Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Pattern Recognition ▮▮▮ (▮▮▮▮) ▮▮▮–▮▮▮

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Locally linear metric adaptation with application to semi-supervised clustering and image retrieval

Hong Chang, Dit-Yan Yeung*

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

Received 28 July 2005; received in revised form 8 December 2005; accepted 18 December 2005

Abstract

Many computer vision and pattern recognition algorithms are very sensitive to the choice of an appropriate distance metric. Some recent research sought to address a variant of the conventional clustering problem called *semi-supervised clustering*, which performs clustering in the presence of some background knowledge or supervisory information expressed as pairwise similarity or dissimilarity constraints. However, existing metric learning methods for semi-supervised clustering mostly perform global metric learning through a linear transformation. In this paper, we propose a new metric learning method that performs nonlinear transformation globally but linear transformation locally. In particular, we formulate the learning problem as an optimization problem and present three methods for solving it. Through some toy data sets, we show empirically that our *locally linear metric adaptation* (LLMA) method can handle some difficult cases that cannot be handled satisfactorily by previous methods. We also demonstrate the effectiveness of our method on some UCI data sets. Besides applying LLMA to semi-supervised clustering, we have also used it to improve the performance of content-based image retrieval systems through metric learning. Experimental results based on two real-world image databases show that LLMA significantly outperforms other methods in boosting the image retrieval performance.

Keywords: Metric learning; Locally linear metric adaptation; Linear transformation; Semi-supervised clustering; Gradient method; Iterative majorization; Spectral method; UCI repository; Content-based image retrieval

1. Introduction

Many computer vision and pattern recognition algorithms rely on a distance metric. Some commonly used methods are nearest neighbor classifiers, radial basis function networks and support vector machines for classification (or supervised learning) tasks and the k-means algorithm for clustering (or unsupervised learning) tasks. The performance of these methods often depends critically on the choice of an appropriate metric. Instead of choosing the metric manually, a promising approach is to learn the metric from data automatically. This idea can be dated back to some early work on optimizing the metric for k-nearest neighbor density estimation [1]. Later, optimal local metric [2] and optimal global metric [3] were also developed for nearest neighbor classification. More recent research along this line continued to develop various locally adaptive metrics for nearest neighbor classifiers, e.g., Refs. [4–8]. 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 [9].

While class label information is available for metric learning in classification tasks, such information is generally unavailable in conventional clustering tasks. To adapt the metric appropriately to improve the clustering results, some additional background knowledge or supervisory information should be made available. This learning paradigm between the supervised and unsupervised learning extremes is referred to as *semi-supervised clustering*, as contrasted to another type of semi-supervised learning tasks called *semi-supervised classification*, which solves the classification problem with the aid of additional unlabeled data.

[*] Corresponding author. Tel.: +852 2358 6977; fax: +852 2358 1477.
E-mail address: dyyeung@cs.ust.hk (D.-Y. Yeung).

0031-3203/$30.00 © 2006 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2005.12.012

One type of supervisory information is in the form of limited labeled data.[1] The set of labeled examples is typically very small compared with the set of unlabeled examples. Based on such information, Sinkkonen and Kaski [10] proposed a local metric learning method to improve clustering and visualization results. Basu et al. [11] explored using labeled data to generate initial seed clusters for the k-means clustering algorithm. Also, Zhang et al. [12] proposed a parametric distance metric learning method for both classification and clustering tasks.

Another type of supervisory information is in the form of pairwise similarity or dissimilarity constraints. This type of supervisory information is weaker than the first type, in that pairwise constraints can be derived from labeled data but not vice versa. Wagstaff and Cardie [13] and Wagstaff et al. [14] proposed using such pairwise constraints to improve clustering results. Klein and Kamvar [15] introduced spatial generalizations to pairwise constraints, so that the pairwise constraints can also have influence on the neighboring data points. However, both methods do not incorporate metric learning into the clustering algorithms. Xing et al. [16] proposed using pairwise side information in a novel way to learn a global Mahalanobis metric before performing clustering with constraints. Both Klein et al.'s and Xing et al.'s methods generally outperform Wagstaff et al.'s method in the experiments reported. Instead of using an iterative algorithm as in Ref. [16], Bar-Hillel et al. [17] devised a more efficient, non-iterative algorithm called relevant component analysis (RCA) for learning a global Mahalanobis metric. However, their method can only incorporate similarity constraints. Shental et al. [18] extended the work of Bar-Hillel et al. [17] by incorporating both pairwise similarity and dissimilarity constraints into the expectation-maximization (EM) algorithm for model-based clustering based on Gaussian mixture models. Kwok and Tsang [19] established the relationship between metric learning and kernel matrix adaptation.

To summarize, we can categorize metric learning 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 under the semi-supervised learning paradigm. Supervisory information 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 semi-supervised metric learning method with pairwise similarity side information. While our method is local in the sense that it performs metric learning through locally linear transformation, it also achieves global consistency through interaction between adjacent local neighborhoods.

The rest of this paper is organized as follows. In Section 2, we present our metric learning method based on locally linear transformation. We also formulate the learning problem as an optimization problem and present two methods for solving it. A more efficient optimization method based on the spectral approach is then proposed in Section 3. Section 4 presents some experimental results on semi-supervised clustering, comparing our method with some previous methods. We then apply our metric learning method to content-based image retrieval in Section 5. Finally, some concluding remarks are given in the last section.

2. Locally linear metric adaptation

2.1. Basic ideas

Let us denote a set of n data points in a d-dimensional input space by $\mathscr{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$. As in Ref. [17], we only consider pairwise similarity constraints which are given in the form of a set $\mathscr{S}$ of similar point pairs. Intuitively, we want to transform the n data points to a new space in which the points in each similar pair will get closer to each other. To preserve the topological relationships between data points, we move not only the points involved in the similar pairs but also other points. For computational efficiency, we resort to linear transformation. One promising approach is to apply locally linear transformation so that the overall transformation of all points in $\mathscr{X}$ is linear locally but nonlinear globally, generalizing previous metric learning methods based on applying linear transformation globally [16,17]. We call this new metric learning method *locally linear metric adaptation* (LLMA). However, caution should be taken when applying linear transformation to reduce the distance between similar points, as a degenerate transformation will simply map all points to the same location so that all inter-point distances vanish (and hence become the smallest possible). Obviously, this degenerate case is undesirable and should be avoided.

2.2. Metric adaptation as an optimization problem

We now proceed to devise the metric learning algorithm more formally. For each point $\mathbf{x}_r$ involved in some similar point pair, say $(\mathbf{x}_r, \mathbf{x}_s)$, we apply a linear transformation to the vector $(\mathbf{x}_s - \mathbf{x}_r)$ to give $\mathbf{A}_r(\mathbf{x}_s - \mathbf{x}_r) + \mathbf{c}_r$ for some $d \times d$ matrix $\mathbf{A}_r$ and d-dimensional vector $\mathbf{c}_r$. The same linear transformation is also applied to every data point $\mathbf{x}_i$ in the neighborhood set $\mathscr{N}_r$ of $\mathbf{x}_r$. In other words, every data point $\mathbf{x}_i \in \mathscr{N}_r$ is transformed to

$$\begin{aligned}\mathbf{y}_i &= \mathbf{A}_r(\mathbf{x}_i - \mathbf{x}_r) + \mathbf{c}_r + \mathbf{x}_r \\ &= \mathbf{x}_i + (\mathbf{A}_r - \mathbf{I})\mathbf{x}_i + (\mathbf{I} - \mathbf{A}_r)\mathbf{x}_r + \mathbf{c}_r \\ &= \mathbf{x}_i + (\mathbf{A}_r - \mathbf{I})\mathbf{x}_i + \mathbf{b}_r,\end{aligned}$$

[1] Semi-supervised clustering with the aid of labeled data is essentially the same as semi-supervised classification with the aid of unlabeled data.

where $\mathbf{b}_r = (\mathbf{I} - \mathbf{A}_r)\mathbf{x}_r + \mathbf{c}_r$ is the translation vector for all points $\mathbf{x}_i$'s in $\mathcal{N}_r$.

