Semi-Supervised Distance Metric Learning for Collaborative Image Retrieval and Clustering

Steven C.H. Hoi
School of Computer Engineering, Nanyang Technological University
chhoi@ntu.edu.sg
Wei Liu
Department of Electrical Engineering, Columbia University
wliu@ee.columbia.edu
Shih-Fu Chang
Department of Electrical Engineering, Columbia University
sfchang@ee.columbia.edu

Learning a good distance metric plays a vital role in many multimedia retrieval and data mining tasks. For example, a typical content-based image retrieval (CBIR) system often relies on an effective distance metric to measure similarity between any two images. Conventional CBIR systems simply adopting Euclidean distance metric often fail to return satisfactory results mainly due to the well-known semantic gap challenge. In this paper, we present a novel framework of **Semi-Supervised Distance Metric Learning** for learning effective distance metrics by exploring the historical relevance feedback log data of a CBIR system and utilizing unlabeled data when log data are limited and noisy. We formally formulate the learning problem into a convex optimization task and then present a new technique, named as "Laplacian Regularized Metric Learning" (LRML). Two efficient algorithms are then proposed to solve the LRML task. Further, we apply the proposed technique to two applications. One direct application is for Collaborative Image Retrieval (CIR), which aims to explore the CBIR log data for improving the retrieval performance of CBIR systems. The other application is for Collaborative Image Clustering (CIC), which aims to explore the CBIR log data for enhancing the clustering performance of image pattern clustering tasks. We conduct extensive evaluation to compare the proposed LRML method with a number of competing methods, including 2 standard metrics, 3 unsupervised metrics, and 4 supervised metrics with side information. Encouraging results validate the effectiveness of the proposed technique.

Categories and Subject Descriptors: H3.3 [**Information Systems**]: Information Search and Retrieval; H.2.8 [**Database Management**]: Database ApplicationsłData mining

General Terms: Algorithm, Experimentation

Additional Key Words and Phrases: distance metric learning, content-based image retrieval, multimedia data clustering

A short version of this work has been published in the *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR2008), Alaska, 24-26 June, 2008.

Contact Author: Steven C.H. Hoi is with the School of Computer Engineering of the Nanyang Technological University, Singapore. E-mail: chhoi@ntu.edu.sg, Tel: (+65) 6513-8040 Fax: (+65) 6792-6559

1. INTRODUCTION

Determination of appropriate distance metrics plays a key role in many multimedia applications, including multimedia retrieval and multimedia data mining tasks. For example, choosing a valid distance metric is often critical to building an effective content-based image retrieval (CBIR) system [Smeulders et al. 2000; Lew et al. 2006]. For a regular CBIR system, in order to measure the visual distance/similarity between a query image and an image in database, the CBIR system has to predefine some distance metric for similarity measure, e.g. Euclidean distance is often adopted. Besides CBIR, for unsupervised multimedia data mining tasks, Euclidean distance is often used in conjunction with clustering algorithms, such as k-means clustering [Jain et al. 1999]. Unfortunately, Euclidean distance is often inadequate for these applications primarily because of the well-known semantic gap between low-level features and high-level semantics [Smeulders et al. 2000].

In response to the semantic gap challenge, relevance feedback has been extensively studied in CBIR [Rui et al. 1997; Rui et al. 1998; Tong and Chang 2001; King and Zhong 2003; Hoi and Lyu 2004a; Tao and Tang 2004]. In general, relevance feedback aims to interactively improve the retrieval performance by learning with users' judgements on the retrieval results. More specifically, for a CBIR retrieval task, the CBIR system first returns a short list of top ranked images with respect to a user's query by a regular retrieval approach based on Euclidean distance measure, and then requests the user to make relevance judgement on the retrieval results. Based on the user's feedback, the CBIR system is expected to learn an effective ranking function with the labeled data and retrieve more relevant images for the retrieval task. In the past decade, extensive studies have shown that relevance feedback is a powerful technique to improve the CBIR performance.

Despite the broad interest, regular relevance feedback techniques often suffer from some drawbacks. The most obvious one is the communication overhead imposed on the systems and users. CBIR systems with relevance feedback often require a non-trivial number of iterations before improved search results are obtained; this makes the process inefficient and unattractive for online applications. A useful CBIR system should minimize the times that it needs to engage the user in *online* feedback.

Recently, an increasing number of studies have attempted to attack the above challenge by exploring historical relevance feedback log data [Hoi et al. 2006; Si et al. 2006]. Such systems accumulate feedback information collected in multiple image retrieval sessions possibly conducted by multiple users for different search targets. We refer to a paradigm of utilizing CBIR log data in an image retrieval task as "Collaborative Image Retrieval" (CIR). In literature, there are two kinds of CIR approaches for exploring the historical CBIR log data. One is to reduce the number of relevance feedback iterations by devising the *log-based relevance feedback* technique [Hoi et al. 2006] that improves regular relevance feedback techniques by utilizing the historical log data. The other solution is to learn an effective distance metric for bridging the semantic gap by mining the historical feedback log data [Si et al. 2006; Hoi et al. 2006; Hoi et al. 2008]. In this paper, we focus on investigating distance metric learning techniques for mining the historical feedback log data toward two applications. One direct application is CIR, and the other is to enhance an unsupervised image clustering task by utilizing the log data, which is referred to as "Collaborative Image Clustering" (CIC).

Recently, learning distance metrics from log data or called "side information" [Xing et al. 2002] has been actively studied in machine learning and pattern recognition communities [Xing et al. 2002; Bar-Hillel et al. 2005; Hoi et al. 2006]. Despite active research efforts in the past few years, existing distance metric learning techniques are usually sensitive to noise and unable to learn a reliable metric when dealing with noisy data or only a small amount of log data, which are two common issues in the real-world relevance feedback log data. In this paper, we propose a novel framework of semi-supervised distance metric learning, which incorporates unlabeled data in the distance metric learning task. Specifically, we develop a novel technique of Laplacian Regularized Metric Learning (LRML) to integrate the unlabeled data information through a regularized learning framework. We formally formulate the technique into an optimization task and present two efficient algorithms to solve the task. One is based a Semidefinite Program (SDP) [Hoi et al. 2008], which can efficiently find global optimum for small-scale problems by existing convex optimization techniques, and the other is based on a simple matrix inversion algorithm, which can solve large-scale problems much more efficiently.

Here we highlight the major contributions of this paper: (1) a novel regularization framework for distance metric learning and a new semi-supervised metric learning technique, i.e., LRML; (2) two efficient algorithms to perform Laplacian Regularized Metric Learning (LRML); (3) a comprehensive study of applying the LRML technique to two applications: collaborative image retrieval and collaborative image clustering, through the exploration of real CBIR log data; (4) an extensive experimental evaluation of comparing our technique with a number of competing distance metric learning methods.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 formally defines the distance metric learning problem and proposes the framework of semi-supervised distance metric learning. Section 4 presents the the proposed LRML technique for the CIR application. Section 4 applies the proposed technique to a new application of collaborative image clustering. Section 6 gives experimental evaluations on some testbeds of real CBIR log data. Section 7 concludes this paper.

2. RELATED WORK

Our work is mainly related to two groups of research. One is the studies of exploring users' relevance feedback log data in CBIR. The other is distance metric learning research in machine learning. We briefly review some representative work in both sides.

2.1 CBIR Feedback Log Mining

In recent years, there are some emerging research interests for exploring historical log data of user relevance feedback in CBIR. Hoi et al. [Hoi and Lyu 2004b; Hoi et al. 2006] proposed a log-based relevance feedback technique with support vector machines (SVM) by engaging user feedback log data in traditional online relevance feedback tasks. In the solution, a small set of relevant and irrelevant images are acquired from users by online relevance feedback. Based on the labeled images collected in the relevance feedback sessions, the images in the database that are similar to the current labeled examples are included in the pool of labeled data for training some retrieval models, such as SVMs. In addition to this work, some other solutions, such as the log-based relevance feedback with the coupled SVM method [Hoi et al. 2005], was also proposed, in which every image in the database is represented by two modalities, i.e., visual and log, and then an unified SVM model is learned on the two modalities. Besides SVM based approaches, there were some other

efforts in exploring log data with other machine learning techniques, such as manifold learning [He et al. 2004], which takes into consideration the log data when learning an optimal mapping function via manifold learning. Finally, there are some research work on studying weighting schemes for low-level visual features via mining user log data [Müller et al. 2004]. In [Müller et al. 2004], similar to the TF-IDF scheme in text retrieval [Salton and Buckley 1988], the authors suggested a weighting scheme by exploiting the log data of user's relevance judgments in CBIR.

Different from the foregoing previous work, some recent studies have explored log data for learning distance metrics [Si et al. 2006; Hoi et al. 2006; Hoi et al. 2008], which can be applied to various applications. Following the same direction, our work in this paper mainly investigates a new distance metric learning technique towards two real applications through exploring users' relevance feedback log data.

2.2 Distance Metric Learning

The other major group of related work is distance metric learning research in machine learning, which can be further classified into three major categories. One category is unsupervised learning techniques, most of which attempt to find low-dimensional embeddings from high-dimensional input data. Some well-known techniques include classical Principal Component Analysis (PCA) [Fukunaga 1990] and Multidimensional Scaling (MDS) [Cox and Cox 1994]. In addition, some manifold-based approaches study nonlinear techniques, such as Locally Linear Embedding (LLE) [Roweis and Saul 2000] and Isomap [Tenenbaum and de Silva andJohn C. Langford 2000].

Another category is supervised metric learning techniques for classification tasks. These methods usually learn metrics from training data associated with explicit class labels. The representative techniques include Fisher Linear Discriminant Analysis (LDA) [Fukunaga 1990] and some recently proposed methods, such as Neighbourhood Components Analysis (NCA) [J. Goldberger and Salakhutdinov 2005], Maximally Collapsing Metric Learning [Globerson and Roweis 2005], metric learning for Large Margin Nearest Neighbor classification (LMNN) [Weinberger et al. 2006], and Local Distance Metric Learning [Yang et al. 2006], etc.

Our work is closer to the third category, which learns metrics from log data of pairwise constraints, or called "side information" [Xing et al. 2002], in which each pairwise constraint indicates if two examples are relevant (similar) or irrelevant (dissimilar) in a particular learning task. A popular DML approach was proposed by Xing et al. [Xing et al. 2002], which formulated the task as a convex optimization problem, and applied the technique to clustering. Following their work, there are a group of emerging DML studies. For example, Relevant Component Analysis (RCA) learns a global linear transformation by exploiting only equivalent constraints [Bar-Hillel et al. 2005]. Discriminant Component Analysis (DCA) improves RCA by incorporating negative constraints [Hoi et al. 2006]. Si et al. [Si et al. 2006] proposed a regularized metric learning method for CIR. Recently, Lee et al. [Lee et al. 2008] studied a rank-based distance metric learning method for CBIR. Most existing work often learn only with side information without exploring unlabeled data. To overcome the limitations, this paper proposes a novel semi-supervised distance metric learning framework for learning effective and reliable metrics by incorporating unlabeled data in the DML tasks [Hoi et al. 2008]. To the best of our knowledge, this is the first work to explore unlabeled data explicitly for the DML tasks in this category of research.

3. SEMI-SUPERVISED DISTANCE METRIC LEARNING

3.1 Problem Definition

Suppose we are given a set of n data points in an m-dimensional vector space $\mathcal{C} = \{\mathbf{x}_i\}_{i=1}^n \subseteq \mathbb{R}^m$, and two sets of pairwise constraints among the data points:

$$\mathcal{S} = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are judged to be relevant}\}$$
$$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are judged to be irrelevant}\}$$

where $\mathcal{S}$ is the set of *similar* pairwise constraints and $\mathcal{D}$ is the set of *dissimilar* pairwise constraints. Each pairwise constraint $(\mathbf{x}_i, \mathbf{x}_j)$ indicates if the two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ are relevant or irrelevant judged by users in some application context.

