An information geometry approach for distance metric learning

Shijun Wang
Dept. of Radiology and Imaging Sciences
National Institutes of Health

Rong Jin
Dept. of Computer Science and Engineering
Michigan State University

Abstract

In this paper, we propose a framework for metric learning based on information geometry. The key idea is to construct two kernel matrices for the given training data: one is based on the distance metric and the other is based on the assigned class labels. Inspired by the idea of information geometry, we relate these two kernel matrices to two Gaussian distributions, and the difference between the two kernel matrices is then computed by the Kullback-Leibler (KL) divergence between the two Gaussian distributions. The optimal distance metric is then found by minimizing the divergence between the two distributions. We present two metric learning algorithms, one for linear distance metric and one for nonlinear distance using a kernel function. Unlike many existing algorithms for metric learning that require solving a non-trivial optimization problem and are computationally expensive when the data dimension is high, the proposed algorithms have a closed-form solution and are computationally more efficient. Extensive experiments with data classification and face recognition show that the proposed algorithms are comparable to or better than the state-of-the-art algorithms for metric learning.

1 INTRODUCTION

Metric learning is an important problem in machine learning and pattern recognition. The performance of algorithms for data classification and clustering often depends heavily on the availability of a good metric. In addition, metric learning has found application in a number of real-world problems, including face recognition, visual object recognition, and automated speech recognition.

The objective of metric learning is to learn an optimal mapping, either linear or nonlinear, in the original feature space or the reproducing kernel Hilbert space, from training data. A number of algorithms have been proposed to learn the distance metric from the labeled data. They can be classified into the categories of unsupervised metric learning and supervised metric learning, depending on whether or not label or side-information is used to learn the optimal metric. Unsupervised distance metric learning, or sometimes referred to as manifold learning, aims to learn a underlying low-dimensional manifold where the distance between most pairs of data points are preserved. Example algorithms in this category include ISOMAP (Tenenbaum et al., 2000) and Local Linear Embedding (LLE) (Saul & Roweis, 2003). Supervised metric learning attempts to learn distance metrics that (a) keep data points within the same classes close, and (b) separate data points from different classes far away. Example algorithms in this category include (Xing et al., 2002; Shental et al., 2002; Weinberger et al., 2005; Globerson & Roweis, 2005; Tsang et al., 2005; Yang et al., 2006; Davis et al., 2007). Overall, empirical studies showed that supervised metric learning algorithms usually outperform unsupervised ones by exploiting either the label information or the side information presented in pairwise constraints. However, despite extensive studies, most of the existing algorithms for metric learning require solving a non-trivial optimization problem, and therefore are computationally expensive, particularly when the data dimension is high.

In this paper, we propose a framework for metric learning that is based on the idea of information geometry. The key idea is to construct two Gaussian dis-

Appearing in Proceedings of the 12^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2009, Clearwater Beach, Florida, USA. Volume 5 of JMLR: W&CP 5.

tributions, one based on the distance metric and the other based on the class labels assigned to the training data. The difference between the distance metric and the assigned class labels is measured by the KL divergence two Gaussian distributions. The optimal metric is found by minimizing the KL divergence between two distributions. Based on this idea, we present two algorithms for metric learning, one for linear distance metric, and the other for nonlinear distance metric with the introduction of a kernel function. We show that, for both problems, we can find the closed-form solutions, which result in efficient computation of the distance metric. Our extensive empirical study shows that the proposed algorithms are comparable to and better than the state-of-the-art algorithms for metric learning. Our study also reveals that the proposed algorithms are in general computationally efficient compared to the state-of-the-art approaches for metric learning.

2 RELATED WORK

In this section, we briefly review the existing work on supervised metric learning. Most of these algorithms are designed to learn either from the class labeling information or the side information that is usually cast in the form of pairwise constraints (i.e., must-link constraints and cannot-link constraints). In (Xing et al., 2002), the authors proposed to learn a distance metric from the pairwise constraints. The optimal kernel is found to minimize the distance between data points in must-link constraints and simultaneously maximize the distance between data points in cannot-link constraints. Relevance component analysis (Shental et al., 2002) is another popular approach for distance metric learning. Data points in the same classes are grouped in so called chunklets, and the distance metric is computed based on the covariance matrix that is estimated from each chunklet. Goldberger et al. presented an algorithm, termed Neighborhood Component Analysis, that combines distance metric learning with k-nearest neighbor (kNN) classification (Goldberger et al., 2005). It was extended in the work of maximum-margin nearest neighbor (LMNN) classifier (Weinberger et al., 2005) through a maximum margin framework. Globerson et al. (Globerson & Roweis, 2005) presented an algorithm for metric learning that aims to collapse data samples in the same class into a single point and samples belong to different classes far apart. J. Davis et al. presented an information-theoretic based approach for metric learning (Davis et al., 2007). Local Fisher Discriminant Analysis (Sugiyama, 2006) extends classical LDA to the case when the side information is in the form of pairwise constraints. Finally, for more information about metric learning, we refer the readers to a recent survey on this subject (Yang & Jin, 2006).

