Relaxational Metric Adaptation and Its Application to Semi-Supervised Clustering and Content-Based Image Retrieval

Hong Chang & Dit-Yan Yeung

Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong

William K. Cheung

Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong

Corresponding author: Dit-Yan Yeung, `dyyeung@cs.ust.hk`, +852-2358-1477 (fax)

Abstract

The performance of many supervised and unsupervised learning algorithms is very sensitive to the choice of an appropriate distance metric. Previous work in metric learning and adaptation has mostly been focused on classification tasks by making use of class label information. In standard clustering tasks, however, class label information is not available. In order to adapt the metric to improve the clustering results, some background knowledge or side information is needed. One useful type of side information is in the form of pairwise similarity or dissimilarity information. Recently, some novel methods (e.g., the parametric method proposed by Xing *et al.*) for learning global metrics based on pairwise side information have been shown to demonstrate promising results. In this paper, we propose a nonparametric method, called relaxational metric adaptation (RMA), for the same metric adaptation problem. While RMA is local in the sense that it allows locally adaptive metrics, it is also global because even patterns not in the vicinity can have long-range effects on the metric adaptation process. Experimental results for semi-supervised clustering based on both simulated and real-world data sets show that RMA outperforms Xing

et al.'s method under most situations. Besides applying RMA to semi-supervised learning, we have also used it to improve the performance of content-based image retrieval systems through metric adaptation. Experimental results based on two real-world image databases show that RMA significantly outperforms other methods in improving the image retrieval performance.

Key words: Distance metric, Nonparametric method, Semi-supervised clustering, Constrained k-means, Side information, Pairwise similarity and dissimilarity, Content-based image retrieval

---

1 Introduction

Many machine learning and pattern recognition algorithms involve the use of a distance metric [1]. Commonly used methods include nearest neighbor classifiers, radial basis function networks and support vector machines for classification and the k-means algorithm for clustering. The performance of these methods often depends very much on the choice of an appropriate metric. Instead of determining a metric manually, a more promising approach is to learn an appropriate metric from data automatically. This idea is not new, though. It can be dated back to the early work on optimizing the metric for k-nearest neighbor density estimation [2]. Optimal local metric [3] and optimal global metric [4] were also developed for nearest neighbor classification. More recent research continued to develop various locally adaptive metrics for nearest neighbor classifiers [5–11]. Besides nearest neighbor classifiers, there are other methods that also perform metric learning based on nearest neighbors, e.g., radial basis function networks and variants [12].

While class label information is available for metric learning in classification (or supervised learning) tasks, such information is not available in standard clustering (or unsupervised learning) tasks. In order to adapt the metric to improve the clustering results, some additional background knowledge is needed. One approach to the introduction of additional knowledge is called semi-supervised learning [13], which learns with both labeled and unlabeled data. [1]

[1] Typically the set of labeled patterns is very small compared with the set of unlabeled patterns.

Based on this paradigm, [14,15] proposed local metric learning methods to improve clustering and visualization results. [16,17] proposed a parametric distance metric learning method that learns better metrics for improving both classification and clustering tasks. Another approach to the introduction of background knowledge or side information is in the form of pairwise similarity or dissimilarity information. [18,19] proposed using such pairwise information to improve clustering. However, they did not incorporate metric learning into the clustering algorithms. [20] extended their method nicely by using pairwise side information to learn a global metric before performing clustering with constraints. Instead of using an iterative algorithm as in [20], a more efficient, non-iterative algorithm called relevant component analysis (RCA) [21] was proposed for learning a global Mahalanobis metric. More recently, [22,23] proposed a nonmetric distance function learning algorithm called DistBoost by boosting the hypothesis over the product space with Gaussian mixture models as weak learners. However, both RCA and DistBoost can only incorporate similarity constraints. [24] introduced the use of discriminant kernels for metric learning. [25] established the relationship between metric learning and kernel matrix adaptation.

To summarize, we can categorize metric learning and adaptation methods according to two different dimensions. The first dimension is concerned with whether (*supervised*) classification or (*unsupervised*) clustering is performed. Most methods were proposed for classification tasks, but some recent methods extended metric learning to clustering tasks with the existence of background knowledge. Background knowledge may be in the form of class label information or pairwise (dis)similarity information. The second dimension categorizes metric learning methods into *global* and *local* ones. Provided that sufficient data are available, local metric learning is generally preferred as it is more flexible in allowing different local metrics at different locations of the input space. In this paper, we propose a new metric adaptation method for clustering with pairwise (dis)similarity side information. While our method is local in the sense that it allows locally adaptive metrics, our method is also global because even patterns not in the vicinity can have long-range effects on the metric adaptation process.

2 Our Metric Adaptation Method

Let us denote the set of n patterns or data points by $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, the set of similar pairs by $\mathcal{S}_0$, and the set of dissimilar pairs by $\mathcal{D}_0$. $\mathcal{S}_0$ and $\mathcal{D}_0$ are both represented as sets of pairs of patterns, or *pattern pairs*, where each pattern pair $(\mathbf{x}_i, \mathbf{x}_j)$ indicates that patterns $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar or dissimilar to each other, respectively. Given the two sets $\mathcal{S}_0$ and $\mathcal{D}_0$ of pairwise side information, we first convert them into $\mathcal{S}$ and $\mathcal{D}$, respectively. This conversion consists of two steps. The first step finds the transitive closure of all pattern pairs in $\mathcal{S}_0$ to form $\mathcal{S}$. Suppose $\mathbf{x}_i$ and $\mathbf{x}_j$ form a pair and $\mathbf{x}_j$ and $\mathbf{x}_k$ form another pair in $\mathcal{S}_0$. We introduce $\mathbf{x}_i$ and $\mathbf{x}_k$ as a new similar pair to $\mathcal{S}$ if the pair is not yet explicitly represented in $\mathcal{S}_0$. In the second step, we try to infer from the pairs in $\mathcal{S}$ and $\mathcal{D}_0$. If $\mathbf{x}_i$ and $\mathbf{x}_j$ form a similar pair in $\mathcal{S}$ and $\mathbf{x}_i$ and $\mathbf{x}_k$ form a dissimilar pair in $\mathcal{D}_0$, we introduce $\mathbf{x}_j$ and $\mathbf{x}_k$ as a new dissimilar pair to $\mathcal{D}$ if it is not in $\mathcal{D}_0$. Through these two steps, the implicit knowledge that can be inferred from $\mathcal{S}_0$ and $\mathcal{D}_0$ is represented explicitly in $\mathcal{S}$ and $\mathcal{D}$.

Our metric adaptation method, called relaxational metric adaptation (RMA), is an iterative algorithm that repeatedly adjusts the locations of the patterns in the input space, such that similar patterns tend to get closer while dissimilar patterns tend to move away from each other. Each iteration of the RMA algorithm has two phases. The first phase affects only the patterns involved in $\mathcal{S}$ and $\mathcal{D}$. Based on the pairwise (dis)similarity information, it tries to move similar patterns closer together and dissimilar patterns farther apart. In the second phase, all other patterns are also affected by the pattern movement in the first phase. Since pattern movement is equivalent to changing the metric of the input space implicitly, the RMA algorithm is essentially an iterative metric adaptation procedure. Moreover, it is a nonparametric learning algorithm since there is no parametric model governing metric adaptation.

