Probabilistic Non-linear Distance Metric Learning For Constrained Clustering

Behnam Babagholami-Mohamadabadi
Computer Engineering Department
Sharif University of Technology
Tehran, Iran
babagholami@ce.sharif.edu

Ali Zarghami
Computer Engineering Department
Sharif University of Technology
Tehran, Iran
zarghami@ce.sharif.edu

Hojjat Abdollahi Pourhaghighi
Computer Engineering Department
Sharif University of Technology
Tehran, Iran
habdollahi@ce.sharif.edu

Mohammad T. Manzuri-Shalmani
Computer Engineering Department
Sharif University of Technology
Tehran, Iran
manzuri@sharif.edu

ABSTRACT

Distance metric learning is a powerful approach to deal with the clustering problem with side information. For semi-supervised clustering, usually a set of pairwise similarity and dissimilarity constraints is provided as supervisory information. Although some of the existing methods can use both equivalence (similarity) and inequivalence (dissimilarity) constraints, they are usually limited to learning a global Mahalanobis metric (i.e., finding a linear transformation). Moreover, they find metrics only according to the data points appearing in constraints, and cannot utilize information of other data points. In this paper, we propose a probabilistic metric learning algorithm which uses information of unconstrained data points (data points which do not appear in neither positive nor negative constraints) along with both positive and negative constraints. We also kernelize our metric learning method based on the kernel trick which provides a non-linear version of the learned metric. Experimental results on synthetic and real-world data sets demonstrate the effectiveness of the proposed metric learning algorithm.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*; I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms*

MultiClust '13, August 11, 2013, Chicago, Illinois, USA.

General Terms

Algorithms and Experimentation

Keywords

Metric learning, logistic regression, deterministic annealing EM, kernel trick

1. INTRODUCTION

Over the past few years, distance metric learning has been widely studied in machine learning and pattern recognition community due to its important rule in classification and clustering problems [15, 17]. The goal of distance metric learning is to learn a suitable metric function from data with the constraint that similar (dissimilar) data points should stay closer (further). For supervised learning applications such as classification and regression tasks, class label information of the training data can be used as the supervisory information for learning an appropriate metric. for unsupervised learning applications such as clustering and dimensionality reduction, however, class label information is not generally available. Hence, the distance metric learning problem is an ill-posed problem with no well-defined optimization criteria. Recently, researchers have given much attention to distance metric learning for semi supervised (constrained) clustering tasks. In this class of problems, class label information is not generally available. However, some information in the form of equivalence (similarity) and inequivalence (dissimilarity) constraints is available as supervisory information.

Many algorithms have been proposed for learning an appropriate metric based on the similarity-dissimilarity constraints. Xing et al. [16] proposed a metric learning algorithm by formulating the learning task into the following

constrained convex optimization problem.

$$A^* = \underset{A}{argmin} \sum_{(x_i,x_j)\in S} (x_i - x_j)^T A(x_i - x_j),$$
$$s.t. \sum_{(x_i,x_j)\in D} (x_i - x_j)^T A(x_i - x_j) \geq 1,$$
$$A \succeq 0, \quad (1)$$

where A is a Mahalanobis distance matrix (A must be positive semidifinite matrix to satisfy the non-negativity and triangle inequality conditions), and S and D is the set of positive and negative constraints respectively. Relevant Component Analysis (RCA) is another method which learns a global linear transformation (a Mahalanobis distance) by utilizing only the positive (equivalence) constraints [2]. Hoi et al. [8] extended the RCA method by incorporating the inequivalence constraints into its objective function. Yang et al. [17] proposed a probabilistic method based on the logistic regression classifier parameterized by the distance metric. In this method, the metric is learned using the maximum likelihood estimation which is equal to the following convex optimization problem.

$$[A^*, \mu^*] = \underset{A,\mu}{argmin} \Bigg[\sum_{(x_i,x_j)\in S} \log\big(1 + exp(-\|x_i - x_j\|_A^2 + \mu)\big) + \sum_{(x_i,x_j)\in D} \log\big(1 + exp(\|x_i - x_j\|_A^2 - \mu)\big) \Bigg]$$
$$s.t.\ \mu \geq 0, A \succeq 0, \quad (2)$$

where μ is the threshold parameter, $\|x_i - x_j\|_A^2$ is the squared Mahalanobis distance, and $y_{ij} = 1$, if $(x_i, x_j) \in S$, and $y_{ij} = 0$, if $(x_i, x_j) \in D$ (y_{ij} denotes whether two data points x_i and x_j belong to the same cluster or not). Xiang et al. [15] proposed a method for learning a linear transformation matrix W (Learning the transformation matrix W can yield the Mahalanobis metric $A = WW^T$) by maximizing the trace ratio objective function:

$$W^* = \underset{W^TW=I}{argmax} \frac{W^T S_b W}{W^T S_w W}, \quad (3)$$

where S_b and S_w are the covariance matrices computed from nagative and positive constraints respectively. The constraint $W^TW = I$ is for avoiding degenerate solutions [15]. Davis et al [6] proposed an information theoretic metric learning method (ITML), which its goal is to learn a Mahalanobis distance parameterized by A that has minimum LogDet divergence to a given baseline matrix A_0 while satisfying the similarity-dissimilarity constraints:

$$A^* = \underset{A}{argmin} tr(AA_0^{-1}) - \log|AA_0^{-1}| - m,$$
$$s.t.\ d_A(x_i, x_j) \leq u, \quad (x_i, x_j) \in S,$$
$$d_A(x_i, x_j) \geq l, \quad (x_i, x_j) \in D, \quad (4)$$

where m is the dimension of data points, and $|X|$ denotes the determinant of the matrix X. In the last few years, some non-linear metric learning algorithms for constrained clustering have been proposed. Yeung and Chang [19] introduced a kernel-based metric learning algorithm which can only use positive constraints. Baghshah and shouraki [1] extended the objective function presented in (3) which can learn a non-linear transformation and also preserves the geometrical structure of data using the idea of Locally Linear Embedding (LLE) algorithm [11].