For any two given data points $\mathbf{x}_i$ and $\mathbf{x}_j$, let $d(\mathbf{x}_i, \mathbf{x}_j)$ denote the distance between them. To compute the distance, let $\mathbf{A} \in \mathbb{R}^{m \times m}$ be the distance metric, we can then express the formula of distance measure as follows:

$$d_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}} = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{A}(\mathbf{x}_i - \mathbf{x}_j)} = \sqrt{\mathbf{tr}(\mathbf{A}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top)} \quad (1)$$

where $\mathbf{A}$ is a symmetric matrix of size $m \times m$, and **tr** stands for the *trace* operator. In general, $\mathbf{A}$ is a valid metric if and only if it satisfies the non-negativity and the triangle inequality properties. In other words, the matrix $\mathbf{A}$ must be positive semi-definite (PSD), i.e., $\mathbf{A} \succeq 0$. Generally, the matrix $\mathbf{A}$ parameterizes a family of Mahalanobis distances on the vector space $\mathbb{R}^m$. Specifically, when setting $\mathbf{A}$ to be an identity matrix $\mathbf{I}_{m \times m}$, the distance in Eqn. (1) reduces to the regular Euclidean distance. Note that Euclidean distance metric assumes all variables are independent, the variance across all dimensions is one and that covariances among all variables are zero, a scenario that is hardly achieved in real world. In practice, instead of adopting the regular Euclidean metric, it is important and more desirable to learn an optimal metric from the real data. To this end, we give a formal definition of distance metric learning below.

DEFINITION 1. *The distance metric learning (DML) problem is to learn an optimal distance metric, i.e. a matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$, from a collection of data points $\mathcal{C}$ in a vector space $\mathbb{R}^m$ together with a set of similar pairwise constraints $\mathcal{S}$ and a set of dissimilar pairwise constraints $\mathcal{D}$, which can be in general formulated into an optimization task below:*

$$\min_{\mathbf{A} \succeq 0} f(\mathbf{A}, \mathcal{S}, \mathcal{D}, \mathcal{C}) \quad (2)$$

where $\mathbf{A}$ is a positive semidefinite matrix and f is some objective function defined over the given data.

Given the above definition, the crux of solving the DML problem lies in how to formulate a proper objective function f and then find an efficient algorithm to solve the optimization. In the following subsections, we will discuss some principles for formulating appropriate optimization towards DML. We will then emphasize that it is important to avoid overfitting when solving a real DML problem.

3.2 A Regularization Learning Framework

One common principle for metric learning is to *minimize* the distances between the data points with similar constraints and meanwhile to *maximize* the distances between the data

points with dissimilar constraints. We refer it to a **min-max** learning principle. Some existing DML work can be interpreted within the min-max learning framework. For example, [Xing et al. 2002] formulated the DML problem as a convex optimization problem:

$$\min_{\mathbf{A}\succeq 0} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} \|\mathbf{x}_i-\mathbf{x}_j\|_{\mathbf{A}}^2 \quad \text{subject to} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}} \|\mathbf{x}_i-\mathbf{x}_j\|_{\mathbf{A}} \geq 1 \tag{3}$$

This formulation attempts to find the metric $\mathbf{A}$ by minimizing the sum of squared distances between the similar data points and meanwhile enforcing the sum of distances between the dissimilar data points larger than 1. Although the above method has been shown effective for some experimental tasks of artificial side information, it might not be suitable to solve real-world applications, such as CIR, where the log data could be rather noisy and quite limited at the beginning stage of system development. In practice, the above DML method is likely to overfit the log data in real-world applications.

To develop DML techniques for practical applications, the second principle we would like to highlight is the **regularization** principle, which is the key to enhancing the generalization and robustness performance of the distance metric in practical applications. Regularization has played an important role in many machine learning methods in order to prevent the overfitting issue [Girosi et al. 1995]. For example, in SVMs, regularization is critical to ensuring the excellent generalization performance [Vapnik 1998].

Similar to the idea of regularization used in kernel machine learning [Vapnik 1998], we formulate a general regularization framework for distance metric learning as follows:

$$\min_{\mathbf{A}\succeq 0}\ g(\mathbf{A}) + \gamma_s \mathcal{V}_s(\mathcal{S}) + \gamma_d \mathcal{V}_d(\mathcal{D}) \tag{4}$$

where $g(\mathbf{A})$ is a regularizer defined on the target metric $\mathbf{A}$, and $\mathcal{V}_s(\cdot)$ and $\mathcal{V}_d(\cdot)$ are some loss functions defined on the sets of similar and dissimilar constraints, respectively. γ_s and γ_d are two regularization parameters for balancing the tradeoff between similar and dissimilar constraints as well as the first regularization term. By following the min-max learning principle, the similar loss function $\mathcal{V}_s(\cdot)$ ($\mathcal{V}_d(\cdot)$) should be defined in the way such that the minimization of the loss function will result in minimizing (maximizing) the distances between the data points with the similar (dissimilar) constraints. We adopt the sum of squared distances for defining the two loss functions in terms of its effectiveness and efficiency:

$$\mathcal{V}_s(\cdot) = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} \|\mathbf{x}_i-\mathbf{x}_j\|_{\mathbf{A}}^2, \qquad \mathcal{V}_d(\cdot) = -\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}} \|\mathbf{x}_i-\mathbf{x}_j\|_{\mathbf{A}}^2 \tag{5}$$

Next, we will discuss how to select an appropriate regularizer and how to incorporate the unlabeled data information within the above regularization learning framework.

3.3 Laplacian Regularized Metric Learning

There are a lot of possible ways to choose a regularizer in the above regularization framework. One simple approach used in [Si et al. 2006] is based on the Frobenius norm defined as follows:

$$g(\mathbf{A}) = \|\mathbf{A}\|_{\mathrm{F}} = \sqrt{\sum_{i,j=1}^{m} a_{i,j}^2} \tag{6}$$

This regularizer simply prevents any elements within the matrix $\mathbf{A}$ from being overlarge. However, the regularizer does not take advantage of any unlabeled data information. In practice, the unlabeled data is beneficial to the DML task. By this consideration, we will show how to formulate a regularizer for exploiting the unlabeled data information in the regularization framework.

Consider the collection of n data points $\mathcal{C}$, we can compute a weight matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$ whose element W_{ij} is calculated as follows:

$$W_{ij} = \begin{cases} 1 & \mathbf{x}_i \in \mathcal{N}(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i) \\ 0 & \text{otherwise.} \end{cases}$$

where $\mathcal{N}(\mathbf{x}_j)$ denotes the nearest neighbor list of the data point $\mathbf{x}_j$ that is found by adopting regular Euclidean distance. To learn a metric, one can assume there is some corresponding linear mapping $\mathbf{U}^\top : \mathbb{R}^m \rightarrow \mathbb{R}^r$, where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_r] \in \mathbb{R}^{m \times r}$, for a possible metric $\mathbf{A}$. As a result, the distance between two input examples can be computed as:

$$d(\mathbf{x}_i, \mathbf{x}_j) \;=\; \|\mathbf{U}^\top(\mathbf{x}_i - \mathbf{x}_j)\|^2 = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{U}\mathbf{U}^\top(\mathbf{x}_i - \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{A}(\mathbf{x}_i - \mathbf{x}_j) \quad (7)$$

where $\mathbf{A} = \mathbf{U}\mathbf{U}^\top$ is the desirable metric to be learned. By taking unlabeled data information with the weight matrix $\mathbf{W}$, we can formulate the Laplacian regularizer as follows:

$$g(\mathbf{A}) \;=\; \frac{1}{2}\sum_{i,j=1}^{n} \left\|\mathbf{U}^\top \mathbf{x}_i - \mathbf{U}^\top \mathbf{x}_j\right\|^2 W_{ij} = \sum_{k=1}^{r} \mathbf{u}_k^\top \mathbf{X}(\mathbf{D} - \mathbf{W})\mathbf{X}^\top \mathbf{u}_k \quad (8)$$

$$= \sum_{k=1}^{r} \mathbf{u}_k^\top \mathbf{X}\mathbf{L}\mathbf{X}^\top \mathbf{u}_k = \mathbf{tr}(\mathbf{U}^\top \mathbf{X}\mathbf{L}\mathbf{X}^\top \mathbf{U}) = \mathbf{tr}(\mathbf{X}\mathbf{L}\mathbf{X}^\top \mathbf{U}\mathbf{U}^\top) = \mathbf{tr}(\mathbf{X}\mathbf{L}\mathbf{X}^\top \mathbf{A}) \quad (9)$$

where $\mathbf{D}$ is a diagonal matrix whose diagonal elements are calculated by $D_{ii} = \sum_j W_{ij}$, $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is known as the Laplacian matrix, and **tr** stands for the *trace* operator.

Remark. Regarding the graph Laplacian matrix, in practice, we often adopt the normalized laplacian matrix, which is computed as: $\tilde{\mathbf{L}} = \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-1/2}$.

By adopting the above Laplacian regularizer, we formulate a new distance metric learning technique called "Laplacian Regularized Metric Learning" (LRML) as follows:

$$\min_{\mathbf{A} \succeq 0} \quad \mathbf{tr}(\mathbf{X}\mathbf{L}\mathbf{X}^\top \mathbf{A}) + \gamma_s \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 - \gamma_d \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D}} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 \quad (10)$$

The above formulation can be further improved. In the extreme case, when the dissimilar factor $\gamma_d \rightarrow 0$, the above optimization will result in the trivial solution by shrinking the entire space, i.e. obtaining the solution of $\mathbf{A} = 0$. To prevent obtaining such undesirable results, we can modify the above formulation as follows:

$$\min_{\mathbf{A} \succeq 0} \quad \mathbf{tr}(\mathbf{X}\mathbf{L}\mathbf{X}^\top \mathbf{A}) + \gamma_s \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 - \gamma_d \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D}} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 \quad (11)$$

$$\text{s. t.} \quad \log\det(\mathbf{A}) \geq 0 \quad (12)$$

where the constraint $\log\det(\mathbf{A}) \geq 0$ is introduced to prevent trivial solutions. Note that choosing function $\log\det(\mathbf{A})$ is not unique; other types of regularizers may also be considered.

4. LRML FOR COLLABORATIVE IMAGE RETRIEVAL

4.1 Problem Formulation

We now show how to apply the proposed LRML technique to collaborative image retrieval and investigate its related optimization. Following the previous work in [Hoi et al. 2006; Si et al. 2006], we assume the log data were collected in the forms of *log sessions*, in which every log session corresponds to some particular user query. In each log session, a user first submits an image example to the CBIR system and then judges relevance on the top ranked images. The user relevance judgements will then be saved as the log data.