2.1 Phase 1: Local Changes

For each pattern pair $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}$, the squared Euclidean distance between patterns $\mathbf{x}_i$ and $\mathbf{x}_j$ is $\|\mathbf{x}_i - \mathbf{x}_j\|^2$. The sum of squared Euclidean distances for

all pattern pairs in $\mathcal{S}$ is given by

$$d_{\mathcal{S}} = \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}} \|\mathbf{x}_i - \mathbf{x}_j\|^2.$$

We also define the sum of inverse squared Euclidean distances for all pattern pairs in $\mathcal{D}$ as

$$d_{\mathcal{D}} = \sum_{(\mathbf{x}_k, \mathbf{x}_l) \in \mathcal{D}} \frac{1}{\|\mathbf{x}_k - \mathbf{x}_l\|^2}.$$

If the metric is good, both $d_{\mathcal{S}}$ and $d_{\mathcal{D}}$ should be sufficiently small. Based on this observation, we define an optimization criterion as follows:

$$J = d_{\mathcal{S}} + \alpha d_{\mathcal{D}}, \tag{1}$$

for some parameter $\alpha > 0$ representing the relative importance of the two terms. We would like to move the patterns in $\mathcal{S}$ and $\mathcal{D}$ so as to minimize J. Note that J is defined in such a way that it is affected more by a pattern pair in $\mathcal{S}$ with longer inter-pattern distance but by a pattern pair in $\mathcal{D}$ with shorter inter-pattern distance. This is desirable since a pair of similar patterns that are far apart and a pair of dissimilar patterns that are close to each other are expected to update their locations more significantly.

Consider the gradient vector of J with respect to a pattern $\mathbf{x}_i$ in one pattern pair of $\mathcal{S}$ or $\mathcal{D}$:

$$\nabla_{\mathbf{x}_i} J = \nabla_{\mathbf{x}_i} d_{\mathcal{S}} + \alpha \nabla_{\mathbf{x}_i} d_{\mathcal{D}}. \tag{2}$$

Only the first term of Equation (2) remains if $\mathbf{x}_i$ appears only in $\mathcal{S}$, and only the second term remains if $\mathbf{x}_i$ appears only in $\mathcal{D}$. If $\mathbf{x}_i$ appears in both $\mathcal{S}$ and $\mathcal{D}$, then both terms of Equation (2) do not vanish. Although one pattern may appear in multiple pattern pairs in $\mathcal{S}$ and $\mathcal{D}$, we could consider the effect of one pattern pair at a time. Thus the gradient vector of $d_{\mathcal{S}}$ due to pattern pair $(\mathbf{x}_i, \mathbf{x}_j)$ of $\mathcal{S}$ with respect to pattern $\mathbf{x}_i$ is

$$\nabla_{\mathbf{x}_i} d_{\mathcal{S}}|_{(\mathbf{x}_i, \mathbf{x}_j)} = 2\left(\mathbf{x}_i - \mathbf{x}_j\right),$$

while the gradient vector of $d_{\mathcal{D}}$ due to pattern pair $(\mathbf{x}_k, \mathbf{x}_l)$ of $\mathcal{D}$ with respect to pattern $\mathbf{x}_k$ is

$$\nabla_{\mathbf{x}_k} d_{\mathcal{D}}|_{(\mathbf{x}_k, \mathbf{x}_l)} = -\frac{2}{\|\mathbf{x}_k - \mathbf{x}_l\|^4}\left(\mathbf{x}_k - \mathbf{x}_l\right).$$

Using a gradient-descent procedure to minimize J, the patterns $\mathbf{x}_i$ and $\mathbf{x}_k$ should move in directions opposite to their gradient directions, i.e.,

$$\begin{aligned}
\Delta\mathbf{x}_i|_{(\mathbf{x}_i,\mathbf{x}_j)} &= -\eta\,(\mathbf{x}_i - \mathbf{x}_j) \\
&= \eta\|\mathbf{x}_i - \mathbf{x}_j\|\mathbf{u}_{\mathbf{m}_{ij}\leftarrow\mathbf{x}_i}, \qquad (3) \\
\Delta\mathbf{x}_k|_{(\mathbf{x}_k,\mathbf{x}_l)} &= \frac{\eta\alpha}{\|\mathbf{x}_k - \mathbf{x}_l\|^4}\,(\mathbf{x}_k - \mathbf{x}_l) \\
&= \frac{\eta\alpha}{\|\mathbf{x}_k - \mathbf{x}_l\|^3}\mathbf{u}_{\mathbf{x}_k\leftarrow\mathbf{m}_{kl}}, \qquad (4)
\end{aligned}$$

where $\eta > 0$ is a learning rate parameter, $\mathbf{m}_{ab}$ is the midpoint between $\mathbf{x}_a$ and $\mathbf{x}_b$, and $\mathbf{u}_{\mathbf{x}_a\leftarrow\mathbf{x}_b}$ is the unit vector pointing from $\mathbf{x}_b$ to $\mathbf{x}_a$. It is easy to see that $\mathbf{x}_i$ moves towards $\mathbf{x}_j$ and $\mathbf{x}_k$ moves away from $\mathbf{x}_l$. To consider all patterns in the pattern pairs, Equation (3) is applied $2|\mathcal{S}|$ times and Equation (4) is applied $2|\mathcal{D}|$ times. However, the number of different patterns involved is in general less than $2(|\mathcal{S}| + |\mathcal{D}|)$ since some patterns can appear in multiple pattern pairs. [2]

Although location changes are computed for all patterns involved in $\mathcal{S}$ and $\mathcal{D}$, the actual changes will only be made simultaneously in a batch mode at the end of the second phase.

2.2 Phase 2: Global Changes

The second phase generalizes pattern movement in the first phase to all patterns, allowing long-range effects to influence metric adaptation globally.

For each pattern pair $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}$, pattern movement of $\mathbf{x}_i$ and $\mathbf{x}_j$ towards each other (as a result of the first phase) will influence all other patterns, i.e., all patterns in $\mathcal{X}\backslash\{\mathbf{x}_i, \mathbf{x}_j\}$. Similarly, pattern movement of $\mathbf{x}_k$ and $\mathbf{x}_l$ for each pattern pair $(\mathbf{x}_k, \mathbf{x}_l) \in \mathcal{D}$ will also influence all patterns in $\mathcal{X}\backslash\{\mathbf{x}_k, \mathbf{x}_l\}$.

Consider a pattern $\mathbf{x}_r \in \mathcal{X}\backslash\{\mathbf{x}_i, \mathbf{x}_j\}$. Without loss of generality, let us assume that $\mathbf{x}_r$ is closer to $\mathbf{x}_i$ than $\mathbf{x}_j$, i.e., $\|\mathbf{x}_r - \mathbf{x}_i\| < \|\mathbf{x}_r - \mathbf{x}_j\|$. Pattern $\mathbf{x}_r$ is

[2] This is very likely the case if the pattern pairs are reasonably dense with respect to the data set size n.

updated in a way similar to Equation (3) for $\mathbf{x}_i$, as follows:

$$\Delta \mathbf{x}_r|_{(\mathbf{x}_i,\mathbf{x}_j)} = \eta \|\mathbf{x}_i - \mathbf{x}_j\| \mathcal{N}_{ij}(\mathbf{x}_r) \mathbf{u}_{\mathbf{m}_{ij} \leftarrow \mathbf{x}_r}, \tag{5}$$

where $\mathcal{N}_{ij}(\cdot)$ is a neighborhood function that depends on both $\mathbf{x}_i$ and $\mathbf{x}_j$. It is natural to use a Gaussian neighborhood function so that the influence of $\mathbf{x}_i$ on $\mathbf{x}_r$ decreases with the distance between them. The question is how exactly the Gaussian function should be defined. One possibility is to use a hyperspherical Gaussian function, meaning that the covariance matrix is diagonal with all diagonal entries being the same (σ_h^2). This scheme is very simple to implement. Another possibility is to allow a more general Gaussian function with full covariance matrix. This alternative scheme is slightly more complicated and hence we will give the details below.

