Nonlinear Adaptive Distance Metric Learning for Clustering*

Jianhui Chen   Zheng Zhao   Jieping Ye   Huan Liu
Department of Computer Science and Engineering
Arizona State University
{jianhui.chen, zheng.zhao, jieping.ye, huan.liu}@asu.edu

ABSTRACT

A good distance metric is crucial for many data mining tasks. To learn a metric in the unsupervised setting, most metric learning algorithms project observed data to a low-dimensional manifold, where geometric relationships such as pairwise distances are preserved. It can be extended to the nonlinear case by applying the kernel trick, which embeds the data into a feature space by specifying the kernel function that computes the dot products between data points in the feature space. In this paper, we propose a novel unsupervised **N**onlinear **A**daptive **M**etric **L**earning algorithm, called **NAML**, which performs clustering and distance metric learning simultaneously. NAML first maps the data to a high-dimensional space through a kernel function; then applies a linear projection to find a low-dimensional manifold where the separability of the data is maximized; and finally performs clustering in the low-dimensional space. The performance of NAML depends on the selection of the kernel function and the projection. We show that the joint kernel learning, dimensionality reduction, and clustering can be formulated as a trace maximization problem, which can be solved via an iterative procedure in the EM framework. Experimental results demonstrated the efficacy of the proposed algorithm.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications - Data Mining

General Terms

Algorithms

Keywords

Clustering, distance metric, kernel, convex programming

---

*The first two authors contribute equally to the paper.

1. INTRODUCTION

Good distance metrics are crucial to many areas in data mining, such as clustering, classification, regression, and semi-supervised learning. In distance metric learning, the goal is to achieve better compactness (reduced dimensionality) and separability (inter-cluster distance) on the data, in comparison with usual distance metrics, such as Euclidean distance. With a good distance metric, the construction of the learning models becomes easier and the accuracy of the learning models usually improves [34]. Based on the availability of the constraint information (class label), distance metric learning algorithms fall into two categories: supervised distance metric learning [26, 31, 32, 35] and unsupervised distance metric learning [4, 10, 17, 23, 29].

The performance of unsupervised learning algorithms, such as K-means is largely dependent on the pairwise similarity, which is commonly determined via a pre-specified distance metric. However, learning a good distance metric in the unsupervised setting is challenging due to the absence of any prior knowledge on the data. In this paper, we focus on the problem of unsupervised distance metric learning for clustering. Without any constraint or class label information, most unsupervised metric learning algorithms apply the projection method such that geometric relationships, such as the pairwise distances are preserved in a low-dimensional manifold. Commonly used projection (dimensionality reduction) methods include the Principle Component Analysis (PCA) [17], Locally Linear Embedding (LLE) [23], Laplacian Eigenmap [4], and ISOMAP [29]. Unsupervised learning algorithms, such as K-means can then be applied in the dimensionality-reduced space, avoiding the *curse of dimensionality*.

In unsupervised learning, the goal is to find a collection of clusters in the data, which achieves the maximum inter-cluster separability. Traditionally, dimensionality reduction and clustering are applied in two separate steps. If distance metric learning (via dimensionality reduction) and clustering can be performed together, the cluster separability in the data can be better maximized in the dimensionality-reduced space. In this paper, we propose a novel algorithm for nonlinear adaptive distance metric learning, called NAML for simultaneous distance metric learning and clustering. NAML first maps the data to a high-dimensional space through a kernel function; next applies a linear projection to find a low-dimensional manifold; and then perform clustering in the low-dimensional space. The performance of NAML depends on the selection of the kernel function and the projection. The key idea of NAML is to integrate kernel learning,

dimensionality reduction, and clustering in a joint framework so that the separability of the data is maximized in the low-dimensional space.

One aspect of NAML shares the same goal of supervised metric learning approaches, which try to adjust the distance among the instances to improve the separability of the data. For example, in [26, 32], the distance metric adjusts the geometry of data, so that the distance between data points from the same class under the metric is small. The metric improves the separability of the data and enhances the performance of classifiers, such as K-Nearest-Neighbor (K-NN). In [9, 19, 37], a linear projection is performed to learn the distance metric for clustering, which assumes linear separability of the data as in [26, 32]. However, many real-world applications may involve data with nonlinear and complex patterns. Kernel methods [24, 25] have been commonly used to deal with this problem. They work by embedding the input data into a high-dimensional feature space through the so-called *kernel function*. The key to the success of kernel methods is that the embedding into a feature space can be uniquely determined by specifying the kernel function that computes the dot products between data points in the feature space. One of the central issues in kernel methods is the selection (learning) of a good kernel function. The problem of kernel learning has been an active area of recent research [2, 3, 13, 16, 18, 20, 21, 28, 33, 37]. The novel aspect of the proposed approach in comparison with these approaches is that NAML does not use any class label. In [30], generalized maximum margin clustering was proposed for simultaneous kernel learning and clustering, which was formulated as a semidefinite program (SDP). Besides its high computational cost of solving a SDP problem, the proposed formulation in [30] is restricted to the two-cluster problems only.

We show in this paper that the simultaneous kernel learning, dimensionality reduction, and clustering in NAML can be formulated as a trace maximization problem, which can be solved by an iterative algorithm based on the *EM* framework. In particular, we show that both dimensionality reduction and clustering can be solved by spectral anslysis, while the kernel learning can be formulated as a Quadratically Constrained Quadratic Programming (QCQP) problem, which can be solved more efficiently than SDP. We evaluate the proposed algorithm using benchmark data sets, and the experimental results show the effectiveness of the proposed algorithm.

The remainder of the paper is organized as follows. We introduce the formulation of distance metric learning for the linear case in Section 2. The formulation is then extended to the nonlinear case in Section 3. Experimental results are presented in Section 4. This paper concludes with discussion and future work in Section 5.

For convenience, we present in Table 1 the important notations used in the rest of this paper.

2. ADAPTIVE DISTANCE METRIC LEARNING: THE LINEAR CASE

In this section, we present the linear adaptive distance metric learning algorithm from [37], which will then be extended to the nonlinear case in the next section.

Assume we are given a data set of zero mean, which consists of n samples $\{x_i\}_{i=1}^n$, where $x_i \in \mathbb{R}^m$. Denote $X = [x_1, x_2, \cdots, x_n]$ as the data matrix. Consider the pro-

Table 1: Important notations used in the paper.