However, a data point $\mathbf{x}_i$ may belong to multiple neighborhood sets corresponding to different points involved in $\mathcal{S}$. Thus, the new location $\mathbf{y}_i$ of $\mathbf{x}_i$ is the overall transformation effected by possibly all points involved in all similar pairs (and hence neighborhood sets):

$$\mathbf{y}_i = \mathbf{x}_i + \sum_{\mathbf{x}_r:(\mathbf{x}_r,\cdot)\vee(\cdot,\mathbf{x}_r)\in\mathcal{S}} \pi_{ri}[(\mathbf{A}_r - \mathbf{I})\mathbf{x}_i + \mathbf{b}_r],$$

where $\pi_{ri} = 1$ if $\mathbf{x}_i \in \mathcal{N}_r$ and 0 otherwise.

Let m denote the number of unique points involved in $\mathcal{S}$. Thus, a total of m different transformations have to be estimated from the point pairs in $\mathcal{S}$, requiring $O(md^2)$ transformation parameters in $\{\mathbf{A}_r\}$ and $\{\mathbf{b}_r\}$. When m is small compared with the dimensionality d, we cannot estimate the $O(md^2)$ transformation parameters accurately. One way to get around this problem is to focus on a more restrictive set of linear transformations. The simplest case is to allow only translation, which can be described by md parameters. Obviously, translating all data points in a neighborhood set by the same amount leads to no change in the inter-point distances. Although some data points may fall into multiple neighborhood sets and hence this phenomenon does not hold, we want to incorporate an extra degree of freedom by changing the neighborhood sets to Gaussian neighborhood functions. More specifically, we set $\mathbf{A}_r$ to the identity matrix $\mathbf{I}$ and express the new location $\mathbf{y}_i$ of $\mathbf{x}_i$ as

$$\mathbf{y}_i = \mathbf{x}_i + \sum_{\mathbf{x}_r:(\mathbf{x}_r,\cdot)\vee(\cdot,\mathbf{x}_r)\in\mathcal{S}} \pi_{r_i}\mathbf{b}_r, \tag{1}$$

where π_{ri} is a Gaussian function defined as

$$\pi_{ri} = \exp[-\tfrac{1}{2}(\mathbf{x}_i - \mathbf{x}_r)^{\mathrm{T}}\boldsymbol{\Sigma}_r^{-1}(\mathbf{x}_i - \mathbf{x}_r)],$$

with $\boldsymbol{\Sigma}_r$ being the covariance matrix. For simplicity, we use a hyperspherical Gaussian function, meaning that the covariance matrix is diagonal with all diagonal entries being ω^2. Thus π_{ri} can be rewritten as $\pi_{ri} = \exp(-\|\mathbf{x}_i - \mathbf{x}_r\|^2/(2\omega^2))$. Note that (1) can be expressed as

$$\mathbf{y}_i = \mathbf{x}_i + \mathbf{B}\pi_i, \tag{2}$$

where $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m]$ is a $d \times m$ matrix and $\boldsymbol{\pi}_i = (\pi_{1i}, \pi_{2i}, \ldots, \pi_{mi})^{\mathrm{T}}$ is an m-dimensional column vector. For data points that are far away from all points involved in $\mathcal{S}$ (and hence the centers of the neighborhoods), all π_{ri}'s are close to 0 and hence those points essentially do not move (since $\mathbf{y}_i \approx \mathbf{x}_i$).

We now formulate the optimization problem for finding the transformation parameters. The optimization criterion is defined as

$$J = d_{\mathcal{S}} + \lambda P, \tag{3}$$

where $d_{\mathcal{S}}$ is the sum of squared Euclidean distances for all similar pairs in the transformed space

$$d_{\mathcal{S}} = \sum_{(\mathbf{x}_r,\mathbf{x}_s)\in\mathcal{S}} \|\mathbf{y}_r - \mathbf{y}_s\|^2,$$

and P, a penalty term used to constrain the degree of transformation, is defined as

$$P = \sum_i \sum_j \mathcal{N}_\sigma(d_{ij})(q_{ij} - d_{ij})^2, \tag{4}$$

where $q_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|$ and $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ represent the inter-point Euclidean distances in the transformed and original spaces, respectively. $\mathcal{N}_\sigma(d_{ij})$ is again in the form of a Gaussian function, as $\mathcal{N}_\sigma(d_{ij}) = \exp(-d_{ij}^2/\sigma^2)$, with parameter σ specifying the spread of the Gaussian window. The regularization parameter $\lambda > 0$ in (3) determines the relative significance of the penalty term in the objective function for the optimization problem. Note that the optimization criterion in (3) is analogous to objective functions commonly used in energy minimization models such as deformable models [20], with the penalty term P playing the role of an internal energy term.

The optimization problem formulated above can be solved in an iterative manner, resulting in an iterative metric adaptation procedure [21]. In Ref. [21], we decrease over time the Gaussian window parameters ω and σ, which determine the neighborhood size and the weights in the penalty term, respectively. In so doing, the local specificity is increased gradually to allow global nonlinearity in the transformation. More specifically, given the data point locations $\{\mathbf{y}_i^{(t)}\}$ and the window parameters $\omega^{(t)}$ and $\sigma^{(t)}$ at iteration t, the overall optimization criterion in (3) is rewritten as

$$\begin{aligned} J^{(t)}(\{\mathbf{b}_r\}; \{\mathbf{y}_i^{(t)}\}, \omega^{(t)}, \sigma^{(t)}) &= \sum_{(\mathbf{x}_r,\mathbf{x}_s)\in\mathcal{S}} \|\mathbf{y}_r^{(t+1)} - \mathbf{y}_s^{(t+1)}\|^2 \\ &\quad + \lambda \sum_i \sum_j \mathcal{N}_{\sigma^{(t)}}(d_{ij})(q_{ij}^{(t+1)} - d_{ij})^2. \end{aligned} \tag{5}$$

We seek to minimize $J^{(t)}$ by finding the optimal values of $\{\mathbf{b}_r\}$ as $\{\mathbf{b}_r^{(t)}\}$, which are then used to compute the location changes from $\{\mathbf{y}_i^{(t)}\}$ to $\{\mathbf{y}_i^{(t+1)}\}$.

However, based on the many experiments we have performed on both synthetic and real data sets, we find that the iterative procedure typically terminates after one or two iterations. In fact, the experimental results usually do not change much after the first iteration. In this paper, we consider non-iterative versions of the optimization methods studied in Ref. [21]. With these methods, we can disengage our attention from the consideration of decreasing Gaussian window parameters and setting the stopping criteria. In the next section, we further propose a more efficient method based on the spectral approach.

2.3. Two optimization methods: gradient method and iterative majorization

We solve the optimization problem by minimizing J in Eq. (3). Two different optimization methods based on the gradient method and iterative majorization are discussed in the following two subsections.

2.3.1. Gradient method

While the first term of J in (5) is quadratic in $\{\mathbf{b}_r\}$, the second term is of a more complex form. So we cannot find a closed-form solution for the optimal values of $\{\mathbf{b}_r\}$ simply by solving $\nabla_{\mathbf{b}_r} J = \mathbf{0}$, $1 \leqslant r \leqslant m$. However, by using perturbation value of d_{ij} to approximate q_{ij}, we can obtain an approximate closed-form solution

$$\mathbf{B} = -\mathbf{U}_1\mathbf{U}_2^+,$$

where

$$\mathbf{U}_1 = \sum_i \sum_j [s_{ij} + \lambda \mathcal{N}_\sigma(d_{ij})(1 - d_{ij}/q_{ij})] \times (\mathbf{y}_i - \mathbf{y}_j)(\boldsymbol{\pi}_i - \boldsymbol{\pi}_j)^{\mathrm{T}}$$

$$\mathbf{U}_2 = \sum_i \sum_j [s_{ij} + \lambda \mathcal{N}_\sigma(d_{ij})(1 - d_{ij}/q_{ij})] \times (\boldsymbol{\pi}_i - \boldsymbol{\pi}_j)(\boldsymbol{\pi}_i - \boldsymbol{\pi}_j)^{\mathrm{T}},$$

and $s_{ij} = 1$ if $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}$ and 0 otherwise. $\mathbf{U}_2^+$ denotes the pseudo-inverse of $\mathbf{U}_2$.

