limits_{\mathbf{x}_j\in\Phi_D(\mathbf{x})} f(\mathbf{x},\mathbf{x}_j)} \quad (1)$$

where the kernel function $f(\mathbf{x},\mathbf{x}')$ is defined as

$$f(\mathbf{x},\mathbf{x}') = \exp\left(-\|\mathbf{x}-\mathbf{x}'\|^2_{\mathbf{A}}\right). \quad (2)$$

Using leave-one-out estimation and Equation 1, we can write the log likelihood for both $\mathcal{S}$ and $\mathcal{D}$ as

$$\begin{aligned}\mathcal{L}_l(\mathbf{A}) &= \sum_{\mathbf{x}\in\mathcal{T}} \log\Pr(+|\mathbf{x}) \quad (3)\\ &= \sum_{x\in\mathcal{T}} \log\left(\frac{\sum\limits_{\mathbf{x}_i\in\Phi_S(\mathbf{x})} f(\mathbf{x},\mathbf{x}_i)}{\sum\limits_{\mathbf{x}_i\in\Phi_S(\mathbf{x})} f(\mathbf{x},\mathbf{x}_i) + \sum\limits_{\mathbf{x}_j\in\Phi_D(\mathbf{x})} f(\mathbf{x},\mathbf{x}_j)}\right)\end{aligned}$$

where the set $\mathcal{T} = \{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$ includes all of the data points involved in the constraints given in sets $\mathcal{S}$ and $\mathcal{D}$. Using maximum likelihood estimation, we cast local distance metric estimation into the following optimization problem

$$\begin{aligned}\max_{\mathbf{A}\in\mathbf{R}^{m\times m}} \quad & \mathcal{L}_l(\mathbf{A})\\ \text{s. t.} \quad & \mathbf{A} \succeq 0. \end{aligned} \quad (4)$$

Remark: Note that in Equation 4, it is the *ratio* between the kernel function $f(\mathbf{x},\mathbf{x}_i)$ evaluated at different data points $\mathbf{x}_i$ that determines the probability $\Pr(+|\mathbf{x})$. When a data point $\mathbf{x}_i$ is relatively far from $\mathbf{x}$ compared to other data points in $\Phi(\mathbf{x})_S$ and $\Phi(\mathbf{x})_D$, its kernel value $f(\mathbf{x},\mathbf{x}_i)$ will be relatively smaller than the kernel value of other data points. Hence, data pairs that are far away from each other will have a smaller impact on the objective function $\mathcal{L}_l(\mathbf{A})$ than data pairs that are close to each other.

The Optimization Algorithm

The difficulty with solving Equation 4 lies in the positive semi-definitive constraint $\mathbf{A} \succeq 0$. To simplify our computation, we model the matrix $\mathbf{A}$ using the eigenspace of the training instances. Let $\mathbf{M} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_j^T$ be the pairwise correlation matrix, and $\{\mathbf{v}_i\}_{i=1}^K$ be the top K ($K \le m$) eigenvectors of the matrix $\mathbf{M}$. Then $\mathbf{A}$ is assumed to be a linear combination of the top K eigenvectors:

$$\mathbf{A} = \sum_{i=1}^K \gamma_i \mathbf{v}_i\mathbf{v}_i^T, \gamma_i \ge 0,\ i = 1,\ldots,K \quad (5)$$

where $(\gamma_i,\ldots,\gamma_K)$ are the non-negative weights in the linear combination. Using the parametric form in Equation 5, the log likelihood function in Equation 3 simplifies to:

$$\begin{aligned}&\mathcal{L}^e_l(\{\gamma_i\}_{i=1}^K) = \\ &\sum_{x_i\in\mathcal{T}} \log\left(\sum_{\mathbf{x}_j\in\Phi_S(\mathbf{x}_i)} \exp\left(-\sum_{k=1}^K \gamma_k w^k_{i,j}\right)\right) \quad (6)\\ &- \sum_{x_i\in\mathcal{T}} \log\left(\begin{array}{c}\sum\limits_{\mathbf{x}_j\in\Phi_S(\mathbf{x}_i)} \exp\left(-\sum_{k=1}^K \gamma_k w^k_{i,j}\right) +\\ \sum\limits_{\mathbf{x}_j\in\Phi_D(\mathbf{x}_i)} \exp\left(-\sum_{k=1}^K \gamma_k w^k_{i,j}\right)\end{array}\right).\end{aligned}$$