To apply the DML techniques for CIR, for each log session of user relevance feedback, we can convert it into similar and dissimilar pairwise constraints. Specifically, given a specific query q, for any two images $\mathbf{x}_i$ and $\mathbf{x}_j$, if they are marked as relevant in the log session, we will put them into the set of similar pairwise constraints $\mathcal{S}_q$; if one of them is marked as relevant, and the other is marked as irrelevant, we will put them into the set of dissimilar pairwise constraints. As a result, we denote the collection of user relevance feedback log data as $\mathcal{L} = \{(\mathcal{S}_q, \mathcal{D}_q), q = 1, \ldots, Q\}$, where Q is the number of log sessions in the log dataset. In the CIR context, we can modify the two loss functions and reformulate the LRML formulation as:

$$\min_{\mathbf{A}\succeq 0} \quad \mathbf{tr}(\mathbf{X}\mathbf{L}\mathbf{X}^\top\mathbf{A}) + \gamma_s \sum_{q=1}^{Q} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}_q} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 - \gamma_d \sum_{q=1}^{Q} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}_q} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 \quad (13)$$

$$\text{s. t.} \quad \log\det(\mathbf{A}) \geq 0$$

To solve the above optimization problem, we rewrite the two loss functions as follows:

$$\begin{aligned}\sum_{q=1}^{Q} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}_q} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 &= \sum_{q=1}^{Q} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}_q} \mathbf{tr}\Big(\mathbf{A}\cdot(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^\top\Big) \\ &= \mathbf{tr}\left(\mathbf{A}\cdot\sum_{q=1}^{Q}\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}_q}(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^\top\right) \quad (14)\end{aligned}$$

$$\begin{aligned}\sum_{q=1}^{Q} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}_q} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}^2 &= \sum_{q=1}^{Q} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}_q} \mathbf{tr}\Big(\mathbf{A}\cdot(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^\top\Big) \\ &= \mathbf{tr}\left(\mathbf{A}\cdot\sum_{q=1}^{Q}\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}_q}(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^\top\right) \quad (15)\end{aligned}$$

To simplify the above expressions, we introduce two matrices $\mathbf{S}$ and $\mathbf{D}$:

$$\mathbf{S} = \sum_{q=1}^{Q} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}_q} (\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^\top, \quad \mathbf{D} = \sum_{q=1}^{Q} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}_q} (\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^\top \quad (16)$$

Further, by introducing a slack variable t, we can rewrite the formulation as:

$$\min_{\mathbf{A}\succeq 0} \; t + \gamma_s \mathbf{tr}(\mathbf{A}\cdot\mathbf{S}) - \gamma_d \mathbf{tr}(\mathbf{A}\cdot\mathbf{D}) \quad (17)$$

$$s.t. \quad \mathbf{tr}(\mathbf{X}\mathbf{L}\mathbf{X}^\top\mathbf{A}) \leq t, \quad \log\det(\mathbf{A}) \geq 0 \quad (18)$$

The above optimization belongs to standard Semidefinite Programs (SDP) [Boyd and Vandenberghe 2003], which can be solved with global optimum by existing convex optimization packages, such as SeDuMi [Sturm 1999].

4.2 Fast LRML Algorithm

Solving the LRML problem by an SDP solver is feasible for a small-scale problem, but often becomes impractical when handing real applications, even for moderate-size datasets. This is because the time complexity of a general SDP solver can be as high as $\mathcal{O}(n^{6.5})$, which is clearly inefficient and not scalable for real applications. In this section, we present a simple and significantly more efficient algorithm, which can avoid engaging a general SDP solver in solving the LRML problem.

First of all, instead of enforcing the constraint $\log\det(\mathbf{A}) \geq 0$, we can consider an alternative formulation as follows:

$$\begin{aligned} &\min \ \mathbf{tr}(\mathbf{X}\mathbf{L}\mathbf{X}^\top\mathbf{A}) + \gamma_s\mathbf{tr}(\mathbf{A}\cdot\mathbf{S}) - \gamma_d\mathbf{tr}(\mathbf{A}\cdot\mathbf{D}) - \epsilon\log\det(\mathbf{A}) \\ &s.t. \quad \mathbf{A} \succeq 0 \end{aligned} \tag{19}$$

where $\epsilon \geq 0$ is a small constant, and a regularization term $\epsilon\log\det(\mathbf{A})$ is added into the objective function. It is easy to show that when $\epsilon \to 0$, the above optimization reduces to the equivalent optimization problem. Next we present an efficient technique to solve this optimization. In particular, we first introduce an important proposition as follows.

PROPOSITION 4.1. *Given a symmetric and positive-definite matrix $\mathbf{B} \succ 0$, the solution $\mathbf{A}^*$ to the following optimization:*

$$\min_{\mathbf{A}\succeq 0} \ \mathbf{tr}(\mathbf{A}\mathbf{B}) - \epsilon\log\det(\mathbf{A}) \tag{20}$$

can be expressed as follows:

$$\mathbf{A}^* = \epsilon\mathbf{B}^{-1} \tag{21}$$

PROOF. First of all, by introducing dual variables $\mathbf{Z} \in \mathcal{S}_+^n$ for the constraint $A \succeq 0$, we have the Lagrangian as follows:

$$\mathcal{L}(\mathbf{A};\mathbf{Z}) = \mathbf{tr}(\mathbf{A}\mathbf{B}) - \epsilon\log\det(\mathbf{A}) + \mathbf{tr}(\mathbf{A}\mathbf{Z}) \tag{22}$$

According to the Karush-Kuhn-Tucker (KKT) conditions [Kuhn 1982], we can derive the following equations:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{A}} = \mathbf{B} - \epsilon\mathbf{A}^{-1} + \mathbf{Z} \ = \ 0 \Rightarrow \mathbf{Z} = \epsilon\mathbf{A}^{-1} - \mathbf{B} \tag{23}$$

$$\mathbf{tr}(\mathbf{A}\mathbf{Z}) \ = \ 0 \tag{24}$$

Further, it is not difficult to show that $\mathbf{tr}(\mathbf{A}\mathbf{Z}) = \mathbf{0}$ is equivalent to $\mathbf{A}\mathbf{Z} = 0$. Specifically, given $\mathbf{A} \succeq 0$ and $\mathbf{Z} \succeq 0$, we have $\mathbf{tr}(\mathbf{A}\mathbf{Z}) = \mathbf{tr}(\mathbf{A}^{\mathbf{1/2}}\mathbf{A}^{\mathbf{1/2}}\mathbf{Z}^{\mathbf{1/2}}\mathbf{Z}^{\mathbf{1/2}}) = \|\mathbf{Z}^{\mathbf{1/2}}\mathbf{A}^{\mathbf{1/2}}\|_{\mathbf{F}}^{\mathbf{2}}$. Therefore, by $\mathbf{tr}(\mathbf{A}\mathbf{Z}) = \mathbf{0}$, we should have $\mathbf{Z}^{1/2}\mathbf{A}^{1/2} = 0$, which further leads to $\mathbf{A}\mathbf{Z}$ by multiplying by $\mathbf{Z}^{1/2}$ and $\mathbf{A}^{1/2}$ on both sides of the equation. Putting together with the result in (23), we can derive the equation: $\mathbf{A}\mathbf{B} = \epsilon\mathbf{I}$. Finally, combining it with the PSD constraint,i.e., $\mathbf{A} \succeq 0$, we thus have the solution as: $\mathbf{A}^* = \epsilon\mathbf{B}^{-1}$. □

Based on Proposition 4.1, we can apply it to solve the above optimization efficiently, which only involves simple matrix inversion. In particular, we can solve the optimization

in (19) by letting $\mathbf{B} = \mathbf{X}\mathbf{L}\mathbf{X}^\top + \gamma_s\mathbf{S} - \gamma_d\mathbf{D}$ and assuming that $B \succ 0$. Following Proposition 4.1, the optimal solution can be expressed as follows:

$$\mathbf{A}^* = \epsilon\Big(\mathbf{X}\mathbf{L}\mathbf{X}^\top + \gamma_s\mathbf{S} - \gamma_d\mathbf{D}\Big)^{-1} \tag{25}$$

In practice, the assumption that $B \succ 0$ may not always hold. To handle the non-positive definite issue, we suggest to add a regularization of an identity matrix, which results in the following solution:

$$\mathbf{A}^* = \epsilon\Big(\mathbf{X}\mathbf{L}\mathbf{X}^\top + \gamma_s\mathbf{S} - \gamma_d\mathbf{D} + \gamma_I\mathbf{I}_{m\times m}\Big)^{-1} \tag{26}$$

where γ_I is a regularization parameter of an identity matrix $\mathbf{I}_{m\times m}$. We note that the resulting solution in this situation is sub-optimal to the original optimization problem.

Remark I. Regarding the solutions in (25) and (26), the parameter ϵ generally should be a small constant. However, since scaling does not affect the performance of distance metric learning, we can simply fix ϵ to 1 for metric learning tasks in practice.

Remark II. The above result enjoys some interesting connections to the solution of relevant component analysis (RCA) [Bar-Hillel et al. 2005], in which the optimal metric learned by RCA is $\mathbf{A} = \mathbf{C}^{-1}$, where $\mathbf{C}$ is the chunklet average covariance matrix. Similarly, for the result in Eq.(26), if we set $\gamma_d = 0$ and ignore the regularizer of unlabeled data, the solution reduces to $\mathbf{S}^{-1}$, which is essentially equivalent to RCA by noting $\mathbf{S} \approx \mathbf{C}$ (RCA forms chunklets while we do not use). Therefore, RCA can be viewed as a special case of the proposed semi-supervised DML technique without considering dissimilar constraints and unlabeled data.

4.3 Complexity Analysis

In this part, we analyze the computational complexity of the proposed LRML algorithms. We denote by LRML$^{\mathrm{SDP}}$ the proposed LRML method solved by a general SDP solver, and denote by LRML$^{\mathrm{INV}}$ the proposed fast LRML method solved by simple matrix inversion.

First, in terms of space complexity, both algorithms have the worst case complexity of $\mathcal{O}(n^2)$, where n is the dataset size. The major space is used for storing the matrices W and L when computing the graph Laplacian, and the matrices S and D when computing the pairwise similar and dissimilar matrices.

Second, in terms of time complexity, LRML$^{\mathrm{INV}}$ is significantly more efficient and scalable than LRML$^{\mathrm{SDP}}$. This is because the time complexity of a general SDP solver based on the interior-point approach can be high of $\mathcal{O}(m^{6.5})$ [Sturm 1999], while the LRML$^{\mathrm{INV}}$ algorithm often involves simple matrix computation for matrix inversions, leading to the worst time complexity of $\mathcal{O}(m^3)$, where m is the matrix dimension.

5. LRML WITH APPLICATION TO COLLABORATIVE IMAGE CLUSTERING

In this section, we investigate the proposed DML learning technique for another application in multimedia data mining, which aims to discover image cluster patterns from image databases by exploring the historical log data of users' relevance feedback in CBIR. We refer to such an image clustering scheme as "Collaborative Image Clustering" (CIC) that utilizes the users' log data information in improving clustering performance. The CIC scheme can be beneficial to a lot of real applications by discovering the cluster patterns. For example, it can help to enhance image browsing experience and improve the retrieval quality for an image retrieval system.

Most clustering techniques require an effective distance metric to measure distance (dissimilarity) between data examples. For example, conventional k-means clustering often adopts a simple Euclidean metric for distance measure, which, however, is not always effective for real problems. Applying distance metric learning techniques to existing clustering algorithms has been explored in literature [Xing et al. 2002]. Below, we introduce a clustering technique by exploiting side information in extending the popular k-means algorithm, which is known as the constrained k-means algorithm, denoted by "CKmeans" for short.

K-means is a well-known and efficient clustering algorithm, which assigns n data examples into k clusters by some iterative refinement processes [Jain et al. 1999]. In particular, a typical k-means algorithm starts by defining k initial centroids, and then repeats an iterative refinement procedure until convergence is achieved. For each step, every example is assigned to its closest centroid based on some distance measure such as Euclidean distance, and the means of the updated clusters are refined in every step.

The idea of the constrained k-means algorithm is two-fold: (1) replacing the Euclidean metric by the metric learned by the proposed LRML technique; and (2) enforcing certain pairwise examples to be grouped in the same cluster when they are linked by similar (must-link) constraints. Similar to [Xing et al. 2002], only similar pairwise constraints are enforced during clustering. Finally, Figure 1 summarizes the constrained k-means clustering algorithm using LRML for collaborative image clustering.