Let us assume that the covariance matrix of the Gaussian function has the first principal component direction along the direction $(\mathbf{x}_i - \mathbf{x}_j)$ with variance $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ while all other principal component directions have the same variance σ^2. Let $\mathbf{\Sigma}_{ij}$ denote the covariance matrix that corresponds to the pattern pair $(\mathbf{x}_i, \mathbf{x}_j)$. The neighborhood function is defined as

$$\mathcal{N}_{ij}(\mathbf{x}_r) = \exp\left[-\frac{1}{2}(\mathbf{x}_r - \mathbf{x}_i)^T \mathbf{\Sigma}_{ij}^{-1} (\mathbf{x}_r - \mathbf{x}_i)\right]. \tag{6}$$

By eigendecomposition, we can express $\mathbf{\Sigma}_{ij}$ as

$$\mathbf{\Sigma}_{ij} = \mathbf{V} \mathbf{\Lambda}_{ij} \mathbf{V}^T,$$

where $\mathbf{\Lambda}_{ij} = \text{diag}(\|\mathbf{x}_i - \mathbf{x}_j\|^2, \sigma^2, \ldots, \sigma^2)$ is the diagonal matrix of eigenvalues of $\mathbf{\Sigma}_{ij}$ and $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_d]$ is the matrix of corresponding (orthonormal) eigenvectors normalized such that $\mathbf{v}_m^T \mathbf{v}_m = 1$ for all $1 \leq m \leq d$.[3] Note that $\mathbf{v}_1$ is a unit vector along the direction $\mathbf{x}_i - \mathbf{x}_j$, i.e.,

$$\mathbf{v}_1 = \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|}.$$

[3] More correctly, $\mathbf{V}$ and its eigenvectors should depend on $\mathbf{x}_i$ and $\mathbf{x}_j$. For notational simplicity, however, the subscripts showing the dependency are omitted. Also note that $\mathbf{V}^T = \mathbf{V}^{-1}$.

Since

$$\mathbf{\Sigma}_{ij}^{-1} = \mathbf{V}\mathbf{\Lambda}_{ij}^{-1}\mathbf{V}^T,$$

Equation (6) can be rewritten as

$$\begin{aligned}\mathcal{N}_{ij}(\mathbf{x}_r) &= \exp\left\{-\frac{1}{2}\left[\mathbf{V}^T(\mathbf{x}_r - \mathbf{x}_i)\right]^T \mathbf{\Lambda}_{ij}^{-1}\left[\mathbf{V}^T(\mathbf{x}_r - \mathbf{x}_i)\right]\right\} \\ &= \exp(-\frac{1}{2}D^2), \end{aligned} \tag{7}$$

where $\mathbf{\Lambda}_{ij}^{-1} = \text{diag}(\|\mathbf{x}_i - \mathbf{x}_j\|^{-2}, \sigma^{-2}, \ldots, \sigma^{-2})$ and D^2 denotes the squared Mahalanobis distance between $\mathbf{x}_r$ and $\mathbf{x}_i$. We simplify D^2 as follows:

$$\begin{aligned} D^2 &= \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|^2}\left\|\mathbf{v}_1^T(\mathbf{x}_r - \mathbf{x}_i)\right\|^2 + \frac{1}{\sigma^2}\sum_{m=2}^{d}\left\|\mathbf{v}_m^T(\mathbf{x}_r - \mathbf{x}_i)\right\|^2 \\ &= \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|^2}\left\|\mathbf{v}_1^T(\mathbf{x}_r - \mathbf{x}_i)\right\|^2 - \frac{1}{\sigma^2}\left\|\mathbf{v}_1^T(\mathbf{x}_r - \mathbf{x}_i)\right\|^2 + \frac{1}{\sigma^2}\sum_{m=1}^{d}\left\|\mathbf{v}_m^T(\mathbf{x}_r - \mathbf{x}_i)\right\|^2 \\ &= \frac{1}{\sigma^2}\|\mathbf{x}_r - \mathbf{x}_i\|^2 - \left(\frac{1}{\sigma^2} - \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|^2}\right)\left\|\mathbf{v}_1^T(\mathbf{x}_r - \mathbf{x}_i)\right\|^2. \end{aligned} \tag{8}$$

Hence, given $\mathbf{x}_i$, $\mathbf{x}_j$, $\mathbf{x}_r$, and σ, the neighborhood function value $\mathcal{N}_{ij}(\mathbf{x}_r)$ for pattern $\mathbf{x}_r$ can be computed by applying Equations (7) and (8).

The effect of dissimilar pattern pairs in $\mathcal{D}$ can also be modeled similarly. As in Equation (5), the update equation for any pattern $\mathbf{x}_s \in \mathcal{X}\backslash\{\mathbf{x}_k, \mathbf{x}_l\}$, as a result of the influence of pattern pair $(\mathbf{x}_k, \mathbf{x}_l) \in \mathcal{D}$, is given by

$$\Delta\mathbf{x}_s|_{(\mathbf{x}_k,\mathbf{x}_l)} = \frac{\eta\alpha}{\|\mathbf{x}_k - \mathbf{x}_l\|^3}\mathcal{N}_{kl}(\mathbf{x}_s)\mathbf{u}_{\mathbf{x}_s \leftarrow \mathbf{m}_{kl}}, \tag{9}$$

where $\mathcal{N}_{kl}(\cdot)$ is defined in the same way as $\mathcal{N}_{ij}(\cdot)$.

Location changes computed from this phase, together with location changes computed from the previous phase, will take effect all at once before moving on to the next iteration. [4]

[4] This is analogous to the Jacobi method, as opposed to the Gauss-Seidel method, for boundary value problems.

2.3 Annealing and Stopping Criteria

Based on update equations (3)–(5) and (9) in the two previous subsections, the locations of all patterns in the input space are adjusted iteratively. To ensure convergence, the learning rate parameter η should decrease monotonically with time. Moreover, the variances of the neighborhood functions should also decrease with time to gradually increase the specificity of the neighborhood functions. We apply a simple annealing procedure by using a fixed decay rate $\tau = 0.95$ for both η and σ_h^2 (for hyperspherical Gaussian) or $1/D^2$ (for full Gaussian).

The iterative metric adaptation procedure of RMA eventually has to stop. One possible stopping criterion can be defined based on the ratio λ of the average Euclidean distance over all pattern pairs in $\mathcal{S}$ to that over all pattern pairs in $\mathcal{D}$:

$$\lambda = \frac{|\mathcal{D}| \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} \|\mathbf{x}_i - \mathbf{x}_j\|}{|\mathcal{S}| \sum_{(\mathbf{x}_k,\mathbf{x}_l)\in\mathcal{D}} \|\mathbf{x}_k - \mathbf{x}_l\|}.$$

Apparently, a good metric should give a small value of λ. In general, λ decreases with time in the course of the metric adaptation process.

We summarize our RMA metric adaptation algorithm in Figure 1 below.