where $w^k_{i,j} = ((\mathbf{x}_i-\mathbf{x}_j)^T\mathbf{v}_k)^2$. Optimizing the objective function in Equation 6 is challenging because this function is not concave. Hence, standard approaches, such as conjugate gradient and Newton's method, may not converge. We apply the bound optimization algorithm (Salakhutdinov & Roweis 2003) to search for the optimal local distance metric. The main idea is to divide the optimization procedure into multiple steps. In each iteration, we approximate the difference between the log likelihood of the current iteration and of the previous iteration by a concave function. Then, a standard convex programming technique, such as Newton's method, is applied to efficiently find the solution that maximizes the approximate difference. We iterate until the procedure converges at the local maximum. More specifically, for two consecutive iterations parameterized by $\{\gamma'_i\}_{i=1}^K$ and $\{\gamma_i\}_{i=1}^K$, the difference between their log likelihood functions, denoted by $\Delta(\{\gamma'_i\}_{i=1}^K,\{\gamma_i\}_{i=1}^K) = \mathcal{L}^e_l(\{\gamma_i\}_{i=1}^K) - \mathcal{L}^e_l(\{\gamma'_i\}_{i=1}^K)$, can be lower-bounded by the following expression:

$$\begin{aligned}&\Delta(\{\gamma'_i\}_{i=1}^K,\{\gamma_i\}_{i=1}^K)\\ &\ge \Delta_0(\{\gamma'_i\}_{i=1}^K) + \sum_{\mathbf{x}_i\in\mathcal{T}}\sum_{\mathbf{x}_j\in\Phi_S(\mathbf{x}_i)} \phi_{i,j}\sum_{k=1}^K \gamma_k w^k_{i,j}\\ &- \sum_{x_i\in\mathcal{T}} \log\left(\begin{array}{c}\sum\limits_{\mathbf{x}_j\in\Phi_S(\mathbf{x}_i)} \exp\left(-\sum_{k=1}^K \gamma_k w^k_{i,j}\right) +\\ \sum\limits_{\mathbf{x}_j\in\Phi_D(\mathbf{x}_i)} \exp\left(-\sum_{k=1}^K \gamma_k w^k_{i,j}\right)\end{array}\right),\end{aligned}$$

where

$$\phi_{i,j} = \frac{\frac{\exp\left(-\sum_{k=1}^K \gamma_k w^K_{i,j}\right)}{\sum_{\mathbf{x}_j\in\Phi_S(\mathbf{x}_i)}\exp\left(-\sum_{k=1}^K \gamma_k w^K_{i,j}\right)}}{1+\frac{\sum_{\mathbf{x}_j\in\Phi_D(\mathbf{x}_i)}\exp\left(-\sum_{k=1}^K \gamma_k w^K_{i,j}\right)}{\sum_{\mathbf{x}_j\in\Phi_S(\mathbf{x}_i)}\exp\left(-\sum_{k=1}^K \gamma_k w^K_{i,j}\right)}}, \quad (7)$$

and $\Delta_0(\{\gamma'_i\}_{i=1}^K)$ is a constant independent from the parameters γs. Extracting the part of the lower bound depending on γ, we have the following objective function for each iteration:

$$\begin{aligned}Q(\{\gamma_i\}_{i=1}^K) &= \sum_{\mathbf{x}_i\in\mathcal{T}}\sum_{\mathbf{x}_j\in\Phi_S(\mathbf{x}_i)} \phi_{i,j}\sum_{k=1}^K \gamma_k w^k_{i,j} \quad (8)\\ &- \sum_{x_i\in\mathcal{T}} \log\left(\begin{array}{c}\sum\limits_{\mathbf{x}_j\in\Phi_S(\mathbf{x}_i)} \exp\left(-\sum_{k=1}^K \gamma_k w^k_{i,j}\right) +\\ \sum\limits_{\mathbf{x}_j\in\Phi_D(\mathbf{x}_i)} \exp\left(-\sum_{k=1}^K \gamma_k w^k_{i,j}\right)\end{array}\right).\end{aligned}$$

Initialize Assign random values to $\{\gamma_i\}_{i=1}^K$.
Loop
- Compute $\phi_{i,j}$ for each equivalence constraint $(\mathbf{x}_i, \mathbf{x}_j)$ using Equation 7.
- Re-estimate the parameters $\{\gamma_i\}_{i=1}^K$ by optimizing function $Q(\{\gamma_i\}_{i=1}^K)$ in Equation 8.

