A Multimedia Retrieval Framework Based on Semi-Supervised Ranking and Relevance Feedback

Yi Yang, Feiping Nie, Dong Xu, Jiebo Luo, *Fellow*, *IEEE*, Yueting Zhuang, and Yunhe Pan

Abstract—We present a new framework for multimedia content analysis and retrieval which consists of two independent algorithms. First, we propose a new semi-supervised algorithm called ranking with Local Regression and Global Alignment (LRGA) to learn a robust Laplacian matrix for data ranking. In LRGA, for each data point, a local linear regression model is used to predict the ranking scores of its neighboring points. A unified objective function is then proposed to globally align the local models from all the data points so that an optimal ranking score can be assigned to each data point. Second, we propose a semi-supervised long-term Relevance Feedback (RF) algorithm to refine the multimedia data representation. The proposed long-term RF algorithm utilizes both the multimedia data distribution in multimedia feature space and the history RF information provided by users. A trace ratio optimization problem is then formulated and solved by an efficient algorithm. The algorithms have been applied to several content-based multimedia retrieval applications, including cross-media retrieval, image retrieval, and 3D motion/pose data retrieval. Comprehensive experiments on four data sets have demonstrated its advantages in precision, robustness, scalability, and computational efficiency.

Index Terms—Content-based multimedia retrieval, semi-supervised learning, ranking algorithm, relevance feedback, cross-media retrieval, image retrieval, 3D motion data retrieval.

1 Introduction

To effectively manage the rapidly growing multimedia data, a large number of methods have been proposed for multimedia content analysis and retrieval. These works include content-based image retrieval [15], [16], [40], audio retrieval [21], and video retrieval [12]. In recent years, researchers also proposed numerous content-based retrieval systems for new media types, including 3D data, cultural artifacts, motion data, and biological data [4], [20], [22]. One of the well-known challenges in multimedia content analysis and retrieval is the so-called semantic gap, i.e., the low-level features are not sufficient to characterize the high-level semantics of multimedia data. As a way to bridge the semantic gap, many machine learning algorithms have been proposed and made remarkable improvements in content-based multimedia retrieval [20]. To achieve better multimedia retrieval performance, we focus in this work on two research issues: 1) given a query, how to rank the database multimedia data that are represented by feature vectors and return the most relevant ones to the user, and 2) how to infer an effective vector representation of multimedia data according to their features and user feedback.

Given a query example provided by a user, the retrieval process is to rank the database multimedia data according to their relevance to the query example and return the top-ranked ones. Thus, a good ranking algorithm is crucial for multimedia retrieval. The existing ranking algorithms can be roughly grouped into two categories, namely, query-independent ranking and query-dependent ranking. A representative work of query-independent ranking algorithm is PageRank algorithm [19], which ranks the importance of webpages by mining the link structure among them. The most frequently used query-dependent ranking method in the field of multimedia retrieval is distance-based ranking. Such ranking is usually performed according to the euclidean distance between the database multimedia data and the query, either in the original feature space or in a lower dimensional space of multimedia features. For example, in [17], Huang et al. have proposed an image retrieval system where the ranking is directly based on euclidean distance of image color features. Other researchers also proposed ranking database images according to the euclidean distance in a lower dimensional space derived from the original feature space by using linear or nonlinear mapping methods [16], [37]. In [8], researchers suggested to learn a parameterized similarity function among multimedia data for retrieval. However, similarly to distance-based ranking, the algorithm proposed in [8] only focuses on the

- Y. Yang is with the College of Computer Science, Zhejiang University, and the School of Computer Science, Carnegie Mellon University, 5000 Forbes Ave., NSH 4629, Pittsburgh, PA 15213. E-mail: yiyang@cs.cmu.edu.
- F. Nie is with the Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019. E-mail: feipingnie@gmail.com.
- D. Xu is with the School of Computer Engineering, Nanyang Technological University, 50 Nanyang Avenue, Blk N4, Singapore 639798. E-mail: dongxu@ntu.edu.sg.
- J. Luo is with the Kodak Research Laboratories, Eastman Kodak Company, 1999 Lake Avenue, Rochester, NY 14650. E-mail: jiebo.luo@kodak.com.
- Y. Zhuang and Y. Pan are with the College of Computer Science, Caoguangbiao Building, Yuquan Campus, Zhejiang University, Hangzhou 310027, China. E-mail: yzhuang@zju.edu.cn, panyh@cae.cn.

Manuscript received 15 Oct. 2009; revised 17 Jan. 2011; accepted 4 May 2011; published online 8 Aug. 2011.
Recommended for acceptance by A. Torralba.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2009-10-0691.
Digital Object Identifier no. 10.1109/TPAMI.2011.170.

pairwise similarity among multimedia data and the distribution of the whole data set is not considered. In the meantime, other distance metrics, such as Earth Mover's Distance and Mahalanobis distance, have also been used for multimedia data analysis and retrieval [25], [35]. In addition, researchers developed a specific distance metric for shape images [13]. However, these ranking algorithms cannot work well for the cases in which the distribution of multimedia data is too complicated to be measured by the commonly used distance or similarity functions.

Besides the distance-based ranking algorithms, researchers have applied inductive learning methods to multimedia retrieval. These algorithms usually learn a classifier to separate the relevant and irrelevant training data and employ the decision values from the learned classifier as the ranking scores of multimedia data [28], [38]. However, the performance of these algorithms depends heavily on the labeled training data. For example, the image retrieval performance of the system in [28] may drop considerably if the negative examples are not properly chosen. It remains an open problem how to choose the proper training samples during the relevance feedback (RF) process. Moreover, as indicated in [45], the inductive learning methods may overfit the training data if only a limited number of labeled data are provided.

In [43], Zhou et al. proposed a transductive ranking algorithm, namely, Manifold Ranking (MR), to rank the data with respect to the intrinsic data distribution. MR is intrinsically different from distance-based ranking methods because it exploits the data distribution of all the samples for ranking rather than only considering the pairwise distances or inner products [15], [2], [43], [46] between the query example and database multimedia data. During recent years, MR has been applied to various applications by other researchers, and the results on different applications, including image retrieval [15], shape retrieval [2], and cross-media retrieval [46], have demonstrated that it is beneficial to utilize the data distribution for ranking. Compared with inductive learning algorithms, it is useful to perform ranking according to data distribution in content-based multimedia retrieval because it provides a way to address the small sample size problem by employing unlabeled data [15] without requiring labeled data for training. As discussed in [30], the performance of MR is sensitive to the bandwidth parameter of the Gaussian function based on which the Laplacian matrix in MR is calculated. Therefore, MR is less effective in real-world multimedia retrieval applications because the ground truth labels are generally not available for tuning the bandwidth parameter.