3 INFORMATION GEOMETRY OF POSITIVE DEFINITE MATRICES

Information geometry studies probability and information from the viewpoint of differential geometry (Amari & Nagaoka, 2000). It treats a space of probabilities as a differential manifold endowed with a Riemannian metric and a family of affine connections. In order to learn a distance metric, in this section, we follow the work (Tsuda et al., 2003), and introduce the information geometry in the space of positive definite matrices. To relate a positive definite matrix P of size $d \times d$ to a probability distribution, we treat P as the covariance matrix of a Gaussian distribution, i.e.,

$$\Pr(x|P) = \frac{1}{(2\pi)^{d/2}\,|P|^{1/2}} \exp\left(-\frac{1}{2}x^\top P^{-1}x\right),$$

where $x \in \mathbb{R}^d$. In the above, we assume that the mean of the Gaussian distribution is zero. By defining

$$r(x) = -\left(\frac{1}{2}x_1^2, \ldots, \frac{1}{2}x_d^2, x_1x_2, \ldots, x_{d-1}x_d\right)^\top$$
$$\theta = \left(\left[P^{-1}\right]_{11}, \ldots, \left[P^{-1}\right]_{dd}, \left[P^{-1}\right]_{12}, \ldots, \left[P^{-1}\right]_{d-1,d}\right)^\top,$$

the Gaussian distribution can be expressed in the canonical form of an exponential family, i.e.,

$$\Pr(x|\theta) = \exp\left(\theta^T r(x) - \psi(\theta)\right)$$

where $\psi(\theta)$ is the logarithm of the partition function. θ is usually referred to as the natural parameter, which provides a coordinate system (i.e., e-coordinate system) for specifying a positive definite matrix. The expectation of elements in $r(x)$ provides another coordinate system called η-coordinate system (Amari & Nagaoka, 2000).

Given two positive definite (PD) matrices P and Q, we define two Gaussian distributions, denoted by $\Pr(x|P)$ and $\Pr(x|Q)$. The distance between the two positive definite matrices P and Q, denoted by $d(P\|Q)$, can be derived by the Kullback-Leibler (KL) divergence between two distributions $\Pr(x|P)$ and $\Pr(x|Q)$, i.e.,

$$\begin{aligned} d(P\|Q) &= KL(\Pr(x|P)\|\Pr(x|Q)) \\ &= \int dx \Pr(x|P) \log \frac{\Pr(x|P)}{\Pr(x|Q)} \end{aligned} \quad (1)$$

The following theorem allows us to compute $d(P\|Q)$ in a closed form.

Theorem 1. *The distance between two positive definite matrices $P \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^{n\times n}$, defined in (1), is equal to the following expression:*

$$d(P\|Q) = \frac{1}{2}\left(tr(Q^{-1}P) + \log|Q| - \log|P| - n\right) \quad (2)$$

The following proposition relates the distance function defined in (2) to the Bregman distance function that is widely used in the study of information theory.

Proposition 1. *The distance function in (2) is equivalent to the following Bregman distance function, i.e.,*

$$d_B(Q\|P) = \phi(Q) - \phi(P) - tr((Q-P)^\top \nabla\phi(P)) \quad (3)$$

where $\phi(P) = -\frac{1}{2}\log|P|$.

Finally, the analysis below reveals the relation between the matrix distance function in (2) and the Wishart distribution. Let Q be the scale matrix, and the Wishart distribution of degree q, denoted by $\Pr(P|Q,q)$, is expressed as

$$\Pr(P|Q,q) = \frac{|P|^{\frac{q-n-1}{2}}}{2^{\frac{qn}{2}}|Q|^{\frac{q}{2}}\Gamma_n\left(\frac{q}{2}\right)} \exp\left(-\frac{1}{2}\mathrm{tr}(Q^{-1}P)\right) \quad (4)$$

The negative log-likelihood $\log\Pr(P|Q,q)$, i.e.,

$$\log\Pr(P|Q,q) \propto \frac{1}{2}\left(\mathrm{tr}(Q^{-1}P) + q\log|Q| - (q-n-1)\log|P|\right),$$

is clearly similar to the distance function in (2) except for a different weight is assigned to $\log|P|$ and $\log|Q|$ in the above expression.