Although some of the metric learning algorithms [15, 16, 18, 8, 9, 20, 1] can use information of both similarity and dissimilarity constraints, most of them [15, 16, 18, 8] learn a Mahalanobis metric which corresponds to a linear transformation. Furthermore, some recent kernel-based metric learning methods [8, 9, 20] cannot take advantage of unconstrained data points.

In this paper, we propose an efficient non-linear metric learning method which uses both similarity and dissimilarity constraints along with the unconstrained data. More precisely, we extend the idea of [17] by incorporating a new term into the objective function presented in (2) using the information of the unconstrained data.

The reminder of this paper is organized as follows. In section 2, we introduce our metric learning algorithm which considers the unconstrained data as well as the positive and the negative constraints. In section 3, we describe the optimization procedure for solving the proposed objective function based on the deterministic annealing EM algorithm [14]. Section 4 presents experimental results on some synthetic and real-world data sets. Concluding remarks are described in the final section.

2. PROPOSED METHOD

In this section, first we introduce our global Mahalanobis metric learning method using pairwise constraints while utilizing information of the unlabeled (unconstrained) data. Then, we present a non-linear extension of this method.

2.1 Linear Metric Learning

We are given a set of N data points $X = \{x_i \in \mathcal{R}^m\}_{i=1}^N$ and two sets of pairwised constraints which are defined as

$$S = \{(x_i, x_j) \mid x_i \text{ and } x_j \text{ are in the same class}\}, \quad (5)$$

$$D = \{(x_i, x_j) \mid x_i \text{ and } x_j \text{ are in two different class}\}, \quad (6)$$

where S is the set of similar pairwise constraints, and D is the set of dissimilar pairwise constraints. We also define U as a set of all pairs of data points which do not appear in S and D. Hence, we have

$$U = \{(x_i, x_j) \mid i \neq j, (x_i, x_j) \notin S \cup D\}. \quad (7)$$

For any pair of points x_i and x_j , let $d_A(x_i, x_j)$ denote the distance between them parameterized by $A \in \mathcal{R}^{m\times m}$, and is defined as

$$d_A(x_i, x_j) = \|x_i - x_j\|_A = \sqrt{(x_i - x_j)^T A(x_i - x_j)}, \quad (8)$$

where A is the Mahalanobis distance matrix. To learn a distance metric, one can assume there exits a corresponding linear mapping $W^T : \mathcal{R}^m \to \mathcal{R}^w$ $(w \leq m)$, where $W \in \mathcal{R}^{m\times w}$, and $A = WW^T$. Hence, the distance between two data points x_i and x_j under A can be computed as

$$\|x_i - x_j\|_A = \sqrt{(x_i - x_j)^T WW^T (x_i - x_j)}$$
$$= \|W^T(x_i - x_j)\|. \quad (9)$$

In other words, the Mahalanobis distance between data points is equal to the Euclidian distance between them after transforming data points by W^T. So, A represents a suitable Mahalanobis metric if clusters of data points are well-separated

after mapping them using its corresponding linear transformation (W^T), and all positive and negative constraints are satisfied.
Following the idea of [17], we denote $y_{ij} \in \{+1, -1\}$ as the class label of data points x_i and x_j, and define it as

$$y_{ij} = \begin{cases} +1 & \text{if } (x_i, x_j) \in S \\ -1 & \text{if } (x_i, x_j) \in D. \end{cases} \quad (10)$$

We model the probability distribution of y_{ij}, given x_i and x_j based on the logistic regression model parameterized by A as [17]

$$P(y_{ij} \mid x_i, x_j) = \frac{1}{1 + exp\big(-y_{ij}(\|x_i - x_j\|_A^2 - \mu)\big)}, \quad (11)$$

where $\mu \geq 0$ is the threshold parameter. The intuition of Eq. 11 is that two samples x_i and x_j belong to the same cluster if their distance $\|x_i - x_j\|_A^2$ is less than μ. Given S and D, A and μ can be estimated by maximizing the overall likelihood function for all the constraints in S and D (Eq. 2). The problem with this model is that it cannot exploit the unlabeled (unconstrained) data points. Motivated by semi-supervised logistic regression algorithm [7], we can include unlabeled data based on the following intuition: if the clusters are well-separated (after transforming data points by W^T), then the classification on any pairs of unlabeled data points $(x_i, x_j) \in U$ should be confident. More precisely, any pairs of unlabeled data points should clearly belong to the same cluster, or to different clusters. Equivalently, the posterior probability $P(y_{ij} \mid x_i, x_j)$ should be either close to 1, or close to 0. One appropriate way for measuring the confidence is the Shannon's conditional entropy which is defined as

$$H(X) = \begin{cases} \sum_x P(X = x) \log \frac{1}{P(X=x)} & \text{if } X \text{ is discrete,} \\ \int_x p(x) \log \frac{1}{p(x)} & \text{if } X \text{ is continuous.} \end{cases} \quad (12)$$