RF has been proven helpful for multimedia content analysis and retrieval [6], [15], [16], [37], [20], [26], [46]. Recently, statistical learning methods are used to improve the performance of content-based multimedia retrieval in which the labels are typically obtained via RF. The RF algorithms can be divided into two groups, short-term RF and long-term RF. Short-term RF only considers RF information provided by the current user, and is usually used to better understand the user's search intension and can only improve the accuracy of the current search. Long-term RF makes use of the RF information provided by all the users. For example, He et al. proposed a manifold learning-based image retrieval system in [16]. In their system, a heuristic long-term RF algorithm is proposed to boost the performance of image retrieval by directly modifying the distances between the vertices on the image manifold. However, a large number of images may need to be labeled in [16] in order to achieve significant retrieval performance improvement because the RF algorithm in [16] can only modify the edge weights between labeled data. As a supervised learning method, Linear Discriminant Analysis (LDA) generally outperforms Principal Component Analysis (PCA) for image classification or retrieval [14]. However, the performance of LDA may also drop considerably when only a limited number of labeled samples are marked by the user in the RF. To solve this problem, Cai et al. proposed Semi-supervised Discriminant Analysis (SDA) [6], in which a geometrical regularizer is introduced into the objective function of LDA in order to utilize the manifold structure of both labeled and unlabeled training data. The experiments in [6] demonstrate that SDA outperforms LDA for image retrieval, especially when the number of labeled data is limited. The objective functions in LDA and SDA are formulated in the ratio trace form[1] of $\max_W Tr((W^TBW)^{-1}(W^TAW))$. However, the solutions may lead to poor performance in multimedia retrieval due to the deviation from the original objectives [31].

In this paper, we propose a framework consisting of two algorithms for multimedia content analysis and retrieval. First, a new transductive ranking algorithm, namely, ranking with Local Regression and Global Alignment (LRGA) [34], is proposed. Differently from distance-based ranking methods, the distribution of the samples in the whole data set is exploited in LRGA. Compared with the inductive methods, such as [28], only the query example is required. In contrast to the MR algorithm [43] that directly adopts the Gaussian kernel to compute the Laplacian matrix [10], LRGA learns a Laplacian matrix for data ranking. For each data point, we employ a local linear regression model to predict the ranking scores of its neighboring points. In order to assign an optimal ranking score to each data point, we propose a unified objective function to globally align local linear regression models from all the data points. In retrieval applications, there is no ground truth to tune the parameters of ranking algorithms like MR. Therefore, it is meaningful to develop a new method that learns an optimal Laplacian matrix for data ranking.

Second, we propose a semi-supervised learning algorithm for long-term RF. A system log is constructed to record the history RF information marked by all of the users. We refine the vector representation of multimedia data according to the log information via a statistical approach. To that end, we convert the RF information into pairwise constraints, which are classified into two groups. The data pairs in the first group are semantically similar to each other, while the data pairs in the second group are dissimilar to each other. While LDA can be used to exploit these two types of information, the valuable information in the unlabeled data is not utilized. In this paper, we propose a semi-supervised learning algorithm to refine the vector

[1] Taking LDA [14] as an example, A and B can be regarded as the between-class scatter matrix and the within-class scatter matrix, respectively.

representation by considering the history RF information as well as the multimedia data distribution of both labeled and unlabeled samples. Compared with the existing semi-supervised learning method SDA [6], our algorithm can learn a better multimedia data representation because we formulate the objective function in a trace ratio form of $\max_W \frac{Tr(W^TAW)}{Tr(W^TBW)}$ (see Section 3 for more details). After performing the long-term RF, the vector representation of multimedia data is more effective and thus a higher multimedia retrieval performance can be achieved.

The two proposed algorithms can be applied to many multimedia content-based classification and retrieval applications. In this paper, we evaluate the performance of our algorithms in content-based cross-media retrieval [36], [46], where the query example and retrieval results can be of different media types. For example, the user can search images either by an example image or an example audio record. We also apply the proposed algorithms to content-based image retrieval and 3D motion/pose data retrieval. Extensive experiments demonstrate that our algorithms achieve better retrieval performance when compared with the existing related works.

The remainder of this paper is organized as follows: In Section 2, we describe the proposed ranking algorithm, i.e., ranking with Local Regression and Global Alignment. Section 3 presents the long-term RF algorithm. In Section 4, we apply the proposed framework to three different multimedia retrieval applications. Section 5 shows the experimental results, and conclusions are given in Section 6.

2 Ranking with Local Regression and Global Alignment

2.1 Notations

Given a set of multimedia data represented by their feature vectors $\chi = \{x_1, x_2, \dots, x_N\}$, the LRGA ranking algorithm aims to find a function f that assigns each data point $x_i \in \mathbb{R}^d$ a ranking score $f_i \in \mathbb{R}$ according to its relevance to the user query and the data distribution. Let us denote $\mathcal{N}_k(x_i) = \{x_i, x_{i_1}, x_{i_2}, \dots x_{i_k}\}$ as the set of k-nearest neighbors of x_i plus x_i, and $v_i = [i, i_1, i_2, \dots i_k]$ is a vector comprising the indices of samples in $\mathcal{N}_k(x_i)$. We also define $f = [f_1, f_2, \dots, f_N]^T \in \mathbb{R}^N$, in which f_i is the ranking score of x_i, and $y = [y_1, y_2, \dots, y_N]^T \in \mathbb{R}^N$, where $y_i = 1$ if x_i is the query and $y_i = 0$, otherwise.

2.2 LRGA Ranking Algorithm

To rank the data points in χ, we employ two sources of information: 1) the query/queries provided by users and 2) the relationship of all the multimedia data points. The final ranking results can be obtained by balancing the two sources of information. In other words, there are two constraints on f. First, it should be consistent with query example (examples) provided by the user. Second, it should be consistent with multimedia data distribution. As we will show later, learning from query/queries and from data distribution can be formulated as two different minimization problems, which are then linearly combined to obtain the final ranking results. To utilize the information from query/queries and data distribution for data ranking, a straightforward way is to minimize the following objective function:

$$\min_{f \in \mathbb{R}^N} \sum_{i=1}^{N} (f_i - y_i)^2 + \Omega(f), \tag{1}$$

where $\Omega(f)$ is a regularization function on f to enforce manifold smoothness (i.e., the neighboring samples in high-dimensional space should share similar ranking scores). The first term of (1) enforces that the ranking results will be consistent with the queries because the queries provided by the user reflect his or her search intention. Given a multimedia sample that is not the query example, we have no prior knowledge of whether it meets the user's search intention. We therefore define a diagonal matrix U to assign different weights to different data points. We set U_{ii} as a very large value[2] if x_i is the query, and set $U_{ii} = 1$ otherwise. We therefore propose to minimize the following objective function:

$$\begin{aligned}&\min_{f \in \mathbb{R}^N} \sum_{i=1}^{N} U_{ii}(f_i - y_i)^2 + \Omega(f)\\ &= \min_{f \in \mathbb{R}^N} (f - y)^T U (f - y) + \Omega(f).\end{aligned} \tag{2}$$