4 DISTANCE METRIC LEARNING

In this section, we present a general framework for distance metric learning. The key idea is to first construct two kernel matrices for the given training data, one based on the distance metric to be learned and the other based on the assigned class labels. We then search for the distance metric that minimizes the distance between the two kernel matrices defined in (2). We first present the framework of supervised distance metric learning for a linear distance function, followed by the extension to nonlinear distance function with the introduction of a kernel function.

4.1 LEARNING LINEAR DISTANCE METRIC

Let $X = (x_1, \ldots, x_n)$ denote the collection of input patterns for n training examples. Each $x_i \in \mathbb{R}^m$ is a vector of m dimensions, and therefore X is a matrix of size $m \times n$. Let C be the number of classes, and $Y = (y_1, \ldots, y_n)$ denote the class labels assigned to the n training examples. Each $y_i = (y_i^1, \ldots, y_i^C) \in \{0,1\}^C$ is a binary vector of C elements. In our study, we assume each example is assigned to one and only one class, and therefore have $y_i^\top \mathbf{1} = 1$ where $\mathbf{1}$ is a vector of all ones. Following (Cristianini et al., 2002; Kwok & Tsang, 2003), we introduce so called "ideal kernel", denoted by K_D, that is computed as

$$K_D = Y^\top Y \quad (5)$$

Since K_D is a singular matrix when $C < n$, we further smooth K_D with an identify matrix I_n, i.e.,

$$\bar{K}_D = Y^\top Y + \lambda I_n \quad (6)$$

where $\lambda > 0$ is the smoother parameter.

In addition, we can construct another kernel matrix based on the input pattern X and the distance metric A. We define $M = A^{1/2}$. It is well known that the introduction of distance metric A is equivalent to a linear transform that maps x to Mx with $M = A^{1/2}$. We thus construct a linear kernel K_X as

$$K_X = (MX)^\top(MX) = X^\top AX \quad (7)$$

We then search for the distance metric A that minimizes the matrix distance $d(\bar{K}_D\|K_X)$ defined in (2), i.e.,

$$\begin{aligned} A &= \arg\min_{A\succeq 0} d(K_X\|\bar{K}_D) \\ &= \arg\min_{A\succeq 0} \mathrm{tr}(\bar{K}_D^{-1}X^\top AX) - \log|A| \end{aligned} \quad (8)$$

Proposition 2. *The optimal solution to (8) is*

$$A = \left(X\bar{K}_D^{-1}X^\top\right)^{-1} \quad (9)$$

It is easy to verify the result in the above proposition. The analysis below aims to provide in-depth understanding of the expression for distance metric A in (9). To this end, we introduce the notation z_k to represent the kth row of matrix Y, which represents the assignment of the kth class to all n examples. We furthermore introduce s_k to represent the number of training examples assigned the kth class, i.e., $s_k = |z_k|_1$. Using these notations, we have the following proposition for $\bar{K}_D^{-1}$.

Proposition 3.

$$\begin{aligned} \bar{K}_D^{-1} &= \left(Y^\top Y + \lambda I\right)^{-1} \\ &= \frac{1}{\lambda}\left(I - \sum_{k=1}^{C}\frac{z_k z_k^\top}{\lambda + s_k}\right) \end{aligned} \quad (10)$$

The result in the above proposition follows directly the fact that any z_i and z_j are orthogonal to each other, i.e., $z_i^\top z_j = 0$ for $i \neq j$. Using the result in Proposition 3, we have the following theorem for A.

Theorem 2. *The distance metric A in (9) can also be expressed as follows*

$$A = \lambda \left(\sum_{k=1}^{C} s_k \left[\Sigma_k + \frac{\lambda \bar{x}_k \bar{x}_k^T}{(\lambda + s_k)} \right] \right)^{-1} \tag{11}$$

where $\bar{x}_k$ and Σ_k are the mean and the covariance matrix for the input patterns in the kth class, respectively

$$\bar{x}_k = \frac{1}{s_k} X z_k, \; \Sigma_k = \frac{1}{s_k} \sum_{i=1}^{n} y_i^k \left[x_i - \bar{x}_k \right] \left[x_i - \bar{x}_k \right]^\top \tag{12}$$

Proof of the above theorem can be found in the Appendix A. As revealed in the above theorem, the inverse of the distance matrix A, is the weighted sum of the covariance matrices Σ_k of all the classes, smoothed by the centers of each class.