where X is a random variable, and $P(X = x)$ and $p(x)$ are the probability mass function (PMF) and the probability density function (PDF) of X respectively.
Since the class label of each pairs of data points is a Bernoulli random variable ($y_{ij} \in \{+1, -1\}$) with probability $p = 1/\big(1 + exp(-\|x_i - x_j\|_A^2 + \mu)\big)$, its entropy is defined as

$$H(p) = -p \log p - (1 - p) \log(1 - p). \quad (13)$$

The above entropy H reaches its minimum 0 when $p = 0$ or $p = 1$ (The entropy is small if the classification on the unlabeled samples is certain). So, we propose the following optimization problem based on the unlabeled data and the positive and negative constraints:

$$\begin{aligned}[A^*, \mu^*] = \underset{A,\mu}{argmin} \Bigg[& \sum_{(x_i,x_j)\in S} \log\big(1 + exp(-\|x_i - x_j\|_A^2 + \mu)\big) \\ & + \sum_{(x_i,x_j)\in D} \log\big(1 + exp(\|x_i - x_j\|_A^2 - \mu)\big) \\ & + \lambda \sum_{(x_i,x_j)\in U} H\big(1/\big(1 + exp(-\|x_i - x_j\|_A^2 + \mu)\big)\big)\Bigg] \\ & s.t.\ \mu \geq 0, A \succeq 0, \end{aligned} \quad (14)$$

where λ is the regularization parameter. In other words, the goal of entropy term is to alter the maximum likelihood solution, by biasing it towards low entropy.

In order to simplify the computation for solving the above problem (the difficulty with solving the above problem is due to the positive semi-definitive constraint $A \succeq 0$), we follow the idea of [17] and use the eigenspace of the training samples to approximate A. Let $R = \frac{1}{N}\sum_{i=1}^N x_i x_i^T$ be the sample correlation matrix, and $\{u_i\}_{i=1}^k$ be the top k ($k \leq min(m, N)$) eigenvectors of the matrix R. Then A is assumed to be a linear combination of the top k eigenvectors [17]:

$$A = \sum_{i=1}^k \theta_i u_i u_i^T, \ \ \theta_i \geq 0, \quad (15)$$

where $\Theta = [\theta_1, ..., \theta_K]^T$ is a vector of the non-negative weights in the linear combination. Low-rank distance metrics are desirable because they not only can drastically reduce the computational requirements for working with the data, but also they can often provide noise reduction as well.
By replacing A with the right-hand side of Eq. 15, the proposed optimization problem can be reformulated as

$$\begin{aligned}[\Theta^*, \mu^*] = \underset{\Theta,\mu}{argmin} \Bigg[& \sum_{(x_i,x_j)\in S} \log\big(1 + exp(-\Theta^T V_{ij} + \mu)\big) \\ & + \sum_{(x_i,x_j)\in D} \log\big(1 + exp(\Theta^T V_{ij} - \mu)\big) \\ & + \lambda \sum_{(x_i,x_j)\in U} H\big(1/\big(1 + exp(-\Theta^T V_{ij} + \mu)\big)\big)\Bigg] \\ & s.t.\ \mu \geq 0, \Theta \geq 0, \end{aligned} \quad (16)$$

where $V_{ij} = [v_{ij}^1, v_{ij}^2, ..., v_{ij}^k]^T$ is a k dimentional vector with

$$v_{ij}^l = u_l^T (x_i - x_j)(x_i - x_j)^T u_l, \ \ l = 1, ..., k. \quad (17)$$

2.2 Kernel-Based (non-linear) Metric Learning

In this section, we introduce a kernelized version of the proposed linear metric learning method presented in the previous section. In order to learn our linear metric in Reproducing Kernel Hilbert Space (RKHS), we map the data into a high-dimensional feature space $\mathcal{F}$ equipped with an inner product as

$$\phi : \mathcal{R}^m \rightarrow \mathcal{F}, \quad (18)$$

where ϕ is a non-linear function which maps data points from input space to the high-dimentional feature space $\mathcal{F}$. Suppose we use a kernel function $K(.,.)$ which satisfies Mercer's condition (many choices for kernel functions satisfy Mercer's condition, such as polynomial, RBF, and exponential kernels), then it can be proved [13] that there exist a nonlinear mapping ϕ_K which can be implicitly specified by K as

$$\phi_K(x_i)^T \phi_K(x_j) = K(x_i, x_j), \ \forall x_i, x_j \in \mathcal{R}^m. \quad (19)$$

Using Eq. 19, we can make use of *kernel trick* idea which is also known as *kernel substitution*. Precisely speaking, if we have an algorithm, in which input data points enters only in the form of inner products, we can replace that inner product with some other choice of kernels (kernels which satisfy Mercer's condition).
Let $\mathbf{\Phi} = [\phi(x_1), ..., \phi(x_N)]$ be the matrix containing the transformed training data, so that each data point $x_i (i =$