In many real applications, the local structure is more important than the global structure [24]. Meanwhile, it has been reported that the local learning (LL) algorithms often outperform global learning algorithms [5], [32]. To make use of the data distribution, i.e., to learn a regularization function $\Omega(f)$, we employ the local structure of each data point in $\chi = \{x_1, x_2, \dots, x_N\}$. For each data point x_i, we adopt a local linear regression model $h_i(x) = w_i^T x + b_i$, where $w_i \in \mathbb{R}^d$ is the local projection matrix, $b_i \in \mathbb{R}$ is the bias term. While it is possible to use other complex nonlinear models, we use the linear model because: 1) It is fast and more suitable for practical applications and 2) the local structure of manifold is approximately linear [24]. The linear regression model $h_i(x_j) = w_i^T x_j + b_i$ is used to predict the ranking score f_j of each data point $x_j \in \mathcal{N}_k(x_i)$. The *local prediction error* of the model with respect to a single data point $x_j \in \mathcal{N}_k(x_i)$ is given by [34]

$$\left(w_i^T x_j + b_i - f_j\right)^2. \tag{3}$$

The *local model error* of the local regression model $h_i(x) = w_i^T x + b_i$ can be computed by summing the local prediction errors from all the data points in $\mathcal{N}_k(x_i)$, which is formulated as

$$\sum_{x_j \in \mathcal{N}_k(x_i)} \left(w_i^T x_j + b_i - f_j\right)^2 + \lambda w_i^T w_i, \tag{4}$$

where the regularization term (i.e., the second term) is imposed to avoid overfitting. We minimize the local model error of $h_i(x)$ and then we arrive at

$$\min_{f_{(i)}, b_i, w_i} \|X_i^T w_i + b_i \mathbf{1}_{k+1} - f_{(i)}\|^2 + \lambda w_i^T w_i, \tag{5}$$

where $X_i = [x_i, x_{i_1}, x_{i_2}, \dots x_{i_k}]$ is a data matrix comprising all the data points in the set $\mathcal{N}_k(x_i)$, $f_{(i)} = [f_i, f_{i_1}, f_{i_2}, \dots f_{i_k}]^T$ is a

[2] 2. Theoretically, it should be ∞. The retrieval performance is not sensitive to this value provided that it is large enough. In our experiment, we set it to 10,000 to ensure that the query examples always have the highest ranking scores.

vector comprising the ranking scores of all the data points in the set $\mathcal{N}_k(x_i)$, and $\mathbf{1}_{k+1} \in \mathbb{R}^{k+1}$ is a column vector with all ones. In order to assign an optimal ranking score to each data point, we globally align all the local regression models by summing (5) over all the data. Then, we arrive at

$$\min_{f_{(i)}|_{i=1}^{N}, b_i|_{i=1}^{N}, w_i|_{i=1}^{N}} \sum_{i=1}^{N} \left(\left\| X_i^T w_i + b_i \mathbf{1}_{k+1} - f_{(i)} \right\|^2 + \lambda w_i^T w_i \right). \tag{6}$$

By setting the derivatives of (6) to be zero w.s.t. b_i and w_i, we have

$$\begin{aligned} & w_i^T X_i \mathbf{1}_{k+1} + (k+1) b_i - f_{(i)}^T \mathbf{1}_{k+1} = 0 \\ & \quad \Rightarrow b_i = \frac{1}{k+1} \left(f_{(i)}^T \mathbf{1}_{k+1} - w_i^T X_i \mathbf{1}_{k+1} \right) \\ & \quad = \frac{1}{k+1} \left(\mathbf{1}_{k+1}^T f_{(i)} - \mathbf{1}_{k+1}^T X_i^T w_i \right), \end{aligned} \tag{7}$$

$$\begin{aligned} & X_i X_i^T w_i + X_i \mathbf{1}_{k+1} b_i - X_i f_{(i)} + \lambda w_i = 0 \\ & \quad \Rightarrow w_i = \left(X_i H X_i^T + \lambda I \right)^{-1} X_i H f_{(i)}, \end{aligned} \tag{8}$$

where $H = I - \frac{1}{k+1} \mathbf{1}_{k+1} \mathbf{1}_{k+1}^T \in \mathbb{R}^{(k+1)\times(k+1)}$ is the centering matrix. Note that $H = H^T = HH^T$. Substituting (7) and (8) for w_i and b_i , we then have

$$\begin{aligned} & X_i^T w_i + \mathbf{1}_{k+1} b_i - f_{(i)} \\ & \quad = H X_i^T (X_i H X_i^T + \lambda I)^{-1} X_i H f_{(i)} - H f_{(i)}. \end{aligned} \tag{9}$$

The objective function in (6) becomes

$$\begin{aligned} \min_{f_{(i)}|_{i=1}^{N}} \sum_{i=1}^{N} & \left[\left\| \left(H X_i^T (X_i H X_i^T + \lambda I \right)^{-1} X_i H f_{(i)} - H f_{(i)} \right\|^2 \right. \\ & \left. + \lambda f_{(i)}^T H X_i^T \left(X_i H X_i^T + \lambda I \right)^{-2} X_i H f_{(i)} \right]. \end{aligned} \tag{10}$$

Theorem 1. *The objective function in (10) is equivalent to the following objective function:*

$$\min_{f_{(i)}|_{i=1}^{N}} \sum_{i=1}^{N} f_{(i)}^T L_i f_{(i)}, \tag{11}$$

where $L_i \in \mathbb{R}^{(k+1)\times(k+1)}$ *is defined as*

$$L_i = H - H X_i^T (X_i H X_i^T + \lambda I)^{-1} X_i H. \tag{12}$$

Proof. See the Appendix, which can be found in the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2011.170. □

Let us denote $S_i \in \mathbb{R}^{N\times(1+k)}$ in which $(S_i)_{pq} = 1$, if $p = (v_i)_q$, and $(S_i)_{pq} = 0$, otherwise. Recall that $v_i = [i, i_1, i_2, \ldots i_k]$ is a vector comprising the indices of samples in $\mathcal{N}_k(x_i)$. Then, we have $f_{(i)}^T = f^T S_i$. The objective function becomes

$$\min_{f_{(i)}|_{i=1}^{N}} \sum_{i=1}^{N} f_{(i)}^T L_i f_{(i)} = \min_{f} f^T L f, \tag{13}$$

where L is defined as

$$\begin{aligned} L &= [S_1, S_2, \ldots, S_N] \begin{bmatrix} L_1 & & \\ & \ldots & \\ & & L_N \end{bmatrix} [S_1, S_2, \ldots, S_N]^T \\ &= SAS^T, \end{aligned} \tag{14}$$

where $S = [S_1, S_2, \ldots, S_N]$, and

$$A = \begin{bmatrix} L_1 & & \\ & \ldots & \\ & & L_N \end{bmatrix}.$$

Substituting $\Omega(f)$ in (2) by (13), we have the following objective function:

$$\min_{f} f^T L f + (f - y)^T U (f - y). \tag{15}$$

The optimal solution can then be obtained by solving the following linear equation:

$$(L + U) f = Uy. \tag{16}$$

2.3 A Fast Matrix Computation Algorithm

To calculate the matrix L defined in (14) for data ranking, we need to compute the matrix L_i for each multimedia datum x_i according to (12). In this section, we propose a new approach to accelerate the computation of matrix L.