2.3.2. Iterative majorization

Let us define two $d \times n$ matrices $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]$ for n data points before and after transformation, respectively. From (2), we have

$$\mathbf{Y} = \mathbf{X} + \mathbf{B}\boldsymbol{\Pi} = (\mathbf{X}\boldsymbol{\Pi}^+ + \mathbf{B})\boldsymbol{\Pi} = \mathbf{L}\boldsymbol{\Pi},$$

where $\boldsymbol{\Pi} = [\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots, \boldsymbol{\pi}_n]$ is an $m \times n$ matrix. The optimization problem is then equivalent to minimization of J with respect to $\mathbf{L}$.

The optimization criterion $J(\mathbf{L})$ can be rewritten as

$$\begin{aligned} J(\mathbf{L}) &= \sum_{i,j} s_{ij} q_{ij}^2(\mathbf{L}) + \lambda \sum_{i,j} \mathcal{N}_\sigma(d_{ij})(q_{ij}(\mathbf{L}) - d_{ij})^2 \\ &= \sum_{i,j} (s_{ij} + \lambda \mathcal{N}_\sigma(d_{ij})) \\ &\quad \times \left(q_{ij}(\mathbf{L}) - \frac{\lambda \mathcal{N}_\sigma(d_{ij})}{s_{ij} + \lambda \mathcal{N}_\sigma(d_{ij})} d_{ij} \right)^2 \\ &\quad + \lambda \sum_{i,j} \mathcal{N}_\sigma(d_{ij}) \left(1 - \frac{\lambda \mathcal{N}_\sigma(d_{ij})}{s_{ij} + \lambda \mathcal{N}_\sigma(d_{ij})} \right) d_{ij}^2. \end{aligned}$$

We can omit the second term since it does not depend on $\mathbf{L}$. The equivalent optimization criterion is

$$\sum_i \sum_j \alpha_{ij} (q_{ij}(\mathbf{L}) - p_{ij})^2,$$

where

$$\alpha_{ij} = s_{ij} + \lambda \mathcal{N}_\sigma(d_{ij}),$$
$$p_{ij} = \frac{\lambda \mathcal{N}_\sigma(d_{ij})}{s_{ij} + \lambda \mathcal{N}_\sigma(d_{ij})} d_{ij}.$$

Since this form is the same as that for multidimensional scaling for discriminant analysis [22], we can solve the optimization problem by *iterative majorization*, which can be seen as an EM-like algorithm for problems with no missing data. We define

$$\mathbf{C} = \sum_i \sum_j \alpha_{ij} (\boldsymbol{\pi}_i - \boldsymbol{\pi}_j)(\boldsymbol{\pi}_i - \boldsymbol{\pi}_j)^{\mathrm{T}}$$

and

$$\mathbf{D}(\mathbf{L}) = \sum_i \sum_j e_{ij}(\mathbf{L})(\boldsymbol{\pi}_i - \boldsymbol{\pi}_j)(\boldsymbol{\pi}_i - \boldsymbol{\pi}_j)^{\mathrm{T}}$$

with

$$e_{ij}(\mathbf{L}) = \begin{cases} \dfrac{\lambda \mathcal{N}_\sigma(d_{ij}) d_{ij}}{q_{ij}(\mathbf{L})}, & q_{ij}(\mathbf{L}) > 0, \\ 0, & q_{ij}(\mathbf{L}) = 0. \end{cases}$$

Then the optimization problem consists of the following steps:

(1) Initialize $\mathbf{L}^{(0)}$; $u = 0$.
(2) $u = u + 1$; and compute

$$\mathbf{L}^{(u)} = \mathbf{L}^{(u-1)}(\mathbf{D}(\mathbf{L}^{(u-1)}))^{\mathrm{T}}(\mathbf{C}^{-1})^{\mathrm{T}}.$$

(3) If converged, then stop; otherwise go to Step 2.

3. A more efficient optimization method: spectral method

Recall that the penalty term P in (3) serves to constrain the degree of transformation, partly to prevent the occurrence of a degenerate transformation and partly to preserve the local topological relationships between data points. Besides defining the penalty term as in (4), there also exist other ways to achieve this goal. One possibility is to preserve the locally linear relationships between nearest neighbors, as in a nonlinear dimensionality reduction method called *locally linear embedding* (LLE) [23]. Specifically, we seek to find the best reconstruction weights for all data points, represented as an $n \times n$ weight matrix $\mathbf{W} = [w_{ij}]$, by minimizing the following cost function

$$\begin{aligned} \mathscr{E} &= \sum_i \left\| \mathbf{x}_i - \sum_{\mathbf{x}_j \in \mathcal{N}_i} w_{ij} \mathbf{x}_j \right\|^2 \\ &= \mathrm{Tr}[\mathbf{X}(\mathbf{I} - \mathbf{W})^{\mathrm{T}}(\mathbf{I} - \mathbf{W})\mathbf{X}^{\mathrm{T}}] \end{aligned}$$

with respect to $\mathbf{W}$ subject to the constraints $\sum_{\mathbf{x}_j \in \mathcal{N}_i} w_{ij} = 1$, where $\mathcal{N}_i$ denotes the set of K nearest neighbors of $\mathbf{x}_i$ and

Tr is the trace operator. This can be solved as a constrained least squares problem. With the optimal weight matrix $\mathbf{W}$ found, the penalty term P is defined to ensure that points $\mathbf{y}_i$'s in the transformed space preserve the local geometry of the corresponding points $\mathbf{x}_i$'s, i.e.

$$P = \mathrm{Tr}[\mathbf{Y}(\mathbf{I}-\mathbf{W})^{\mathrm{T}}(\mathbf{I}-\mathbf{W})\mathbf{Y}^{\mathrm{T}}],$$

subject to the constraints $(1/n)\sum_i \mathbf{y}_i = \frac{1}{n}\mathbf{1}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}} = 0$ and $(1/n)\sum_i \mathbf{y}_i\mathbf{y}_i^{\mathrm{T}} = (1/n)\mathbf{Y}\mathbf{Y}^{\mathrm{T}} = \mathbf{I}_d$, where $\mathbf{1}$ represents a vector of 1's and $\mathbf{I}_d$ is the $d \times d$ identity matrix.

The first term $d_{\mathscr{S}}$ of J in (3) can be rewritten as

$$\sum_{(\mathbf{x}_r,\mathbf{x}_s)\in\mathscr{S}} \|\mathbf{y}_r - \mathbf{y}_s\|^2 = \sum_i \sum_j u_{ij}\mathbf{y}_i^{\mathrm{T}}\mathbf{y}_j = \mathrm{Tr}[\mathbf{Y}\mathbf{U}\mathbf{Y}^{\mathrm{T}}],$$

where u_{ij} is the (i, j)th element in an $n \times n$ matrix $\mathbf{U}$ with u_{ij} defined as

$$u_{ij} = u_{ij} = \tau_{ij}\sum_{r=1}^{n} s_{ir} - (1-\tau_{ij})s_{ij}.$$

$\tau_{ij} = 1$ if $i = j$ and 0 otherwise, and $s_{ij} = 1$ if $(\mathbf{x}_i, \mathbf{x}_j) \in \mathscr{S}$ and 0 otherwise. Thus the optimization criterion can be expressed as

$$\begin{aligned} J &= \mathrm{Tr}[\mathbf{Y}\mathbf{U}\mathbf{Y}^{\mathrm{T}}] + \lambda\mathrm{Tr}[\mathbf{Y}(\mathbf{I}-\mathbf{W})^{\mathrm{T}}(\mathbf{I}-\mathbf{W})\mathbf{Y}^{\mathrm{T}}] \\ &= \mathrm{Tr}[\mathrm{L}\mathbf{\Pi}(\mathrm{U}+\lambda(\mathbf{I}-\mathbf{W})^{\mathrm{T}}(\mathbf{I}-\mathbf{W}))\mathbf{\Pi}^{\mathrm{T}}\mathrm{L}^{\mathrm{T}}], \qquad (6) \end{aligned}$$

subject to the constraints $(1/n)\mathbf{1}^{\mathrm{T}}\mathbf{\Pi}^{\mathrm{T}}\mathbf{L}^{\mathrm{T}} = 0$ and $(1/n)$ $\mathbf{L}\mathbf{\Pi}\mathbf{\Pi}^{\mathrm{T}}\mathbf{L}^{\mathrm{T}} = \mathbf{L}\mathbf{B}\mathbf{L}^{\mathrm{T}} = \mathbf{I}_d$.