$1, ..., N)$ is projected onto a point $\phi(x_i)$, and $\boldsymbol{K} = [K(x_i, x_j)] = \boldsymbol{\Phi}^T\boldsymbol{\Phi}$ be the corresponding kernel matrix. We now perform our linear metric learning in the feature space, which implicitly defines a nonlinear metric in the original data space.
Let $R_\phi = \frac{1}{N}\sum_{i=1}^{N}\phi(x_i)\phi(x_i)^T$ be the sample correlation matrix in feature space, and $\{a_i\}_{i=1}^{k}$ be the top k $(k \leq N))$ eigenvectors of the matrix R_ϕ. Our goal is to reformulate the objective function of Eq. 16 in the feature space without having to work explicitly in that space. Motivated by the idea of Kernel PCA algorithm [12], we propose an optimization problem that is formulated based on the elements of the kernel matrix $\boldsymbol{K}$. We know that each eigenvector $a_i(i = 1, ..., k)$ can be computed by eigenvector expansion of R_ϕ as

$$R_\phi a_i = \sigma_i a_i, \;\; i = 1, ..., k, \tag{20}$$

where σ_i is the corresponding eigenvalue of a_i. From the definition of R_ϕ, Eq. 20 can be reformulated as

$$\frac{1}{N}\sum_{n=1}^{N}\phi(x_n)(\phi(x_n)^T a_i) = \sigma_i a_i, \;\; i = 1, ..., k. \tag{21}$$

From Eq. 21, we can see that each eigenvector a_i can be represented by a linear combination of the $\{\phi(x_n)\}_{n=1}^{N}$ as

$$a_i = \boldsymbol{\Phi} z_i, \;\; i = 1, ..., k, \tag{22}$$

where z_i is an N dimensional coefficient vector. By substituting the above expansion back into Eq. 21, we obtain

$$\frac{1}{N}\sum_{n=1}^{N}\phi(x_n)\phi(x_n)^T\boldsymbol{\Phi} z_i = \sigma_i\boldsymbol{\Phi} z_i, \;\; i = 1, ..., k. \tag{23}$$

By multiplying both side of the above equation by $\boldsymbol{\Phi}^T$, we obtain

$$\boldsymbol{K}^2 z_i = N\sigma_i\boldsymbol{K} z_i, \;\; i = 1, ..., k \tag{24}$$

where we used the fact that $\boldsymbol{K}(x_i, x_j) = \phi(x_i)^T\phi(x_j)$. Now we can compute each z_i by solving the following eigenvector problem

$$\boldsymbol{K} z_i = N\sigma_i z_i, \;\; i = 1, ..., k, \tag{25}$$

where we have removed a factor of $\boldsymbol{K}$ from both sides of Eq. 24. Having computed the vectors $z_i(i = 1, ..., k)$ based on the above eigenvector problem, the optimization problem in Eq. 16 can be cast in terms of the kernel matrix as

$$\begin{aligned}\Delta^* = \underset{\Delta}{argmin} \Bigg[& \sum_{(x_i,x_j)\in S} \log\big(1 + exp(-\Delta^T V_{ij}^\phi)\big) \\ & + \sum_{(x_i,x_j)\in D} \log\big(1 + exp(\Delta^T V_{ij}^\phi)\big) \\ & + \lambda \sum_{(x_i,x_j)\in U} H\big(1/\big(1 + exp(-\Delta^T V_{ij}^\phi)\big)\big)\Bigg] \\ & s.t.\ \Delta \geq 0,\end{aligned} \tag{26}$$

where $\Delta = [\Theta^T, \mu]^T$, and $V_{ij}^\phi = [\mathrm{v}_{ij}^1, \mathrm{v}_{ij}^2, ..., \mathrm{v}_{ij}^l, ..., \mathrm{v}_{ij}^k, -1]^T$ is a $k+1$ dimensional vector with

$$\begin{aligned}\mathrm{v}_{ij}^l &= a_l^T(\phi(x_i) - \phi(x_j))(\phi(x_i) - \phi(x_j))^T a_l \\ &= z_l^T\boldsymbol{\Phi}^T(\phi(x_i) - \phi(x_j))(\phi(x_i) - \phi(x_j))^T\boldsymbol{\Phi} z_l \\ &= z_l^T(\boldsymbol{K}_{(.,i)} - \boldsymbol{K}_{(.,j)})(\boldsymbol{K}_{(.,i)} - \boldsymbol{K}_{(.,j)})^T z_l,\end{aligned} \tag{27}$$

and $\boldsymbol{K}_{(.,i)}$ is the i-th column of matrix $\boldsymbol{K}$.

3. OPTIMIZATION PROCEDURE

In this section, we describe the optimization procedure for the proposed objective function (Eq. 26). Solving (26) is a challenging task because it is not convex. Precisely speaking, although the first two component of the proposed objective function is convex, the entropy component of the objective function is concave, hence their weighted sum is usually not convex, except for $\lambda = 0$. Hence, the optimization surface is expected to possess local minima.
In order to solve (26), we reformulate the minimization problem in Eq. 26 as the following maximization problem.

$$\begin{aligned}\Delta^* = \underset{\Delta}{argmax} \Bigg[& \sum_{(x_i,x_j)\in S} \log P(y_{ij} = 1 \mid V_{ij}^\phi; \Delta) \\ & + \sum_{(x_i,x_j)\in D} \log P(y_{ij} = -1 \mid V_{ij}^\phi; \Delta) \\ & + \lambda \sum_{(x_i,x_j)\in U} P(y_{ij} = 1 \mid V_{ij}^\phi; \Delta) \log P(y_{ij} = 1 \mid V_{ij}^\phi; \Delta) \\ & + \lambda \sum_{(x_i,x_j)\in U} P(y_{ij} = -1 \mid V_{ij}^\phi; \Delta) \log P(y_{ij} = -1 \mid V_{ij}^\phi; \Delta)\Bigg] \\ & s.t.\ \Delta > 0.\end{aligned} \tag{28}$$