Lemma 1. *For any matrix* A, $A(A^T A + \lambda I)^{-1} = (AA^T + \lambda I)^{-1} A$ *holds.*

Proof. Note $(AA^T + \lambda I) A = A(A^T A + \lambda I)$. Then, we have

$$\begin{aligned} A(A^T A + \lambda I)^{-1} &= (AA^T + \lambda I)^{-1} (AA^T + \lambda I) A (A^T A + \lambda I)^{-1} \\ &= (AA^T + \lambda I)^{-1} A (A^T A + \lambda I)(A^T A + \lambda I)^{-1} \\ &= (AA^T + \lambda I)^{-1} A. \end{aligned}$$

□

Theorem 2. *For any matrix* $X_i \in \mathbb{R}^{n\times(k+1)}$,

$$H - H X_i^T (X_i H X_i^T + \lambda I)^{-1} X_i H = \lambda H (H X_i^T X_i H + \lambda I)^{-1} H$$

holds.

Proof. Note that $H = HH$. Therefore, we have

$$\begin{aligned} & H - H X_i^T (X_i H X_i^T + \lambda I)^{-1} X_i H \\ & \quad = H - H H X_i^T (X_i H X_i^T + \lambda I)^{-1} X_i H H. \end{aligned}$$

According to Lemma 1, we have

$$\begin{aligned} & H - H H X_i^T (X_i H X_i^T + \lambda I)^{-1} X_i H H \\ & \quad = H - H H X_i^T X_i H \left(H X_i^T X_i H + \lambda I \right)^{-1} H \\ & \quad = H - H \left(H X_i^T X_i H + \lambda I - \lambda I \right) \left(H X_i^T X_i H + \lambda I \right)^{-1} H \\ & \quad = H - H \left(I - \lambda \left(H X_i^T X_i H + \lambda I \right)^{-1} \right) H \\ & \quad = \lambda H \left(H X_i^T X_i H + \lambda I \right)^{-1} H. \end{aligned}$$

□

According to Theorem 2, we can calculate L_i defined in (12) by

$$L_i = \lambda H\big(HX_i^T X_i H + \lambda I\big)^{-1} H. \tag{17}$$

Let d be the dimension of multimedia feature vector. It is clear that $X_i H X_i^T$ is a matrix of size $d \times d$. If we compute L_i according to (12), we need to compute the inverse of a matrix of size $d \times d$ for each multimedia datum. On the other hand, if we compute L_i according to (17), we only need to compute the inverse of a matrix of size $(k+1) \times (k+1)$ for each multimedia datum. Considering that multimedia data are always of high dimensions that are much larger than k, the computation of L will be faster in the later case.

In summary, given the multimedia data represented by their feature vectors $\chi = \{x_1, x_2, \ldots, x_N\}$ and query information $y = [y_1, y_2, \ldots y_N]^T$, the algorithm of ranking with Local Regression and Global Alignment is listed below. Note that the Laplacian matrix L defined in (14) can be precomputed. When the user submits a query, our system only needs to solve a linear equation (i.e., step 4 in Algorithm-1[3]) to obtain the ranking scores for multimedia retrieval. Note that L is a sparse matrix. As shown in [7], [27], such a linear system can be solved in linear time. In our experiments, we show that LRGA gains good performance when k, which specifies the size of the nearest neighbors, is no larger than 20. Therefore, the time complexity of LRGA to perform multimedia retrieval is *linear* with respect to the total number of multimedia data N, making LRGA ranking algorithm suitable for large-scale multimedia retrieval.

Algorithm-1. The Procedure of LRGA

```
1) Define the diagonal matrix $U$: set $U_{ii} = \infty$, if $y_i = 1$,
   and $U_{ii} = 1$ otherwise.
2) Compute $L_i, i = 1, \ldots, N$, according to (17).
3) Compute $L$ according to (14).
4) Solve the linear equation in (16) and the ranking result
   is given by: $f = (L + U)^{-1} Uy$.
5) Sort multimedia data according to $f = [f_1, f_2, \ldots, f_N]^T$
   in a descending order and return the top ranked ones
   as results.
```

As a transductive method, the proposed algorithm shown above can only deal with the cases when the query examples are inside the database. If the query example submitted by a user is not in the database, a straightforward way is to extend the size of database to $N+1$ and recompute the matrix L. After that, the situation is the same as when the query example is inside the database. However, it is time consuming to compute L online. Alternatively, we propose a simple but effective algorithm to deal with the query examples outside the database. After a user submits a query example Q which is outside database, the system first finds its k-nearest neighbors set $QN = \{qn_1, qn_2, \ldots, qn_k\}$ according to the euclidean distance of multimedia data feature. If $x_i \in QN$, we set $y_i = 1$. Otherwise, we set $y_i = 0$. Then, the LRGA ranking algorithm shown above is performed and top-ranked multimedia data are returned to the user. The user judges the results and then performs relevance feedback. For each multimedia datum x_i, if it is a positive sample marked by the user, we set $y_i = 1$ and, otherwise, we set $y_i = 0$. Then we reperform the LRGA ranking algorithm using the updated y to rerank the multimedia data for a new round of retrieval. In this case, we only need to solve the linear equation online, for which the time complexity is approximately linear [7], [27].

2.4 Relation with Previous Work

Relation with manifold ranking [43]. We first compare LRGA with Manifold Ranking [43]. LRGA and MR can be unified into the same formulation in (15), but the Laplacian matrices are different. The Laplacian matrix L_{MR} in MR is defined based on the Gaussian function. Let us denote an adjacency matrix W as

$$W_{ij} = \begin{cases} \exp(-\frac{1}{\delta}\|x_i - x_j\|^2), & \text{if } x_i \in \mathcal{N}_k(x_j) \text{ or } x_j \in \mathcal{N}_k(x_i), \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

The normalized Laplacian matrix L_{MR} in MR is then defined as

$$L_{MR} = I - D^{-1/2} W D^{-1/2}, \tag{19}$$

where D is a diagonal matrix with its element $D_{ii} = \sum_j W_{ij}$.

As reported in [30], the performance of the Laplacian matrix defined in (19) is sensitive to the bandwidth parameter δ in (18). In contrast, the Laplacian matrix in LRGA is learned by local regression and global alignment and our experiments demonstrate that the Laplacian matrix used in LRGA ranking algorithm is relatively robust to the parameter λ. The experiments also demonstrate that LRGA outperforms MR [43] for multimedia retrieval.

Relation with local learning [32]. The ranking algorithm in [43] has been successfully converted to a classification algorithm [44]. Similarly, we can also extend LRGA ranking algorithm to develop a transductive classification algorithm. To this end, another related approach is the Transductive Classification via Local Learning (Local Learning) algorithm [32]. For each data point x_i, another local linear regression model $h_i'(x) = w_i^T(x - x_i) + b_i$ was used in Local Learning [32]. Local Learning adopted a two-step approach to learn a Laplacian matrix. In the first step, each local linear model $h_i'(x)$ was learned by minimizing the following objective function:

$$\min_{w_i \in \mathbb{R}^d, b_i \in \mathbb{R}} \lambda\|w_i\|^2 + \sum_{x_j \in \mathcal{N}_k(x_i) \wedge j \neq i} \big(w_i^T(x_j - x_i) + b_i - f_j\big)^2. \tag{20}$$

Once the local linear model $h_i'(x)$ was learned, it was applied to predict the class label $h_i'(x_i)$ for the sample x_i only. In the second step, the sum of local prediction errors of all the data points was minimized to compute the Laplacian matrix. The objective function of this step can be written as

$$\min_f \sum_{i=1}^N \big(f_i - h_i'(x_i)\big)^2. \tag{21}$$

[3] The matlab code of the LRGA ranking algorithm can be downloaded from http://feipingnie.googlepages.com/LRGA_ranking.m.