Let

$$\mathbf{E} = \mathbf{\Pi}[\mathbf{U}+\lambda(\mathbf{I}-\mathbf{W})^{\mathrm{T}}(\mathbf{I}-\mathbf{W})]\mathbf{\Pi}^{\mathrm{T}},$$

$$\mathbf{F} = \frac{1}{n}\mathbf{\Pi}\mathbf{\Pi}^{\mathrm{T}}.$$

The solution to the optimization problem with respect to $\mathbf{L}$ is given by the second to $(d+1)$st smallest generalized eigenvectors $\mathbf{v}$ with $\mathbf{E}\mathbf{v} = \hat{\lambda}\mathbf{F}\mathbf{v}$. Minimization of J in the form of (6) by the spectral approach is analogous to minimization of (3) based on the gradient method and iterative majorization. We present some experimental results based on both gradient method and spectral method in Section 4.

4. Experiments on semi-supervised clustering

To assess the efficacy of LLMA, we perform extensive experiments on toy data as well as real data from the UCI Machine Learning Repository.[2]

4.1. Illustrative examples

Fig. 1 demonstrates the power of our LLMA method by comparing it with the RCA method [17] on three toy data sets.[3] RCA, as a metric learning method, changes the feature space by a global linear transformation, which assigns large weights to relevant dimensions and low weights to irrelevant dimensions. The relevant dimensions are estimated based on connected components composed of similar patterns. For each data set, we randomly select 10 similar pairs to form $\mathscr{S}$. For LLMA, the gradient method is used to obtain the transformed results. More details about these experiments will be given in Section 4.3.

Notice that although the original Euclidean metric is not good for the first data set, even applying a linear transformation (RCA) can give a new Euclidean metric that is significantly better in grouping data points from the same class together. However, this is no longer the case for the second and third data sets which are more difficult than the first data set, demonstrating the limitations of linear metric learning methods. On the other hand, LLMA, as a nonlinear metric learning method, can give satisfactory results for all three data sets.

4.2. Clustering algorithms and performance measures for comparative study

In order to assess the efficacy of LLMA for semi-supervised clustering, we compare the clustering results based on k-means with and without metric learning. Besides RCA method, we also repeat the experiments using the constrained k-means algorithm [14]. Constrained k-means algorithm is based on default Euclidean metric subject to the constraints that patterns in a pair $(\mathbf{x}_r, \mathbf{x}_s) \in \mathscr{S}$ are always assigned to the same cluster. As for LLMA, we use both the gradient method and the spectral method as presented in Section 2 and Section 3, respectively, to solve the optimization problem. More specifically, the following five clustering algorithms are compared:

69

(1) k-means without metric learning;
(2) Constrained k-means without metric learning;
(3) k-means with RCA for metric learning;
(4) k-means with LLMA for metric learning (gradient method);
(5) k-means with LLMA for metric learning (spectral method).

The Rand index [24] is used to measure the clustering quality in our experiments. It reflects the agreement of the clustering result with the ground truth. Let n_s be the number of point pairs that are assigned to the same cluster (i.e.,

[2] http://www.ics.uci.edu/ mlearn/MLRepository.html

[3] The MATLAB code for RCA was downloaded from the web page of an author of Ref. [17].

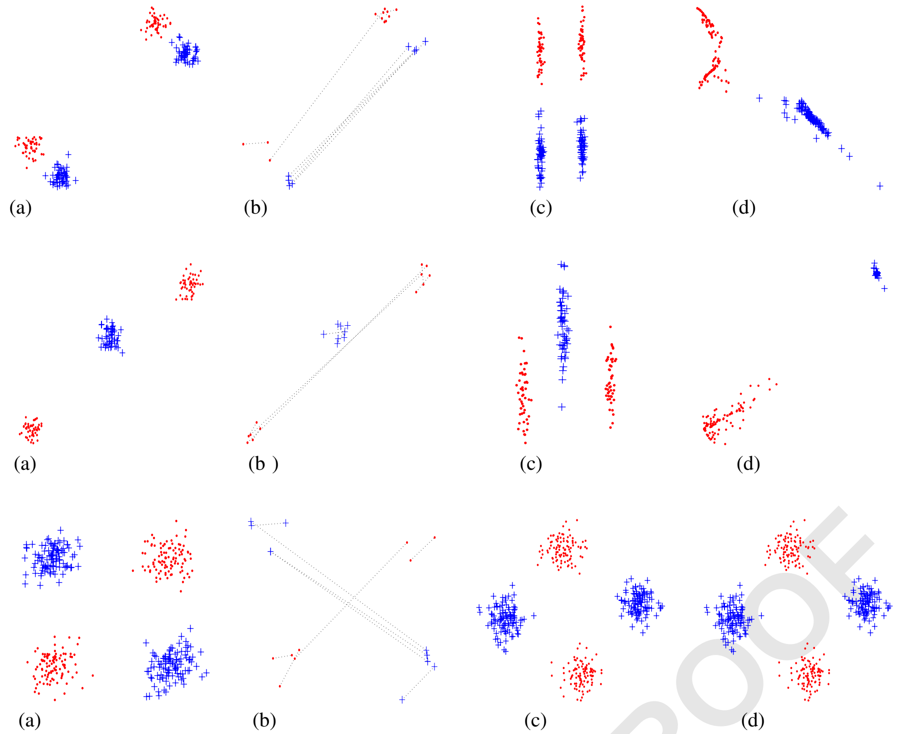

Fig. 1. Comparison of LLMA with RCA on three toy data sets. Subfigures in the first column show the data sets each with two classes, while subfigures in the second column show 10 similar pairs in $\mathscr{S}$ for each data set. The third and fourth columns show the data sets after applying RCA and LLMA, respectively, for metric learning.

matched pairs) in both the resultant partition and the ground truth, and n_d be the number of point pairs 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 point pairs, i.e., $n(n-1)/2$. When there are more than two clusters, however, the standard Rand index will favor assigning data points to different clusters. We modify the Rand index as in [16] so that matched pairs and mismatched pairs are assigned weights to give them equal chance of occurrence (0.5).

To see how different algorithms vary their performance with the background knowledge provided, we use 20 randomly generated $\mathscr{S}$ sets for each data set. Moreover, we compute the average Rand index over 20 random runs of (constrained) k-means for each $\mathscr{S}$ set. The results for all five algorithms are then shown as box-plots using MATLAB.

4.3. Semi-supervised clustering on toy and UCI data sets

In the LLMA algorithm, there are a few parameters that need to be set. For the gradient method described in Section 2, we make the Gaussian window parameters ω and σ depend on $\overline{d_0}$, which is the average initial Euclidean distance between all point pairs in $\mathscr{X}$ (i.e., $\overline{d_0} = 2/(n(n-1))\sum_{i<j}\|\mathbf{x}_i - \mathbf{x}_j\|$), as $\omega = \beta\overline{d_0}$ and $\sigma = \gamma\omega$. β and γ are constant parameters set to [0.1,3] and (0,1), respectively, in our experiments. For the spectral method described in Section 3, the only Gaussian window parameter ω is set in the same way. The regularization parameter λ adjusting the tradeoff between local transformation and geometry preservation is set to 5. All data sets are normalized before applying the five algorithms.

Fig. 2 shows the clustering results for the three toy data sets as illustrated in Section 4.1. Obviously, all the three data sets cannot be clustered well using the standard and constrained k-means algorithms. Even RCA can give good result only on the first data set. On the other hand, LLMA can handle all these cases and perform particularly well on the second and third data sets which cannot be handled satisfactorily by the other methods. For our LLMA method, the spectral approach leads to slightly better clustering results than the gradient method.