Input: $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, $\mathcal{S}_0$, $\mathcal{D}_0$
Begin
  Convert $\mathcal{S}_0$ and $\mathcal{D}_0$ into $\mathcal{S}$ and $\mathcal{D}$
  $t = 0$
  Repeat {
    For each $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}$, compute $\Delta\mathbf{x}_i|_{(\mathbf{x}_i,\mathbf{x}_j)}$ according to (3)
    For each $(\mathbf{x}_k, \mathbf{x}_l) \in \mathcal{D}$, compute $\Delta\mathbf{x}_k|_{(\mathbf{x}_k,\mathbf{x}_l)}$ according to (4)
    For each $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}$,
      For each $\mathbf{x}_r \in \mathcal{X}\backslash\{\mathbf{x}_i, \mathbf{x}_j\}$, compute $\Delta\mathbf{x}_r|_{(\mathbf{x}_i,\mathbf{x}_j)}$ according to (5)
    For each $(\mathbf{x}_k, \mathbf{x}_l) \in \mathcal{D}$,
      For each $\mathbf{x}_s \in \mathcal{X}\backslash\{\mathbf{x}_k, \mathbf{x}_l\}$, compute $\Delta\mathbf{x}_s|_{(\mathbf{x}_k,\mathbf{x}_l)}$ according to (9)
    Compute total location change $\Delta\mathbf{x}_i$ for each $\mathbf{x}_i \in \mathcal{X}$
    Update each $\mathbf{x}_i \in \mathcal{X}$ simultaneously
    $t = t + 1$
    $\lambda = \frac{|\mathcal{D}| \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} \|\mathbf{x}_i - \mathbf{x}_j\|}{|\mathcal{S}| \sum_{(\mathbf{x}_k,\mathbf{x}_l)\in\mathcal{D}} \|\mathbf{x}_k - \mathbf{x}_l\|}$
  } until λ is small enough
End

Fig. 1. Summary of RMA algorithm

3 Experiments on Semi-Supervised Clustering

To assess the effectiveness of RMA for clustering tasks, we perform extensive experiments on both simulated data and real-world data from the UCI Machine Learning Repository. [5]

3.1 Eight Clustering Algorithms

Similar to [20], we compare the clustering results based on k-means with and without metric learning. We also repeat the experiments using the constrained k-means algorithm [19]. More specifically, the following eight clustering algorithms are compared:

(1) k-means algorithm based on default Euclidean metric without using the constraints in $\mathcal{S}$ and $\mathcal{D}$
(2) constrained k-means algorithm based on default Euclidean metric subject to the constraints that patterns in a pair $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}$ are always assigned to the same cluster and patterns in a pair $(\mathbf{x}_k, \mathbf{x}_l) \in \mathcal{D}$ are always assigned to different clusters
(3) k-means algorithm + Xing *et al.*'s diagonal metric learning method
(4) constrained k-means algorithm + Xing *et al.*'s diagonal metric learning method
(5) k-means algorithm + Xing *et al.*'s full metric learning method
(6) constrained k-means algorithm + Xing *et al.*'s full metric learning method
(7) k-means algorithm + RMA
(8) constrained k-means algorithm + RMA

3.2 Performance Measures

The Rand index [26] is a clustering quality measure that measures the agreement of the clustering result with the ground truth. Let n_s be the number of pairs of patterns that are assigned to the same cluster (i.e., matched pairs) in both the resultant partition and the ground truth, and n_d be the number

[5] `http://www.ics.uci.edu/~mlearn/MLRepository.html`

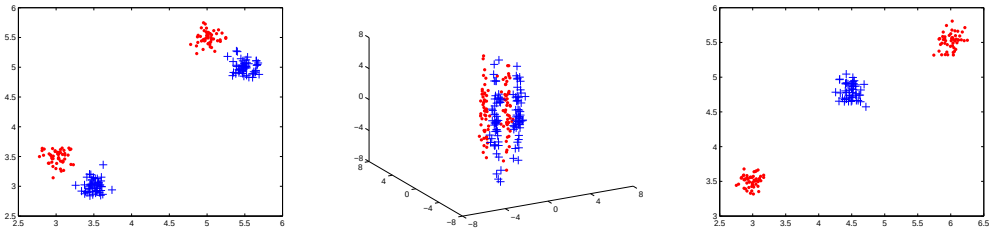
of pairs of patterns that are assigned to different clusters (i.e., mismatched pairs) in both the resultant partition and the ground truth. The Rand index is defined as the ratio of $(n_s + n_d)$ to the total number of pattern pairs, i.e., $n(n-1)/2$. When there are more than two clusters, however, the standard Rand index will favor assigning patterns to different clusters. We modify the Rand index as in [20] so that matched pairs and mismatched pairs are assigned weights to give them equal chance of occurrence (0.5).

Note that the result depends on the $\mathcal{S}$ and $\mathcal{D}$ sets. To see how different algorithms vary their performance with the background knowledge provided, we randomly generate different sets of $\mathcal{S}$ and $\mathcal{D}$ for our experiments. Specifically, we use 100 different $\mathcal{S}-\mathcal{D}$ sets for each data set. Moreover, since the result from k-means or constrained k-means can also vary slightly with random initialization, we compute the average Rand index over 20 random runs of (constrained) k-means for each $\mathcal{S}-\mathcal{D}$ pair. The results from all eight algorithms are then shown using box plots from S-PLUS.

3.3 Experiments on Simulated Data

We first perform some experiments on simulated data. We generate three simulated data sets as shown in Figure 2. The data points with the same color and type belong to the same class. The first two data sets, with 200 patterns (i.e., sample points) each (100 points for each class), are similar to those used by [20]. We also introduce a third set which better demonstrates the advantages of our method. There are 150 patterns in this data set, with 50 points for each of the three natural groupings (i.e., one class has 100 points while the other class has 50). For each data set, we randomly select 0.5% of the similar pairs to form $\mathcal{S}_0$. Similarly, we also randomly select 0.5% of the dissimilar pairs to form $\mathcal{D}_0$.

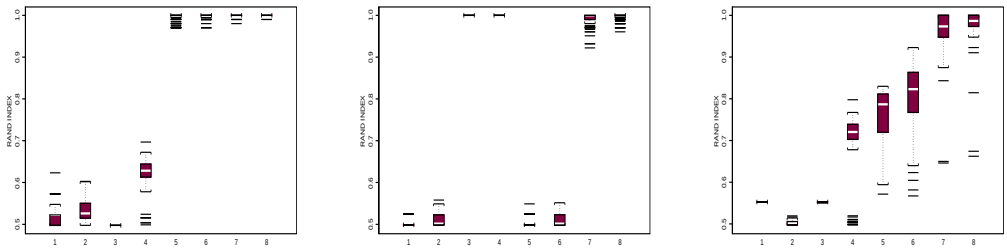
Figure 3 shows the clustering results of the eight algorithms for the three simulated data sets. Obviously, all the three data sets cannot be clustered well using the standard k-means algorithm. In fact, even the constrained k-means algorithm does not work well. For the first two data sets, good results can be obtained by using both RMA and Xing *et al.*'s diagonal or full metric learning method. For the third data set, RMA still gives fairly good results



(a) Data set 1 (b) Data set 2 (c) Data set 3

Fig. 2. Simulated data sets

but Xing *et al.*'s method cannot cluster the data well. It is easy to understand why Xing *et al.*'s method does not work well for this data set. Since both the lower-left cluster and the upper-right cluster belong to the same class but the middle one belongs to the other class, the adapted metric tries to scale the data globally so that patterns from the lower-left and upper-right clusters are projected linearly to lie close to each other. However, this projection based on a global metric makes the two classes overlap significantly with each other and hence the clustering results are not satisfactory.