Notation	Description
n	number of samples
m	number of features (dimensions)
k	number of clusters
X	data matrix of size m by n
l	reduced dimensionality
W	transformation in the linear case
S	covariance matrix of data in X
μ	mean of the data in X
C_i	the i-th cluster in X
n_i	size of the i-th cluster C_i
μ_i	mean of i-th cluster C_i
λ	regularization parameter
L	cluster indicator matrix of size n by k
K	kernel function
G	kernel Gram matrix of size n by n
Q	transformation in the nonlinear case
$\mathcal{L}$	Laplacian matrix of size n by n

jection of the data via a linear transformation $W \in \mathbb{R}^{m \times l}$. Thus, each x_i in the m-dimensional space is mapped to a vector $\hat{x}_i$ in the l-dimensional space as follows:

$$x_i \in \mathbb{R}^m \rightarrow \hat{x}_i = W^T x_i \in \mathbb{R}^l \ (l < m). \qquad (1)$$

It has been shown [8, 15] that for most high-dimensional data sets, almost all low dimensional projections are nearly normal. That is, for large m the projected data $\{\hat{x}_i\}_{i=1}^n$ is expected to be nearly normal. In this case, a good distance measure is the well-known *Mahalanobis* distance measure defined as follows:

$$d_M(\hat{x}_i, \hat{x}_j) = \sqrt{(\hat{x}_i - \hat{x}_j)^T \hat{S}^{-1} (\hat{x}_i - \hat{x}_j)}, \qquad (2)$$

where $\hat{S}$ is the covariance matrix defined as follows:

$$\hat{S} = \frac{1}{n} \sum_{i=1}^n (\hat{x}_i - \hat{\mu})(\hat{x}_i - \hat{\mu})^T, \qquad (3)$$

and $\hat{\mu} = \frac{1}{n} \sum_{i=1}^n \hat{x}_i$ is the mean of $\{\hat{x}_i\}_{i=1}^n$. It follows that

$$\hat{S} = \frac{1}{n} \sum_{i=1}^n W^T (x_i - \mu)(x_i - \mu)^T W = W^T S W, \qquad (4)$$

where $\mu = \frac{1}{n} \sum_{i=1}^n x_i$ is the mean of $\{x_i\}_{i=1}^n$, and

$$S = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T \qquad (5)$$

is the class covariance matrix of the original data in X. For high-dimensional data, the estimation of the covariance matrix in Eq. (5) is often not reliable. Thus, the regularization technique [11] is applied to improve the estimation as follows:

$$S = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T + \lambda I_m, \qquad (6)$$

where I_m is the identity matrix of size m and $\lambda > 0$ is a regularization parameter.

Under this new distance measure, K-means clustering can be applied to assign the data $\{\hat{x}_i\}_{i=1}^n$ into k disjoint clusters, $\{C_1, C_2, \cdots, C_k\}$, which minimize the following *Sum of Squared Error* (SSE):

$$\text{SSE}(\{C_j\}_{j=1}^k) = \sum_{j=1}^{k} \sum_{\hat{x}_i \in C_j} d_M(\hat{x}_i, \mu_j)^2, \qquad (7)$$

where the Mahalanobis distance $d_M(\cdot, \cdot)$ is defined as in Eq. (2), and μ_j is the mean of the j-th cluster C_j.

As the summation of all pair-wise distances is a constant for a fixed W. The minimization of the SSE is equivalent to the maximization of *Sum of Squared Intra-cluster Error* (SSIE) defined as follows:

$$\text{SSIE}\left(\{C_j\}_{j=1}^k\right) = \sum_{j=1}^{k} n_j\, d_M(\mu_j, \hat{\mu})^2, \qquad (8)$$

where n_j is the sample size of the j-th cluster C_j, μ_j is the mean of the j-th cluster C_j, and $\hat{\mu}$ is the global mean as defined above. SSIE can be expressed in a compact matrix form as follows. Let $F \in \mathbb{R}^{n \times k}$ be the cluster indicator matrix defined as follows:

$$F = \{f_{i,j}\}_{n \times k}, \text{ where } f_{i,j} = \begin{cases} 1 & \text{if } x_i \in C_j \\ 0 & \text{otherwise} \end{cases}. \qquad (9)$$

The *weighted cluster indicator* matrix $L = [L_1, L_2, \cdots, L_K]$ is defined as [7, 9]:

$$L = F(F^T F)^{-\frac{1}{2}}, \qquad (10)$$

where the i-th column of L is given by

$$L_i = (0, \ldots, 0, \overbrace{1, \ldots, 1}^{n_i}, 0, \ldots, 0)^T / n_i^{\frac{1}{2}}. \qquad (11)$$

With the weighted cluster indicator matrix L, the Sum of Squared Intra-cluster Error (SSIE) can be expressed as:

$$\begin{aligned} \text{SSIE}\left(\{C_j\}_{j=1}^k\right) &= \text{trace}\left(L^T X^T W \hat{S}^{-1} W^T X L\right) \\ &= \text{trace}\left(L^T X^T W (W^T S W)^{-1} W^T X L\right). \end{aligned}$$

The joint metric learning and clustering problem can be formulated as follows [37]:

$$\max_{W,L} \text{ trace}\left(L^T X^T W (W^T S W)^{-1} W^T X L\right). \qquad (12)$$

The optimization problem in Eq. (12) maximizes the inter-cluster distance under the Mahalanobis distance measure determined by the transformation W. Thus, it computes the distance metric and performs the clustering simultaneously.

3. ADAPTIVE DISTANCE METRIC LEARNING: THE NONLINEAR CASE

In this section, we first review the basics of kernel methods. We then present the nonlinear formulation of the adaptive metric learning algorithm from the last section using the kernel trick.

Kernel methods [24, 25] work by mapping the data into a high-dimensional *Hilbert space* (feature space) $\mathcal{F}$ equipped with an inner product through a nonlinear mapping ϕ_K as:

$$\phi_K : \mathbb{R}^m \to \mathcal{F}.$$

The nonlinear mapping can be implicitly specified by a symmetric *kernel function* K, which computes the inner product of the images of each data pair in the feature space, that is

$$K(x_i, x_j) = (\phi_K(x_i),\ \phi_K(x_j)),$$

where $x_i, x_j \in \mathbb{R}^m$ are training data points. A kernel function K satisfies the finitely positive semidefinite property: for any $x_1, \cdots, x_n \in \mathbb{R}^m$, the so-called *kernel Gram matrix* G, defined as $G_{ij} = K(x_i, x_j)$, is symmetric and positive semidefinite.