We further conduct experiments on nine UCI data sets. The number of data points n, the number of features d, the number of classes c, and the number of randomly selected similar pairs $|\mathscr{S}|$ are shown under each subfigure in Fig. 3. From the clustering results, we can see that LLMA outperforms the other methods for most of these data sets. As for the iris and Boston housing data sets, RCA can improve the clustering results most. For LLMA, the clustering results obtained using the gradient and spectral methods are comparable.

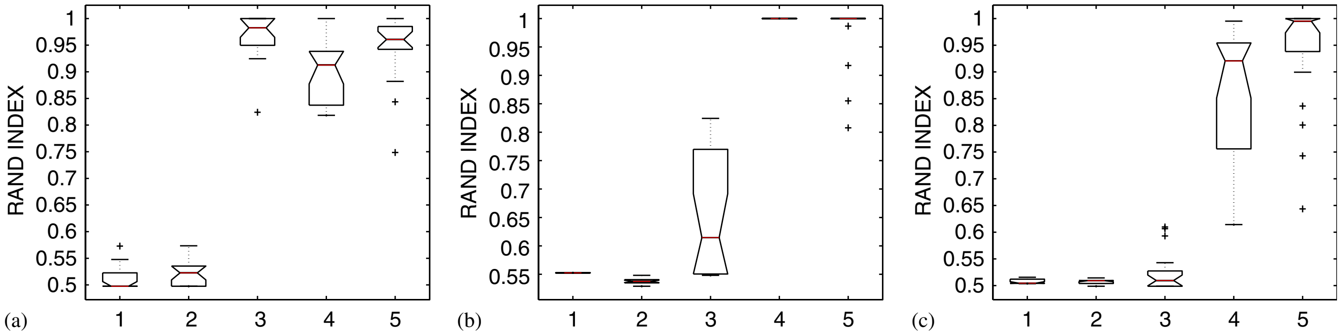

Fig. 2. Clustering results for toy data sets shown as box-plots for 20 different $\mathscr{S}$ sets with $|\mathscr{S}| = 10$ (the five clustering algorithms are numbered as in Section 4.2). (a) Toy data set 1, (b) toy data set 2; (c) toy data set 3.

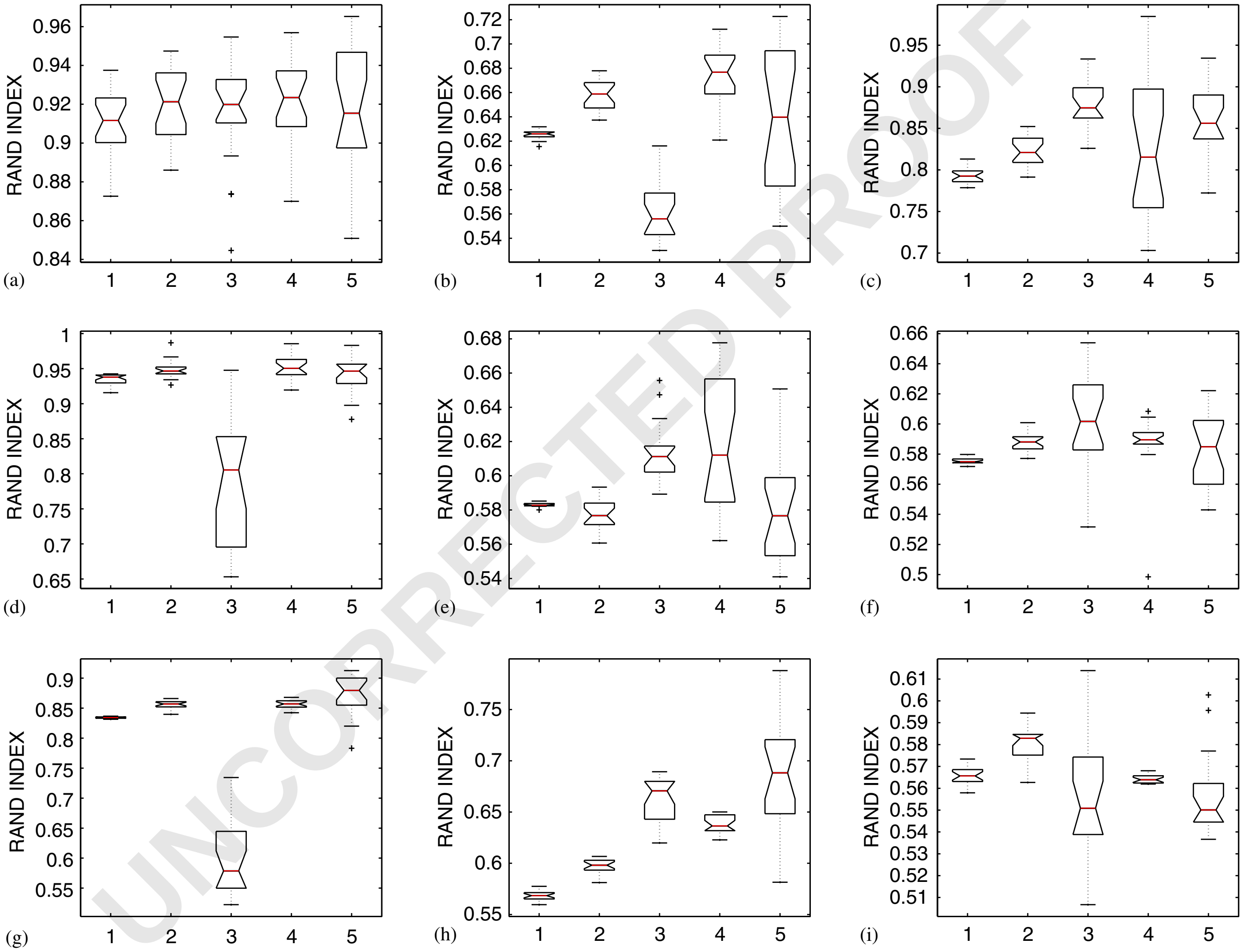

Fig. 3. Clustering results for UCI data sets shown as box-plots for 20 different $\mathscr{S}$ sets (the five clustering algorithms are numbered as in Section 4.2). (a) Soybean $n = 47, d = 35, c = 4, |S| = 10$; (b) protein $n = 116, d = 20, c = 6, |S| = 20$; (c) iris plants $n = 150, d = 4, c = 3, |S| = 30$; (d) wine $n = 178, d = 13, c = 3, |S| = 20$; (e) ionosphere $n = 351, d = 34, c = 2, |S| = 30$; (f) boston housing $n = 506, d = 13, c = 3, |S| = 40$; (g) breast cancer $n = 569, d = 31, c = 2, |S| = 50$; (h) balance $n = 625, d = 4, c = 3, |S| = 40$; (i) diabetes $n = 768, d = 8, c = 2, |S| = 40$.

To summarize, these experimental results on both toy and real data sets demonstrate the effectiveness of our LLMA method.

5. Experiments on image retrieval

5.1. 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 interest [25]. 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 [17,26] has been used to improve image retrieval performance in CBIR tasks.

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

5.2. 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) [27] 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 $(\alpha_1, \beta_1, \ldots, \alpha_{512}, \beta_{512})^{\mathrm{T}}$, with α_i and β_i representing the numbers of coherent and noncoherent 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 LLMA as described above.

5.3. Comparative study and performance measures

We compare the image retrieval performance of LLMA with the baseline method of using Euclidean distance without distance learning, as well as some other distance learning methods. In particular, we consider two distance learning methods: Mahalanobis distance with whitening transform and RCA.

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 is based on *cumulative neighbor purity* curves. Cumulative neighbor purity measures the percentage of correctly retrieved 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 five 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. In LLMA, we use the spectral method (Section 3) because it is more efficient than the other two optimization methods.

5.4. Experimental results

5.4.1. Basic retrieval results

Fig. 4 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 LLMA 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 Fig. 5. Note that this database is highly unbalanced as the class sizes vary significantly. For this database, both whitening transform and RCA cannot improve the retrieval performance. On the other hand, LLMA significantly outperforms the other methods in improving the retrieval performance.