Until $\{\gamma_i\}_{i=1}^K$ converges to a stable solution.

Figure 2: Algorithm for local distance metric learning

Note that the objective function $Q(\{\gamma_i\}_{i=1}^K)$ is a concave function in terms of all γs because of the convexity of the log-sum-of-exponential function (Boyd & Vandenberghe 2004). Hence, the optimal solution that maximizes $Q(\{\gamma_i\}_{i=1}^K)$ can be efficiently obtained using Newton's method. We refer to this algorithm as "**Local Distance Metric Learning**", or **LDM** for short. Figure 2 summarizes the detailed steps for automatically learning a local distance metric from pairwise constraints.

Remark 1: In the objective function $Q(\{\gamma_i\}_{i=1}^K)$, each pairwise equivalence constraint $(\mathbf{x}_i, \mathbf{x}_j)$ is weighted by $\phi_{i,j}$. As shown in Equation 7, the closer a data point $\mathbf{x}_j$ is to $\mathbf{x}_i$, the larger is its kernel value $\phi_{i,j}$. Hence, by multiplying $\phi_{i,j}$ with the corresponding equivalence constraint, we are able to weight the local constraints more than the constraints that involve pairs of distant points. As a result, the local constraints have more impact than other constraints on the optimal solution to the function $Q(\{\gamma_i\}_{i=1}^K)$. Thus, the step of computing $\phi_{i,j}$ can also be viewed as the step of identifying the local constraints based on the current distance metric.

Remark 2: The simplified form of the matrix $\mathbf{A}$ also allows us to exploit unlabeled training data. Rather than computing the matrix $\mathbf{M}$ based on the labeled data alone, we can incorporate additional unlabeled data $\mathcal{T}'(\mathbf{x}'_1, \ldots, \mathbf{x}'_{n'})$ using $\mathbf{M} = \frac{1}{n+n'}\left(\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T + \sum_{i=1}^{n'} \mathbf{x}'_i(\mathbf{x}'_i)^T\right)$. Note that the top K eigenvectors now depend on both the labeled $\mathcal{T}$ and the unlabeled data $\mathcal{T}'$. Our experiments show the significant improvement provided by unlabeled data for distance metric learning.

Application to Multi-Class Classification

This section discusses the application of distance metric learning to multi-class classification. Let $\mathcal{C}\{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$ be the training examples for multi-class learning, where $\mathbf{x} \in \mathbf{R}^m$ and $y_i \in \{1, \ldots, C\}$. We convert the training set $\mathcal{C}$ into pairwise equivalence and inequivalence constraints as follows:

$$\mathcal{S} = \{(\mathbf{x}_i, \mathbf{x}_j) | y_i = y_j\}, \quad \mathcal{D} = \{(\mathbf{x}_i, \mathbf{x}_j) | y_i \neq y_j\}.$$

These constraints are used to train the local distance metrics described above. To predict the class label for a test data point $\mathbf{x}$, we follow the kernel-based KNN algorithm by estimating the probability that $\mathbf{x}$ belongs to the j-th class as

$$\Pr(j|\mathbf{x}) = \frac{\sum_{i=1}^n \delta(j, y_i) \exp\left(-\|\mathbf{x} - \mathbf{x}_i\|_{\mathbf{A}}^2\right)}{\sum_{i=1}^n \exp\left(-\|\mathbf{x} - \mathbf{x}_i\|_{\mathbf{A}}^2\right)}.$$

Evaluation

This section presents experiments that evaluate the effectiveness of local distance metric learning. To this end, two evaluation metrics are used in our experiments:

- *Classification Accuracy*. We apply the multi-class learning approach outlined above with a 10-fold cross validation to estimate the average classification accuracy and its variance.
- *Retrieval accuracy*. In addition to classification accuracy, which is determined collectively by all of the training data points in the neighborhood of the test points, we evaluate the retrieval accuracy for a distance metric, which is more focused on individual training examples in the neighborhood of test points. In particular, for a test point $\mathbf{x}$, we rank all of the training data in ascending order of distance to $\mathbf{x}$. If the class label of the test point $\mathbf{x}$ is y, and the class labels of the ranked training examples are $(y'_1, \ldots, y'_n)$, then the retrieval accuracy at a given rank position k $(1 \leq k \leq n)$ is calculated as: $r(k|\mathbf{x}) = \frac{1}{k}\sum_{i=1}^k \delta(y, y'_i)$. The reported retrieval accuracy is averaged over 10-fold cross validation.

Experimental Data

Our experiments employ two real-world datasets:

- *Image Classification.* We randomly choose five categories from the COREL dataset and randomly select 100 examples from each category, resulting in an image collection of 500 images. Each image is represented by 36 different visual features that belong to three categories: color, edge, and texture.
- *Text Categorization.* We randomly select five categories from the Newsgroup dataset (Yang 1999) for text categorization and randomly select 100 documents for each category, resulting in 500 documents. The mutual information (Yang 1999) is used to identify the top 100 most informative words for the five classes.

Baseline Approaches

We compare the proposed method against three established approaches. The first is kernel-based KNN using the Euclidean distance metric. Given two data points $\mathbf{x}$ and $\mathbf{x}'$, their similarity under the RBF kernel is calculated as: $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{\lambda}\right)$. Parameter λ is determined by a cross validation using a $20/80$ split of the training data. The second baseline is the Support Vector Machine (SVM) using the RBF kernel (Vapnik 1998). The third baseline approach is based on the global distance metric learning algorithm (Xing *et al.* 2003). Specifically, we assume a logistic regression model for estimating the probability that two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same class:

$$\Pr(y_{i,j}|\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{1 + \exp\left(-y_{i,j}(\|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 - \mu)\right)}, \quad (9)$$

$$\text{where} \qquad y_{i,j} = \begin{cases} 1 & (\mathbf{x}_i, \mathbf{x}_j) \in \mathbf{S} \\ -1 & (\mathbf{x}_i, \mathbf{x}_j) \in \mathbf{D} \end{cases}$$

Table 1: Classification accuracy (%) on image data comparing our method (LDM) vs. Euclidean (EDM), probabilistic global metric (PGDM) and support vector machine (SVM).

Distance Metrics		Accuracy (%)
EDM KNN		79.8 ± 8.94
PGDM KNN		77.8 ± 8.60
SVM with RBF kernel		73.8 ± 6.14
LDM KNN	w/o unlabeled data	$\mathbf{81.0 \pm 6.78}$
	w/ unlabeled data	$\mathbf{82.0 \pm 8.77}$

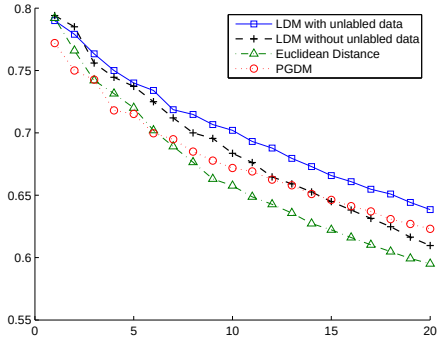


Figure 3: Retrieval accuracy for image data.

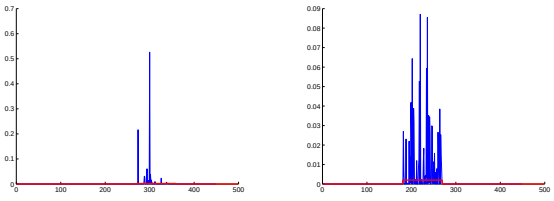
and the parameter μ is a threshold. Then, the overall log likelihood for both the equivalence constraints $\mathcal{S}$ and the inequivalence constraints $\mathcal{D}$ can be written as:

$$\begin{aligned}\mathcal{L}_g(\mathbf{A}, \mu) &= \log \Pr(\mathcal{S}) + \log \Pr(\mathcal{D}) \\ &= -\sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}} \log\left(1 + \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 + \mu\right)\right) \\ &\quad -\sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D}} \log\left(1 + \exp\left(\|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 - \mu\right)\right) \quad (10)\end{aligned}$$

Using the maximum likelihood estimation, the global distance metric $\mathbf{A}$ is determined by maximizing the log-likelihood in Equation 10. Similar to the local distance metric learning algorithm, we can assume the parametric form for the distance metric $\mathbf{A}$ as in Equation 5. We refer to this algorithm as "**Probabilistic Global Distance Metric Learning**", or **PGDM** for short.