4.2 LEARNING METRIC WITH NONLINEAR KERNELS

In this subsection, we extend the above analysis to the nonlinear distance metric learning with the introduction of a kernel function. We introduce a nonlinear kernel function $\kappa(x, x') : \mathbb{R}^m \times \mathbb{R}^m \mapsto \mathbb{R}$. This kernel function defines a mapping $\Phi : \mathbb{R}^m \mapsto \mathcal{H}_\kappa$, i.e., $x \in \mathbb{R}^m \rightarrow \Phi(x) \in \mathcal{H}_\kappa$. We denote by $X_1 = (\Phi(x_1), \ldots, \Phi(x_n))$ the new data representation resulting from the kernel function, and by $K \in \mathbb{R}^{n \times n}$ the kernel matrix with $K_{i,j} = \kappa(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$ for $i, j = 1, \ldots, n$. We then introduce the linear operator $M : \mathcal{H}_\kappa \mapsto \mathbb{R}^n$, which results in another data representation $X_2 = MX_1 = (M\Phi(x_1), M\Phi(x_2), \ldots, M\Phi(x_n))$. The resulting kernel matrix based on the transformed representation X_2, denoted by K_X, is computed as

$$K_X = X_2^\top X_2 = X_1^\top M^\top M X_1 = X_1^\top A X_1 \tag{13}$$

where $A : \mathcal{H}_\kappa \mapsto \mathcal{H}_\kappa$ is a linear operator that represents a metric in the reproducing kernel Hilbert space $\mathcal{H}_\kappa$. Using the result in Proposition 2, we have

$$A = (X_1 \bar{K}_D^{-1} X_1^\top)^{-1} \tag{14}$$

Since the operator $X_1 \bar{K}_D^{-1} X_1$ may not be a one-to-one mapping, and as a result, its inverse operator A may not be well defined, we further smooth the expression in (14) as follows

$$A = \left(X_1 \bar{K}_D^{-1} X_1^\top + \lambda I_\kappa \right)^{-1} \tag{15}$$

where I_κ is the identity operator in the space of $\mathcal{H}_\kappa$. This linear operator A essentially defines a new kernel function, denoted by $\hat{\kappa} : \mathbb{R}^m \times \mathbb{R}^m \mapsto \mathbb{R}$. The following theorem gives the explicit expression for the new kernel function $\hat{\kappa}(x, x')$.

Theorem 3.

$$\begin{aligned} \hat{\kappa}(x, x') &= \langle \Phi(x), A\Phi(x') \rangle \\ &= \frac{1}{\lambda} \left(\kappa(x, x') - \mathbf{k}(x)^\top \left[\lambda \bar{K}_D + K \right]^{-1} \mathbf{k}(x') \right) \end{aligned} \tag{16}$$

where $\mathbf{k}(x) = (\kappa(x_1, x), \ldots, \kappa(x_n, x))^\top$.

Proof of this theorem can be found in Appendix B.

Corollary 4.

$$\begin{aligned} K_X &= \frac{1}{\lambda} \left(K - K \left[\lambda \bar{K}_D + K \right]^{-1} K \right) \\ &= \left(\bar{K}_D^{-1} + \lambda K^{-1} \right)^{-1} \end{aligned} \tag{17}$$

Proof.

$$\begin{aligned} K_X &= \frac{1}{\lambda} \left(K - K \left[\lambda \bar{K}_D + K \right]^{-1} K \right) \\ &= \frac{1}{\lambda} K^{1/2} \left(I - \left[K^{-1/2} \lambda \bar{K}_D K^{-1/2} + I \right]^{-1} K \right) K^{1/2} \\ &= \frac{1}{\lambda} K^{1/2} \left(K^{-1/2} \left[K^{-1} + (\lambda \bar{K}_D)^{-1} \right]^{-1} K^{-1/2} \right) K^{1/2} \\ &= (\bar{K}_D^{-1} + \lambda K^{-1})^{-1} \end{aligned}$$

□

As revealed in the above corollary, the new kernel matrix K_X is the harmonic mean of K and $\lambda \bar{K}_D$.

4.3 RELATION TO KERNEL RELEVANT COMPONENT ANALYSIS

Relevance component analysis was first presented in (Shental et al., 2002; Bar-Hillel et al., 2003), and was later on extended to a kernel version (KRCA) in (Tsang et al., 2005). In KRCA, the new kernel function learned from the side information, is computed as

$$\begin{aligned} &\hat{\kappa}(x, x') \\ &= \frac{1}{\lambda} \left(\kappa(x, x') - \mathbf{k}(x)^T \left[H \left(\lambda I + KH \right)^{-1} \right] \mathbf{k}(x') \right) \end{aligned} \tag{18}$$

where H is computed as

$$H = \frac{1}{n} \sum_{c=1}^{C} \left(I - \frac{1}{n_c} 1_c 1_c^T \right).$$

I is a identity matrix and 1_c is a vector of length n whose values are 1 if the corresponding samples belong to chunklet c and zeros otherwise. To connect (18)

with the kernel function defined (16), we soften H as follows to avoid its singularity

$$H = \frac{1}{n}\sum_{c=1}^{C}\left(I - \frac{\delta}{n_c}1_c 1_c^T\right) \quad (19)$$

where $\delta \in [0, 1)$. Given the non-singular H defined above, we have the kernel function in (18) written as

$$\hat{\kappa}(x, x') \quad (20)$$
$$= \frac{1}{\lambda}\left(\kappa(x, x') - \mathbf{k}(x)^T\left[\left(\lambda H^{-1} + K\right)^{-1}\right]\mathbf{k}(x')\right)$$

To connect with (16), we have to relate H^{-1} to $\bar{K}_D$ defined in (6). The following proposition shows the result for H^{-1}.