The adaptive metric learning problem in Eq. (12) can be extended to the nonlinear case using the kernel trick. Denote $\phi_K(X)$ as the data matrix in the feature space. For a given kernel function K, the nonlinear adaptive metric learning problem can be formulated as the following trace maximization problem:

$$\max_{W_K, L} \text{ trace}\left(L^T \phi_K(X)^T W_K (W_K^T S_K W_K)^{-1} W_K^T \phi_K(X) L\right),$$

where W_K is the transformation in the feature space. Assume the data in the feature space has been centered, i.e., $\sum_{i=1}^n \phi_K(x_i) = 0$. Otherwise the kernel centering technique in [24] can be used. Thus the covariance matrix S_K can be expressed as

$$S_K = \phi_K(X)\phi_K(X)^T. \qquad (13)$$

It follows from the *Representer Theorem* [24] that the optimal transformation W_K is in the span of the images of the data points in the feature space. That is,

$$W_K = \phi_K(X)Q, \qquad (14)$$

for some matrix $Q \in \mathbb{R}^{n \times l}$. Thus, the objective function for NAML can be rewritten as

$$\max_{Q,L} \text{ trace}\left(L^T G Q \left(Q^T (GG + \lambda G)Q\right)^{-1} Q^T G L\right), \qquad (15)$$

where $G = \phi_K(X)^T \phi_K(X)$ is the kernel matrix. Here we assume that the matrix $GG + \lambda G$ is nonsingular, and we can use pseudo-inverse [14] to deal with the singular case.

In essence, NAML maps the data into a high-dimensional feature space through a nonlinear mapping, where linear projection and clustering are performed to maximize the cluster separability. The representation of the data in the feature space is determined by the nonlinear mapping, which can be implicitly specified by a kernel matrix. The performance of NAML is dependent on the choice of the kernel matrix. We propose to learn an appropriate kernel matrix for NAML in a joint framework, which leads to the following joint trace optimization problem:

$$\max_{Q,L,G} \text{ trace}\left(L^T G Q \left(Q^T (GG + \lambda G)Q\right)^{-1} Q^T G L\right), \qquad (16)$$

where the kernel matrix G is restricted to be a convex combination of a given set of p kernel matrices, defined as

$$G \in \mathcal{G} = \left\{ \sum_{i=1}^p \theta_i G_i \ \middle|\ \sum_{i=1}^p \theta_i \text{trace}(G_i) = 1, \theta_i \geq 0\ \forall i \right\}. \qquad (17)$$

The formulation in Eq. (16) performs kernel learning, dimensionality reduction, and clustering simultaneously. However, the joint optimization problem is highly nonlinear and difficult to solve. One key observation is that if two of the three components L, G, and Q are fixed, the optimization problem

is easy to solve. This enables us to solve the problem in the **EM** framework, in which we update L, G, and Q iteratively to find a local solution.

3.1 The computation of L for given Q and G

For a given matrix Q and a given kernel matrix G, computing the optimal L in Eq. (16) is equivalent to solving the following trace maximization problem:

$$\max_L \ \text{trace}(L^T \tilde{G} L), \tag{18}$$

where $\tilde{G}$ is defined as

$$\tilde{G} = G\, Q \left(Q^T (GG + \lambda G) Q \right)^{-1} Q^T G. \tag{19}$$

Recall that the entries of the i-th column of the weighted cluster indicator matrix L are either 0 or $1/\sqrt{n_i}$ as defined in Eq. (11). It follows that $L^T L = I_k$, i.e., the columns of L are orthonormal. We apply the spectral relaxation technique [38] for the computation of the optimal L, which is given by the eigenvectors of $\tilde{G}$ as follows:

THEOREM 3.1. *(Ky Fan) Let $\tilde{G}$ be a symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, and the corresponding eigenvectors $U = [u_1, \cdots, u_n]$. Then*

$$\lambda_1 + \cdots + \lambda_k = \max_{L^T L = I_k} \ trace(L^T \tilde{G} L).$$

Moreover, the optimal L^* is given by $L^* = [u_1, \cdots, u_k]\, P$, for an arbitrary orthogonal matrix $P \in \mathbb{R}^{k \times k}$.

In the implementation, we choose the first k eigenvectors of $\tilde{G}$ corresponding the largest k eigenvalues, where k is the number of clusters. For simplicity, we set P to be the identity matrix. Note that we compute the trace value of the matrix $L^T \tilde{G} L$ in each iteration as the measure of convergence. When the relative change of the trace value is smaller than a pre-specified threshold ϵ, the iterative process stops.

3.2 The computation of Q for given L and G

For a given kernel matrix G and a given relaxed weighted cluster indicator matrix L, the trace maximization problem in Eq. (16) is equivalent to the maximization of the following objective function:

$$F_1(G, Q) = \text{trace} \left(\left(Q^T S_{K_2} Q \right)^{-1} Q^T S_{K_1} Q \right). \tag{20}$$

where the matrices S_{K_1} and S_{K_2} are defined as

$$S_{K_1} = GLL^T G, \quad S_{K_2} = GG + \lambda G. \tag{21}$$

The optimal Q^* which maximizes $F_1(G, Q)$ in Eq. (20) is given by solving an eigenvalue problem associated with S_{K_1} and S_{K_2}, as summarized below:

THEOREM 3.2. *Let S_{K_1} and S_{K_2} be defined in Eq. (21), and $V = [v_1, \cdots, v_q]$ be the matrix consisting of the first q eigenvectors of $S_{K_2}^+ S_{K_1}$ corresponding to the largest q eigenvalues, where $q = rank(S_{K_1})$. Let $Q^* \equiv argmax_Q F_1(G, Q)$. Then $Q^* = V$.*

PROOF. Let $G = U \Sigma U^T$ be the Singular Value Decomposition (SVD) [14] of G, where $U \in \mathbb{R}^{n \times n}$ is orthogonal and $\Sigma = \text{diag}\,(\Sigma_t, 0) \in \mathbb{R}^{n \times n}$ is diagonal, $\Sigma_t \in \mathbb{R}^{t \times t}$ is diagonal with positive diagonal entries, and $t = \text{rank}(G)$. Let $U_1 \in \mathbb{R}^{n \times t}$ consist of the first t columns of U. Then

$$G = U \Sigma U^T = U \text{diag}\,(\Sigma_t, 0)\, U^T = U_1 \Sigma_t U_1^T. \tag{22}$$

Denote $P = (\Sigma_t^2 + \lambda \Sigma_t)^{-\frac{1}{2}} \Sigma_t U_1^T L$ and let $P = M \Sigma_P N^T$ be the SVD of P, where M and N are orthogonal and Σ_P is diagonal with $\text{rank}(\Sigma_P) = \text{rank}(S_{K_1}) = q$. Let Z be a nonsingular matrix defined as

$$Z = U \left(\begin{array}{cc} (\Sigma_t^2 + \lambda \Sigma_t)^{-\frac{1}{2}} M & 0 \\ 0 & I_{n-t} \end{array} \right), \tag{23}$$

where I_{n-t} is the identity matrix of size $n - t$. It follows that

$$Z^T S_{K_1} Z = \left(\begin{array}{cc} \tilde{\Sigma} & 0 \\ 0 & 0 \end{array} \right), \ Z^T S_{K_2} Z = \left(\begin{array}{cc} I_t & 0 \\ 0 & 0 \end{array} \right), \tag{24}$$

where $\tilde{\Sigma} = (\Sigma_P)^2 \in \mathbb{R}^{t \times t}$ is diagonal with the diagonal entries sorted in non-increasing order. It is clear that the optimal Q^*, which maximizes $F_1(G, Q)$, consists of the first q columns of Z. It can be verified that the first q columns of Z gives $V = [v_1, \cdots, v_q]$ which consists of the first q eigenvectors of $S_{K_2}^+ S_{K_1}$ corresponding to the largest q eigenvalues. This completes the proof of the theorem. $\square$

It is worth noting that the above trace maximization problem is similar to the well-known linear discriminant analysis (LDA) [12]. However, they are fundamentally different, as S_{K_1} is different from the so-called *between-class scatter* matrix in LDA, due to the spectral relaxation in L.