In order to find a suitable local maxima of the above objective function, we use the Deterministic Annealing EM (DAEM) algorithm [14]. This algorithm is a simple generalization of the standard EM algorithm that doesn't have the initialization dependence problem. More precisely, the DAEM algorithm includes a *temperature* parameter ($T = 1 - \lambda$) which controls the influence of unreliable model parameters, and this annealing process can reduce the dependency on initial model parameters [14]. DAEM starts from a possibly concave conditional likelihood ($\lambda = 0$, i.e., $T = 1$) and the temperature is gradually decreased (gradually increasing λ) until it reaches some predetermined value $1 - \lambda_0 = T_0 \geq 0$, to return a good local maximum of the objective function. For each trial value of λ, the corresponding solution is computed by a two-step iterative process, where the soft assignments for class labels of unlabeled data are calculated at the E-step, and at M-step, the expected log-likelihood is maximized.
Using the DAEM algorithm for the proposed model (Eq. 28), the posterior distribution for the class label of each $(x_i, x_j) \in U$ given the current value of Δ at iteration t (Δ^t) can be computed as

$$\begin{aligned}&Q(y_{ij} = 1 \mid V_{ij}^\phi, \Delta^t) = \\ &\frac{P(y_{ij} = 1 \mid V_{ij}^\phi, \Delta^t)^{\frac{1}{1-\lambda}}}{P(y_{ij} = 1 \mid V_{ij}^\phi, \Delta^t)^{\frac{1}{1-\lambda}} + \big(1 - P(y_{ij} = 1 \mid V_{ij}^\phi, \Delta^t)\big)^{\frac{1}{1-\lambda}}}.\end{aligned} \tag{29}$$

The M-step then consists in maximizing the expected log-likelihood with respect to the Δ,

$$\begin{aligned}\Delta^{t+1} = \underset{\Delta}{argmax} \Bigg[& \sum_{x_i,x_j,i\neq j} Q(y_{ij} = 1 \mid V_{ij}^\phi, \Delta^t) \log P(y_{ij} = 1 \mid V_{ij}^\phi, \Delta) \\ & + \big(1 - Q(y_{ij} = 1 \mid V_{ij}^\phi, \Delta^t)\big) \log\big(1 - P(y_{ij} = 1 \mid V_{ij}^\phi, \Delta)\big)\Bigg] \\ & s.t.\ \Delta \geq 0,\end{aligned} \tag{30}$$

where $Q(y_{ij} = 1 \mid V_{ij}^\phi, \Delta^t)$ is equal to the right-hand side

of Eq. 29 for unlabeled data points ($(x_i, x_j) \in U$), and $Q(y_{ij} = 1 \mid V_{ij}^{\phi}, \Delta^t) = \delta_{y_{ij}=1}$ for labeled samples. It can be seen that the optimization problem (30) is concave with respect to the Δ. Hence, it can be solved efficiently using convex optimization tools [3]. Algorithm 1 presents the pseudocode of DAEM algorithm for solving (28).

Algorithm 1 DAEM algorithm for solving (28)

Input: $x_1, x_2, ..., x_N$, $\boldsymbol{K}$, λ^0, and Δ^0
Output: Δ^t
initialization: Set $\lambda = \lambda^0$, and $t = 0$

1. Compute $z_1, ..., z_k (k \leq N)$, where $\{z_i\}_{i=1}^{k}$ are the k eigenvectors corresponding to the non-zero eigenvalues of $\boldsymbol{K}$;
2. Compute $V_{ij}^{\phi} (i = 1, ..., N,\ j = 1, ..., N)$ using Eq. 27;
3. Iterate EM-steps with λ fixed until the objective function of Eq. 30 converged:
   E step: compute $Q(y_{ij} \mid V_{ij}^{\phi}, \Delta^t)$ for each unlabeled pair using Eq. 29;
   M step: compute Δ^{t+1} using Eq. 30;
   t = t+1;
4. Increase λ;
5. If $\lambda > 1$, stop the procedure. Otherwise go to step 3.

4. EXPERIMENTAL RESULTS

In this section, we explain experiments that we have conducted to evaluate the performance of our method. We demonstrate results of the proposed method on some synthetic and real-world datasets and compare them with results of some recently introduced metric learning methods.

4.1 Experimental Setup

We compare our non-linear method with the metric learning algorithms introduced in [15, 18], as they are the most effective methods considering both positive and negative constraints. We also include the methods introduced in [4, 19, 1] as non-linear metric learning methods in our evaluations. As in [18, 4, 19, 1], we use the Euclidean distance (without metric learning) for the baseline comparison and apply the k-means clustering algorithm with different distance metrics to evaluate the efficiency of these metrics. For the kernel-based methods, we apply the kernel k-means algorithm on the obtained kernels.
Thus, we compare the performance of the following algorithms:

- k-means without metric learning (Euclidean);
- k-means with the metric learning method introduced in [15] (Xiang's);
- k-means with the LLMA [4] method for metric learning (LLMA);
- k-means with the extended RCA [18] method for metric learning (ERCA);
- kernel k-means with the kernel obtained by the kernel-β method [19] (Kernel-β);
- kernel k-means with the non-linear metric learning method introduced in [1] (Baghshah'c);
- k-means with our probabilistic non-linear metric learning method (PM);

In our experiments, we use the exponentional kernel $K(x, y) = exp(-\|x - y\|/\eta)$ for all datasets. We also set the kernel parameter η as

$$\eta = 2 \sum_{i<j} \frac{\|x_i - x_j\|_2^2}{N(N-1)}. \tag{31}$$

The regularization parameter λ was updated by $\lambda^i = \sqrt{i/T_N}$, where λ^i is the value of λ at i-th iteration, and T_N is the total number of the iterations. From preliminary experiments, we used $T_N = 20$, and 10 iterations of the EM-steps were conducted at each temperature.
The number of pairwise similarity and dissimilarity constraints is set to be equal $|S| = |D|$. We also generate 20 different S and D sets for each data set. Moreover, we run the k-means algorithm 20 times with different random initializations for each S and D set, and report the average rand index over the 20 runs.