Algorithm 1: Constrained k-means algorithm for collaborative image clustering
Input
$X = \{\mathbf{x}_i\}_{i=1}^N$ - a collection of image examples
$\mathcal{S}, \mathcal{D}$ - sets of pairwise similar and dissimilar constraints from feedback log data
k - a predefined number of clusters
Output
$\{\mathcal{C}_i\}_{i=1}^k$ - partitions of clustering results
Procedure

(1) Compute Laplacian matrix from the image database: $L = \mathsf{laplacian}(X)$;
(2) Learn the metric by the proposed LRML method: $\mathbf{A} = \mathbf{LRML}(X, L, \mathcal{S}, \mathcal{D})$;
(3) Initialize a random partition: $\{\mathcal{C}_i\}_{i=1}^k$;
(4) Update assignments until convergence:
  (a) For each $\mathbf{x}_j$, assign $\mathbf{x}_j \rightarrow \mathcal{C}_p$, where $p = \arg\min_i \|\mathbf{x}_j - \mu_i\|_{\mathbf{A}}$ and $\mu_i = \frac{1}{|\mathcal{C}_i|} \sum_{\mathbf{x} \in \mathcal{C}_i} \mathbf{x}$
  (b) For each $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S} \quad (i \leq j)$, if $\mathcal{C}(\mathbf{x}_i) \neq \mathcal{C}(\mathbf{x}_j)$, then assign $\mathbf{x}_j \rightarrow \mathcal{C}(\mathbf{x}_i)$.

End of Algorithm

Fig. 1. The constrained K-means clustering algorithm for collaborative image clustering.

6. EXPERIMENTAL RESULTS

6.1 Overview

In our experiments, we evaluate the effectiveness of LRML for both CIR and CIC applications. We design the experiments for performance evaluation in several aspects. First of all, we extensively compare it with a number of state-of-the-art DML techniques. Second, we carefully examine if the proposed algorithms are effective to learn reliable metrics by exploiting unlabeled data for limited log data. Third, we study if the proposed algorithms are robust to large noisy log data. Finally, we evaluate the computational efficiency.

6.2 Experimental Testbed

We employed a standard CBIR testbed used in [Hoi et al. 2006]. The image testbed consists of real-world photos from COREL image CDs. It has two datasets: 20-Category (20-Cat) that includes images from 20 different categories, and 50-Category (50-Cat) that includes images from 50 different categories. Each category contains exactly 100 images that are randomly selected from relevant examples in the COREL image CDs. Every category represents certain semantic topic, such as *antelope*, *balloon*, *butterfly*, *car*, *cat*, *dog*, and *horse*, etc. The way of using the images with semantic categories is able to help us to evaluate the retrieval performance automatically, which significantly reduces the subjective errors relative to manual evaluations.

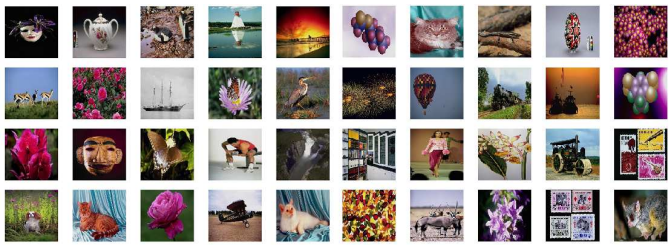


Fig. 2. Some image examples from the datasets used in our experiments.

6.3 Image Representation

Image representation is an important step for building a CBIR system. In our experiment, we employ three types of visual features to represent the images: color, edge and texture.

Color features are widely adopted for their simplicity. The color feature in our experiments is color moment, which is close to natural human perception and whose effectiveness has been shown in many previous CBIR studies. Three different color moments are used: color mean, color variance and color skewness in each color channel (H, S, and V), respectively. Thus, a 9-dimensional color moment is adopted as the color feature.

Edge features can be effective for CBIR when contour lines of images are evident. The edge feature in our experiments is edge direction histogram [Jain and Vailaya 1998]. In our approach, an image is first translated to a gray image, and a Canny edge detector is applied to obtain its edge image. Based on the edge image, the edge direction histogram can be computed. Each edge direction histogram is quantized into 18 bins of 20 degrees each. Hence an 18-dimensional edge direction histogram is used as the edge feature.

Texture features are proven to be an important cue for CBIR. In our experiments, we employ the wavelet-based texture [Manjunath et al. 2001]. A color image is first transformed to a gray image. Then the Discrete Wavelet Transformation (DWT) is performed on the gray image using a Daubechies-4 wavelet filter. Each wavelet decomposition on a gray 2D-image results in four subimages with a $0.5 * 0.5$ scaled-down image of the input image and the wavelets in three orientations: horizontal, vertical and diagonal. The scaled-down image is then fed into the DWT to produce the next four subimages. In total, we perform a 3-level decomposition, which produces 10 subimages in different scales and orientations. Among nine of the subimages, we compute the entropy of each subimage separately. Hence, a wavelet-based texture feature of 9 dimensions is used to describe the texture information.

In sum, a 36-dimensional feature vector is used to represent an image, including 9-dimensional color histogram, 18-dimensional edge direction histogram, and 9-dimensional wavelet-based texture.

6.4 Real Log Data of User Relevance Feedback

In our experiments, we adopt the real log data related to the COREL testbed collected by a real CBIR system with an interactive relevance feedback mechanism from [Hoi et al. 2006]. In the log data collection, there are two sets of relevance feedback logs. One is a set of normal log data, which contains small noise. The other is a set of noisy log data of relatively large noise. For log data, a *log session* is defined as the basic unit. Each log session corresponds to a regular relevance feedback session, in which 20 images were judged by a user. Thus, each log session contains 20 labeled images that are marked as either "relevant (positive)" or "irrelevant (negative)".

Regarding the noise of the log data, it is mainly caused by subjective judgments from human subjects. Given the fact that different users may have different opinions on judging the same image, the noise problem in collaborative image retrieval is almost inevitable in real applications. According to the previous study [Hoi et al. 2006], the noise of log data is measured by its percentage of incorrect relevance judgments P_{noise}, i.e.,$P_{noise} = \frac{\text{tot \# wrong judgements}}{N_l \times N_{log}} \times 100\%$, where N_l and N_{log} stand for the number of labeled examples acquired for each log session and the number of log sessions, respectively.

Table I. The log data collected from users on both datasets

Datasets	Normal Log		Noisy Log	
	#Log Sessions	Noise	# Log Sessions	Noise
20-Cat	100	7.8%	100	16.2%
50-Cat	150	7.7%	150	17.1%

Finally, Table I shows the information of the log data on the two testbeds. More details on the collection of the users' relevance feedback log data can be found [Hoi et al. 2006].

6.5 Compared Methods and Experimental Setup

We compare the proposed LRML method extensively with two groups of major metric learning techniques: unsupervised approaches and metric learning with side information. We do not compare the DML techniques for supervised classification as they often require explicit class labels, which is unsuitable for CIR. Although it may be unfair to directly compare the unsupervised methods with supervised/semi-supervised metric learning using side information, we still include the unsupervised results. The results could help us examine how effective is the proposed method compared with traditional approaches since there was still limited comprehensive study for applying DML in CIR before. Specifically, the compared schemes include:

—Euclidean: the baseline denoted as "EU" in short.

—Mahalanobis: a standard Mahalanobis metric, $\mathbf{A} = \mathbf{P}^{-1}$, where $\mathbf{P}$ is the covariance matrix, denoted as "Mah" in short.

—PCA: classical PCA method [Fukunaga 1990]. For all unsupervised methods, the number of reduced dimensions r is set to 15 in all experiments.

—MDS: classical Multidimensional Scaling method [Cox and Cox 1994].

—Isomap: unsupervised method finding low-dimensional manifolds with geometrical information [Tenenbaum and de Silva andJohn C. Langford 2000].

—LLE: unsupervised method computing low-dimensional and neighborhood-preserving embeddings [Roweis and Saul 2000].

—DML: a popular DML method solving by an iterative convex optimization technique [Xing et al. 2002].

—RCA: relevant component analysis [Bar-Hillel et al. 2005], which learns with only equivalent constraints.

—DCA: discriminative component analysis, which improves RCA by including dissimilar constraints [Hoi et al. 2006].

—RML: regularized metric learning algorithm with the Frobenius norm as the regularizer [Si et al. 2006].

—LRML$^{\mathrm{SDP}}$: the proposed Laplacian Regularized Metric Learning method by an SDP based algorithm.

—LRML$^{\mathrm{INV}}$: the proposed Laplacian Regularized Metric Learning method solved by the matrix inversion based algorithm.

In sum, the compared schemes include 2 standard metrics, 3 unsupervised metrics, 4 supervised DML, and 2 variants of the proposed semi-supervised DML method.

Regarding the setup of our experiments, we follow a standard procedure for CBIR experiments. Specifically, a query image is picked from the database and then queried with the evaluated distance metric. The retrieval performance is then evaluated based on the top ranked images ranging from top 10 to top 100 images. The average precision (AP) and mean average precision (MAP) are engaged as the performance metrics, which are widely used in CBIR experiments. For the implementation of the proposed LRML algorithm, we use a standard method for computing a normalized Laplacian matrix with 6 nearest neighbors.

Table II. Average precision of top ranked images on the 20-Category testbed over 2,000 queries with the *normal* log data. For each scheme, the first row shows the AP (%) and the second row shows the relative improvement over the baseline (Euclidean) method.

TOP	20	30	40	50	60	70	80	90	100	MAP
EU	39.91	35.62	32.73	30.55	28.84	27.53	26.40	25.39	24.44	31.93
Mah	40.24 + 0.8 %	35.22 -1.1 %	31.52 -3.7 %	28.85 -5.6 %	26.71 -7.4 %	24.94 -9.4 %	23.42 -11.3 %	22.19 -12.6 %	21.09 -13.7 %	30.36 -4.9 %
PCA	39.50 -1.0 %	35.33 -0.8 %	32.57 -0.5 %	30.45 -0.3 %	28.76 -0.3 %	27.44 -0.3 %	26.32 -0.3 %	25.35 -0.2 %	24.42 -0.1 %	31.76 -0.5 %
MDS	39.80 -0.3 %	35.69 + 0.2 %	32.85 + 0.4 %	30.63 + 0.3 %	28.90 + 0.2 %	27.61 + 0.3 %	26.47 + 0.3 %	25.47 + 0.3 %	24.50 + 0.2 %	31.99 + 0.2 %
LLE	31.52 -21.0 %	28.43 -20.2 %	26.26 -19.8 %	24.67 -19.2 %	23.40 -18.9 %	22.34 -18.9 %	21.46 -18.7 %	20.68 -18.6 %	19.87 -18.7 %	25.72 -19.4 %
Isomap	27.34 -31.5 %	23.74 -33.4 %	21.52 -34.2 %	20.04 -34.4 %	18.92 -34.4 %	18.04 -34.5 %	17.23 -34.7 %	16.56 -34.8 %	15.88 -35.0 %	21.38 -33.0 %
XING	40.85 + 2.4 %	36.86 + 3.5 %	34.26 + 4.7 %	32.22 + 5.5 %	30.51 + 5.8 %	29.05 + 5.5 %	27.74 + 5.1 %	26.64 + 4.9 %	25.61 + 4.8 %	33.23 + 4.1 %
RCA	43.16 + 8.1 %	38.41 + 7.8 %	35.19 + 7.5 %	32.70 + 7.0 %	30.64 + 6.2 %	29.01 + 5.4 %	27.56 + 4.4 %	26.21 + 3.2 %	24.96 + 2.1 %	33.94 + 6.3 %
DCA	44.11 + 10.5 %	39.24 + 10.2 %	35.95 + 9.8 %	33.36 + 9.2 %	31.27 + 8.4 %	29.58 + 7.4 %	28.13 + 6.6 %	26.81 + 5.6 %	25.51 + 4.4 %	34.66 + 8.6 %
RML	43.80 + 9.7 %	39.46 + 10.8 %	36.37 + 11.1 %	34.06 + 11.5 %	32.33 + 12.1 %	30.74 + 11.7 %	29.45 + 11.6 %	28.26 + 11.3 %	27.20 + 11.3 %	35.38 + 10.8 %
LRML$^{\mathrm{SDP}}$	46.51 + 16.5 %	42.03 + 18.0 %	38.71 + 18.3 %	36.18 + 18.4 %	34.05 + 18.1 %	32.44 + 17.8 %	30.95 + 17.2 %	29.66 + 16.8 %	28.36 + 16.0 %	37.38 + 17.1 %
LRML$^{\mathrm{INV}}$	46.86 **+ 17.4 %**	42.12 **+ 18.2 %**	38.87 **+ 18.8 %**	36.37 **+ 19.1 %**	34.23 **+ 18.7 %**	32.58 **+ 18.3 %**	31.11 **+ 17.8 %**	29.82 **+ 17.4 %**	28.52 **+ 16.7 %**	37.54 **+ 17.6 %**

Table III. Average precision of top ranked images on the 50-Category testbed over 5,000 queries with the *normal* log data. For each scheme, the first row shows the AP (%) and the second row shows the relative improvement over the baseline (Euclidean) method.