Proposition 4. *If we assume each data point is only assigned to one chunklet, we have H^{-1} for the softened H in (19) computed as*

$$H^{-1} = n\left(\frac{1}{C}I + \frac{\delta}{(C-\delta)C}\sum_{c=1}^{C}\frac{1}{n_c}\mathbf{1}_c\mathbf{1}_c^T\right) \quad (21)$$

Compared to (6), we clearly see the commonality sharing between H^{-1} and $\bar{K}_D$, which allows us to establish the explicit connection between our method and RCA.

5 EXPERIMENTS

We conduct an extensive study to verify the efficacy of the proposed algorithms for metric learning. For the convenience of discussion, we refer to the proposed algorithm for linear distance metric as **IGML**, and the one for kernel distance metric as **KIGML**. To examine the efficacy of the learned distance metric, we employed the k Nearest Neighbor (k-NN) classifier. Our hypothesis is that the better the distance metric is, the higher the classification accuracy of k-NN will be. We set $k = 4$ for k-NN for all the experiments according to our empirical experience.

In addition to the two proposed algorithms, the following six algorithms are employed in our study as baselines for comparison:

- **Euclidean** distance metric.
- **Mahalanobis** distance metric, which is computed as the inverse of covariance matrix of training samples, i.e., $(\sum_{i=1}^{n} x_i x_i)^{-1}$. This method does not utilize the label information, and therefore help reveal the value of the label information.
- **Xing**'s algorithm proposed in (Xing et al., 2002).
- **LMNN**, a distance metric learning algorithm based on the large margin nearest neighbor classification (Weinberger et al., 2005). Empirical study has shown that LMNN outperforms many existing approaches for metric learning.
- **ITML**, an Information-theoretic metric learning based on (Davis et al., 2007). This approach is closely related to the proposed algorithms in that both studies are based on information-theoretic methods.
- **KRCA**, the kernel relevant component analysis (Tsang et al., 2005). This approach is closely related to the proposed approaches, as revealed before.

For ITML method, parameter γ was tuned by cross validation over the range from 10^{-4} to 10^4. For KIGML and KRCA, we first normalized each feature into the range [0 1], and then deploy the RBF kernel with kernel width $\sigma = 1$ for all experiments.

5.1 EXPERIMENT (I): DATA CLASSIFICATION

We conducted experiments of data classification over ten UCI datasets. Table 1 summarizes the properties of the 10 datasets. For all the datasets, we randomly selected 50% samples for training, and use the remaining samples for testing. We run each experiment 30 times. Table 2 shows the classification error of eight methods over 10 datasets averaged over 30 runs together with the standard deviation. The best result is highlighted by a bold font.

Based on the results in Table 2, we draw the following observations. First, we observed that overall the two proposed metric learning algorithms, i.e., IGML and KIGML, achieve the classification accuracy of k-NN that is comparable to the state-of-the-art algorithms for metric learning. In particular, IGML achieves the best performance over one dataset, while KIGML wins over four out of ten datasets. For most of the datasets, although the proposed algorithms do not outperform the six baseline algorithms, their classification accuracy is in general close to the best performance. We thus conclude that the proposed algorithms are effective for distance metric learning. Second, we observed that KIGML usually outperforms IGML. For six out of ten datasets, KIGML performs significantly better than IGML. Only for two datasets, IGML outperforms KIGML significantly. This observation indicates the importance of learning a nonlinear distance metric.

5.2 EXPERIMENT (II): FACE RECOGNITION

The AT&T database of faces contains grey images of 40 distinct subjects (AT&T, 2002). Each subject has 10 pictures. For each subject, the images were taken at different times, with varied the lighting condition and different facial expressions (open/closed-eyes, smiling/not-smiling) and facial details (glasses/no-

Table 2: Classification error (%) of a k-NN ($k = 4$) classifier on the ten UCI datasets using eight different metrics. Standard deviation is included. The best performance of each dataset is highlighted by bold font.