3.3 The computation of G for given Q and L

Given Q and L, the optimal G can be computed by maximizing $F_1(G, Q)$ in Eq. (20), where the kernel matrix G is restricted to be a convex combination of a set of pre-specified kernel matrices as in Eq. (17). One key observation for the computation of the optimal G is that the Q matrix in $F_1(G, Q)$ can be replaced by its optimal value Q^* given in Theorem 3.2. This significantly simplifies the derivation. Denote

$$F_1^*(G) = \max_G F_1(G, Q) = F_1(G, Q^*). \tag{25}$$

It follows from Theorem 3.2 that

$$
\begin{aligned}
F_1^*(G) &= \text{trace}(\tilde{\Sigma}) = \text{trace}\left((S_{K_2})^+ S_{K_1} \right) \\
&= \text{trace}\left(L^T G (GG + \lambda G)^+ GL \right) \\
&= \text{trace}\left(L^T U \left(\begin{array}{cc} (I_t + \lambda \Sigma_t^{-1})^{-1} & 0 \\ 0 & 0 \end{array} \right) U^T L \right).
\end{aligned}
\tag{26}
$$

Since $\left(I_t + \lambda \Sigma_t^{-1} \right)^{-1} + (I_t + \frac{1}{\lambda} \Sigma_t)^{-1} = I_t$, and

$$U \left(\begin{array}{cc} (I_t + \frac{1}{\lambda} \Sigma_t)^{-1} & 0 \\ 0 & I_{n-t} \end{array} \right) U^T = \left(I + \frac{1}{\lambda} G \right)^{-1},$$

we have

$$F_1^*(G) = \text{trace}\left(L^T L \right) - \text{trace}\left(L^T \left(I + \frac{1}{\lambda} G \right)^{-1} L \right). \tag{27}$$

Thus, the optimal G^*, which maximizes $F_1^*(G)$, is given by minimizing the following objective function:

$$F_2(G) = \text{trace}\left(L^T \left(I + \frac{1}{\lambda} G \right)^{-1} L \right). \tag{28}$$

The minimization of $F_2(G)$ can be solved by gradient descent methods [6]. However, the computation of its gradient

is expensive for each iteration. Following the recent work in [36], we can show that this minimization problem can be formulated as a Quadratically Constrained Quadratic Programming (QCQP) problem as follows:

THEOREM 3.3. *Given a set of p centered kernel matrices $G_1, \cdots, G_p$ as defined in Eq. (17), the minimization of $F_2(G)$ defined above can be formulated as a QCQP problem as follows:*

$$\max_{\beta_1, \cdots, \beta_k, t} \quad \sum_{j=1}^{k} \beta_j^T L_j - \frac{1}{4} \sum_{j=1}^{k} \beta_j^T \beta_j - \frac{1}{4\lambda} t$$

$$subject\ to \quad t \geq \frac{1}{r_i} \sum_{j=1}^{k} \beta_j^T G_i \beta_j, \ i = 1, \cdots, p, \quad (29)$$

where $L = [L_1, \cdots, L_k]$ and $r_i = trace(G_i)$.

The coefficient θ_i for the i-th kernel G_i is given by the dual variable corresponding to the i-th constraint in Eq. (29) divided by r_i. Note that the general-purpose optimization software packages like MOSEK [1] also report the dual variables by solving the dual problem.

3.4 The Main Algorithm

Based on the discussion described above, we propose to develop an iterative algorithm, called NAML, for **N**onlinear **A**daptive **M**etric **L**earning. The pseudo-code of the NAML algorithm is given as below.

Algorithm : NAML

  Input: $X, k, \lambda, \{K_i\}_{i=1}^{p}, \epsilon$
  Output: G, L and Q
1. Randomly choose one kernel matrix G from $\{K_i\}_{i=1}^{p}$;
2. Compute the initial cluster indicatior matrix L by applying kernel K-means on the initial kernel G;
3. **While** relative change of the trace value $\geq \epsilon$ **do**
4.   Update Q as in Section 3.2;
5.   Update G as in Section 3.3;
6.   Update L as in Section 3.1;
7.   Compute the trace of $L^T \tilde{G} L$ as in Eq. (19);
8. **End**
9. return G, L and Q;

The final clustering result is obtained by applying K-means on the relaxed cluster indicator matrix L. The convergence of the NAML algorithm is guaranteed, as summarized in the following theorem:

THEOREM 3.4. *Algorithm NAML converges in a finite number of steps.*

PROOF. The NAML algorithm updates G, Q, and L iteratively, by maximizing the same objective function, i.e., trace($L^T \tilde{G} L$). As the objective value is non-decreasing and is bounded from above by a finite number, the algorithm converges in a finite number of steps. □

In the implementation, we set $\epsilon = 10^{-5}$ for checking the convergence. We observe from our experiments that the NAML algorithm typically converges within 3 to 4 iterations. The time complexity of the NAML algorithm is dominated by the QCQP problem in Theorem 3.3, whose worst-case complexity is $O(pk^3 n^3)$.