TOP	20	30	40	50	60	70	80	90	100	MAP
EU	36.29	31.93	28.90	26.68	24.90	23.43	22.15	21.06	20.13	27.99
Mah	37.32 + 2.8 %	32.39 1.4 %	29.00 0.3 %	26.52 -0.6 %	24.50 -1.6 %	22.89 -2.3 %	21.49 -3.0 %	20.33 -3.5 %	19.30 -4.1 %	28.02 0.1 %
PCA	35.33 -2.6 %	31.24 -2.2 %	28.29 -2.1 %	26.17 -1.9 %	24.50 -1.6 %	23.08 -1.5 %	21.84 -1.4 %	20.79 -1.3 %	19.87 -1.3 %	27.44 -2.0 %
MDS	36.01 -0.8 %	31.77 -0.5 %	28.80 -0.3 %	26.61 -0.3 %	24.86 -0.2 %	23.38 -0.2 %	22.13 -0.1 %	21.04 -0.1 %	20.10 -0.1 %	27.87 -0.4 %
LLE	26.01 -28.3 %	22.24 -30.3 %	19.79 -31.5 %	18.09 -32.2 %	16.79 -32.6 %	15.75 -32.8 %	14.88 -32.8 %	14.14 -32.9 %	13.48 -33.0 %	19.52 -30.3 %
Isomap	25.35 -30.1 %	22.09 -30.8 %	20.01 -30.8 %	18.50 -30.7 %	17.27 -30.6 %	16.33 -30.3 %	15.47 -30.2 %	14.74 -30.0 %	14.10 -30.0 %	19.60 -30.0 %
XING	37.98 + 4.7 %	33.91 + 6.2 %	31.14 + 7.8 %	29.02 + 8.8 %	27.25 + 9.4 %	25.80 + 10.1 %	24.54 + 10.8 %	23.41 + 11.2 %	22.44 + 11.5 %	30.12 + 7.6 %
RCA	40.84 + 12.5 %	36.05 + 12.9 %	32.67 + 13.0 %	30.05 + 12.6 %	27.98 + 12.4 %	26.23 + 12.0 %	24.74 + 11.7 %	23.47 + 11.4 %	22.33 + 10.9 %	31.41 + 12.2 %
DCA	41.28 + 13.8 %	36.42 + 14.1 %	33.00 + 14.2 %	30.37 + 13.8 %	28.25 + 13.5 %	26.51 + 13.1 %	25.00 + 12.9 %	23.69 + 12.5 %	22.56 + 12.1 %	31.72 + 13.3 %
RML	41.90 + 15.5 %	37.20 + 16.5 %	33.86 + 17.2 %	31.19 + 16.9 %	29.09 + 16.8 %	27.33 + 16.6 %	25.80 + 16.5 %	24.46 + 16.1 %	23.29 + 15.7 %	32.47 + 16.0 %
LRML$^{\mathrm{SDP}}$	42.70 **+ 17.7 %**	37.96 **+ 18.9 %**	34.43 + 19.1 %	31.82 + 19.3 %	29.72 + 19.4 %	27.94 + 19.2 %	26.43 + 19.3 %	25.10 + 19.2 %	23.91 + 18.8 %	33.13 + 18.4 %
LRML$^{\mathrm{INV}}$	42.62 + 17.4 %	37.93 + 18.8 %	34.49 **+ 19.3 %**	31.86 **+ 19.4 %**	29.77 **+ 19.6 %**	28.02 **+ 19.6 %**	26.51 **+ 19.7 %**	25.20 **+ 19.7 %**	24.02 **+ 19.3 %**	33.13 **+ 18.4** %

6.6 Experiment I: Evaluation on Normal Log Data

For of all, we evaluate the compared schemes on the normal log data. This is to examine if the proposed algorithm is comparable or better than the previous DML techniques in a normal situation. Table II shows the experimental results on the 20-category testbed averaging over 2,000 queries with the normal log data. From the results, we can draw several observations. Firstly, we found that a simple Mahalanobis distance does not always outperform Euclidean distance. In fact, it only improved slightly on top 10 and top 20 ranked images, but failed to obtain improvements on other cases. Secondly, comparing with several unsupervised methods, it is interesting to find that only the MDS method achieved a marginal improvement over the baseline. Two manifold based unsupervised methods performed very poor in this retrieval task. Further, comparing several previous DML methods with the normal log data, the RML method achieved the best overall performance, which obtained 10.8% improvement on MAP over the baseline. The RCA performed the worst among the four compared methods. Finally, comparing with all the metrics, the proposed LRML method achieved the best performance, which significantly improves the baseline with about 17% improvement on MAP. This shows that the proposed method is more effective than the previous methods with normal log data. We also conducted the same comparisons on the 50-category dataset with normal log data.

Table III shows the results on the 50-category dataset, which were obtained by averaging over 5,000 queries. Similar to the previous results, most unsupervised methods fail to improve the retrieval performance compared with the baseline Euclidean approach. Among all the compared DML methods, the proposed LRML methods, including LRML$^{\mathrm{SDP}}$ and LRML$^{\mathrm{INV}}$, achieved the best performance. Compared with the two semi-supervised methods, LRML$^{\mathrm{SDP}}$ tends to achieve slightly better results on the top ranked results, while LRML$^{\mathrm{INV}}$ tends to obtain better results when returning more than top 30 ranked images.

In addition, we found that Xing's method did not perform well in this situation. One possible reason is that the regular DML method might be too sensitive to noise. To evaluate the robustness comprehensively, in the subsequent sections, we will conduct experiments on two tough situations: (1) small amount of log data, and (2) large noisy log data.

6.7 Experiment II: Evaluation on Small Log Data

In this experiment, we evaluate the robustness performance for learning metrics with small amount of normal log data. This situation usually occurs at the beginning stage of developing a CBIR system. Table IV shows the experimental results on the 20-Category testbed with a small subset of normal log data containing only 30 log sessions which were randomly selected from the normal log dataset. From the results, we can see that most supervised DML methods achieved considerably lower improvements compared with their results obtained on the relatively large amount of log data in the previous situation. Among all the four compared supervised metric learning methods, the RML method achieves the best performance, which achieved around 5% improvement over the baseline. Further, when comparing with the semi-supervised DML methods, we observe that the two LRML algorithms significantly outperform the other supervised DML approaches. For example, the improvement achieved by the proposed LRML$^{\mathrm{INV}}$ algorithm almost doubles that achieved by the RML method. This again shows that the proposed method is more effective to learn better metrics by engaging unlabeled data, particularly for limited log data.

Table IV. Average precision of top ranked images on the 20-Category testbed over 2,000 queries with *small* log data of only 30 log sessions. For each scheme, the first row shows the AP (%) and the second row shows the relative improvement over the baseline (Euclidean).

TOP	20	30	40	50	60	70	80	90	100	MAP
EU	39.91	35.62	32.73	30.55	28.84	27.53	26.40	25.39	24.44	31.93
MAH	40.24 + 0.8 %	35.22 -1.1 %	31.52 -3.7 %	28.85 -5.6 %	26.71 -7.4 %	24.94 -9.4 %	23.42 -11.3 %	22.19 -12.6 %	21.09 -13.7 %	30.36 -4.9 %
XING	40.17 0.7 %	36.26 + 1.8 %	33.54 + 2.5 %	31.52 + 3.2 %	29.89 + 3.6 %	28.55 + 3.7 %	27.44 + 3.9 %	26.38 + 3.9 %	25.36 + 3.8 %	32.69 + 2.4 %
RCA	42.41 + 6.3 %	37.78 + 6.1 %	34.54 + 5.5 %	32.20 + 5.4 %	30.22 + 4.8 %	28.57 + 3.8 %	27.11 + 2.7 %	25.79 + 1.6 %	24.62 + 0.7 %	33.41 + 4.6 %
DCA	41.38 + 3.7 %	37.13 + 4.2 %	34.29 + 4.8 %	32.00 + 4.7 %	30.29 + 5.0 %	28.95 + 5.2 %	27.77 + 5.2 %	26.71 + 5.2 %	25.70 + 5.2 %	33.34 + 4.4 %
RML	42.16 + 5.6 %	37.69 + 5.8 %	34.69 + 6.0 %	32.37 + 6.0 %	30.53 + 5.9 %	29.13 + 5.8 %	27.91 + 5.7 %	26.84 + 5.7 %	25.78 + 5.5 %	33.72 + 5.6 %
LRML$^{\mathrm{SDP}}$	44.03 **+ 10.3 %**	39.41 **+ 10.6 %**	36.17 + 10.5 %	33.75 + 10.5 %	31.74 + 10.1 %	30.08 + 9.3 %	28.69 + 8.7 %	27.42 + 8.0 %	26.27 + 7.5 %	35.01 + 9.7 %
LRML$^{\mathrm{INV}}$	43.56 + 9.1 %	39.29 + 10.3 %	36.26 **+ 10.8 %**	33.93 **+ 11.1 %**	32.11 **+ 11.3 %**	30.53 **+ 10.9 %**	29.19 **+ 10.6 %**	27.96 **+ 10.1 %**	26.78 **+ 9.6 %**	35.15 **+ 10.1** %

Similarly, we also evaluated the small log data case on the 50-Category testbed with only 50 log sessions, as shown in Table V. The relative improvements by the LRML methods over the RML method in this dataset are less significant than the 20-Category case, but the proposed two semi-supervised algorithms still achieved the best improvement, which are considerably better than other compared metric learning schemes.

Table V. Average precision of top ranked images on the 50-Category testbed over 5,000 queries with *small* log data of only 50 log sessions. For each scheme, the first row shows the AP (%) and the second row shows the relative improvement over the baseline (Euclidean).

TOP	20	30	40	50	60	70	80	90	100	MAP
EU	36.29	31.93	28.90	26.68	24.90	23.43	22.15	21.06	20.13	27.99
MAH	37.32 + 2.8 %	32.39 + 1.4 %	29.00 + 0.3 %	26.52 -0.6 %	24.50 -1.6 %	22.89 -2.3 %	21.49 -3.0 %	20.33 -3.5 %	19.30 -4.1 %	28.02 + 0.1 %
Xing	36.29 0.00 %	31.92 -0.03 %	28.90 0.00 %	26.68 0.00 %	24.90 0.00 %	23.42 -0.04 %	22.15 0.00 %	21.05 -0.05 %	20.12 -0.05 %	27.99 -0.02 %
RCA	39.81 + 9.7 %	35.02 + 9.7 %	31.63 + 9.4 %	29.07 + 9.0 %	27.02 + 8.5 %	25.28 + 7.9 %	23.87 + 7.8 %	22.63 + 7.5 %	21.54 + 7.0 %	30.47 + 8.8 %
DCA	38.58 + 6.3 %	34.12 + 6.9 %	30.98 + 7.2 %	28.59 + 7.2 %	26.59 + 6.8 %	25.00 + 6.7 %	23.62 + 6.6 %	22.43 + 6.5 %	21.40 + 6.3 %	29.84 + 6.6 %
RML	39.26 + 8.2 %	34.43 + 7.8 %	31.08 + 7.5 %	28.57 + 7.1 %	26.58 + 6.7 %	24.88 + 6.2 %	23.48 + 6.0 %	22.25 + 5.7 %	21.22 + 5.4 %	30.00 + 7.2 %
LRML$^{\mathrm{SDP}}$	40.49 **+ 11.6 %**	35.68 **+ 11.7 %**	32.22 + 11.5 %	29.61 + 11.0 %	27.52 + 10.5 %	25.83 + 10.2 %	24.32 + 9.8 %	23.05 + 9.4 %	21.93 + 8.9 %	31.00 + 10.7 %
LRML$^{\mathrm{INV}}$	39.99 + 10.2 %	35.38 + 10.8 %	32.06 **+ 10.9 %**	29.53 **+ 10.7 %**	27.55 **+ 10.6 %**	25.83 **+ 10.2 %**	24.38 **+ 10.1 %**	23.11 **+ 9.7 %**	22.00 **+ 9.3 %**	30.86 **+ 10.3** %

6.8 Experiment III: Evaluation on Noisy Log Data

To further validate the robustness performance, the third experiment is to evaluate the compared schemes with noisy log data of relatively large noise. Table VI and Table VII show the results on the 20-Category and 50-Category testbeds with the log data of large noise, respectively. We can draw some observations from the results as follows.