4.2 Performance Measure

In order to measure the performance of the clustering methods in our experiments, we use the *Rand index* which is the most widely used measure for evaluating the performance of metric learning algorithms. It shows how well the clustering results agree with the ground truth clusters [1]. This measure is defined as [16]

$$RI = 2\frac{N_s + N_d}{N(N-1)}, \tag{32}$$

where N_s is the number of data pairs assigned to the same cluster, both in the ground truth and the resultant clustering, and N_d is the number of data pairs assigned to different clusters both in the ground truth and the resultant clustering. Since this index is biased toward assigning data points to different clusters when there are more than two clusters [16], we use the modified Rand index introduced in [16], and is defined as

$$\begin{aligned} \hat{RI} =& \frac{1}{2} \times \frac{\sum_{i>j} \delta(c_i = c_j \wedge \hat{c}_i = \hat{c}_j)}{\sum_{i>j} \delta(\hat{c}_i = \hat{c}_j)} \\ &+ \frac{1}{2} \times \frac{\sum_{i>j} \delta(c_i \neq c_j \wedge \hat{c}_i \neq \hat{c}_j)}{\sum_{i>j} \delta(\hat{c}_i \neq \hat{c}_j)}, \end{aligned} \tag{33}$$

where $\delta(.)$ is an indicator function (i.e., $\delta(true) = 1$ and $\delta(false) = 0$), $\hat{c}_i$ is the cluster to which x_i is assigned by the clustering algorithm, and c_i is the correct cluster assignment.
Using this measure, the matched pairs and mismatched pairs are assigned weights to give them equal chances of occurrence ($\frac{1}{2}$) [4].

4.3 Experiments on Synthetic Data Sets

We first perform some experiments on three synthetic datasets (circle, cross, moon) demonstrated in Fig. 1. Data points shown with the same color and point style belong to the same cluster. For these datasets, we set the number of constraints to $nc = |S| = |D| = 15$. Fig. 2 shows the results of applying different algorithms on these datasets as box-plots. From that figure, we can see that the K-means algorithm has a poor performance on these datasets. Furthermore, these datasets cannot be clustered well using the linear methods due to the non-linear structure of these datasets. We can also observe that our method has better performance

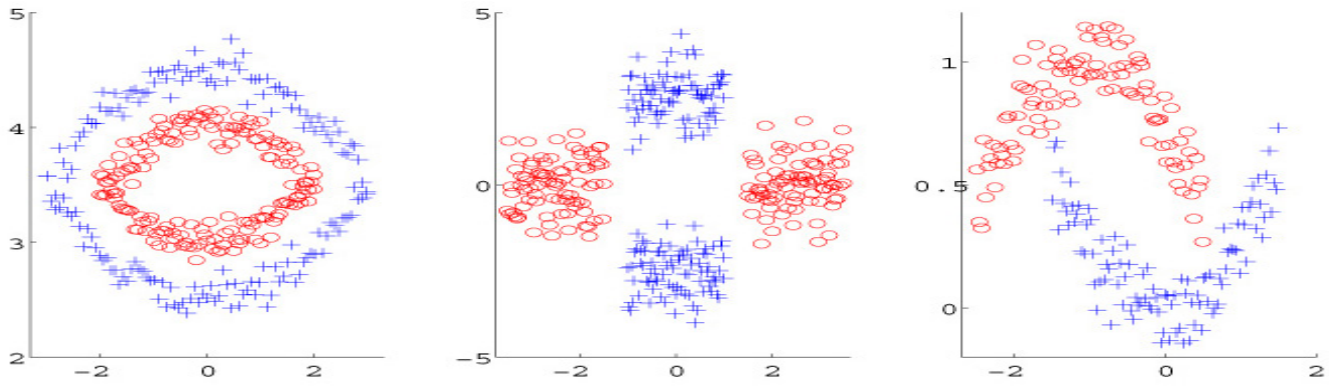

Figure 1: Synthetic datasets. From left to right: dataset 1 (circle), dataset 2 (cross), dataset 3 (moon).