Some typical retrieval results on the first and second databases are shown in Fig. 6(a) and (b), respectively. For each query image, we show the retrieved images in three rows, corresponding, from top to bottom, to the use of Euclidean distance without distance learning and distance learning with RCA and LLMA. Each row shows the seven 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. We can see that both distance learning methods improve the retrieval performance, with LLMA outperforming RCA slightly.

5.4.2. Results with relevance feedback

As in traditional information retrieval, *relevance feedback* from users on the retrieval results is considered as a

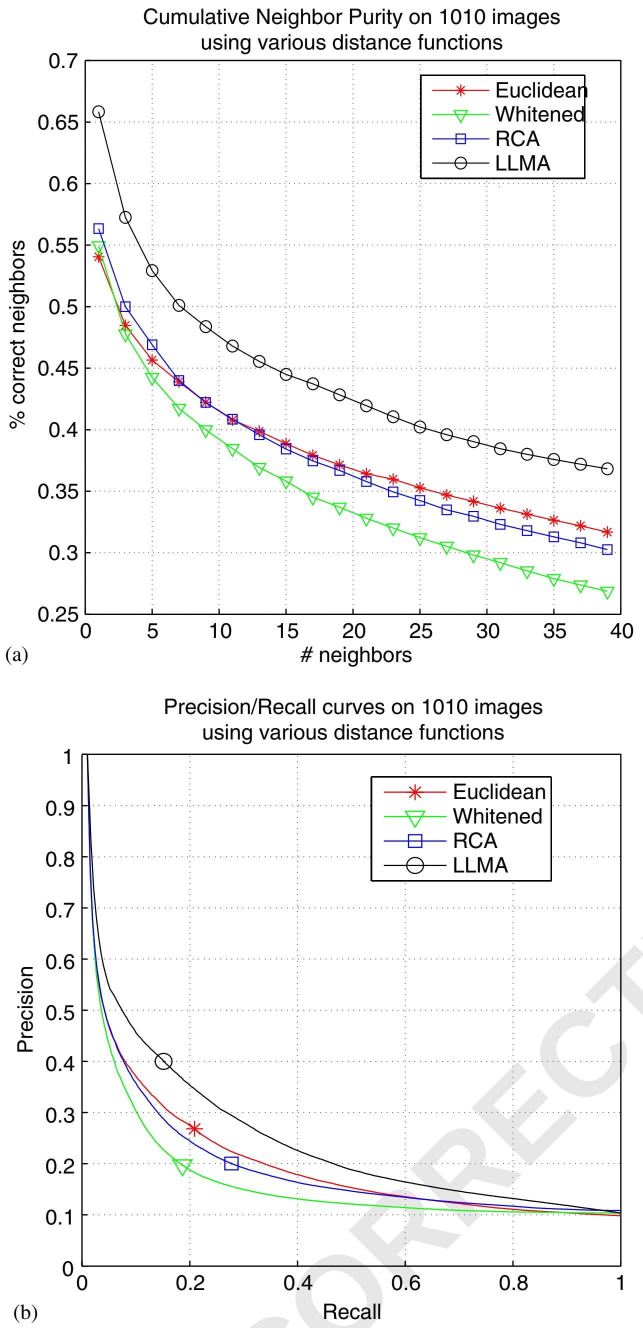

Fig. 4. Retrieval results on the first image database (1010 images, 10 classes). (a) Cumulative neighbor purity curves; (b) precision/recall curves.

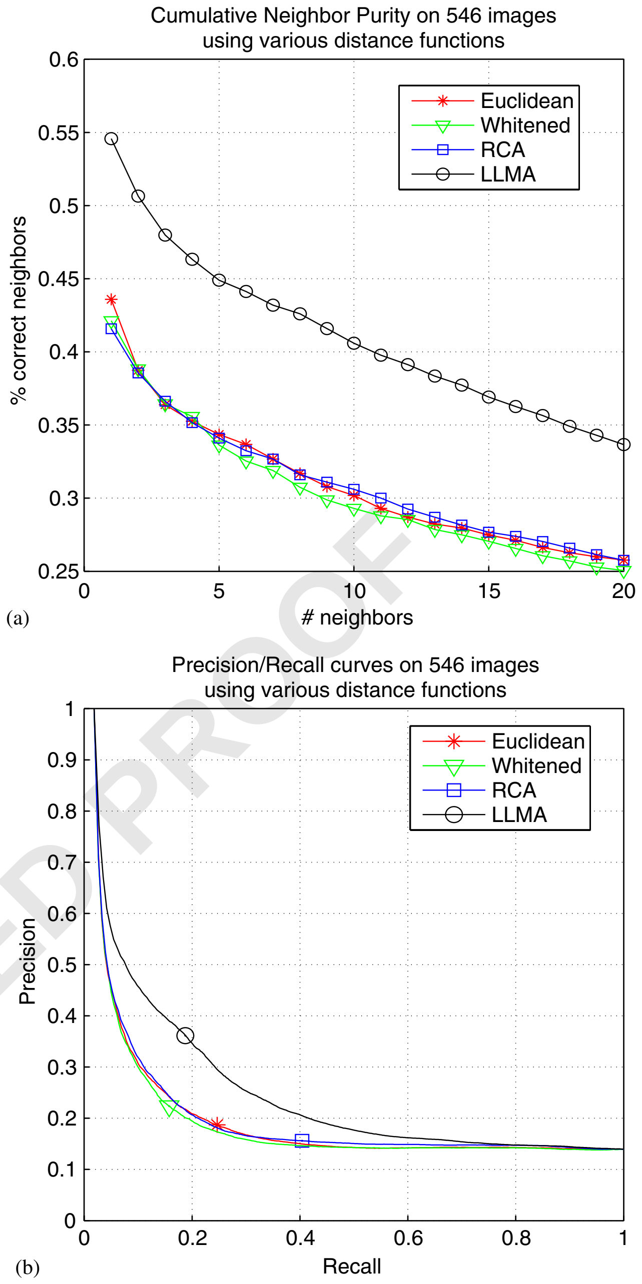

Fig. 5. Retrieval results on the second image database (546 images, 10 classes). (a) Cumulative neighbor purity curves; (b) precision/recall curves.

powerful tool to bridge the gap between low-level features and high-level semantics in CBIR systems [28]. When displayed images are retrieved in response to the query image(s), the user is allowed to label some or all of the retrieved images as either relevant or irrelevant. Based on the relevance feedback, the system modifies either the query or the distance function and then carries out another retrieval attempting to improve the retrieval performance. Most existing systems only make use of relevance feedback within a single query session.

Similarity constraints used in LLMA can be obtained from users' relevance feedback, with each relevant image and the query image forming a similar pair. We accumulate the similarity constraints over multiple query sessions before applying LLMA. 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 120 images from four classes. Fig. 7 shows the results in terms of cumulative neighbor purity curves for different numbers of pairwise similarity