Table VI. Average precision of top ranked images on the 20-Category testbed over 2,000 queries with *noisy* log data of 100 log sessions. For each scheme, the first row shows the AP (%) and the second row shows the relative improvement over the baseline (Euclidean).

TOP	20	30	40	50	60	70	80	90	100	MAP
EU	39.91	35.62	32.73	30.55	28.84	27.53	26.40	25.39	24.44	31.93
MAH	40.24 + 0.8 %	35.22 -1.1 %	31.52 -3.7 %	28.85 -5.6 %	26.71 -7.4 %	24.94 -9.4 %	23.42 -11.3 %	22.19 -12.6 %	21.09 -13.7 %	30.36 -4.9 %
XING	39.87 -0.1 %	35.56 -0.2 %	32.70 -0.1 %	30.52 -0.1 %	28.82 -0.1 %	27.49 -0.1 %	26.37 -0.1 %	25.36 -0.1 %	24.41 -0.1 %	31.90 -0.1 %
RCA	42.59 + 6.7 %	37.75 6.0 %	34.45 5.3 %	32.00 4.7 %	30.00 4.0 %	28.31 2.8 %	26.97 2.2 %	25.69 1.2 %	24.45 0.0 %	33.34 4.4 %
DCA	43.60 + 9.2 %	38.66 + 8.5 %	35.33 + 7.9 %	32.86 + 7.6 %	30.84 + 6.9 %	29.17 + 6.0 %	27.84 + 5.5 %	26.58 + 4.7 %	25.37 + 3.8 %	34.26 + 7.3 %
RML	42.21 + 5.8 %	37.92 + 6.5 %	35.01 + 7.0 %	32.76 + 7.2 %	30.99 + 7.5 %	29.54 + 7.3 %	28.34 + 7.3 %	27.30 + 7.5 %	26.30 + 7.6 %	34.09 + 6.8 %
LRML$^{\mathrm{SDP}}$	45.95 **+ 15.1 %**	41.07 **+ 15.3 %**	37.85 **+ 15.6 %**	35.37 **+ 15.8 %**	33.43 **+ 15.9 %**	31.83 **+ 15.6 %**	30.40 **+ 15.2 %**	29.15 **+ 14.8 %**	27.89 **+ 14.1 %**	36.69 **+ 14.9 %**
LRML$^{\mathrm{INV}}$	45.55 + 14.1 %	40.88 + 14.8 %	37.67 + 15.1 %	35.17 + 15.1 %	33.14 + 14.9 %	31.53 + 14.5 %	30.16 + 14.2 %	29.00 + 14.2 %	27.75 + 13.5 %	36.49 + 14.3 %

Table VII. Average precision of top ranked images on the 50-Category testbed over 5,000 queries with *noisy* log data of 150 log sessions. For each scheme, the first row shows the AP (%) and the second row shows the relative improvement over the baseline (Euclidean).

TOP	20	30	40	50	60	70	80	90	100	MAP
EU	36.29	31.93	28.90	26.68	24.90	23.43	22.15	21.06	20.13	27.99
MAH	37.32 + 2.8 %	32.39 + 1.4 %	29.00 + 0.3 %	26.52 -0.6 %	24.50 -1.6 %	22.89 -2.3 %	21.49 -3.0 %	20.33 -3.5 %	19.30 -4.1 %	28.02 + 0.1 %
XING	36.25 -0.1 %	31.88 -0.2 %	28.85 -0.2 %	26.64 -0.1 %	24.86 -0.2 %	23.39 -0.2 %	22.11 -0.2 %	21.02 -0.2 %	20.08 -0.2 %	27.95 -0.2 %
RCA	39.00 + 7.5 %	34.24 + 7.2 %	30.99 + 7.2 %	28.44 + 6.6 %	26.38 5.9 %	24.69 + 5.4 %	23.27 + 5.1 %	22.06 + 4.7 %	21.03 + 4.5 %	29.79 + 6.4 %
DCA	39.53 + 8.9 %	34.66 + 8.5 %	31.32 + 8.4 %	28.75 + 7.8 %	26.69 + 7.2 %	25.01 + 6.7 %	23.55 + 6.3 %	22.32 + 6.0 %	21.26 + 5.6 %	30.14 + 7.7 %
RML	40.34 + 11.2 %	35.60 + 11.5 %	32.35 + 11.9 %	29.74 + 11.5 %	27.75 + 11.4 %	26.01 + 11.0 %	24.55 + 10.8 %	23.27 + 10.5 %	22.15 + 10.0 %	31.08 + 11.0 %
LRML$^{\mathrm{SDP}}$	41.58 **+ 14.6 %**	37.11 **+ 16.2 %**	33.86 **+ 17.2 %**	31.25 **+ 17.1 %**	29.21 **+ 17.3 %**	27.50 **+ 17.4 %**	26.01 **+ 17.4 %**	24.74 **+ 17.5 %**	23.59 **+ 17.2 %**	32.48 **+ 16.0 %**
LRML$^{\mathrm{INV}}$	41.28 + 13.8 %	36.76 + 15.1 %	33.43 + 15.7 %	30.86 + 15.7 %	28.81 + 15.7 %	27.09 + 15.6 %	25.62 + 15.7 %	24.31 + 15.4 %	23.18 + 15.2 %	32.10 + 14.7 %

First of all, we found that the performance of most supervised DML methods were considerably degraded when being tested on the noisy data situation when comparing with the normal data situation. Further, we found that the Xing's DML method failed to improve over the baseline method due to the noise problem. The results validated our previous conjecture that the Xing's DML method may be too sensitive to noise. Compared with the Xing's method, the other three DML methods including RCA, DCA and RML are relatively less sensitive to noise, but they still suffered a lot from the noise. For example, on the 50-Category dataset, RCA achieved 12.2% improvement on MAP with the normal log data as shown in Table IV, but only achieved 6.4% improvement on MAP with the same amount of log data of larger noise as shown in Table VII. In contrast, for the same dataset, the proposed LRML$^{\mathrm{SDP}}$ method achieved 18.4% improvement on MAP with normal log data, and is still able to keep 16.0% improvement on MAP with the larger noisy log data without too much dropping. Similarly, the two proposed semi-supervised algorithm performed similarly, which are considerably less sensitive to the noise.

All of the above results again validate that the proposed LRML method is effective to learn reliable metrics on real noisy log data by exploiting unlabeled data information.

6.9 Qualitative comparisons of CIR performance

In addition to the above quantitative results, we also include some experimental results for qualitatively evaluating the visual retrieval performance by different metric learning methods. Figure 5 to Figure 7 (in the last two pages) show the results of visual comparison for several different query cases. In the figures, the first image is each diagram is the query image. Each diagram in each figure shows the top 10 ranked images returned by a distance metric learning method and the relevant images are marked by a "tick" symbol. From the results, we can see that in most situations, the proposed LRML technique (based on the LRML$^{\mathrm{INV}}$ algorithm) returned considerably more relevant images in the top ranked results, which are consistent to the previous quantitative evaluation results.

6.10 Experiment IV: Application to Collaborative Image Clustering

In addition to the CIR application, we also evaluate the performance of applying the proposed semi-supervised DML techniques to the collaborative image clustering application in exploiting user feedback log data for improving the clustering performance.

6.10.1 *Compared Methods and Experimental Setup.* Similar to the CIR experiments, we use the same datasets for the CIC experiments. To evaluate the effectiveness of the proposed techniques, we implemented and compared the following schemes:

—Kmeans: the baseline k-means clustering algorithm with the Euclidean distance;

—CKmeans: the constrained k-means clustering with the Euclidean distance;

—RCA: the constrained k-means with the RCA metric proposed in [Bar-Hillel et al. 2005];

—DCA: the constrained k-means with the DCA metric proposed in [Hoi et al. 2006];

—DML: the constrained k-means with the DML metric proposed in [Xing et al. 2002];

—LRML$^{\mathrm{SDP}}$: the constrained k-means algorithm with the proposed Laplacian Regularized Metric Learning method that is solved by an SDP based algorithm [Hoi et al. 2008].

—LRML$^{\mathrm{INV}}$: the constrained k-means clustering algorithm with the Laplacian Regularized Metric Learning method that is solved by the matrix inversion based algorithm.

To conduct the clustering experiments, we set the number of clusters k to the number of image categories in the datasets, i.e., $k = 20$ in the 20-category dataset and $k = 50$ in the 50-category dataset, respectively. For the experiments on each dataset, we randomly sample k initial examples as the cluster centroids, and then use them as the input of initial cluster centroids for all of the compared clustering methods. We repeat the above experiment 10 times for each dataset and report the average clustering performance.

6.10.2 *Performance Metrics.* To evaluate the clustering performance, we consider two external clustering validation metrics that utilize the explicit category labels of the images in the dataset. Specifically, the two measurements adopted in our CIC experiments are the normalized mutual information [Strehl et al. 2000; Dom 2001] and the pairwise F1 measurement [Liu et al. 2007]. We briefly introduce them as follows.

The normalized mutual information (NMI) measurement [Strehl et al. 2000; Dom 2001] estimates the quality of a clustering with respect to some given underlying class labeling of the data by measuring how closely the clustering algorithm can reconstruct the the

underlying labeling distribution in the data. The formula of NMI is given as follows:

$$NMI = \frac{2 * I(X; X_0)}{H(X) + H(X_0)}$$

where X_0 and X denote the random variables of cluster memberships from the ground truth and the output of clustering algorithm, respectively, $I(X;Y) = H(X) - H(X|Y)$ represents the mutual information between random variables X and Y, and $H(X)$ represents the Shannon entropy of random variable X.

The second measurement is the pairwise F1 (PF1) metric [Liu et al. 2007]. It is the harmonic mean of pairwise precision and pairwise recall, which are defined as follows:

$$precision = \frac{\#\text{pairs correctly grouped in the same cluster}}{\text{total } \#\text{ pairs in the same cluster in the ground truth}} \quad (27)$$

$$recall = \frac{\#\text{pairs correctly grouped in the same cluster}}{\text{total } \#\text{pairs actually grouped in the same cluster}} \quad (28)$$

$$PF1 = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (29)$$

The PF1 measurement is similar to the definition of clustering accuracy in [Xing et al. 2002] that measures the percentage of example pairs correctly groupped in the same clusters. The main drawback of the metric in [Xing et al. 2002] is that it equally counts two types of example pairs: pairs that belong the same clusters and pairs that belong to different clusters. This may be problematic in reflecting the true clustering performance as most data pairs in a clustering experiment come from different clusters. Thus, the PF1 measurement could be more effective for validating the clustering performance.