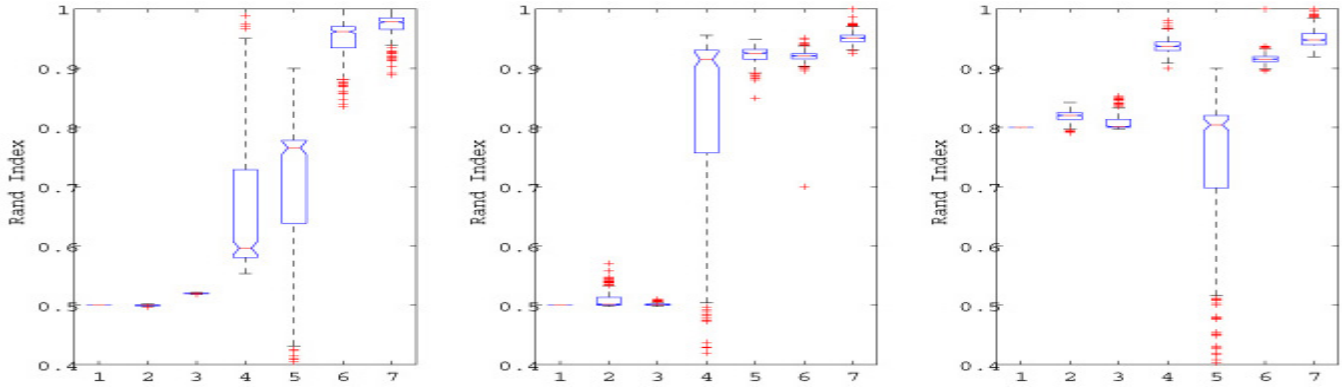

Figure 2: From left to right : clustering results on circle, cross, and moon datasets using different metric learning methods: (1) Euclidean, (2) Xiang's, (3) ERCA, (4) LLMA, (5) Kernel-β, (6) Baghshah's, (7) PM.

than the LLMA and kernel-β as non-linear metric learning algorithms, because our method utilizes the information of unlabeled data but these algorithms do not consider the unlabeled samples. Although the Baghshah's algorithm gives good results on these data sets, our method performs better than this algorithm on all of them.

4.4 Experiments on UCI datasets

In this section, we conduct experiments on nine real-world datasets obtained from the Machine Learning Repository[1] of the University of California, Irvine (UCI). The properties of these datasets are shown in Table 1, where N denotes the number of data points, C denotes the number of classes, and m denotes the number of features for each dataset. All of the nine data sets are normalized before use in the clustering algorithms (each feature is normalized to zero mean and unit variance).

The average Rand index of each method vs. the number of constraints on the nine UCI data sets has been demonstrated in Fig. 3, from which we can see that the proposed non-linear method generally outperforms all the other methods. Again, this is due to the fact that our method makes use of unlabeled data while most of the other methods do not. By comparing the Baghshah's method with our method, we can also observe that the proposed regularization term (entropy term) is more appropriate than the Baghshah's regularization term (geometrical structure term).

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

Table 1: Properties of the UCI datasets used in our experiments

dataset	N	C	m
Soybean	47	4	35
Protein	116	6	20
Wine	178	3	13
Sonar	208	2	60
Glasses	214	6	10
Ionosphere	351	2	34
Boston housing	506	3	13
Breast cancer	569	2	31
Balance	625	3	4

4.5 Experiments on MNIST dataset

Finally, we apply our method on the MNIST database [10] which consists of 70000 28 $\times$ 28 grayscale images. In our experiments, we choose 500 images randomly for each digit. We also set the number of constraints to $nc = |S| = |D| = 30$ for all experiments performed on subsets of the MNIST dataset. Table 2 shows the results of different clustering algorithms for three-digit subsets. For each algorithm, we show the mean rand index and standard deviation over different runs (corresponding to different sets of constraints and