Experimental Results for Image Classification

Classification Accuracy The classification accuracy using Euclidean distance, the probabilistic global distance metric (PGDM), and the local distance metric (LDM) is shown in Table 1. Clearly, LDM outperforms the other two algorithms in terms of the classification accuracy. Surprisingly, the classification accuracy of the PGDM algorithm is slightly worse than EDM, indicating that the global distance metric learned by the PGDM algorithm may be less effective than the straightforward Euclidean distance on this task. Finally, compared to the support vector machine, we observe that all of the KNN methods (using any distance metric) outperform the SVM noticeably. This may be because the decision



(a) (b)

Figure 4: Examples of distribution for $\phi_{i,j}$ in LDM (blue curves). In both (a) and (b), the red curves show the uniform distribution of the posterior probability in PGDM.

boundaries in image classification are complicated (Goh, Chang, & Cheng 2001) and difficult to capture using SVMs.

Retrieval Accuracy The retrieval accuracy results on the image data are shown in Figure 3. First, we observe that both of the learned distance metrics achieve significantly better retrieval accuracy than the Euclidean distance metric for most ranks. It is interesting to observe that, for the first few ranks (ranks 1 to 4), the retrieval accuracy of the global distance metric learned by PGDM is slightly worse than the Euclidean distance metric. Only after rank 5 does the global distance metric start to outperform the Euclidean distance metric. This observation can be explained by the tradeoff between the *local* and *global* compactness. Since a global distance metric attempts to simultaneously satisfy global compactness and global separability, it may sacrifice local compactness for improved global separability. This tradeoff leads to PGDM's poor performance for the first few ranks. In contrast, we note that the retrieval accuracy based on the local distance metric is always better than that based on the Euclidean distance metric. Second, we observe that the local distance metric achieves higher retrieval accuracy than the global one until rank 15. This is consistent with the motivation of LDM — to learn a metric from the local pairwise constraints where data points are not far apart from each other. To highlight this point, we show the examples of the distribution of $\phi_{i,j}$ in Figure 4. In both panels of Figure 4, the blue curves show us that, among all the same-labeled pairs, $\phi_{i,j} = 0$ for most of the pairs in LDM, while $\phi_{i,j}$ is uniformly distributed for PGDM, as illustrated by the flat red curves.

Incorporation of Unlabeled Data We also incorporate unlabeled data into the local distance metric learning algorithm, by randomly selecting 1500 images from 15 categories of the COREL database in addition to the 500 images for classification. We estimate the top eigenvectors based on the mixture of labeled and unlabeled images, and these eigenvectors are used to learn the local distance metric. The classification accuracy and the retrieval accuracy of the local distance metric learning with unlabeled data are presented in Table 1 and Figure 3. We observe that both the classification and retrieval accuracy improve noticeably when unlabeled data is available.

Experimental Results for Text Categorization

Table 2 shows the classification accuracy for the three distance metrics. We clearly see that LDM achieves signifi-

Table 2: Classification accuracy (%) on text comparing our method (LDM) vs. Euclidean (EDM), probabilistic global metric (PGDM) and support vector machine (SVM).