3.5 Connection to Regularized Spectral Clustering

In this subsection, we show the close connection between the proposed formulation and regularized spectral clustering [27]. From Eq. (26), the eigenvectors of G corresponding to the zero eigenvalues can be removed without affecting the value of the objective function. In the following discussions, we assume that G is nonsingular. It follows from Eq. (26) that

$$F_1^*(G) = \text{trace}\left(L^T \left(I + \lambda G^{-1} \right)^{-1} L \right). \quad (30)$$

Next, we consider a specific choice of G by setting $G = \mathcal{L}^{-1}$, where $\mathcal{L}$ is the Laplacian matrix [4] defined as follows. Let $\mathcal{W} \in R^{n \times n}$ be a symmetric similarity matrix, and $\mathcal{D} \in R^{n \times n}$ be a diagonal matrix with $\mathcal{D}_{ii} = \sum_{j=1}^{n} \mathcal{W}_{ij}$. The *Laplacian* $\mathcal{L}$ is defined as [4]

$$\mathcal{L} = \mathcal{D} - \mathcal{W}. \quad (31)$$

The centering of the kernel matrix, which is equivalent to the data centering step in NAML, is not required when $G = \mathcal{L}^{-1}$. It is based on the fact that the inverse of the Laplacian matrix is already centered, as summarized in the following proposition:

PROPOSITION 3.1. *Let $\mathcal{L}$ be the Laplacian matrix defined above. Then the inverse of Laplacian, denoted as $\mathcal{L}^{-1}$, has zero row and column means. In other words, let e_n be the vector of all ones of size n, then $e_n^T \mathcal{L}^{-1} e_n = 0$.*

PROOF. Since $\mathcal{L}$ is symmetric and positive semidefinite, let $\mathcal{L} = U_n \Sigma_n U_n^T$ be SVD of $\mathcal{L}$, where U_n is orthogonal and Σ_n has nonnegative diagonal entries. It follows that

$$e_n^T \mathcal{L} e_n = e_n^T \mathcal{D} e_n - e_n^T \mathcal{W} e_n = 0. \quad (32)$$

Thus, $e_n^T U_n \Sigma_n U_n^T e_n = e_n^T \mathcal{L} e_n = 0$, and $e_n^T U_n = 0$. It follows that $e_n^T \mathcal{L}^{-1} e_n = e_n^T U_n \Sigma_n^{-1} U_n^T e_n = 0$. This completes the proof of the proposition. □

We have assumed that $\mathcal{L}$ is nonsingular in the above derivation. For singular $\mathcal{L}$, we can use its pseudo-inverse and the result in the proposition above still holds. With this particular choice of G, the objective function in Eq. (30) becomes:

$$F(\mathcal{L}) = \text{trace}\left(L^T (I + \lambda \mathcal{L})^{-1} L \right), \quad (33)$$

which corresponds to clustering with a regularized Laplacian matrix [27].

4. EXPERIMENT

We now empirically evaluate the performance of the NAML algorithm in comparison with representative algorithms, and conduct a sensitivity study to evaluate its various components, such as the effect of the regularization parameter λ, and the input kernels. These studies will help us better understand the proposed algorithm, and delineate new challenges and research issues.

4.1 Experiment Setup

To evaluate the performance of NAML, we use the K-means algorithm as the baseline for comparison. We also compare the proposed algorithm with three representative unsupervised distance metric learning algorithms: Principle Component Analysis (PCA), Local Linear Embedding

(LLE), and Laplacian Eigenmap (Leigs). The Matlab implementations of these algorithms are obtained from corresponding authors' websites respectively. NAML is also implemented in the Matlab environment and we solve the QCQP problem using MOSEK [1]. All experiments were conducted on a PENTIUM IV 2.4G PC with 1.5GB RAM.

We test the distance metric learning algorithms and K-means on eight benchmark data sets. They are six UCI data sets [5]: iris, lymph, promoter, satimage, solar, wine, and two image data sets: AR03P[1] and ORL10P[2]. Since MOSEK gives memory overflow error when the number of instances is large, for the satimage data set, we randomly sample 80 instances from each class. The information on the eight test data sets is summarized in Table 2.

Table 2: Summary of the benchmark data sets.

Data set	Dimension	Instance	Class
iris	4	150	3
lymph	18	148	4
promoter	57	106	2
satimage	36	6435	6
solar	12	323	6
wine	13	178	3
AR03P	2400	39	3
ORL10P	10000	100	10

We compare the performance of the algorithms as follows. For each data set, we first run K-means and record its clustering results as a baseline. To make the results of different distance metric learning algorithms comparable, the clustering result of K-means is used to construct $\mathcal{C}$, the set of k initial centroids, for later experiments. Here k is the number of clusters of the data. We apply PCA, LLE, and Leigs on each data set to learn distance metrics, which are used by K-means to learn clusters with the initial centroid set $\mathcal{C}$. Their clustering results are recorded. We also run NAML with $\mathcal{C}$ and record its clustering results. This process is repeated for 20 times with different initial centriods for each data set.

4.1.1 Performance Measures

As we have the label information of all eight benchmark data sets, the clustering results are evaluated by comparing the obtained label of each data point with the ground truth. We use two standard measurements: the accuracy (ACC) and the normalized mutual information (MI) measures defined as below. Given a data point x_i, let c_i and y_i be the obtained cluster indicator and the true class label from the data, respectively. The accuracy measure is defined as:

$$\text{ACC} = \frac{1}{n} \sum_{i=1}^{n} f\left(y_i, \text{map}(c_i)\right), \quad (34)$$

where

$$f(y_i, y_j) = \begin{cases} 0, & y_i \neq y_j \\ 1, & y_i = y_j \end{cases} . \quad (35)$$

[1]http://rvl1.ecn.purdue.edu/~aleix/face_DB.html. Data set is subsampled down to the size of $60 \times 40 = 2400$.
[2]http://www.uk.research.att.com/facedatabase.html. Data set is subsampled down to the size of $100 \times 100 = 10000$

In the equation above, n is the total number of data points and $\text{map}(c)$ is the permutation mapping function that maps each cluster indicator c to its equivalent class label. The mapping is found by using the Kuhn-Munkres algorithm [22].

Let C and Y be the set of cluster indicators and the set of class labels, respectively. The normalized mutual information is defined as:

$$\text{MI}(Y, C) = \frac{2 \times \left(\sum_{y_i \in Y, c_j \in C} p(y_i, c_j) \cdot \log \frac{p(y_i, c_j)}{p(y_i) \cdot p(c_j)} \right)}{\text{H}(Y) + \text{H}(C)} , \quad (36)$$

where $p(c_i)$ (or $p(y_j)$) denotes the probability that an instance randomly selected from X belongs to cluster c_i (or class y_j), $p(y_i, c_j)$ denotes the joint probability, and $\text{H}(Y)$ (or $\text{H}(C)$) is the entropy of Y (or C).

Since each algorithm is tested for 20 times on each data set, we obtain 20 performance evaluations from ACC and MI measures, respectively. These performance evaluations are averaged and yield 2 final performance evaluations per algorithm on each data set. In the experiment, the reduced dimensionality (l) of PCA is selected to retain at least 95% information of the original data, and the reduced dimensionality (l) of NAML is set to k. For each data set, we construct 10 RBF kernels for NAML.