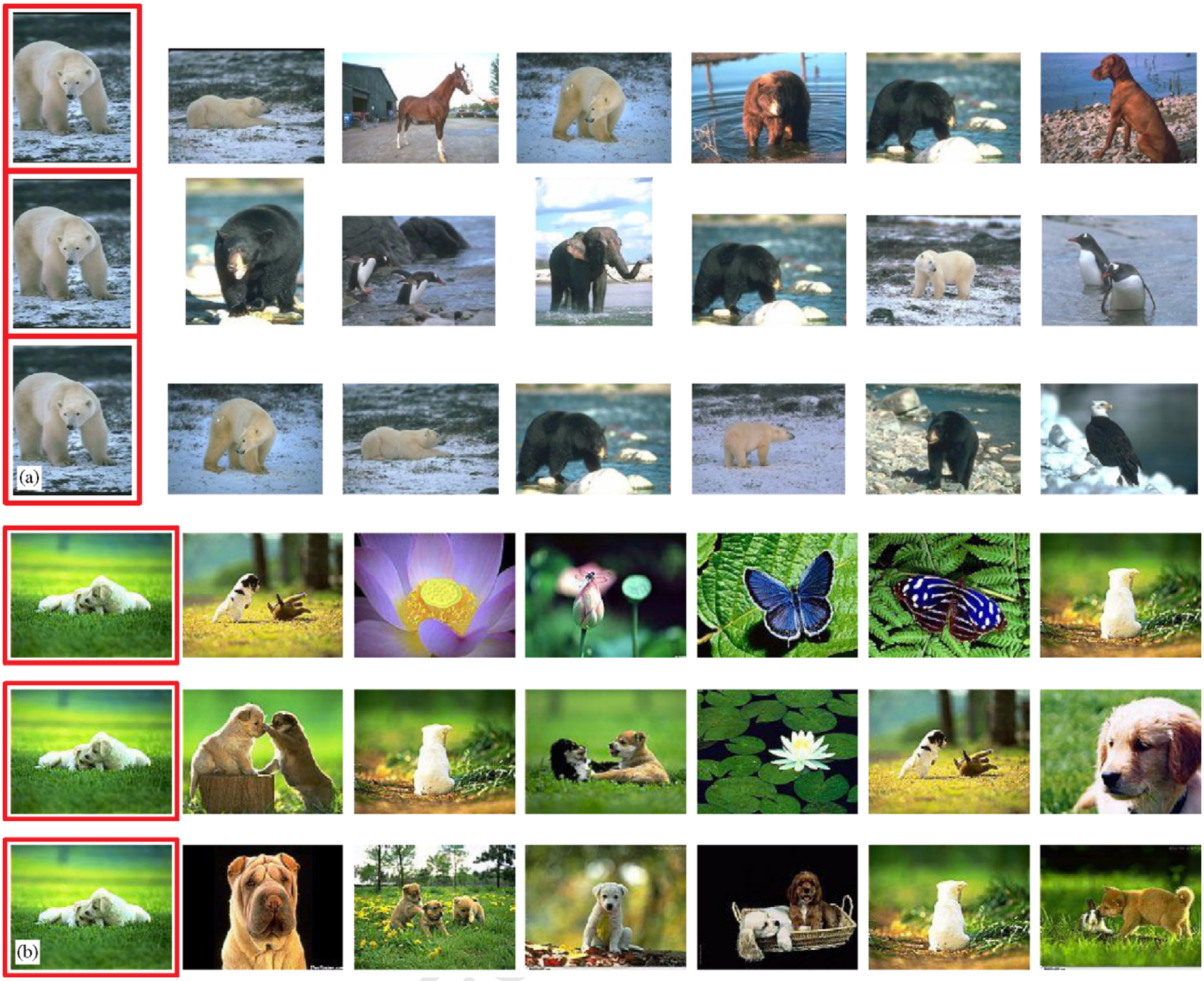

Fig. 6. Typical retrieval results on the two databases (a) and (b) bases on Euclidean distance (first row), RCA (second row), and LLMA (third row). Each row shows the seven nearest neighbors including the query image (framed).

constraints. It is clear that more pairwise constraints can lead to greater improvement.

However, using pairwise constraints collected from many query sessions also implies higher computational demand. As a compromise, we can perform stepwise LLMA by incorporating the pairwise constraints in reasonably small, incremental batches each of a certain size ρ. Whenever the batch of newly collected pairwise constraints reaches this size, LLMA will be performed with this batch to obtain a new metric. The batch of similarity constraints is then discarded. This process will be repeated continuously with the arrival of more relevance feedback from users. In so doing, knowledge acquired from relevance feedback in one session can be best utilized to give long-term improvement in subsequent sessions.

We conduct some experiments on the first image database to verify the effectiveness of this method. For a prespecified maximum batch size ρ, we randomly select ρ images from the database as query images. In each query session based on one of the ρ images, the system returns the top 20 images from the database based on the current distance function, which is Euclidean initially. Of these 20 images, five relevant images are then randomly chosen, simulating the relevance feedback process performed by a user. LLMA is performed once after every ρ sessions. Fig. 8 shows the cumulative neighbor purity curves for the retrieval results based on stepwise LLMA with maximum batch sizes $\rho = 10$ sessions. As we can see, long-term metric learning based on stepwise LLMA can result in continuous improvement of retrieval performance.

5.4.3. Results with noisy pairwise constraints

So far, we have assumed that the pairwise constraints available for metric learning are all correct. However, this

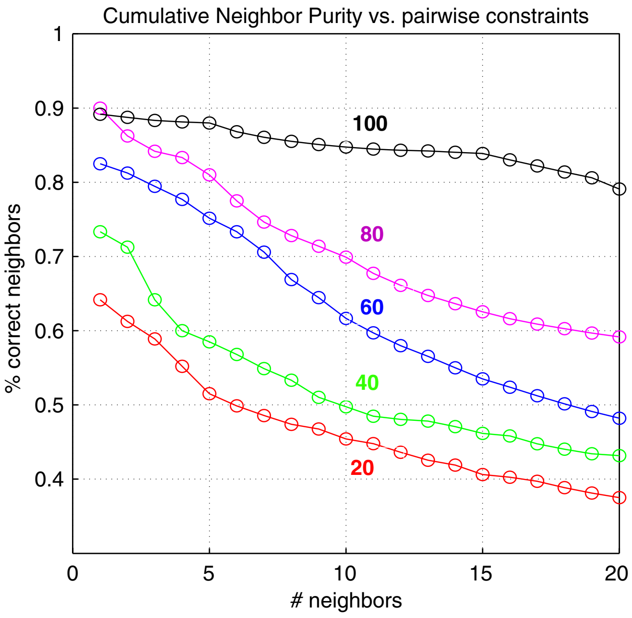

Fig. 7. Cumulative neighbor purity curves for different numbers of pairwise similarity constraints, ranging from 20 to 100.

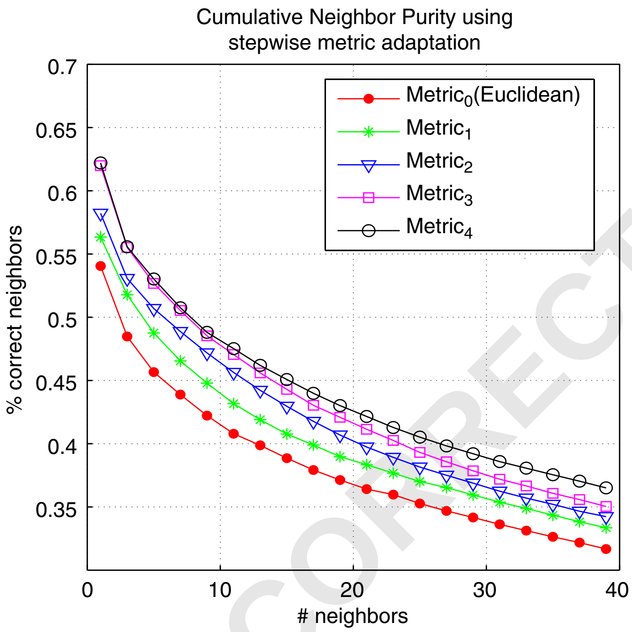

Fig. 8. Cumulative neighbor purity curves based on stepwise LLMA with maximum batch size $\rho = 10$ sessions.

assumption may not hold in some applications. For example, in CBIR, some pairwise constraints provided as relevance feedback to the users may not be correct, in the sense that they do not agree with the high-level semantics. We perform some preliminary experiments here to study the robustness of a CBIR system when there exist noisy pairwise constraints in the relevance feedback.