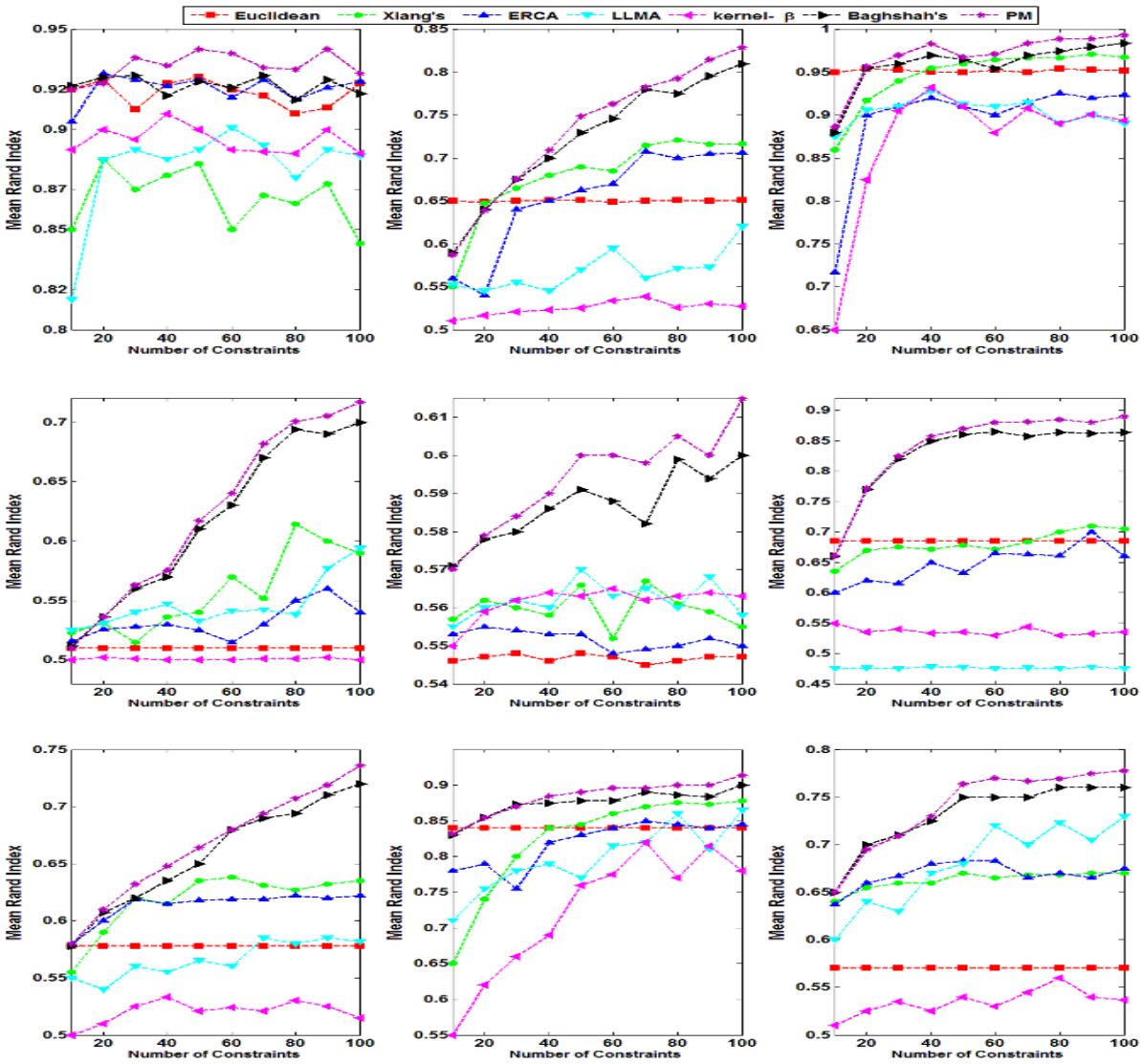

Figure 3: Average Rand index curves of different methods on some UCI datasets. First row (from left to right) : Soybean, Protein, Wine. Second row (from left to right) : Sonar, Glasses, Ionosphere. Last row (from left to right) : Boston housing, Breast cancer, Balance.

Table 2: Mean and variance of the Rand index values obtained for different methods on some subsets of the MNIST database.

subset	Euclidean	Xiang's	ERCA	LLMA	Kernel-β	Baghshah's	PM
$\{1,2,3\}$	**0.868 ± 0.04**	0.862 ± 0.05	0.853 ± 0.07	0.636 ± 0.05	0.569 ± 0.08	0.865 ± 0.04	0.867 ± 0.03
$\{4,5,6\}$	0.771 ± 0.06	0.828 ± 0.08	0.871 ± 0.07	0.715 ± 0.04	0.561 ± 0.06	0.902 ± 0.05	**0.914 ± 0.05**
$\{7,8,9\}$	0.708 ± 0.02	0.745 ± 0.05	0.774 ± 0.04	0.645 ± 0.06	0.523 ± 0.01	0.819 ± 0.01	**0.830 ± 0.01**

different initializations of the k-means clustering algorithm). From that table, we can see that our metric learning algorithm gives the best clustering results for two out of three subsets.

5. CONCLUSION

In this paper, we introduced a novel non-linear metric learning algorithm for constrained clustering. The proposed method uses the information of unlabeled data along with positive and negative constraints to find an appropriate metric. Some experiments on the synthetic, UCI, and MNIST datasets demonstrated the superiority of our method over some existing linear and non-linear metric learning methods. As the future work, we will apply the proposed methods on other real-world problems such as content-based image retrieval [5].

6. REFERENCES

[1] M. S. Baghshah and S. B. Shouraki. Semi-supervised metric learning using pairwise constraints. *IJCIA*, 2009.

[2] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *JMLR*, 6:937–965, 2005.

[3] S. Boyd and L. Vandenberge. *Convex Optimization*. . Cambridge University Press, , Cambridge, UK, 2003.

[4] H. Chang and D. Yeung. Locally linear metric adaptation with application to semi-supervised clustering and image retrieval. *Pattern Recognition*, 39:1255–1264, 2006.

[5] H. Chang, D. Y. Yeung, and W. K. Cheung. Relaxational metric adaptation and its application to semi-supervised clustering and content-based image retrieval. *,âĂİ Pattern Recognition*, 39(10):1905 –1917, 2006.

[6] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. *ICML*, pages 1253 – 1264, June 2007.

[7] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. *Advances in Neural Information Processing Systems (NIPS)*, 11, 2005.

[8] S. C. Hoi, W. Liu, M. R. Lyu, and Y. Ma. Learning distance metrics with contextual constraints for image retrieval. *CVPR*, 2006.

[9] S. C. H. Hoi, R. Jin, and M. R. Lyu. Learning non parametric kernel matrices from pairwise constraints. *International Conference on Machine Learning (ICML)*, pages 361–368, 2007.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *IEEE Conference Proceedings*, 1998.

[11] S. T. Roweis and L. K. Saul. ,non-linear dimensionality reduction by locally linear embedding. *, Science290*, 290:2323–2326, 2000.

[12] B. Scholkopf, A. Smola, and K. R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299 –1319, 1998.

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

[14] N. Ueda and R. Nakano. Deterministic anealing em algorithm. *Neural Networks*, 11:271–282, 1998.

[15] S. Xiang, F. Nie, and C. Zhang. Learning a mahalanobis distance metric for data clustering and classification. *Pattern Recognition*, 2008.

[16] E. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side information. *NIPS*, 2003.

[17] L. Yang and R. Jin. Distance metric learning : a comprehensive survey. *Technical Report, Michigan State University*, 2006.

[18] D. Y. Yeung and H. Chang. Extending the relevant component analysis algorithm for metric learning using both positive and negative equivalence constraints. *Pattern Recognition*, 39:1007–1010, 2006.

[19] D. Y. Yeung and H. Chang. A kernel approach for semi-supervised metric learning. *IEEE Transactions on Neural Networks*, 18(1):141–149, 2007.

[20] J. Zhuang, I. W. Tsang, and S. C. H. Hoi. Simple npkl : simple non-parametric kernel learning. *International Conference on Machine Learning (ICML)*, pages 1273–1280, 2009.