4.2 Experimental Results

Table 3 presents the accuracy (ACC) and normalized mutual information (MI) results on each data set. The results of NAML using 3 different λ values (10^{-6}, 10^{-4}, and 10^{-2}) are shown. NAML with $\lambda = 10^{-2}$ performs the best (including the second best without a significant difference with the best) on 6 data sets in terms of accuracy. On the eight data sets, NAML with $\lambda = 10^{-2}$ performs the best with an average accuracy of 0.747, which is followed by NAML with $\lambda = 10^{-4}$ with an average accuracy of 0.743. LLE performs the third best and PCA is the fourth. Similar trends can also be observed in the MI results.

Experimental results also show that NAML does improve the performance of K-means on all eight data sets. For example, on AR03P data, NAML with $\lambda = 10^{-2}$ improves its accuracy from 0.462 to 0.615, a 15.3% improvement. In our experiment, NAML converges in less than eight iterations and usually converges in 3-4 iterations.

4.3 Sensitivity Study

In this subsection, we study the effects of various components of NAML. More specifically, we study the effect of the input kernels and the regularization parameter λ.

4.3.1 Input Kernels

In Table 4, we compare the performance of NAML with that of K-means using each of ten input kernels, respectively. Thus, we can obtain 10 clustering results from K-means with respect to 10 kernels, and further calculate *max Ker*, *min Ker*, and *ave Ker* corresponding to the best, worst, and average performance. NAML uses the same 10 kernels as its input. We can observe from the table that in most cases, NAML performs much better than *min Ker* (K-means using the worst kernel) and is comparable to *max Ker* (K-means using the best kernel). This set of results has its ramifications for unsupervised metric learning. When we do not have the prior knowledge about the kernel quality,

Table 3: Comparison of accuracy (ACC) and normalized mutual information (MI) on eight benchmark data sets. The numbers behind NAML are the λ values used. For each data set, the first row and the second row list ACC (or MI) and p-val, respectively. The p-val of each algorithm is generated by comparing its ACC with the highest one. ACCs in boldface are the highest ones or the second highest without significant difference with the highest one, according to p-val$>$0.1.

Meas.	Data Set	NAML$^{10^{-6}}$	NAML$^{10^{-4}}$	NAML$^{10^{-2}}$	K-means	PCA	LLE	Leigs
ACC	iris	**0.908**	**0.901**	0.883	0.865	0.865	0.848	0.829
		-	0.801	0.359	0.020	0.020	0	0.004
	lymph	0.707	0.709	**0.716**	0.701	0.701	0.686	0.703
		0.002	0	-	0.008	0.008	0	0.025
	promoter	**0.689**	**0.698**	**0.698**	0.672	0.673	0.677	0.656
		-	-	-	0.074	0.089	0.053	0.004
	satimage	0.658	**0.742**	**0.743**	0.633	0.628	0.599	0.640
		0.014	0.587	-	0.026	0.018	0	0
	solar	**0.593**	**0.587**	0.584	0.576	0.575	0.511	0.562
		-	0.424	0.326	0.136	0.129	0	0.010
	wine	0.413	**0.972**	**0.972**	0.954	0.954	0.961	0.964
		0	-	-	0	0	0	0.061
	AR03P	0.572	0.572	**0.615**	0.462	0.462	0.518	0.403
		0	0	-	0	0	0.012	0
	ORL10P	**0.766**	**0.766**	**0.768**	0.722	0.722	0.715	0.333
		0.168	0.168	-	0.003	0.003	0.001	0
	Average	0.663	0.743	**0.747**	0.698	0.698	0.689	0.636
MI	iris	**0.767**	**0.780**	0.752	0.726	0.726	0.631	0.726
		0.428	-	0	0	0	0	0.001
	lymph	0.183	**0.187**	**0.197**	0.179	0.178	0.173	0.164
		0	0	-	0.169	0.150	0.001	0
	promoter	0.156	**0.168**	0.148	0.104	0.106	0.103	0.087
		0.031	-	0.020	0.013	0.018	0.005	0.002
	satimage	0.520	**0.586**	**0.587**	0.515	0.515	0.452	0.520
		0.006	0.482	-	0.018	0.015	0	0
	solar	**0.395**	0.386	0.342	**0.389**	0.388	0.290	0.350
		-	0.427	0.064	0.163	0.103	0.001	0.107
	wine	0.011	**0.893**	**0.893**	0.850	0.850	0.837	0.875
		0	-	-	0	0	0	0.090
	AR03P	0.212	0.211	**0.301**	0.089	0.089	0.196	0.040
		0.003	0.002	-	0	0	0.032	0
	ORL10P	**0.819**	**0.819**	**0.822**	0.801	0.801	0.793	0.368
		0.264	0.264	-	0.015	0.015	0	0
	Average	0.383	0.504	**0.505**	0.457	0.457	0.434	0.391

NAML provides a way to learn from *multiple* input kernels and generate a metric, with which an unsupervised learning algorithm, like K-means, is more likely to perform as well as with the best input kernel. Hence, NAML has interesting applications in solving real-world clustering problems. For example, in a learning task, the pairwise instance relationship is calculated by a RBF kernel function, and the kernel parameter σ is estimated by several domain experts. According to different understandings of the problem, different experts can assign different values for σ. In this case, NAML's capability of combining different perspectives from multiple experts and learning a good metric can be essential for unsupervised learning of nonlinear patterns. Figure 1 shows two sample cases that NAML converges to a good result, even though the quality of the initial kernel is low.

4.3.2 Regularization Parameter λ

As discussed in Section 2, a regularization parameter λ is introduced to improve the reliability of the estimation of the
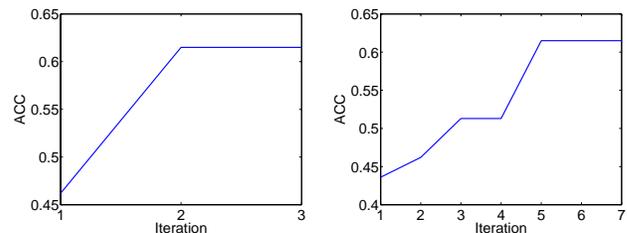

Figure 1: Clustering performance improves during the iterative process of NAML on AR03P data set: $\lambda = 0.01$ (left plot) and $\lambda = 100$ (right plot).

covariance matrix. The performance of NAML is dependent on the value of λ. In the following experiment, we study the effect of this parameter on the clustering performance of NAML. We can observe from Table 3 that, in general, the regularization helps to improve the performance of NAML.

Table 4: Comparison of the performance achieved by NAML and K-means using each input kernel on benchmark data sets. In the table "init Ker" stands for the average accuracy achieved by K-means on the initial kernel. "max Ker", "min Ker" and "ave Ker" stand for the highest, lowest and average accuracy achieved by K-mean on ten input kernels.