Data	Eclidean	Mahala	Xing	LMNN	ITML	KRCA	IGML	KIGML
1	5.0 ± 2.9	10.8 ± 3.3	3.5 ± 1.9	4.5 ± 2.1	4.3 ± 2.7	4.1 ± 1.6	$\mathbf{2.7 \pm 1.7}$	3.9 ± 2.8
2	29.6 ± 3.6	7.5 ± 2.2	10.8 ± 4.6	$\mathbf{4.1 \pm 1.8}$	7.7 ± 3.0	4.6 ± 1.5	5.0 ± 1.6	6.1 ± 1.9
3	23.6 ± 3.1	16.9 ± 3.6	23.2 ± 3.4	14.7 ± 1.9	16.6 ± 5.0	15.0 ± 2.7	12.9 ± 3.4	$\mathbf{12.4 \pm 3.5}$
4	19.5 ± 0.6	36.1 ± 0.8	$\mathbf{17.0 \pm 0.8}$	19.1 ± 0.7	19.7 ± 0.7	20.1 ± 0.7	30.6 ± 0.7	21.1 ± 0.6
5	2.1 ± 0.3	5.9 ± 0.5	12.3 ± 0.9	1.6 ± 0.3	2.1 ± 0.3	2.1 ± 0.3	3.2 ± 0.3	$\mathbf{1.4 \pm 0.2}$
6	6.0 ± 5.1	2.8 ± 3.2	1.1 ± 2.2	2.2 ± 2.1	0.7 ± 1.0	$\mathbf{0.1 \pm 0.8}$	1.8 ± 2.1	0.4 ± 1.3
7	17.8 ± 1.6	18.4 ± 2.0	$\mathbf{10.3 \pm 1.3}$	15.0 ± 1.9	11.1 ± 2.6	17.2 ± 1.6	16.6 ± 1.8	14.2 ± 1.6
8	28.9 ± 4.2	28.9 ± 3.8	28.9 ± 4.2	20.3 ± 4.4	28.3 ± 6.3	26.5 ± 4.6	28.1 ± 4.5	$\mathbf{14.6 \pm 4.0}$
9	28.0 ± 1.8	27.8 ± 2.0	27.9 ± 1.7	$\mathbf{27.1 \pm 1.7}$	27.8 ± 1.7	27.8 ± 1.6	27.6 ± 1.9	27.8 ± 2.0
10	35.5 ± 3.5	34.9 ± 3.2	41.7 ± 4.9	34.9 ± 3.2	36.2 ± 3.4	36.9 ± 2.7	35.8 ± 2.3	$\mathbf{33.3 \pm 3.1}$

Figure 2: Example results for the AT&T face database. Top row: ten test face images misclassified by k-NN using the Mahalanobis distance. Second row: the nearest neighbors found by the Mahalanobis distance from the training set. Images in the third row and the last row are the nearest neighbors found from the training set by the proposed IGML and KIGML methods, respectively.

Table 1: Description of UCI datasets

No	Dataset	Size	Classes	Features
1	iris	150	3	4
2	wine	178	3	13
3	segmentation	210	7	19
4	waveform	5000	3	21
5	optdigits	3823	10	64
6	soybean-small	47	4	35
7	ionosphere	351	2	34
8	sonar	208	2	60
9	pima	768	2	8
10	glass	214	6	9

glasses). The size of each image is 92×112 pixels, with 256 grey levels per pixel. For each subject in the database, we randomly selected 5 images for training, and tested on the remaining 5 images. Figure 1 shows the classification accuracy of k-NN using the distance metrics learned by eight different algorithms. We observed that both KIGML and LMNN achieve the best classification accuracy among eight competitors. Again, the comparison between IGML and KIGML reveals the significant advantage of KIGML in identifying the right face, indicating the importance of learning a nonlinear distance metric. In order to further illustrate the difference between the KIGML and IGML, in Figure 2, we show ten test images (in the first row) that are misclassified by the Mahalanobis distance (in the second row). The nearest neighbors of the ten test images identified by IGML and KIGML are shown in the third and the last row in Figure 2. It is clear that KIGML is able to find more right faces than IGML. Finally, it is surprising to observe that the Xing's algorithm performs significantly worse than the Euclidean distance. We attributes the failure of the

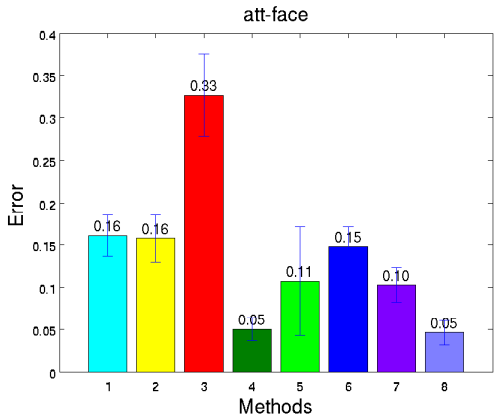

Figure 1: Classification errors of a k-NN ($k = 4$) for the AT&T face dataset using eight different metrics. From left to right, the eight metric learning algorithms are Eclidean, Mahalanobis, Xing, LMNN, ITML, KRCA, IGML, KIGML. The standard deviation of each classification error is on the top of the bar.