LRGA is different from Local Learning [32] in the following aspects: 1) Local Learning adopts a *two-step approach* to learn a Laplacian matrix and it does not have a unified objective function for optimization. If the models are not well trained in the first step, the performance of the second step will degrade as well. In contrast, LRGA has a unified objective function (i.e., (6)), and the Laplacian matrix L can be directly obtained within *only one step* using (14). 2) In Local Learning, each local model $h'_i(x)$ is only applied to a single data point, i.e., x_i, to minimize its *local prediction error* with respect to $h'_i(x)$, i.e., $(f_i - h'_i(x_i))^2$ in (21). In contrast, we minimize *local model error* of $h_i(x)$ in LRGA, which is the sum of the local prediction errors from all the samples in the set $\mathcal{N}_k(x_i)$, as defined in (4). We argue that the local model error can better characterize the capability of the local linear model by counting the errors from all the neighboring data points rather than just a single point as in [32]. The experiments demonstrate that LRGA outperforms Local Learning for data ranking in different multimedia retrieval applications.

Relation with local linear embedding (LLE) [24]. In [24], Roweis and Saul have proposed a manifold learning algorithm Local Linear Embedding. Given a datum x_i and its k nearest neighbors $\{x_{i_1}, \dots x_{i_k}\}$, LLE first learns a weight vector $\tilde{w}_i = [\tilde{w}_{i_1}, \dots, \tilde{w}_{i_k}] \in \mathbb{R}^k$ for each datum x_i by minimizing the following:

$$\left\| x_i - \sum_{j=1}^{k} \tilde{w}_{i_j} x_{i_j} \right\|^2, \quad s.t. \sum_{j=1}^{k} \tilde{w}_{i_j} = 1. \tag{22}$$

Then, the low-dimensional embeddings $z_1, \dots, z_n$ of all the data $x_1, \dots, x_n$ can be obtained by minimizing the following:

$$\sum_{i=1}^{n} \left\| z_i - \sum_{j=1}^{k} \tilde{w}_{i_j} z_{i_j} \right\|^2, \tag{23}$$

where z_{i_j} is the low-dimensional embedding of the jth element of x_i's k nearest neighbors. Our algorithm differs from LLE mainly in the following aspects: First, LLE is a manifold learning algorithm whereas LRGA is a ranking algorithm. Second, LLE is a two-step approach, but LRGA has a unified objective which can be optimized in one step. Third, for any given datum x_i, LLE only uses its neighbors $\{x_{i_1}, \dots x_{i_k}\}$ to reconstruct this datum. In contrast, LRGA optimizes the local model error, i.e., the sum of prediction errors of all the data in the set $\{x_i, x_{i_1}, \dots x_{i_k}\}$. Again we argue that the local model error can better characterize the capability of local linear model by counting the errors from all the neighboring data points.

2.5 Interpretation of LRGA

First, the Laplacian matrix proposed in this paper can be also interpreted in terms of local discriminative analysis [33], and potentially applied to many other applications, such as clustering [33], semi-supervised classification, etc. Next, we provide a Bayesian interpretation of the proposed LRGA ranking algorithm [42]. Let $p(f)$ denote the prior probability of f and $p(y|f)$ denote the conditional probability of y given f. The Maximum A Posteriori (MAP) estimation is given by

$$\max_f \{\log p(y|f) + \log p(f)\}. \tag{24}$$

Suppose a model for the prior distribution $p(f)$ is given by

$$p(f) = \frac{1}{Z_r} \exp(-f^T S A S^T f), \tag{25}$$

where Z_r is a normalization constant. Further, the conditional probability is given by

$$p(y|f) = \frac{1}{Z_c} \exp(-(f-y)^T U (f-y)), \tag{26}$$

where Z_c is another normalization constant. Thus, the MAP estimator yields

$$\min_f \{f^T S A S^T f + (f-y)^T U (f-y)\}, \tag{27}$$

which is exactly the same as (15). Besides, LRGA can be interpreted in terms of label prorogation, which is given in the Appendix, which can be found online in the Computer Society Digital Library.

3 Multimedia Representations Refinement by Long-Term Relevance Feedback

In most of the existing content-based multimedia retrieval systems, multimedia data are usually represented by low-level features. Because of the well-known semantic gap, such multimedia vector representations may not reflect the multimedia semantics accurately. Given that RF provides semantic information from users, we can refine the multimedia vector representation accordingly. In this section, we present a method, namely, Trace Ratio Relevance Feedback (TRRF), for refining the multimedia representations via long-term Relevance Feedback. TRRF is a general algorithm which can be applied to many applications to infer a better multimedia representation for retrieval, classification, and so on.

Given a set of multimedia data represented by vectors $\chi = \{x_1, x_2, \dots, x_N\}$, the refinement process for multimedia representation is to learn a linear transformation from $\chi = \{x_1, x_2, \dots, x_N\}$ to $\chi' = \{x'_1, x'_2, \dots, x'_N\}$ such that for any $x'_i \in \chi'$ $(1 \le i \le N)$, we have $x'_i = W^T x_i$, where $W \in \mathbb{R}^{d \times d'}$ is the projection matrix. After the projection, each multimedia vector representation $x_i \in \mathbb{R}^d$ is transformed into $x'_i \in \mathbb{R}^{d'}$. For the sake of relevance feedback, users can either mark the results as positive or negative ones or provide label information of the returned results. We first discuss the case in which the feedback provided by users is positive and negative examples. Let $\mathcal{P}_t$ and $\mathcal{N}_t$ be the positive sample set and negative sample set marked by the user in the tth round of RF. We define

$$\mathcal{S}_t = \{(x_i, x_j) | x_i, x_j \in \mathcal{P}_t\}, \tag{28}$$

$$\mathcal{D}_t = \{(x_i, x_j) | (x_i \in \mathcal{P}_t \wedge x_j \in \mathcal{N}_t) \vee (x_i \in \mathcal{N}_t \wedge x_j \in \mathcal{P}_t)\}. \tag{29}$$

Let $\mathcal{S} = \bigcup_t \mathcal{S}_t$ and $\mathcal{D} = \bigcup_t \mathcal{D}_t$. If $(x_i, x_j) \in \mathcal{S}$ and $(x_j, x_k) \in \mathcal{S}$, we add the data pair (x_i, x_k) into $\mathcal{S}$. $\mathcal{S}$ is repeatedly updated until there is no change. After that, for any data pair $(x_i, x_j) \in \mathcal{S}, \forall (x_i, x_d) \in \mathcal{D}$, we add the data pair (x_j, x_d) into $\mathcal{D}$.