Meas.	Data Set	NAML$^{10^{-6}}$	NAML$^{10^{-4}}$	NAML$^{10^{-2}}$	init Ker	max Ker	min Ker	ave Ker
	iris	0.908	0.901	0.883	0.887	0.927	0.887	0.903
	lymph	0.707	0.709	0.716	0.681	0.716	0.681	0.708
ACC	promoter	0.689	0.698	0.698	0.585	0.734	0.585	0.686
	satimage	0.658	0.742	0.743	0.594	0.746	0.594	0.715
	solar	0.593	0.587	0.584	0.593	0.600	0.572	0.584
	wine	0.413	0.972	0.972	0.933	0.972	0.933	0.967
	AR03P	0.572	0.572	0.615	0.418	0.626	0.418	0.541
	ORL10P	0.766	0.766	0.768	0.738	0.762	0.738	0.760
	iris	0.767	0.780	0.752	0.729	0.792	0.729	0.771
	lymph	0.183	0.187	0.197	0.159	0.197	0.159	0.187
MI	promoter	0.156	0.168	0.148	0.128	0.223	0.109	0.161
	satimage	0.520	0.586	0.587	0.485	0.590	0.485	0.568
	solar	0.395	0.368	0.342	0.372	0.375	0.344	0.360
	wine	0.011	0.893	0.893	0.773	0.893	0.773	0.876
	AR03P	0.212	0.211	0.301	0.170	0.362	0.170	0.267
	ORL10P	0.819	0.819	0.822	0.800	0.819	0.800	0.817

In terms of accuracy, on four of eight data sets, the performance of NAML with $\lambda = 10^{-2}$ is significantly better than that with a very small value of regularization ($\lambda = 10^{-6}$). Similar improvement can be observed in MI results.

To obtain a better understanding of the effect of the regularization parameter, we tried a series of different λ values ranging from 10^{-8} to 10^{5}. The ACC and MI results using various λ values are plotted in Figure 2. We can observe from the figure that, in general, NAML is not very sensitive to the value of λ, except for the case when λ is very large ($> 10^{2}$) or very small ($< 10^{-6}$). The use of a λ value in the range of $[10^{-4}, 10^{2}]$ is helpful in most cases.

We can observe from Figure 2 that a small λ value is less effective than a large λ value. In most cases, a large λ value does not significantly degrade the performance, which is not the case when λ value is very small. Further studies show that when λ is very small, the kernel weights learnt by NAML become close to each other; while when λ is very large, the kernel weight vector becomes sparse (many of them are zero) and only the best kernels or those close to the best ones have non-zero weights. We show in Table 5 the weight of each input kernel, when using different λ values on AR03P data. The result suggests that when λ is set to 0, the weights of all kernels are not zero; while when λ is large, the weight vector becomes sparse and only a very small number of kernels has non-zero weights.

Recall that the optimal combination of kernels is obtained by maximizing trace $\left(L^{T} G (GG + \lambda G)^{+} GL\right)$. It is clear that when λ approaches to 0, $G(GG + \lambda G)^{+}G$ approaches to a matrix, which contains 1 as the only nonzero eigenvalue. In this case, the optimization in NAML becomes degenerate. On the other hand, when λ becomes large, the λG term in $(GG + \lambda G)^{+}$ dominates. In this case, the optimization problem is reduced to the maximization of trace $\left(L^{T} GL\right)$, which is essentially equivalent to the selection of a single kernel that maximizes trace $\left(L^{T} KL\right)$. These explain the behavior of NAML for different λ values in Table 5. However, we can also observe from Figure 2 and Table 5 that NAML performs the best when the value of λ is neither too small nor too large. This is partly due to the complementary information that exists among the given collection of kernels, which may be exploited by NAML.

5. CONCLUSIONS

In this paper, we propose a nonlinear adaptive distance metric learning algorithm, called NAML. We show that the joint kernel learning, metric learning, and clustering can be formulated as a trace maximization problem, which can be solved iteratively in an EM framework. More specifically, we show that both dimensionality reduction and clustering can be solved by spectral analysis, while the kernel learning can be formulated as a Quadratically Constrained Quadratic Programming problem.

Experimental results on a collection of benchmark data sets demonstrate that NAML is effective in learning a good distance metric and improving the clustering performance. In general, approaches based on learning a convex combination of kernels can be applied for heterogeneous data integration from different data sources. We plan to apply the proposed algorithm for clustering from multiple biological data, *e.g.*, amino acid sequences, hydropathy profiles, and gene expression data. We reveal the close connection between the proposed algorithm and regularized spectral clustering. The selection of a good Laplacian matrix, which is determined by several parameters such as the number of nearest neighbors, is one of the key issues in spectral clustering. Another line of future work is to study how to combine a set of pre-specified Laplacian matrices to achieve better performance in spectral clustering.
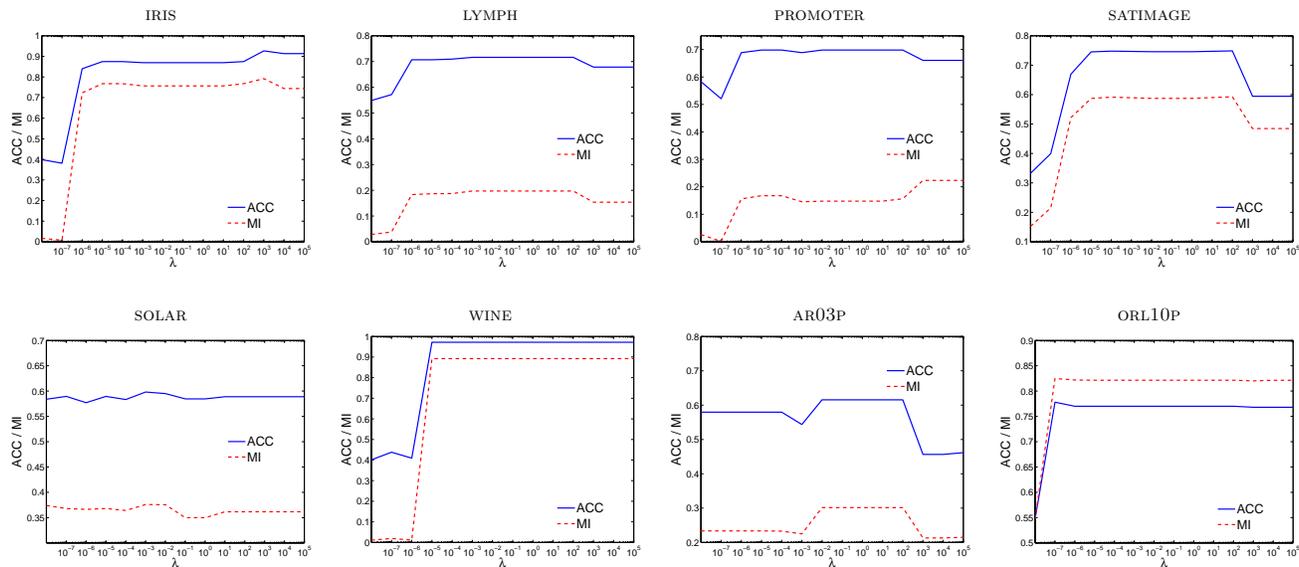
Acknowledgments

Figure 2: ACC and MI vs. different λ values. The x-axis corresponds to different λ values and the y-axis corresponds to ACC or MI values.