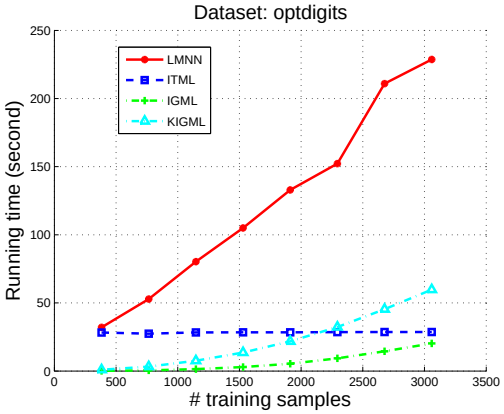

Figure 3: Running time of LMNN, ITML and the proposed IGML, KIGML algorithms for the "optdigit" dataset. Each point in the figure is the average result of 30 random tests.

Xing's algorithm to the high dimensionality, which is $92 \times 112 = 10304$ in this study.

5.3 EXPERIMENT (III): COMPUTATION EFFICIENCY

Computational efficiency is an important issue in the study of distance metric learning, as pointed out in the introduction section. In this experiment, we examine the computational efficiency of the proposed algorithms. Since LMNN appears to be the best learner in our studies, we focus our comparison on LMNN. We also include ITML in our comparison since both our methods and ITML are based on information theoretics. All the algorithms are implemented in Matlab and the experiments are run on an AMD 2.2G computer with 4GMB RAM. Fig. 3 shows the running time of the four comparative algorithms on "optdigit" dataset with varied number of training examples. We clearly see that the computational time of LMNN increase dramatically as the number of training samples is increased. In contrast, our methods and ITML suffer small increase in their running time with an increasing number of training examples. We have similar observation for the other datasets. Due to the space limit, we omit the results of running time for the other datasets.

6 CONCLUSION

In this paper, we propose a novel framework for metric learning that is based on information geometry. The key idea is to construct two kernel matrices for given training data, one based on the distance metric and the other based on the assigned class labels. We relate the two matrices to two different Gaussian distributions, and measure the difference between them by a KL divergence. The optimal distance metric is found by minimizing the distance between the two kernel matrices. We present two metric learning algorithms based on this idea, one for linear distance metric and the other for nonlinear distance metric with the introduction of a kernel function. Extensive experiments with data classification and face recognition show promising results of the proposed approach.

ACKNOWLEDGEMENTS

The work was supported in part by the National Science Foundation (IIS-0643494) and the U. S. Army Research Laboratory and the U. S. Army Research Office (ARO W911NF-08-1-0403). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF and ARO.

Appendix A: Proof for Theorem 2

Proof. First, substituting the result in Proposition 3 into (9), we have

$$\begin{aligned} A &= \left(X\bar{K}_D^{-1}X^T\right)^{-1} \\ &= \left(X\left(\frac{1}{\lambda} - \sum_{k=1}^{C}\frac{z_k z_k^T}{\lambda\left(\lambda + s_k\right)}\right)X^T\right)^{-1} \\ &= \lambda\left(XX^T - \sum_{k=1}^{C}\frac{Xz_k z_k^T X^T}{\lambda + s_k}\right)^{-1} \end{aligned}$$

Using

$$\bar{x}_k = \frac{1}{s_k} X z_k,\ \Sigma_k = \frac{1}{s_k} \sum_{i=1}^{n} y_i^k \left(x_i - \bar{x}_k\right) \left(x_i - \bar{x}_k\right)^T,$$

we have

$$\begin{aligned} A &= \lambda \left(\sum_{k=1}^{C} \sum_{i=1}^{n} y_i^k x_i x_i^T - \sum_{k=1}^{C} \frac{s_k^2 \bar{x}_k \bar{x}_k^T}{(\lambda + s_k)} \right)^{-1} \\ &= \lambda \left(\begin{array}{l} \sum_{k=1}^{C} \sum_{i=1}^{n} y_i^k x_i x_i^T - \sum_{k=1}^{C} s_k \bar{x}_k \bar{x}_k^T \\ + \sum_{k=1}^{C} s_k \bar{x}_k \bar{x}_k^T - \sum_{k=1}^{C} \frac{s_k^2 \bar{x}_k \bar{x}_k^T}{(\lambda + s_k)} \end{array} \right)^{-1} \end{aligned}$$