(a) Data set 1 (b) Data set 2 (c) Data set 3

Fig. 3. Clustering results for simulated data sets shown as box plots for 100 different $\mathcal{S}-\mathcal{D}$ sets (the eight clustering algorithms are numbered as in Section 3.1)

3.4 Experiments on UCI Data

We further perform experiments on some UCI real-world data sets. For comparison, we use the same nine UCI benchmark data sets as in [20]. Each attribute is standardized by subtracting the mean from each value and dividing it by the corresponding standard deviation (i.e., standardized to zero mean and unit variance). Table 1 tabulates some characteristics of the nine data sets. The three columns show the number of patterns n, the number of

classes c, and the number of attributes d for each data set. The number of similar pairs $|\mathcal{S}_0|$ and the number of dissimilar pairs $|\mathcal{D}_0|$ randomly selected from each data set are shown in the subfigure captions of Figure 4.

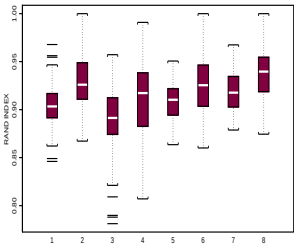
Table 1
Nine UCI data sets used in our experiments

Data set	n	c	d
Soybean	47	4	35
Protein	116	6	20
Iris plants	150	3	4
Wine	178	3	13
Ionosphere	351	2	34
Boston housing	506	3	13
Breast cancer	569	2	31
Balance	625	3	4
Diabetes	768	2	8

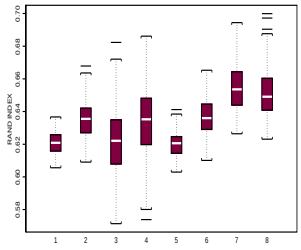
The clustering results of the eight algorithms are shown in Figure 4. As we can see, RMA gives significant improvement in clustering performance over the k-means and constrained k-means algorithms for all data sets. To compare our method with that of Xing *et al.*, we find that RMA is clearly better than Xing *et al.*'s method for six out of the nine data sets (Protein, Ionosphere, Boston housing, Breast cancer, Balance, and Diabetes), while RMA has performance in between the diagonal and full metric learning methods of Xing *et al.* for the Iris plants data set.

3.5 Significance Tests

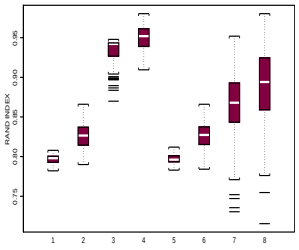
For data sets in which both RMA and Xing *et al.*'s method give comparable results, we would like to compare them more carefully under the hypothesis testing framework. Specifically, we would like to perform two-side paired t-test on the clustering results for the first two simulated data sets and the Soybean and Wine UCI data sets.



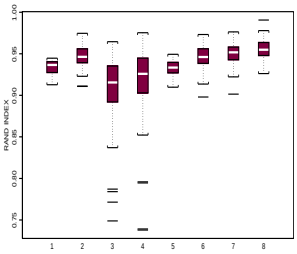
(a) Soybean $|\mathcal{S}_0| = 5$, $|\mathcal{D}_0| = 16$



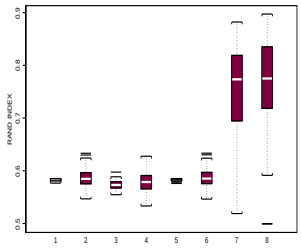
(b) Protein $|\mathcal{S}_0| = 11$, $|\mathcal{D}_0| = 55$



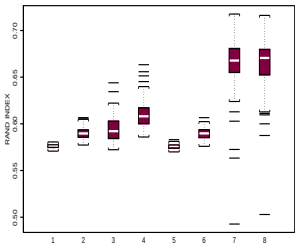
(c) Iris plants $|\mathcal{S}_0| = 18$, $|\mathcal{D}_0| = 37$



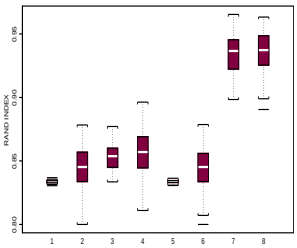
(d) Wine $|\mathcal{S}_0| = 10$, $|\mathcal{D}_0| = 20$



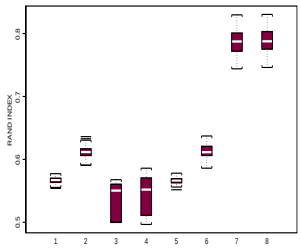
(e) Ionosphere $|\mathcal{S}_0| = 33$, $|\mathcal{D}_0| = 28$



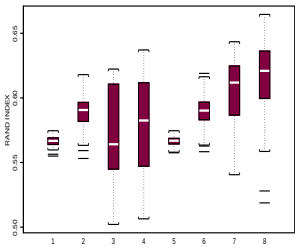
(f) Boston housing $|\mathcal{S}_0| = 53$, $|\mathcal{D}_0| = 74$



(g) Breast cancer $|\mathcal{S}_0| = 85$, $|\mathcal{D}_0| = 75$



(h) Balance $|\mathcal{S}_0| = 41$, $|\mathcal{D}_0| = 55$



(i) Diabetes $|\mathcal{S}_0| = 80$, $|\mathcal{D}_0| = 67$

Fig. 4. Clustering results for UCI data sets shown as box plots for 100 different $\mathcal{S}-\mathcal{D}$ sets (the eight clustering algorithms are numbered as in Section 3.1)

Table 2 shows the paired t-test statistics for the four data sets. Each sample X (or Y) in the pair (X, Y) is a 100-element vector of Rand index values for 100 different $\mathcal{S}-\mathcal{D}$ sets obtained using algorithm X (or Y), where the algorithms are numbered as in Section 3.1. R_X (or R_Y) refers to the Rand index value obtained using algorithm X (or Y). The last column concludes which algorithm is better, with the symbols $<$ and $>$ denoting 'is worse than' and 'is better than', respectively. From the results of paired t-test with significance level 0.05, we can conclude that RMA is better than Xing *et al.*'s full metric

learning method for the first simulated data set. For the second simulated data set, Xing *et al.*'s diagonal metric learning method is better than ours. As for the two UCI data sets, the clustering performance of RMA is the best.

Table 2
Paired t-test on the clustering results for four data sets

Data set	Paired sample (X, Y)	Mean of $R_X - R_Y$	t	p value	Remark
Simulated 1	$(6, 8)$	-1.944×10^{-3}	-3.4757	0.0008	$X < Y$
Simulated 2	$(4, 8)$	4.077×10^{-3}	4.6887	0	$X > Y$
Soybean	$(2, 8)$	-1.033×10^{-2}	-3.3116	0.0013	$X < Y$
	$(4, 8)$	-2.798×10^{-2}	-6.4397	0	$X < Y$
	$(6, 8)$	-1.350×10^{-2}	-4.2601	0	$X < Y$
Wine	$(2, 8)$	-8.893×10^{-3}	-5.8209	0	$X < Y$
	$(4, 8)$	-3.679×10^{-2}	-8.6770	0	$X < Y$
	$(6, 8)$	-8.748×10^{-3}	-5.5408	0	$X < Y$

To summarize, RMA outperforms the other methods for all data sets except the second simulated data set and the Iris plants data set.

3.6 Efficiency

For each iteration of the RMA algorithm as summarized in Figure 1, we have to compute the neighborhood function $\mathcal{N}_{ij}(\mathbf{x}_r)$ for each pattern $\mathbf{x}_r, r = 1, \ldots, n$ for each similar or dissimilar pattern pair $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S} \bigcup \mathcal{D}$, and then decide the location change for each pattern. Therefore, the time complexity for each iteration of the RMA algorithm is $\mathcal{O}(n(|\mathcal{S}| + |\mathcal{D}|))$. Since the number of pairwise constraints $|\mathcal{S}| + |\mathcal{D}|$ is usually much smaller than the total number of patterns n, our method is more efficient than Xing *et al.*'s method which has an $\mathcal{O}(n^2)$ time complexity. In our experiments, we have noticed that the diagonal metric learning method can take much longer time than the full metric learning method for some data sets. In general, the speed of RMA is less sensitive to the data sets considered.