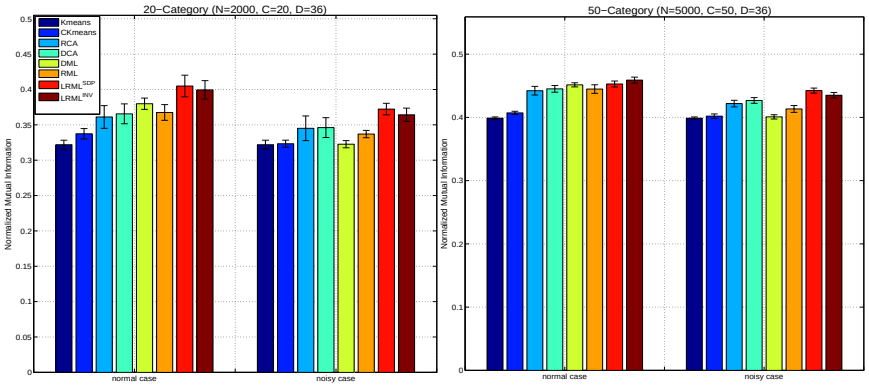


Fig. 3. Comparison of different clustering methods based on the NMI measurement. The left diagram shows the results on the 20-Category dataset and the right one shows the results on the 50-Category dataset. In each diagram, the bars in the left side shows the measurement results obtained in the case of normal log data, while the bars in the right side shows the ones obtained in the case of noisy log data.

6.10.3 *Evaluation of Clustering Results.* Figure 3 shows the evaluation results of the NMI measurement on the two datasets. We can draw a few observations from the results. First of all, similar to the results obtained in CIR experiments, among most test cases, the two proposed LRML algorithms achieved the best clustering performance in term of the NMI measurement. Secondly, we found that the relative improvements obtained by the two proposed LRML algorithms in the noisy log data situation are more significant than

the results obtained in the normal log data situation. This again validates the importance of exploiting unlabeled data in learning more effective and reliable metrics. Finally, comparing the two LRML algorithms, their performance are comparable, in which the LRML$^{\mathrm{SDP}}$ algorithm tends to outperform the LRML$^{\mathrm{INV}}$ algorithm sightly in the 20-Category dataset.

Further, Figure 4 gives the evaluation results of the pairwise F1 measurement on the two datasets. Similar observations were obtained. From the figures, we found that the LRML$^{\mathrm{SDP}}$ algorithm tends to outperform LRML$^{\mathrm{INV}}$ slightly in the 20-Category dataset, while the LRML$^{\mathrm{INV}}$ performs slightly better than LRML$^{\mathrm{SDP}}$ in the 50-Category dataset. From both clustering validation metrics, the two proposed LRML algorithms obtained considerably better improvements than other competing metric learning methods.

Finally, Table VIII and Table IX give the details of the experimental results for the CIC clustering experiments, in which the relative improvements over the baseline are clearly indicated within the parentheses. We can see that the improvements are as significant as the ones obtained in the previous CIR experiments.

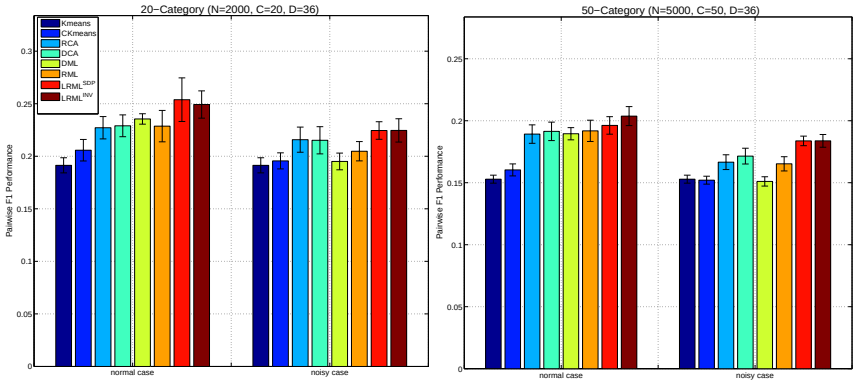


Fig. 4. Comparison of different clustering methods based on the pairwise F1 measurement. The left diagram shows the results on the 20-Category dataset and the right one shows the results on the 50-Category dataset. In each diagram, the bars in the left side shows the measurement results obtained in the case of normal log data, while the bars in the right side shows the ones obtained in the case of noisy log data.

Table VIII. Evaluation of clustering results based on the NMI measurement. For the compared metrics, the constrained kmeans algorithm is used as the clustering algorithm.

Methods	20-Category		50-Category	
	Normal Case	Noisy Case	Normal Case	Noisy Case
Kmeans	0.322 ± 0.006	0.322 ± 0.006	0.399 ± 0.002	0.399 ± 0.002
Ckmeans	0.337±0.007(+4.8%)	0.323±0.005(+0.4%)	0.407±0.003(+2.1%)	0.402±0.004(+0.8%)
RCA	0.361±0.016(+12.2 %)	0.345±0.018(+7.2%)	0.442±0.007(+10.9%)	0.422±0.005(+5.8%)
DCA	0.366±0.014(+13.6%)	0.346±0.014(+7.5%)	0.445±0.005(+11.6%)	0.427±0.005(+7.0%)
DML	0.380±0.008(+18.0%)	0.323±0.005(+0.2%)	0.452±0.003(+13.2%)	0.401±0.003(+0.5%)
RML	0.367±0.011(+14.2%)	0.337±0.005(+4.7%)	0.445±0.007(+11.6%)	0.413±0.005(+3.7%)
LRML$^{\mathrm{SDP}}$	0.405±0.015(+25.8%)	0.372±0.008(+15.6%)	0.453±0.005(+13.6%)	0.442±0.004(+10.9%)
LRML$^{\mathrm{INV}}$	0.400±0.013(+24.1%)	0.364±0.009(+13.1%)	0.459±0.005(+15.1%)	0.435±0.004(+9.1%)

6.11 Experiment V: Time Performance Evaluation

The last experiment is to evaluate the time efficiency of the LRML algorithms. All experiments were run on a PC of 3.4GHz CPU and 3GB RAM in the matlab environment.

Table IX. Evaluation of clustering results based on the pairwise F1 measurement. For the compared metrics, the constrained kmeans algorithm is used as the clustering algorithm.

Methods	20-Category		50-Category	
	Normal Case	Noisy Case	Normal Case	Noisy Case
Kmeans	0.191±0.007	0.191 ± 0.007	0.153±0.003	0.153±0.003
Ckmeans	0.206±0.010(+7.5%)	0.196±0.008(+2.2%)	0.160±0.005(+4.9%)	0.152±0.003(-0.5%)
RCA	0.227±0.011(+18.7%)	0.216±0.012(+12.8%)	0.189±0.007(+23.8%)	0.167±0.006(+9.0 %)
DCA	0.229±0.010(+19.7%)	0.215±0.013(+12.4%)	0.191±0.007(+25.2%)	0.172±0.006(+12.2%)
DML	0.236±0.005(+23.1%)	0.195±0.008(+1.9%)	0.190±0.005(+24.0%)	0.151±0.004(-1.2%)
RML	0.229±0.015(+19.5%)	0.205±0.009(+7.0%)	0.192±0.009(+25.5%)	0.165±0.006(+8.1%)
LRML$^{\mathrm{SDP}}$	0.254±0.021(+32.6%)	0.225±0.008(+17.3%)	0.196±0.007(+28.4%)	0.184±0.004(+20.2%)
LRML$^{\mathrm{INV}}$	0.249±0.013(+30.2%)	0.225±0.011(+17.4%)	0.204±0.008(+33.3%)	0.184±0.005(+20.2%)

Table X shows the evaluation results of time efficiency by different metric learning methods on both datasets with the same amount of normal log data. The time cost in the table includes all preprocessing cost, such as the time cost of computing the Laplacian matrix.

Several observations can be drawn from the results. First of all, we found that the two LRML algorithms are less efficient than some unsupervised and supervised methods, including MAH, RCA and DCA, while it is considerably more efficient than the regular DML method that is solved by a convex optimization method. Secondly, by comparing the proposed LRML methods with the RML approach, we found that the two LRML algorithms took smaller time cost on the 20-Category dataset, however, they took significantly more time when being tested on the 50-Category dataset. The key reason for their difference is that for both semi-supervised methods, we need to compute the Laplacian matrix and its related matrix computation, which takes more time for larger datasets.

Table X. Comparisons of Time Performance (seconds)

Algorithm	MAH	RCA	DCA	DML	RML	LRML$^{\mathrm{SDP}}$	LRML$^{\mathrm{INV}}$
20-Category	0.015	0.036	0.045	199.174	11.310	9.860	9.130
50-Category	0.032	0.078	0.081	2004.479	12.448	71.335	70.704

To justify the efficiency of the proposed algorithm, we further inspect the time cost taken in different stages of the proposed distance metric learning algorithms. Table XI shows the results of time cost in different stages of the compared algorithms. In the table, t_{L} and t_{XLX} represent the time cost of computing the Laplacian matrix and its related matrix computation respectively, which are engaged only in the semi-supervised methods; t_{SD} represents the time cost of computing the two similarity matrices S and D in (16) and t_{OPT} denotes the time cost of solving the optimization problem. We can draw some observations from the results. First of all, we found the major computation of the two LRML algorithms is paid for computing the Laplacian matrix. Further, for comparing the time cost used in solving the optimization problems, we found that the proposed LRML$^{\mathrm{INV}}$ algorithm achieved a significant speedup compared with the SDP based approaches. For example, on the 20-Category dataset, the time cost for solving the optimization by LRML$^{\mathrm{INV}}$ is about 70 times faster than LRML$^{\mathrm{SDP}}$, and is almost over 700 times faster than RML$^{\mathrm{INV}}$. The speedup results are even more significant on the 50-Category dataset.

Table XI. Evaluation of time cost taken in different stages (seconds)

Algorithms	20-Category					50-Category				
	t_{L}	t_{XLX}	t_{SD}	t_{OPT}	total	t_{L}	t_{XLX}	t_{SD}	t_{OPT}	total
RML	0.000	0.000	1.239	13.203	14.442	0.000	0.000	3.426	9.022	12.448
LRML$^{\mathrm{SDP}}$	9.022	0.200	1.282	1.331	11.835	66.534	0.689	3.525	0.587	71.335
LRML$^{\mathrm{INV}}$	9.250	0.189	1.275	0.017	10.732	66.565	0.684	3.452	0.004	70.704

6.12 Discussions

We discuss two important issues that were found from our empirical experiences, and provide some suggestions for further improvements.

First, we notice that one disadvantage of the proposed LRML method lies in the stage of computing the Laplacian matrix, which will take non-trivial time cost, $O(n^2 \log n)$, for large applications. This, however, is not very critical for real DML applications because the stage of computing the Lapalcian matrix often can be done in an offline manner. Hence, with a pre-computed Lapalcian matrix, the LRML method can be solved very efficiently by our proposed algorithms. Further, to efficiently compute the Laplacian matrix, we adopt some efficient data structure to speed up the computation. In particular, we propose to adopt the Cover tree technique [Beygelzimer et al. 2006] to speed up the computation of Laplacian matrix. The construction of the cover tree structure takes $O(n \log n)$ time, and the batch query of searching for k-nearest neighbors on the whole data set can be found in $O(n)$ time. Hence, using the cover tree data structure to find the nearest neighbors, we can considerably reduce the time complexity of computing Laplacian from $O(n^2 \log n)$ to $O(n \log n)$, making large-scale applications feasible.

Second, we discuss some advantages and disadvantages for the two proposed algorithms, LRML$^{\mathrm{SDP}}$ and LRML$^{\mathrm{INV}}$. First of all, in terms of computational cost, LRML$^{\mathrm{INV}}$ is clearly more efficient than LRML$^{\mathrm{SDP}}$ for solving the optimization. In particular, LRML$^{\mathrm{SDP}}$ is only feasible for small applications due to the bottleneck of standard SDP solvers, while LRML$^{\mathrm{SDP}}$ can solve significantly larger problems. Further, in terms of empirical accuracy for retrieval and clustering, we found that the two algorithms are essentially comparable. No one method is significantly better than the other. But, in some situation, we found that the solution of LRML$^{\mathrm{SDP}}$ tends to be slightly more stable than the solution of LRML$^{\mathrm{INV}}$.