We use the second image database in our study. In addition to the 150 similar image pairs, we randomly select

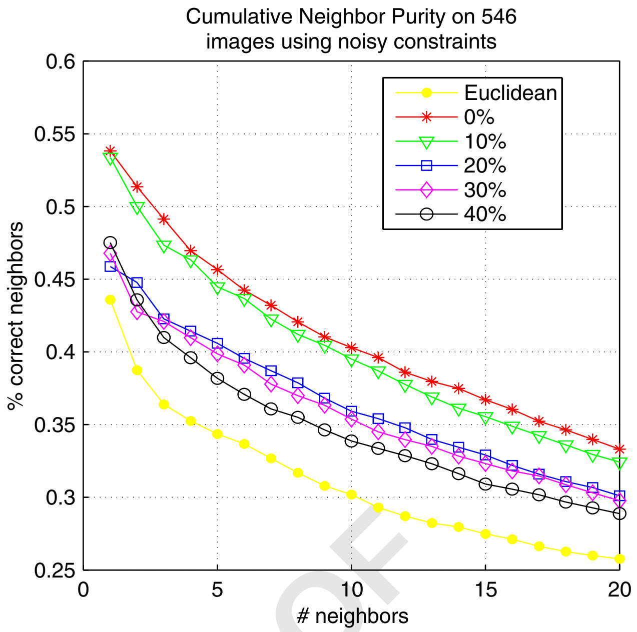

Fig. 9. Cumulative neighbor purity curves for different numbers of noisy pairwise similarity constraints, ranging from 0 to 40%.

some dissimilar image pairs and add them to the set $\mathcal{S}$ as noisy pairs. Fig. 9 shows the retrieval results reported by cumulative neighbor purity curves with different numbers of noisy pairwise similarity constraints incorporated. As expected, the retrieval performance degrades with the number of noisy constraints added. However, even with 40% noisy constraints added, LLMA still gives better retrieval performance than the baseline Euclidean metric.

6. Concluding remarks

In this paper, we have proposed a new metric adaptation method called LLMA based on semi-supervised learning. Unlike previous methods which can only perform linear transformation globally, LLMA performs nonlinear transformation globally but linear transformation locally. This generalization makes it more powerful for solving some difficult clustering tasks as demonstrated through the toy and UCI data sets.

We have simplified the optimization methods presented in [21], and have proposed a more efficient optimization method for LLMA based on the spectral approach. Besides performing semi-supervised clustering on toy and real data sets, we have also demonstrated the promising performance of LLMA for CBIR tasks. Not only does LLMA based on semi-supervised metric learning improve the retrieval performance of Euclidean distance without distance learning, it also outperforms other distance learning methods significantly due to its higher flexibility in metric learning.

Note that in LLMA, the original input space and the transformed space are explicitly related via a mapping, as $\mathbf{Y}{=}\mathbf{L}\mathbf{\Pi}$, where $\mathbf{\Pi}$ is a nonlinear function with respect to $\mathbf{X}$. Although it is not necessary for clustering problems, it is possible for new data points added to the input space to be mapped onto the transformed space. One example is the CBIR application if the query image is not from the image database. We will also explore other applications that can make use of this favorable property.

Currently, our method can only utilize similarity constraints. A natural question to ask is whether we can extend LLMA by incorporating dissimilarity constraints. In principle this is possible, but the optimization criterion has to be modified in order to incorporate the new constraints. One challenge to face is to maintain the form of the objective function so that the optimization problem remains tractable.

Moreover, we have only considered a restrictive form of locally linear transformation, namely, translation. A potential direction to pursue is to generalize it to more general linear transformation types. Other possible research directions include improving the current LLMA algorithm such as performing globally linear transformation first and then LLMA only when necessary.

Acknowledgments

The research described in this paper has been supported by HKUST6174/04E from the Research Grants Council of the Hong Kong Special Administrative Region, China.

References

[1] K. Fukunaga, L. Hostetler, Optimization of k-nearest neighbor density estimates, IEEE Trans. Inf. Theory 19 (3) (1973) 320–326.

[2] R.D. Short, K. Fukunaga, The optimal distance measure for nearest neighbor classification, IEEE Trans. Inf. Theory 27 (5) (1981) 622–627.

[3] K. Fukunaga, T.E. Flick, An optimal global nearest neighbor metric, IEEE Trans. Pattern Anal. Mach. Intelligence 6 (3) (1984) 314–318.

[4] C. Domeniconi, J. Peng, D. Gunopulos, Locally adaptive metric nearest-neighbor classification, IEEE Trans. Pattern Anal. Mach. Intell. 24 (9) (2002) 1281–1285.

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

[6] T. Hastie, R. Tibshirani, Discriminant adaptive nearest neighbor classification, IEEE Trans. Pattern Anal. Mach. Intell. 18 (6) (1996) 607–616.

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

[8] J. Peng, D.R. Heisterkamp, H.K. Dai, Adaptive kernel metric nearest neighbor classification, in: Proceedings of the 16th International Conference on Pattern Recognition, 11–15 August 2002, Québec City, Québec, Canada, vol. 3, pp. 33–36.

[9] T. Poggio, F. Girosi, Networks for approximation and learning, Proc. IEEE 78 (9) (1990) 1481–1497.

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

[11] S. Basu, A. Banerjee, R. Mooney, Semi-supervised clustering by seeding, in: Proceedings of the 19th International Conference on Machine Learning, Sydney, Australia, 8–12 July 2002, pp. 19–26.

[12] Z. Zhang, J.T. Kwok, D.Y. Yeung, Parametric distance metric learning with label information, in: Proceedings of the 18th International Joint Conference on Artificial Intelligence, 9–15 August 2003, Acapulco, Mexico, pp. 1450–1452.

[13] K. Wagstaff, C. Cardie, Clustering with instance-level constraints, in: Proceedings of the 17th International Conference on Machine Learning, Standord, CA, USA, 2000, pp. 1103–1110.

[14] K. Wagstaff, C. Cardie, S. Rogers, S. Schroedl, Constrained k-means clustering with background knowledge, in: Proceedings of the 18th International Conference on Machine Learning, Williamstown, MA, USA, 2001, pp. 577–584.

[15] D. Klein, S.D. Kamvar, From instance-level constraints to space-level constraints: making the most of prior knowledge in data clustering, in: Proceedings of the 19th International Conference on Machine Learning, 2002.

[16] E.P. Xing, A.Y. Ng, M.I. Jordan, S. Russell, Distance metric learning, with application to clustering with side-information, in: S. Becker, S. Thrun, K. Obermayer (Eds.), Advances in Neural Information Processing Systems 15, MIT Press, Cambridge, MA, USA, 2003, pp. 505–512.

[17] A. Bar-Hillel, T. Hertz, N. Shental, D. Weinshall, Learning distance functions using equivalence relations, in: Proceedings of the 20th International Conference on Machine Learning, 21–24 August 2003, Washington DC, USA, pp. 11–18.

[18] N. Shental, A. Bar-Hillel, T. Hertz, D. Weinshall, Computing Gaussian mixture models with EM using equivalence constraints, in: Advances in Neural Information Processing Systems 16, MIT Press, Cambridge, MA, USA, 2004.

[19] J.T. Kwok, I.W. Tsang, Learning with idealized kernels, in: Proceedings of the 20th International Conference on Machine Learning, 21–24 August 2003, Washington DC, USA, pp. 400–407.

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

[21] H. Chang, D.Y. Yeung, Locally linear metric adaptation for semi-supervised clustering, in: Proceedings of the 21st International Conference on Machine Learning, 4–8 August 2004, Banff, Alberta, Canada, pp. 153–160.

[22] A.R. Webb, Multidimensional scaling by iterative majorization using radial basis functions, Pattern Recognition 28 (5) (1995) 753–759.

[23] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.

[24] W.M. Rand, Objective criteria for the evaluation of clustering methods, J. Am. Stat. Assoc. 66 (1971) 846–850.

[25] A. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Trans. Pattern Anal. Mach. Intelligence 22 (12) (2000) 1349–1380.

[26] T. Hertz, N. Shental, A. Bar-Hillel, 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, 18–20 June 2003, Madison, WI, USA, vol. 2, pp. 668–674.

[27] G. Pass, R. Zabih, J. Miller, Comparing images using color coherence vectors, in: Proceedings of the Fourth ACM International Conference on Multimedia, 1996, pp. 65–73.

[28] Y. Rui, T.S. Huang, M. Ortega, S. Mehrotra, Relevance feedback: a power tool for interactive content-based image retrieval, IEEE Trans. Circuits Syst. Video Technol. 8 (5) (1998) 644–655.