Distance Metrics	Accuracy (%)
EDM KNN	91.8 ± 4.90
PGDM KNN	90.8 ± 4.24
SVM	96.8 ± 2.70
LDM KNN with unlabeled data	95.2 ± 1.00

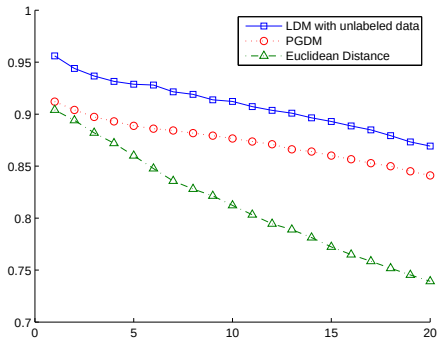


Figure 5: Retrieval accuracy for the text data.

cantly better classification accuracy than the other two distance metrics, and a similar classification accuracy as SVM (which is regarded as the best method for text classification (Joachims 1998)). Figure 5 presents the text retrieval accuracy of the three distance metrics for the top 20 ranks. Again, we observe that the retrieval accuracy of the local distance metric is significantly better than that of the global distance metric and the Euclidean distance metric. It is interesting to note that the retrieval accuracy of LDM at rank 10 compares favorably to that of the other algorithms at the first rank!

Conclusion

This paper proposes a probabilistic framework for local distance metric learning. It differs from the existing approaches in that only the local pairwise constraints impact the calculation of the distance metric. Unlike existing algorithms that must learn a full distance metric from the near neighbors of training examples, the proposed local distance metric learning algorithm avoids the computational difficulty by employing eigenvector analysis. Furthermore, an efficient learning algorithm, based on bound optimization, is employed to automatically learn local distance metrics from pairwise constraints. Experiments on two real-world applications show significant gains over both Euclidean and global learned distance metrics.

References

Belkin, M., and Niyogi, P. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* 15(6).

Boyd, S., and Vandenberghe, L. 2004. *Convex Optimization*. New York, NY, USA: Cambridge University Press.

Domeniconi, C., and Gunopulos, D. 2002. Adaptive nearest neighbor classification using support vector machines. *Proc. NIPS*.

Goh, K.; Chang, E. Y.; and Cheng, K.-T. 2001. SVM binary classifier ensembles for multi-class image classification. In *Proc. CIKM 2001*.

Goldberger, J.; Roweis, S.; Hinton, G.; and Salakhutdinov, R. 2005. Neighbourhood components analysis. In *Proc. NIPS*.

Hastie, T., and Tibshirani, R. 1996. Discriminant adaptive nearest neighbor classification. *IEEE Pattern Analysis and Machine Intelligence* 18(6).

He, X.; King, O.; Ma, W.-Y.; Li, M.; and Zhang, H. J. 2003. Learning a semantic space from user's relevance feedback for image retrieval. *IEEE Trans. Circuits and Systems for Video Technology* 13(1).

He, J.; Li, M.; Zhang, H.-J.; Tong, H.; and Zhang, C. 2004. Manifold ranking based image retrieval. In *Proc. ACM Multimedia*.

Joachims, T. 1998. Text categorization with suport vector machines: Learning with many relevant features. In *Proc. European Conference on Machine Learning*.

Kwok, J. T., and Tsang, I. W. 2003. Learning with idealized kernels. In *Proc. International Conference on Machine Learning*.

Muller, H.; Pun, T.; and Squire, D. 2004. Learning from user behavior in image retrieval: Application of market basket analysis. *International Journal of Computer Vision* 56(1–2).

Salakhutdinov, R., and Roweis, S. T. 2003. Adaptive overrelaxed bound optimization methods. In *Proc. International Conference on Machine Learning*.

Saul, L. K., and Roweis, S. T. 2003. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research* 4.

Shalev-Shwartz, S.; Singer, Y.; and Ng, A. Y. 2004. Online and batch learning of pseudo-metrics. In *Proc. International Conference on Machine Learning*, 94. New York, NY, USA: ACM Press.

Tenenbaum, J.; de Silva, V.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290.

Vapnik, V. N. 1998. *Statistical Learning Theory*. Wiley.

Weinberger, K.; Blitzer, J.; and Saul, L. 2006. Distance metric learning for large margin nearest neighbor classification. In *Proc. NIPS*.

Xing, E.; Ng, A.; Jordan, M.; and Russell, S. 2003. Distance metric learning with application to clustering with side-information. In *Proc. NIPS*.

Yang, Y. 1999. An evaluation of statistical approaches to text categorization. *Inf. Retr.* 1(1-2).

Zhang, K.; Tang, M.; and Kwok, J. T. 2005. Applying neighborhood consistency for fast clustering and kernel density estimation. In *Proc. Computer Vision and Pattern Recognition*.