We further simplify the expression for A as

$$\begin{aligned} A &= \lambda \left(\sum_{k=1}^{C} \sum_{i=1}^{n} y_i^k \left(x_i - \bar{x}_k\right) \left(x_i - \bar{x}_k\right)^T + \sum_{k=1}^{C} \frac{\lambda s_k \bar{x}_k \bar{x}_k^T}{\lambda + s_k} \right)^{-1} \\ &= \lambda \left(\sum_{k=1}^{C} s_k \Sigma_k + \sum_{k=1}^{C} \frac{\lambda s_k \bar{x}_k \bar{x}_k^T}{\lambda + s_k} \right)^{-1} \end{aligned}$$

□

Appendix B: Proof for Theorem 3

Proof. First, according to the definition of $\hat{kappa}(x, x')$, we have

$$\begin{aligned} \hat{\kappa}\left(x, x'\right) &= \left\langle \Phi\left(x\right), A \Phi\left(x'\right) \right\rangle \\ &= \Phi\left(x\right)^T \left(X_1 \bar{K}_D^{-1} X_1^T + \lambda I_\kappa \right)^{-1} \Phi\left(x'\right) \end{aligned}$$

Using the matrix inverse lemma, we have

$$\begin{aligned} &\left(X_1 \bar{K}_D^{-1} X_1^T + \lambda I_\kappa\right)^{-1} \\ &= \frac{1}{\lambda} I_\kappa - \frac{1}{\lambda^2} X_1 \left(\bar{K}_D + \frac{1}{\lambda} X_1^T X_1 \right)^{-1} X_1^T \\ &= \frac{1}{\lambda} I_\kappa - \frac{1}{\lambda^2} X_1 \left(\bar{K}_D + \frac{1}{\lambda} K \right)^{-1} X_1^T \end{aligned}$$

As a result, we have

$$\begin{aligned} &\hat{\kappa}\left(x, x'\right) \\ &= \Phi(x)^T \left(\frac{1}{\lambda} I_\kappa - \frac{1}{\lambda^2} X_1 \left[\bar{K}_D + \frac{1}{\lambda} K \right] X_1^T \right) \Phi(x') \\ &= \frac{1}{\lambda} \left(\kappa(x, x') - \frac{1}{\lambda} \mathbf{k}(x) \left[\bar{K}_D + \frac{1}{\lambda} K \right]^{-1} \mathbf{k}(x') \right) \end{aligned}$$

where $\mathbf{k}\left(x\right) = \left(\kappa\left(x_1, x\right), \ldots, \kappa\left(x_n, x\right)\right)^T$. □

References

Amari, S. & Nagaoka, H. (2000). *Methods of Information Geometry*. Oxford University Press.

AT&T (2002). *AT&T Laboratories Cambridge face dataset*.

Bar-Hillel, A., Hertz, T., Shental, N., & Weinshall, D. (2003). Learning distance functions using equivalence relations. In *Proc. ICML*.

Cristianini, N., Shawe-Taylor, J., Elissee, A., & Kandola, J. (2002). On kernel-target alignment. In *NIPS*.

Davis, J., Kulis, B., Jain, P., Sra, S., & Dhillon, I. (2007). Information-theoretic metric learning. In *Proc. ICML*.

Globerson, A. & Roweis, S. (2005). Metric learning by collapsing classes. In *NIPS*.

Goldberger, J., Roweis, S., Hinton, G., & Salakhutdinov, R. (2005). Neighbourhood components analysis. In *NIPS*.

Kwok, J. & Tsang, I. (2003). Learning with idealized kernels. In *Proc. ICML*.

Saul, L. K. & Roweis, S. T. (2003). Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *J. Mac. Learn. Res.*, **4**, 119–155.

Shental, N., Hertz, T., Weinshall, D., & Pavel, M. (2002). Adjustment learning and relevant component analysis. In *Proc. ECCV*.

Sugiyama, M. (2006). Local fisher discriminant analysis for supervised dimensionality reduction. In *Proc. ICML*.

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

Tsang, I., Cheung, P., & Kwok, J. (2005). Kernel relevant component analysis for distance metric learning. In *Proc. IJCNN*.

Tsuda, K., Akaho, S., Asai, K., & Williams, C. (2003). The em algorithm for kernel matrix completion with auxiliary data. *J. Mac. Learn. Res.*, **4**, 67–81.

Weinberger, K., Blitzer, J., & Saul, L. (2005). Distance metric learning for large margin nearest neighbor classification. In *NIPS*.

Xing, E., Ng, A., Jordan, M., & Russell, S. (2002). Distance metric learning, with application to clustering with side-information. In *NIPS*.

Yang, L. & Jin, R. (2006). Distance metric learning: A comprehensive survey. *MSU, Tech. Rep.*

Yang, L., Jin, R., Sukthankar, R., & Liu, Y. (2006). An efficient algorithm for local distance metric learning. In *Proc. AAAI*.