Similarly, $\forall(x_j, x_d) \in \mathcal{D}$, we add the data pair (x_i, x_d) into $\mathcal{D}$. If the feedback provided by users are class labels of multimedia data, $\mathcal{S}$ is defined as a set of data pairs in which each pair of data are of the same labels and $\mathcal{D}$ is defined as a set of data pairs in which each pair of data are of the different labels. Clearly, the data pairs from $\mathcal{S}$ are semantically similar to each other and those from $\mathcal{D}$ are semantically dissimilar to each other, which can be regarded as pairwise constraints. Using the refined multimedia vector representation $\chi' = \{x'_1, x'_2, \ldots, x'_N\}$, the sum of the squared distances of the data pairs from $\mathcal{S}$ can be calculated as follows:

$$\begin{aligned}&\sum_{(x_i,x_j)\in\mathcal{S}} (W^T x_i - W^T x_j)^T (W^T x_i - W^T x_j)\\ &= \sum_{(x_i,x_j)\in\mathcal{S}} Tr[W^T(x_i - x_j)(x_i - x_j)^T W]\\ &= Tr(W^T S_w W),\end{aligned} \tag{30}$$

where $S_w = \sum_{(x_i,x_j)\in\mathcal{S}} (x_i - x_j)(x_i - x_j)^T$ and Tr is the trace operator. Since the data pairs from $\mathcal{S}$ are semantically similar to each other, the distances among them should be as small as possible. Thus, we minimize the following objective function:

$$\min_{W^T W = I} Tr(W^T S_w W), \tag{31}$$

in which the constraint $W^T W = I$ is imposed to prevent arbitrary scaling of the projection. On the other hand, the distance between the data pairs from $\mathcal{D}$ should be as large as possible. Thus, we have the following objective function:

$$\max_{W^T W = I} Tr(W^T S_b W), \tag{32}$$

where $S_b = \sum_{(x_i,x_j)\in\mathcal{D}} (x_i - x_j)(x_i - x_j)^T$.

It has been observed that the performance of many classification algorithms can be improved by using both unlabeled data and labeled data [45]. To better refine the multimedia vector representation, we not only make use of the pairwise constraints obtained via long-term RF, but also utilize the distributions of both labeled and unlabeled data. Same as in Section 2, the linear regression model $h_i(x_j) = p_i^T x_j + c_i$, where $p_i \in \mathbb{R}^{d\times d'}$ is the local projection matrix and $c_i \in \mathbb{R}^{d'\times 1}$ is the bias vector, is used to predict the refined vector representation x'_j which is the lower dimensional representation of $x_j \in \mathcal{N}_k(x_i)$. Similarly, the local prediction error of the model with respect to the data point $x_j \in \mathcal{N}_k(x_i)$ is given by $\| p_i^T x_j + c_i - x'_j \|^2$. Note that there is a linear transformation between x_j and x'_j, i.e., $x'_j = W^T x_j$. The local prediction error can be written as $\|p_i^T x_j + c_i - W^T x_j\|^2$. Similarly to Section 2, we also minimize the following local model error of all the local models to obtain the transformation matrix W:

$$\begin{aligned}&\min_{p_i|_{i=1}^N, c_i|_{i=1}^N, W} \sum_{i=1}^{N} \sum_{x_j\in\mathcal{N}_k(x_i),} \| p_i^T x_j + c_i - W^T x_j \|^2 + \lambda Tr\big(p_i^T p_i\big)\\ &\quad s.t. \quad W^T W = I.\end{aligned} \tag{33}$$

Using a similar deduction as in Section 2, we obtain the following objective function:

$$\min_{W^T W = I} Tr(W^T X L X^T W), \tag{34}$$

where L is defined in (14). Let us define

$$A = \frac{S_w}{Tr(S_w)} + \alpha \times \frac{XLX^T}{Tr(XLX^T)}, \tag{35}$$

in which α is the tradeoff parameter to balance the two terms. Therefore, we need to minimize $Tr(W^T A W)$ and, in the meantime, maximize $Tr(W^T S_b W)$.

Many algorithms such as LDA [14], SDA [6], Semantic Subspace Projection (SSP) [37], and our TRRF aim to solve for a projection matrix W by simultaneously minimizing a term $Tr(W^T S_m W)$ while maximizing another term $Tr(W^T S_n W)$, where the positive semidefinite matrices S_m and S_n are defined according to different criteria in different algorithms. As shown in [31], it is more natural to formulate such problem as a *trace ratio* optimization problem, namely,

$$W^* = \arg\max_{W^T W = I} \frac{Tr(W^T S_n W)}{Tr(W^T S_m W)}. \tag{36}$$

However, there is no closed-form solution for this trace ratio problem [31]. In LDA [14], SDA [6], and SSP [37], researchers altered it to a more tractable *ratio trace* optimization problem as follows:

$$W^*_{app} = \arg\max_{W^T W = I} Tr[(W^T S_m W)^{-1}(W^T S_n W)], \tag{37}$$

which can be solved by using generalized eigenvalue decomposition. The recent work [31] has experimentally demonstrated that better performances can be achieved in pattern recognition if we solve the original trace ratio optimization problem shown in (36) as opposed to the ratio trace problem shown in (37). Therefore, we propose the following objective function for long-term RF:

$$\max_{W^T W = I} \frac{Tr(W^T S_b W)}{Tr(W^T A W)}. \tag{38}$$

Wang et al. [31] have proposed an iterative algorithm to solve the trace ratio optimization problem in (38), which is summarized in Algorithm 2

In Algorithm 2, the most time consuming operation is the eigenvalue decomposition in step 4, whose time complexity is $O(d^3)$. Denote $ev_1, ev_2, \ldots, ev_d$ as the eigenvectors of $S_b - \lambda_n A$. In [18], researchers have theoretically analyzed Algorithm-2 and they pointed out that a faster convergence can be achieved by modifying step 4 in Algorithm-2 to computing the root of the equation $f(\eta) = 0$, where $f(\eta)$ is defined as

$$f(\eta) = \max_{\breve{W}_n} Tr\big(\breve{W}_n^T (S_b - \eta A)\breve{W}_n\big), \tag{41}$$

where $\breve{W}_n \in \mathbb{R}^{d\times d'}$ and each column of $\breve{W}_n$ is selected from $ev_1, ev_2, \ldots, ev_d$. To solve the above problem, there are several approaches. However, how to compute the root of $f(\eta) = 0$ has not been discussed in [18]. Inspired by Nie et al.

[23], we apply an efficient method to solve this problem and the algorithm is listed in Algorithm 3.