4 Experiments on Content-Based Image Retrieval

4.1 Content-Based Image Retrieval

With the emergence and increased popularity of the World Wide Web (WWW) over the past decade, retrieval of images based on content, often referred to as *content-based image retrieval* (CBIR), has gained a lot of research interests [27]. The two determining factors for image retrieval performance are the features used to represent the images and the distance function used to measure the similarity between a query image and the images in the database. For a specific feature representation chosen, the retrieval performance depends critically on the similarity measure used. Instead of choosing a distance function in advance, a more promising approach is to learn a good distance function from data automatically. Recently, this challenging new direction has aroused great interest in the research community. In particular, RCA [21,28] has been used to improve image retrieval performance in CBIR tasks. More recently, another method called DistBoost [22,23] was demonstrated to give even better image retrieval results.

In this section, we will apply RMA to improve the retrieval performance of CBIR tasks. We will also compare the retrieval performance of this method with other distance learning methods.

4.2 Image Databases and Feature Representation

Our image retrieval experiments are based on two image databases. One database is a subset of the Corel Photo Gallery, which contains 1010 images belonging to 10 different classes. The 10 classes include bear (122), butterfly (109), cactus (58), dog (101), eagle (116), elephant (105), horse (110), penguin (76), rose (98), and tiger (115). Another database contains 546 images belonging to 10 classes that we downloaded from the Internet. The image classes are manually defined based on high-level semantics. Compared with the first database, the class sizes of this database have a much wider range of variations from the smallest class with 24 images to the largest class with 125 images.

We first represent the images in the HSV color space, and then compute the *color coherence vector* (CCV) [29] as the feature vector for each image. Specifically, we quantize each image to $8 \times 8 \times 8$ color bins, and then represent the image as a 1024-dimensional CCV $(\alpha_1, \beta_1, \ldots, \alpha_{512}, \beta_{512})^T$, with α_i and β_i representing the numbers of coherent and non-coherent pixels, respectively, in the ith color bin. The CCV representation gives finer distinctions than the use of color histograms. Thus it usually gives better image retrieval results. For computational efficiency, we first apply principal component analysis (PCA) to retain the 60 dominating principal components before applying RMA as described in the previous section.

4.3 Comparative Study and Performance Measures

We compare the image retrieval performance of RMA with the baseline method of using Euclidean distance without distance learning, as well as some other distance learning methods. Besides Xing *et al.*'s methods, we also include distance learning methods with RCA and DistBoost in our comparative study. [6] RCA makes use of the pairwise similarity constraints to learn a Mahalanobis distance, which essentially assigns large weights to relevant components and low weights to irrelevant components with relevance estimated based on the connected components composed of similar patterns. DistBoost, as discussed in Section 4.1, is a nonmetric distance learning method that makes use of the pairwise constraints and performs boosting. Since both RCA and DistBoost make use of similarity constraints only, for fair comparison, we do not incorporate dissimilarity information in RMA in our image retrieval experiments. As a consequence, we have to modify the stopping criterion slightly by defining λ as the average Euclidean distance over all pattern pairs in $\mathcal{S}$ to all pattern pairs instead of pattern pairs in $\mathcal{D}$ only.

We use two performance measures in our comparative study. The first one, based on *precision* and *recall*, is commonly used in information retrieval. The second one, used in [22,23], is based on *cumulative neighbor purity* curves. Cumulative neighbor purity measures the percentage of correctly retrieved

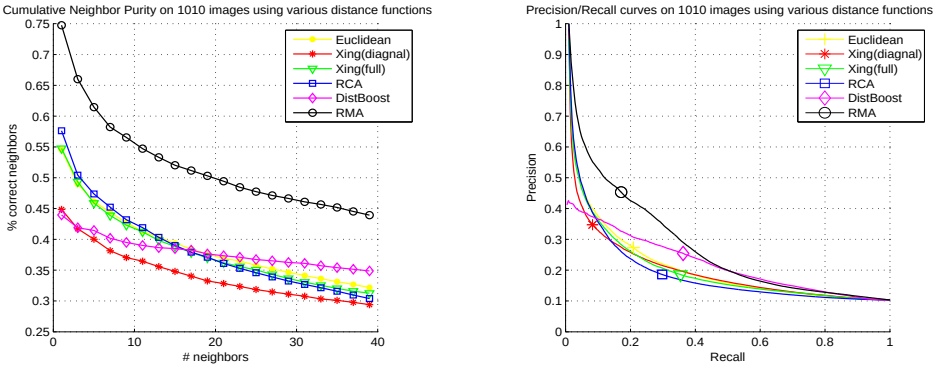
[6] The MATLAB code for RCA and DistBoost was obtained from the authors of [21,23,28].

images in the k nearest neighbors of the query image, averaged over all queries, with k up to some value K ($K = 20$ or 40 in our experiments).

For each retrieval task, we compute the average performance statistics over 5 randomly generated sets of similar image pairs. The number of similar image pairs is set to 150, which is about 0.3% and 0.7% of the total number of possible image pairs in the first and second databases, respectively. For each set of similar image pairs, we set the number of boosting iterations in DistBoost to 50.

4.4 Experimental Results

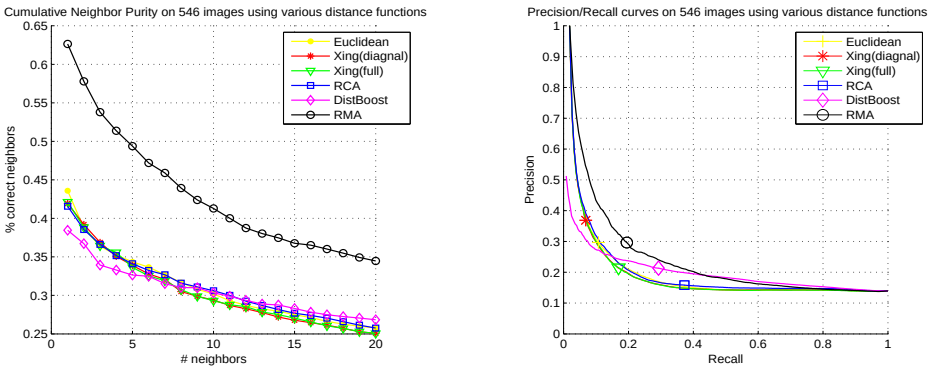
Figure 5 shows the retrieval results on the first image database based on both cumulative neighbor purity and precision/recall. We can see that metric learning with RMA significantly improves the retrieval performance and outperforms other distance learning methods especially with respect to the cumulative neighbor purity measure. The retrieval results on the second image database are shown in Figure 6. Note that this database is highly unbalanced as the class sizes vary significantly. For this database, Xing *et al.*'s methods, RCA and DistBoost cannot improve the retrieval performance. On the other hand, RMA significantly outperforms the other methods in improving the retrieval performance.



(a) (b)

Fig. 5. Retrieval results on the first image database (1010 images, 10 classes). (a) cumulative neighbor purity curves; (b) precision/recall curves.