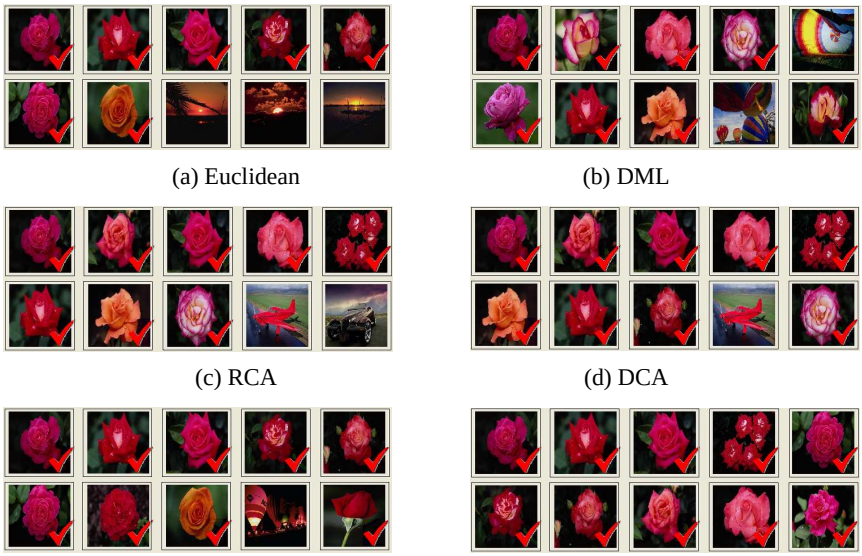
7. CONCLUSIONS

We proposed a novel framework of semi-supervised distance metric learning for solving collaborative image retrieval and clustering, where the real log data of user relevance feedback are analyzed to discover useful information and infer optimal metrics. To fully exploit the unlabeled data, we proposed a Laplacian Regularized Metric Learning (LRML) technique, which leverages the distribution of unlabeled data and ensures the smoothness of metric learning through a regularization framework. Two new algorithms were proposed to resolve the optimization problem efficiently. We conducted extensive experiments over various adverse conditions and compared the proposed method with a large number of standard options and competitive methods. The results show that the LRML method is more effective than the state-of-the-art methods for learning reliable metrics from realistic log data that are probably noisy and scarce.

Despite encouraging results obtained, the current work has some limitations. First, the stage of constructing the graph Laplacian matrix usually takes nontrivial computational cost, which could be further improved by applying some efficient data structures. Second, the distance metric learned by the proposed method is essentially linear, which may be somewhat restrictive to some complicated applications. In future, we may study kernel based techniques [Yan et al. 2006] to consummate the proposed technique.

Acknowledgments

The work was fully supported by Singapore MOE Academic Tier-1 Research Grant (RG67/07).

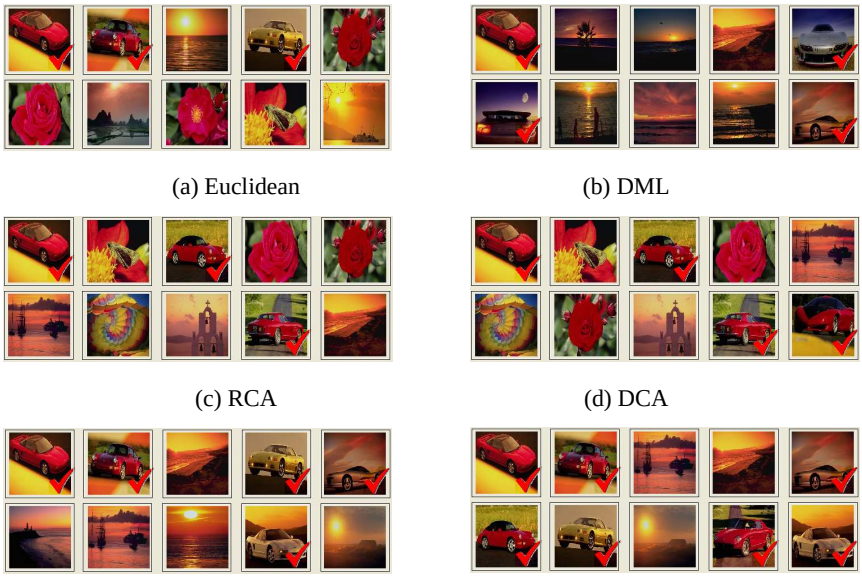


(a) Euclidean (b) DML

(c) RCA (d) DCA

(e) RML (f) LRML

Fig. 5. Comparison of retrieval performance given a “rose” query. Each diagram shows top 10 returned images by one metric learning method. The first image is the query image and the relevant images are marked with a “tick” symbol.



(a) Euclidean (b) DML

(c) RCA (d) DCA

(e) RML (f) LRML

Fig. 6. Comparison of retrieval performance given a “car” query. Each diagram shows top 10 returned images by one metric learning method. The first image is the query image and the relevant images are marked with a “tick” symbol.

(a) Euclidean (b) DML

(c) RCA (d) DCA

(e) RML (f) LRML

Fig. 7. Comparison of retrieval performance given a "bird-net" query. Each diagram shows top 10 returned images by one metric learning method. The first image is the query image and the relevant images are marked with a "tick" symbol.

REFERENCES

BAR-HILLEL, A., HERTZ, T., SHENTAL, N., AND WEINSHALL, D. 2005. Learning a mahalanobis metric from equivalence constraints. *JMLR 6*, 937–965.

BEYGELZIMER, A., KAKADE, S., AND LANGFORD, J. 2006. Cover trees for nearest neighbor. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*. ACM, New York, NY, USA, 97–104.

BOYD, S. AND VANDENBERGHE, L. 2003. *Convex Optimization*. Cambridge University Press.

COX, T. AND COX, M. 1994. *Multidimensional Scaling*. Chapman & Hall, London.

DOM, B. E. 2001. An information-theoretic external cluster-validity measure. In *Research Report RJ 10219, IBM*.

FUKUNAGA, K. 1990. *Introduction to Statistical Pattern Recognition*. Elsevier.

GIROSI, F., JONES, M., AND POGGIO, T. 1995. Regularization theory and neural networks architectures. *Neural Computation 7*, 219–269.

GLOBERSON, A. AND ROWEIS, S. 2005. Metric learning by collapsing classes. In *NIPS'05*.

HE, X., MA, W.-Y., AND ZHANG, H.-J. 2004. Learning an image manifold for retrieval. In *ACM Multimedia*. New York, 17–23.

HOI, C. H. AND LYU, M. R. 2004a. Group-based relevance feeedback with support vector machine ensembles. In *Proceedings 17th International Conference on Pattern Recognition (ICPR'04)*. Cambridge, UK, 874–877.

HOI, C.-H. AND LYU, M. R. 2004b. A novel log-based relevance feedback technique in content-based image retrieval. In *Proceedings of ACM MM 2004*. ACM Press, New York, NY, USA.

HOI, S. C., LIU, W., AND CHANG, S.-F. 2008. Semi-supervised distance metric learning for collaborative image retrieval. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR2008)*.

HOI, S. C., LIU, W., LYU, M. R., AND MA, W.-Y. 2006. Learning distance metrics with contextual constraints for image retrieval. In *Proc. CVPR2006*. New York, US.

HOI, S. C., LYU, M. R., AND JIN, R. 2006. A unified log-based relevance feedback scheme for image retrieval. *IEEE Trans. KDE 18*, 4, 509–204.

HOI, S. C. H., LYU, M. R., AND JIN, R. 2005. Integrating user feedback log into relevance feedback by coupled svm for content-based image retrieval. In *Proc. IEEE ICDE Workshop on EMMA (EMMA2005)*. Japan.

J. GOLDBERGER, S. ROWEIS, G. H. AND SALAKHUTDINOV, R. 2005. Neighbourhood components analysis. In *NIPS17*.

JAIN, A. K., MURTY, M. N., AND FLYNN, P. J. 1999. Data clustering: a review. *ACM Comput. Surv. 31,* 3, 264–323.

JAIN, A. K. AND VAILAYA, A. 1998. Shape-based retrieval: a case study with trademark image database. *Pattern Recognition* 9, 1369–1390.

KING, I. AND ZHONG, J. 2003. Integrated probability function and its application to content-based image retrieval by relevance feedback. *Pattern Recognition 36,* 9, 2177–2186.

KUHN, H. W. 1982. Nonlinear programming: a historical view. *SIGMAP Bull.* 31, 6–18.

LEE, J.-E., JIN, R., AND JAIN, A. K. 2008. Rank-based distance metric learning: An application to image retrieval. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. Anchorage, AK.

LEW, M. S., SEBE, N., DJERABA, C., AND JAIN, R. 2006. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl. 2,* 1, 1–19.

LIU, Y., JIN, R., AND JAIN, A. K. 2007. Boostcluster: boosting clustering by pairwise constraints. In *Proc. 13th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'07)*. San Jose, California, USA, 450–459.

MANJUNATH, B., WU, P., NEWSAM, S., AND SHIN, H. 2001. A texture descriptor for browsing and similarity retrieval. *Signal Processing Image Communication*.

MÜLLER, H., PUN, T., AND SQUIRE, D. 2004. Learning from user behavior in image retrieval: Application of market basket analysis. *Int. J. Comput. Vision 56,* 1-2, 65–77.

ROWEIS, S. AND SAUL, L. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science 290,* 5500, 2323–2326.

RUI, Y., HUANG, T., AND MEHROTRA, S. 1997. Content-based image retrieval with relevance feedback in mars. II: 815–818.

RUI, Y., HUANG, T. S., ORTEGA, M., AND MEHROTRA, S. 1998. Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE Trans. CSVT 8,* 5 (Sept.), 644–655.

SALTON, G. AND BUCKLEY, C. 1988. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage. 24,* 5, 513–523.

SI, L., JIN, R., HOI, S. C., AND LYU, M. R. 2006. Collaborative image retrieval via regularized metric learning. *ACM Multimedia Systems Journal 12,* 1, 34–44.

SMEULDERS, A. W. M., WORRING, M., SANTINI, S., GUPTA, A., AND JAIN, R. 2000. Content-based image retrieval at the end of the early years. *IEEE Trans. PAMI 22,* 12, 1349–1380.

STREHL, E., GHOSH, J., AND MOONEY, R. 2000. Impact of similarity measures on web-page clustering. In *In Workshop on Artificial Intelligence for Web Search (AAAI 2000*. AAAI, 58–64.

STURM, J. F. 1999. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software 11–12*, 625–653.

TAO, D. AND TANG, X. 2004. Random sampling based svm for relevance feedback image retrieval. In *Proceedings IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

TENENBAUM, J. B. AND DE SILVA ANDJOHN C. LANGFORD, V. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science 290,* 5500, 2319–2323.

TONG, S. AND CHANG, E. 2001. Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international conference on Multimedia*. ACM Press, New York, NY, USA, 107–118.

VAPNIK, V. N. 1998. *Statistical Learning Theory*. John Wiley & Sons.

WEINBERGER, K., BLITZER, J., AND SAUL, L. 2006. Distance metric learning for large margin nearest neighbor classification. In *NIPS 18*. 1473–1480.

XING, E. P., NG, A. Y., JORDAN, M. I., AND RUSSELL, S. 2002. Distance metric learning with application to clustering with side-information. In *NIPS2002*.

YAN, R., ZHANG, J., YANG, J., AND HAUPTMANN, A. G. 2006. A discriminative learning framework with pairwise constraints for video object classification. *IEEE Trans. Pattern Anal. Mach. Intell. 28,* 4, 578.

YANG, L., JIN, R., SUKTHANKAR, R., AND LIU, Y. 2006. An efficient algorithm for local distance metric learning. In *AAAI2006*.