Algorithm-2. Solving the trace ratio problem [31]

```
1) Initialize $W_0$ as an arbitrary columnly orthogonal
   matrix such that $W_0^T W_0 = I$ and set $n = 1$.
2) Repeat Step 3 to Step 5 until convergence.
3) Compute $\lambda_n$ defined as follows:
       $$\lambda_n = \frac{Tr(W_{n-1}^T S_b W_{n-1})}{Tr(W_{n-1}^T A W_{n-1})}. \quad (39)$$
4) Solve the following trace difference maximization
   problem to obtain $W_n$ by performing
   eigen-decomposition of $(S_b - \lambda_n A)$:
       $$W_n = \arg\max_{W^T W = I} Tr[W^T (S_b - \lambda_n A) W]. \quad (40)$$
5) Set $n = n + 1$.
6) Output $W^* = W_n$.
```

Note that the difference between the algorithm proposed in [31] and Algorithm-3 is step 4. In Algorithm-3, we have leveraged the trick in [23] to accelerate the convergence speed. Although the algorithm in [23] was originally proposed for feature selection and the scenario is different from ours, the justification in [23] can be borrowed to justify Step-4 in Algorithm-3. Note that Step 4 in Algorithm-3 converges very fast. The time complexity of steps 4-i and 4-ii in Algorithm-3 is $O(d^2)$. Since the time complexity of eigen-decomposition is $O(d^3)$, the total time complexity of step 4 in Algorithm-3 is $O(d^3)$ approximately. Similarly, the time complexity of step 4 in the algorithm proposed in [31] is $O(d^3)$ as well. Therefore, Algorithm-3 is faster than Algorithm-2. In the experiment, we observe for any matrix of size 500×500 (i.e., the multimedia feature dimension is 500), our algorithm converges with fewer iterations and it is faster than the algorithm proposed in [31]. If the feature dimension of the multimedia data increases, more improvement in speed can be gained by using Algorithm-3[4] proposed in this paper.

Algorithm-3. Fast solver of the trace ratio problem

```
1) Initialize $W_0$ as an arbitrary columnly orthogonal
   matrix such that $W_0^T W_0 = I$ and set $n = 1$.
2) Repeat Step 3 to Step 5 until convergence.
3) Compute $\lambda_n$ defined in (39).
4) Compute the $d$ eigenvectors $e_{n1}, e_{n2}, \ldots, e_{nd}$ of
   $S_b - \lambda_n A$. Set $\eta = \lambda_n$, and repeat the following two
   operations until there is no change to $W_n$:
   i) Sort $P_i = e_{ni}^T (S_b - \eta A) e_{ni}, i = 1, \ldots, d$, in
      descending order and select the first $d'$
      eigenvectors to construct $W_n$.
   ii) Compute $\eta = \frac{Tr(W_n^T S_b W_n)}{Tr(W_n^T A W_n)}$.
5) Set $n = n + 1$.
6) Output $W^* = W_n$.
```

4 Applications to Content-Based Multimedia Retrieval

The proposed semi-supervised scheme can be applied to many content-based multimedia classification and retrieval applications. Here, we discuss three diverse and representative problems: image retrieval, 3D motion/pose data retrieval, and cross-media retrieval.

The proposed framework can be readily used for image retrieval and 3D motion/pose data retrieval. Here, we take image retrieval as an example (3D motion/pose data retrieval is similar). The images are first represented by their visual features (e.g., color histogram, texture), and the Laplacian matrix in (14) is then computed for data ranking. When the user submits a query image, we perform the LRGA ranking algorithm to assign ranking scores to database images such that the top-ranked images are returned to the user.

If the query example is outside the database, we first use the kNN method to search inside the database the T nearest neighbors of the query image. We set $y_i = 1$ if x_i is one of the T nearest neighbors of the query example and $y_i = 0$ otherwise [46]. Next, we perform the LRGA algorithm for image retrieval. We can also recalculate the Laplacian matrix by using the database images and the query examples, and then the system can perform ranking using the LRGA algorithm based on the recalculated Laplacian matrix. Note that the latter is not practical for large-scale retrieval applications. In what follows, we denote the two types of search strategies as LRGA-kNN and LRGA-newLap, respectively. The user can also conduct short-term RF by marking some positive images from the top-ranked results. We set $y_i = 1$ for each marked positive image x_i, and then perform another round of retrieval using the LRGA method to further improve the image retrieval precision [46].

Cross-media retrieval deals with an unconventional type of multimedia retrieval tasks in which the returned results can be of different modalities from the query. For example, the user can query images of an animal by submitting its sound. For this reason, here we describe the application in finer detail than the other two. Following [36], [34], and [46], we define a Multimedia Document (MMD) as a set of co-occurring multimedia objects (e.g., images, audio records, and text records) that are of different modalities but carry the same semantics. If a multimedia object obj_q belongs to an MMD MMD_i, obj_q is the ***affiliated multimedia object*** of MMD_i and MMD_i is the ***host MMD*** of obj_q. To apply the proposed framework to cross-media retrieval, we need to compute the vector representations of MMDs. The algorithm proposed in [34] is adopted to construct a Multimedia Correlation Space (MMCS), in which each MMD is represented as a point. In [34], the pairwise distance between any two MMDs is first calculated by linearly fusing the distances based on image, audio record, and text record features. After that, Multidimensional Scaling (MDS) is used to obtain the vector presentation of MMDs in MMCS [34]. In that way, each MMD MMD_i is represented as a data point x_i in MMCS. For more details, please refer to [34]. With the vector representations $\chi = \{x_1, x_2, \ldots, x_N\}$ for all MMDs, we can compute the Laplacian matrix defined in (14) and apply the proposed framework to cross-media retrieval.

First, we assume that the query example submitted by a user is a multimedia object or MMD in the database. We can directly use the LRGA ranking algorithm for cross-media

[4] The matlab code can be downloaded from http://feipingnie.googlepages.com/traceratio_fast.m.

retrieval. For example, if a user queries audio records by an image example Img_q, we first find the host MMD MMD_i of Img_q. We set $y_i = 1$ if MMD_i is the host MMD of the query example, and set $y_i = 0$ otherwise. After performing the LRGA ranking algorithm, each MMD in the database obtains a ranking score. We sort all the MMDs in database according to the ranking scores and find the top c MMDs $MMD_1, \dots MMD_c$. Finally, l affiliated audio objects $A_1, \dots, A_l$ of $MMD_1, \dots, MMD_c$ are returned as results.

Next, we describe the methods for cross-media retrieval when the query examples are outside the database. Suppose the query example is an image Img_q. We first find its T nearest neighbors $TImg = \{Img_1, Img_2, \dots, Img_T\}$ which are in the database according to image feature distances. Then, we set $y_i = 1$ if MMD_i is the corresponding host MMD of at least one of the image in $TImg$, and set $y_i = 0$ otherwise. After that, we perform the LRGA ranking algorithm to obtain the ranking scores for all MMDs and the problem is the same as before. Another possible solution is to treat the query examples outside the database as within the database by recalculating the Laplacian matrix using the database samples and the query examples. When comparing the above two methods, our experiments in Section 5 demonstrate that our LRGA method using the recalculated Laplacian matrix generally leads to better retrieval performance. However, the computational cost increases, thus making it unsuitable for large-scale retrieval applications. After the results are returned to the user, the user marks positive and negative examples and the system performs short-term relevance feedback in order to further improve the retrieval performance. According to the positive examples marked by the users, we can easily find the corresponding positive MMDs. Let $PMMD$ be the set of positive MMDs which are the host MMDs of the positive examples marked by user. The vectors $y \in \mathbb{R}^N$ are initialized as a zero vector. For each MMD $MMD_i \in PMMD$, we set $y_i = 1$. We then run the LRGA algorithm to rerank the data and return top-ranked multimedia objects to the user.

In the above three applications, the system also keeps all the RF records from different users. Long-term RF, discussed in Section 3, can be periodically performed to refine the vector representation for a better retrieval performance afterward.