6. REFERENCES

[1] E. D. Andersen and K. D. Andersen. The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In T. T. H. Frenk, K. Roos and S. Zhang, editors, *High Performance Optimization*, pages 197–232. Kluwer Academic Publishers, 2000.

[2] A. Argyriou, R. Hauser, C. Micchelli, and M. Pontil. A DC-programming algorithm for kernel selection. In *Proceedings of the Twenty-third International Conference on Machine Learning*, pages 41–48, 2006.

[3] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004.

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Advances in Neural Information Processing Systems*, 15, 2003.

[5] C. Blake and C. Merz. UCI repository of machine learning databases, 1998.

[6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[7] I. S. Dhillon, Y. Guan, and B. Kulis. A unified view of kernel k-means, spectral clustering and graph partitioning. Technical report, TR-04-25, UTCS, 2004.

[8] P. Diaconis and D. Freedman. Asymptotics of graphical projection pursuit. *Annuals of Statistics*, 12:793–815, 1984.

[9] C. Ding and T. Li. Adaptive dimension reduction using discriminant analysis and k-means clustering. In *Proceedings of the Twenty-fourth International Conference on Machine Learning*, 2007.

[10] C. Domeniconi, D. Papadopoulos, D. Gunopulos, and S. Ma. Subspace clustering of high dimensonal data. In *Proceedings of the SIAM International Conference on Data Mining*, 2004.

[11] J. H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989.

[12] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 2nd edition, 1990.

[13] G. Fung, M. Dundar, J. Bi, and B. Rao. A fast iterative algorithm for Fisher discriminant using heterogeneous kernels. In *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004.

[14] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.

[15] P. Hall and K. Li. On almost linearity of low dimensional projections from high dimensional data. *Annuals of Statistics*, 21:867–889, 1993.

[16] T. Jebara. Multi-task feature and kernel selection for SVMs. In *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004.

[17] I. Jolliffe. *Principal Component Analysis*. Springer; 2nd edition, 2002.

[18] S.-J. Kim, A. Magnani, and S. Boyd. Optimal kernel selection in kernel Fisher discriminant analysis. In *Proceedings of the Twenty-third International Conference on Machine Learning*, pages 465–472, 2006.

[19] F. D. la Torre Frade and T. Kanade. Discriminative cluster analysis. In *Proceedings of the Twenty-third International Conference on Machine Learning*, pages 241–248, 2006.

[20] G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[21] D. Lewis, T. Jebara, and W. S. Noble. Nonstationary kernel combination. In *Proceedings of the Twenty-third*

Table 5: The weight of each input kernel when using different λ values on the AR03P data set. The weights of most kernels shrink to zero when λ becomes large. The accuracies of K-means using each of the ten kernels are shown in the first row.

AR03P	KER 1	KER 2	KER 3	KER 4	KER 5	KER 6	KER 7	KER 8	KER 9	KER 10
$\lambda \quad ^{Acc}$	0.426	0.456	0.436	0.456	0.595	0.600	0.610	0.610	0.615	0.615
0	0.026	0.027	0.027	0.028	0.029	0.030	0.031	0.032	0.034	0.035
10^{-8}	0.001	0.001	0.001	0.001	0.001	0.002	0.002	0.002	0.002	0.002
10^{-7}	0.001	0.001	0.001	0.001	0.001	0.002	0.002	0.002	0.002	0.002
10^{-6}	0.001	0.001	0.001	0.001	0.001	0.002	0.002	0.002	0.002	0.002
10^{-5}	0.001	0.001	0.001	0.001	0.001	0.002	0.002	0.002	0.002	0.002
10^{-4}	0.001	0.001	0.001	0.001	0.001	0.002	0.002	0.002	0.002	0.002
10^{-3}	0	0	0	0	0	0	0.001	0.001	0.001	0.014
10^{-2}	0	0	0	0	0	0	0	0	0	0.017
10^{-1}	0	0	0	0	0	0	0	0	0	0.017
10^{0}	0	0	0	0	0	0	0	0	0	0.017
10^{1}	0	0	0	0	0	0	0	0	0	0.017
10^{2}	0	0	0	0	0	0	0	0	0	0.017
10^{3}	0	0	0	0.013	0	0	0	0	0	0
10^{4}	0	0	0	0.014	0	0	0	0	0	0
10^{5}	0	0	0	0.013	0	0	0	0	0	0

International Conference on Machine Learning, pages 553–560, 2006.

[22] L. Lovasz and M. Plummer. *Matching Theory*. North Holland, 1986.

[23] L. K. Saul and S. T. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003.

[24] S. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.

[25] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[26] N. Shental, T. Hertz, D. Weinshall, and M. Pavel. Adjustment learning and relevant component analysis. In *Proceedings of the Seventh European Conference on Computer Vision*, pages 776–792, London, UK, 2002. Springer-Verlag.

[27] A. Smola and I. Kondor. Kernels and regularization on graphs. In *Proceedings of the Annual Conference on Computational Learning Theory*, 2003.

[28] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large Scale Multiple Kernel Learning. *Journal of Machine Learning Research*, 7:1531–1565, July 2006.

[29] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[30] H. Valizadegan and R. Jin. Generalized maximum margin clustering and unsupervised kernel learning. In *Advances in Neural Information Processing Systems*, 2006.

[31] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems*, 2005.

[32] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, 2002.

[33] B. Yan and C. Domeniconi. An adaptive kernel method for semi-supervised clustering. In *Proceedings of the Seventeenth European Conference on Machine Learning*, 2006.

[34] L. Yang and R. Jin. Distance metric learning: A comprehensive survey. Technical report, Department of Computer Science and Engineering, Michigan State University, 2006.

[35] L. Yang, R. Jin, R. Sukthankar, and Y. Liu. An efficient algorithm for local distance metric learning. In *Proceedings of the Twenty-first National Conference on Artificial Intelligence*, 2006.

[36] J. Ye, S. Ji, and J. Chen. Learning the kernel matrix in discriminant analysis via quadratically constrained quadratic programming. In *Proceedings of the Thirteenth ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, 2007.

[37] J. Ye, Z. Zhao, and H. Liu. Adaptive distance metric learning for clustering. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[38] H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Spectral relaxation for k-means clustering. In *Advances in Neural Information Processing Systems*, 2001.