Some typical retrieval results on the first and second databases are shown in Figure 7(a) and (b), respectively. For each query image, we show the retrieved



(a) (b)

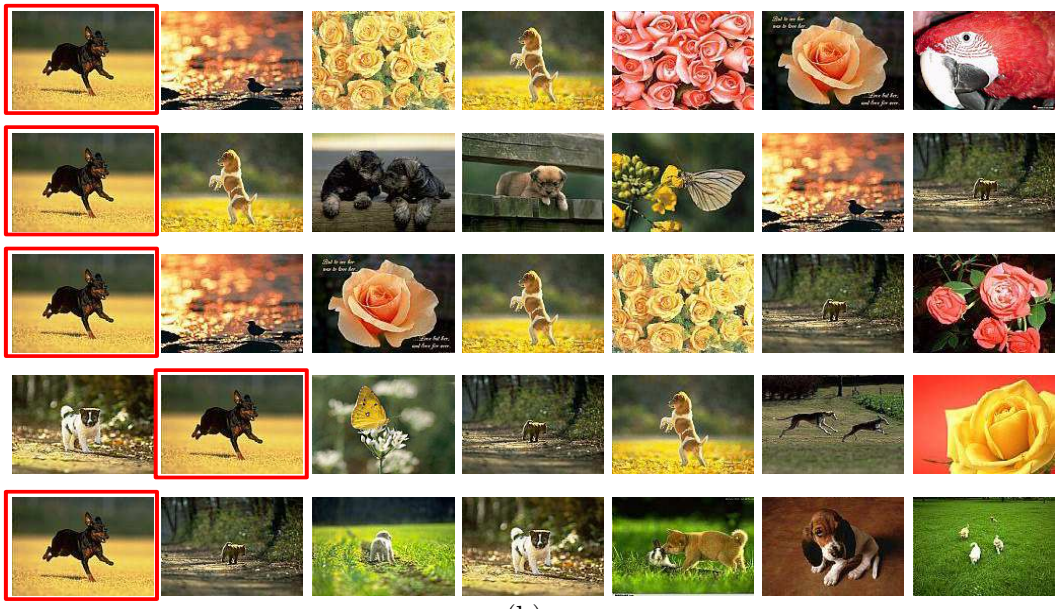
Fig. 6. Retrieval results on the second image database (546 images, 10 classes). (a) cumulative neighbor purity curves; (b) precision/recall curves.

images in five rows, corresponding, from top to bottom, to the use of Euclidean distance without distance learning and distance learning with Xing *et al.*'s full metric learning method, RCA, DistBoost and RMA. Each row shows the 7 nearest neighbors of the query image with respect to the distance used, with dissimilarity based on the distance increasing from left to right. The query image is shown with a frame around it. Note that the query image may not be the nearest neighbor using the DistBoost method since it learns nonmetric distance functions which, among other things, may not satisfy $d(\mathbf{x}, \mathbf{x}) = 0$ and the triangle inequality condition. While Euclidean distance without distance learning tends to retrieve images based mostly on the color coherence features, the distance learning methods can retrieve images with different background colors, as shown in Figure 7(b). This is in fact the main motivation for developing distance learning methods so that the retrieval results correspond better to human expectation in the semantic sense. It can be seen that among the distance learning methods studied, RMA slightly outperforms the other methods.

4.5 Relevance Feedback vs. Pairwise Constraints

Relevance feedback has been used in the traditional information retrieval community to improve the performance of information retrieval systems based on user feedback. This interactive approach has also emerged as a popular approach in CBIR [30]. The user is provided with the option of labeling (some of the) previously retrieved images as either relevant or irrelevant. Based on

(a)



(b)

Fig. 7. Typical retrieval results on the two databases ((a) and (b)) based on Euclidean distance (first row), Xing *et al.*'s full metric learning (second row), RCA (third row), DistBoost (four row) and RMA (fifth row). Each row shows the 7 nearest neighbors including the query image (framed).

this feedback information, the CBIR system can iteratively refine the retrieval results by learning a more appropriate (dis)similarity measure.

Relevance feedback may also be used to obtain the pairwise constraints. The pairwise similarity constraints used by RMA can make better use of the relevance feedback from users, not only from one specific query but also from all previous ones. Specifically, similarity constraints can be obtained from the relevance feedback, with each relevant image and the query image forming a similar image pair. The set of similar image pairs (or pairwise similarity constraints) is incrementally built up as relevance feedback is collected from users. Thus, later retrieval tasks can make use of an increasing set of similar image pairs for metric learning. To verify whether increasing the number of pairwise similarity constraints can improve the retrieval performance, we further perform some experiments on a smaller image database containing 448 images from four classes. Figure 8 shows the results in terms of precision/recall curves for different numbers of pairwise similarity constraints. It is clear that the performance increases with the number of similar image pairs. Thus, RMA can be used for long-term learning to enhance the CBIR performance by accumulating relevance feedback from all previous query sessions. We can also adapt the distance metric when new images are added to the image database. The metric adaptation process will be repeated whenever new queries are put forward.

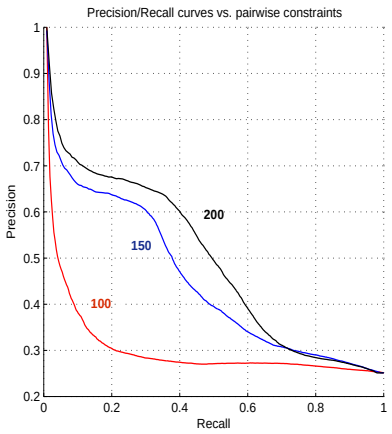


Fig. 8. Precision/recall curves for different numbers of pairwise similarity constraints, ranging from 100 to 200.

5 Related Previous Work

The nonparametric RMA method proposed in this paper for implicit distance metric adaptation has been inspired by previous work in several lines of research, although the problems addressed by them are different from what we intend to solve here.

Deformable shape models [31–34] take a model-based approach to object recognition by varying the shape of a model for representing an object while limiting the degree of model deformation. Our criterion-based optimization method for adjusting the locations of patterns in the input space resembles the energy minimization approach adopted by many deformable models.

We also note that RMA is somewhat related to the nonparametric clustering algorithm based on the so-called 'friends' and 'non-friends' [35], i.e., the proximity relationships between a point and all other points. Unlike the iterative transformation of the proximity matrix in their method, we formulate metric adaptation explicitly in RMA under an optimization framework. Also, RMA is not limited to finding two clusters at a time.

Self-organizing maps (SOM) [36] and their probabilistic extension called generative topographic mapping (GTM) [37] attempt to find a low-dimensional embedding of some data manifold in a high-dimensional space while preserving topographic relationships. The notion of soft neighborhood has inspired us to use Gaussian neighborhood functions to achieve global, long-range metric adaptation effects. A more recent method called stochastic neighbor embedding (SNE) [38] generalizes SOM and GTM by allowing many-to-one mappings in the embedding. Similar to our gradient-descent method, SNE also employs a gradient-based approach for adjusting the locations of points in the low-dimensional embedding space.

RMA also bears similarities with the embedding problem in general, which is of central importance to such techniques as multidimensional scaling (MDS) [39]. As is almost always the case, MDS attempts to embed patterns from one (possibly non-metric, usually high-dimensional) space into another (metric, usually low-dimensional) embedding space. In our case, however, we do not make any difference between the two spaces. Instead, we (implicitly) modify

the metric directly in the input space so that similar patterns tend to get closer and dissimilar patterns tend to move away from each other after metric adaptation. Since the original input space is metric to begin with, the adaptation procedure does not change the metric nature of the space. Hence, we avoid the problem of having to ensure that the embedding space is metric.

6 Concluding Remarks

We have proposed a new metric adaptation method called RMA for improving clustering results using pairwise (dis)similarity information. Since RMA is nonparametric in nature, it can adapt to different local metrics at different locations of the input space. While RMA is local in this sense, it also possesses some global properties in that patterns can induce long-range effects on other patterns through a Gaussian neighborhood.

Note that the original input space and the transformed space after metric adaptation are not explicitly related via some transformation mapping. Thus it is not straightforward for new points added to the original input space to be projected onto the transformed space, although typically clustering tasks do not have to address this problem.

RMA does not attempt to preserve the topological structure of the patterns during the metric adaptation process. We are currently trying to extend it by preserving the topological relationships explicitly. With this new property introduced, it may be possible to extend RMA to visualization and other machine learning tasks.

Acknowledgments

The research described in this paper has been supported by two grants, CA03/04.EG01 (which is part of HKBU2/03/C) and HKUST6174/04E, from the Research Grants Council of the Hong Kong Special Administrative Region, China.

References

[1] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley, New York, NY, USA, 2nd edition, 2001.

[2] K. Fukunaga and L. Hostetler. Optimization of k-nearest neighbor density estimates. *IEEE Transactions on Information Theory*, 19(3):320–326, 1973.

[3] R.D. Short and K. Fukunaga. The optimal distance measure for nearest neighbor classification. *IEEE Transactions on Information Theory*, 27(5):622–627, 1981.

[4] K. Fukunaga and T.E. Flick. An optimal global nearest neighbor metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(3):314–318, 1984.

[5] J.H. Friedman. Flexible metric nearest neighbor classification. Technical report, Department of Statistics, Stanford University, Stanford, CA, USA, November 1994.

[6] D.G. Lowe. Similarity metric learning for a variable-kernel classifier. *Neural Computation*, 7(1):72–85, 1995.

[7] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):607–616, 1996.

[8] C. Domeniconi, J. Peng, and D. Gunopulos. An adaptive metric machine for pattern classification. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 458–464. MIT Press, Cambridge, MA, USA, 2001.

[9] C. Domeniconi and D. Gunopulos. Adaptive nearest neighbor classification using support vector machines. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 665–672. MIT Press, Cambridge, MA, USA, 2002.

[10] C. Domeniconi, J. Peng, and D. Gunopulos. Locally adaptive metric nearest-neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1281–1285, 2002.

[11] J. Peng, D.R. Heisterkamp, and H.K. Dai. Adaptive kernel metric nearest neighbor classification. In *Proceedings of the Sixteenth International Conference on Pattern Recognition*, volume 3, pages 33–36, Québec City, Québec, Canada, 11–15 August 2002.

[12] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[13] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 92–100, Madison, WI, USA, 24–26 July 1998.

[14] S. Kaski, J. Sinkkonen, and J. Peltonen. Bankruptcy analysis with self-organizing maps in learning metrics. *IEEE Transactions on Neural Networks*, 12(4):936–947, 2001.

[15] J. Sinkkonen and S. Kaski. Clustering based on conditional distributions in an auxiliary space. *Neural Computation*, 14(1):217–239, 2002.

[16] Z. Zhang, J.T. Kwok, and D.Y. Yeung. Parametric distance metric learning with label information. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 1450–1452, Acapulco, Mexico, 9–15 August 2003.

[17] Z. Zhang, J.T. Kwok, and D.Y. Yeung. Parametric distance metric learning with label information. Technical Report HKUST-CS03-02, Hong Kong University of Science and Technology, Department of Computer Science, Clear Water Bay, Kowloon, Hong Kong, January 2003. `ftp://ftp.cs.ust.hk/pub/techreport/03/tr03-02.ps.gz`.

[18] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1103–1110, Stanford, CA, USA, 29 June – 2 July 2000.

[19] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, Williamstown, MA, USA, 28 June – 1 July 2001.

[20] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 505–512. MIT Press, Cambridge, MA, USA, 2003.

[21] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 11–18, Washington, DC, USA, 21–24 August 2003.

[22] T. Hertz, A. Bar-Hillel, and D. Weinshall. Boosting margin based distance functions for clustering. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 393–400, Banff, Alberta, Canada, 4–8 July 2004.

[23] T. Hertz, A. Bar-Hillel, and D. Weinshall. Learning distance functions for image retrieval. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 570–577, Washington DC, USA, 27 June–3 July 2004.

[24] Z. Zhang. Learning metrics via discriminant kernels and multidimensional scaling: toward expected Euclidean representation. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 872–879, Washington, DC, USA, 21–24 August 2003.

[25] J.T. Kwok and I.W. Tsang. Learning with idealized kernels. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 400–407, Washington, DC, USA, 21–24 August 2003.

[26] W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.

[27] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[28] T. Hertz, N. Shental, A. Bar-Hillel, and D. Weinshall. Enhancing image and video retrieval: learning via equivalence constraints. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 668–674, Madison, WI, USA, 18–20 June 2003.

[29] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *ACM Multimedia*, pages 65–73, 1996.

[30] Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: a powerful tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):644–655, 1998.

[31] M. Revow, C.K.I. Williams, and G.E. Hinton. Using generative models for handwritten digit recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):592–606, 1996.

[32] K.W. Cheung, D.Y. Yeung, and R.T. Chin. A Bayesian framework for deformable pattern recognition with application to handwritten character

recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1382–1388, 1998.

[33] K.W. Cheung, D.Y. Yeung, and R.T. Chin. Bidirectional deformable matching with application to handwritten character extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1133–1139, 2002.

[34] K.W. Cheung, D.Y. Yeung, and R.T. Chin. On deformable models for visual pattern recognition. *Pattern Recognition*, 35(7):1507–1526, 2002.

[35] S. Dubnov, R. El-Yaniv, Y. Gdalyahu, E. Schneidman, N. Tishby, and G. Yona. A new nonparametric pairwise clustering algorithm based on iterative estimation of distance profiles. *Machine Learning*, 47(1):35–61, 2002.

[36] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, Germany, 3rd edition, 1989.

[37] C.M. Bishop, M. Svensén, and C.K.I. Williams. GTM: the generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.

[38] G. Hinton and S. Roweis. Stochastic neighbor embedding. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 833–840. MIT Press, Cambridge, MA, USA, 2003.

[39] T.F. Cox and M.A.A. Cox. *Multidimensional Scaling*. CRC/Chapman and Hall, 2nd edition, 2001.

About the Author – HONG CHANG received her Bachelor degree and M.Phil. degree in Computer Science from Hebei University of Technology and Tianjin University, China, respectively. She is currently a PhD student at the Department of Computer Science of the Hong Kong University of Science and Technology. Her main research interests include semi-supervised learning, nonlinear dimensionality reduction and related applications.

About the Author – DIT-YAN YEUNG received his B.Eng. degree in Electrical Engineering and M.Phil. degree in Computer Science from the University of Hong Kong, and his Ph.D. degree in Computer Science from the University of Southern California in Los Angeles. He was an Assistant Professor at the Illinois Institute of Technology in Chicago before he joined the Department of Computer Science of the Hong Kong University of Science and Technology, where he is currently an Associate Professor. His current research interests are in machine learning and pattern recognition.

About the Author – WILLIAM K. CHEUNG received his B.Sc. and M.Phil. degrees in Electronic Engineering from the Chinese University of Hong Kong, and his Ph.D. degree in Computer Science from the Hong Kong University of Science and Technology. He is currently an Assistant Professor at the Department of Computer Science of the Hong Kong Baptist University. His current research interests include pattern recognition, machine learning and artificial intelligence with applications to web mining, information extraction, recommender systems and web/grid service management.