5 Experiments

We first use two toy examples to compare our LRGA ranking algorithm with the euclidean distance-based ranking method and the manifold ranking method [43]. Then, we test our proposed methods for three applications including cross-media retrieval, image retrieval, and 3D motion/pose data retrieval. When the query examples are inside the database, we first compare our LRGA ranking algorithm with MR, the manifold learning algorithm LTSA [41], as well as the transductive classification algorithm Local Learning [32]. Even though LTSA and LL are not originally used for data ranking, we still compare our LRGA with them because the Laplacian matrices proposed in LTSA [41] and LL [32] can be readily used in (15) for data ranking. For MR, we set its parameter δ at different values, i.e., $\{0.0001, 0.01, 1, 100, 10000\}$, and report the best results. For LTSA, we observe that the performance is sensitive to the dimension of the local tangent subspace. We tune this parameter as $\{1, 2, 3, \dots, 30\}$ and report the best results. As shown in the initial conference version [34], LRGA is relatively insensitive to the parameter λ in (4), so we only set it as $\{1, 1000\}$ and report the best results. Similarly to LRGA, LL is relatively insensitive to the regularization parameter. In the experiment, we tune the regularization parameter of LL from $\{0.0001, 0.01, 1, 100, 10000\}$ and report the best results. Moreover, we also compare LRGA with the state-of-the-art cross-media retrieval methods in [36], [46] and the image retrieval methods [28], [29]. We are not aware of related learning algorithms that are specifically designed for 3D motion/pose data retrieval.

In order for LGRA to cope with the query examples that are outside the database, we can use the nearest neighbors, which are inside the database, of the query example to initialize y or recalculate the Laplacian matrix by using both the database samples and the query samples (see Section 4 for more details on LRGA-kNN and LRGA-newLap). Moreover, we report the results from our short-term RF method discussed in Section 4. Because LRGA is a transductive ranking algorithm, we use the query samples that are inside the database as an example to compare our long-term relevance feedback method, referred to as Trace Ratio Relevance Feedback, with the related methods, including the supervised algorithm Linear Discriminant Analysis and two semi-supervised learning algorithms, Semi-supervised Discriminant Analysis [6] and Semantic Subspace Projection [37]. Note that after the multimedia vector representation has been refined by long-term RF, LRGA (denoted to as TRRF-R) or euclidean distance-based ranking (denoted to as TRRF-E) can be performed for retrieval. LDA, SDA, and SSP directly utilize euclidean distance for retrieval. To show the advantages of TRRF over other competitors, we report the retrieval results of both TRRF-R and TRRF-E. We also test the sensitivity of our framework with respect to the algorithm parameters and study the scalability of our LRGA ranking algorithm, as well as investigate the effectiveness of using the unlabeled samples in relevance feedback. At last, we compare the computation speed of the proposed algorithm with previous works in [34] and [31].

In our experiments, if a returned result and the query example are in the same semantic category, it is regarded as a correct result. Precision is defined as the percentage of correctly retrieved samples in the top-k returned results. In all the figures, the precision is the averaged precision values from all the query samples.

5.1 Toy Problems

We first use two toy examples to compare our ranking algorithm LRGA with the euclidean distance-based ranking method and the manifold ranking method [43], where we set k to 10. In all the figures of this section, the red point is the query and the marker size of each data point is proportional to its ranking score.

Fig. 1 compares the ranking results on the Swiss Roll data using the LRGA algorithm and the euclidean distance. To rank data according to euclidean distance, the reciprocal of the distance between a given database sample and

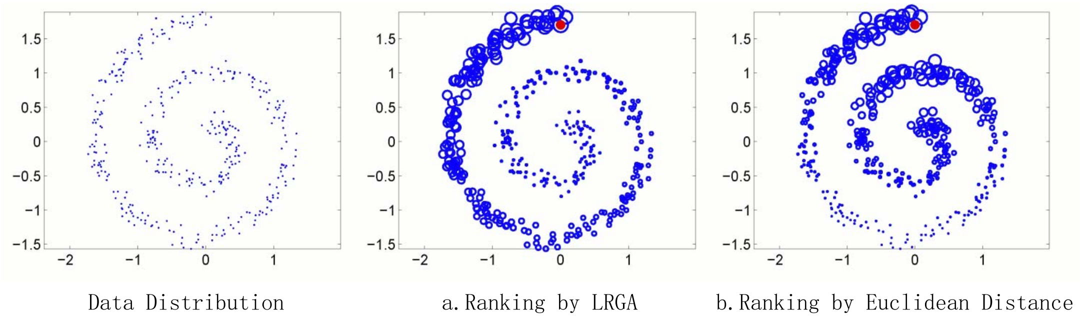
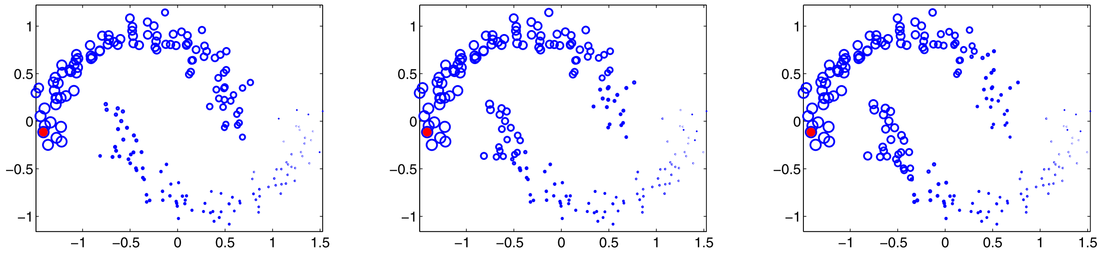
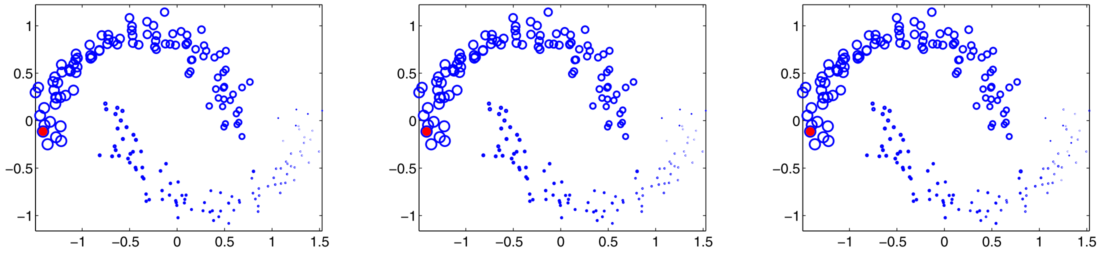


Fig. 1. A comparison of data ranking on the Swiss Roll data set using euclidean distance and the LRGA ranking algorithm. The red point is the query. The marker size of each data point is proportional to the ranking score.



(a) $\delta = 0.25$ (b) $\delta = 0.5$ (c) $\delta = 1$

Fig. 2. Data ranking using the manifold ranking algorithm [43]. This figure shows the performance variation with different parameter δ. The red point is the query. The marker size of each data point is proportional to the ranking score.



(a) $\lambda = 1$ (b) $\lambda = 10^8$ (c) $\lambda = 10^{16}$

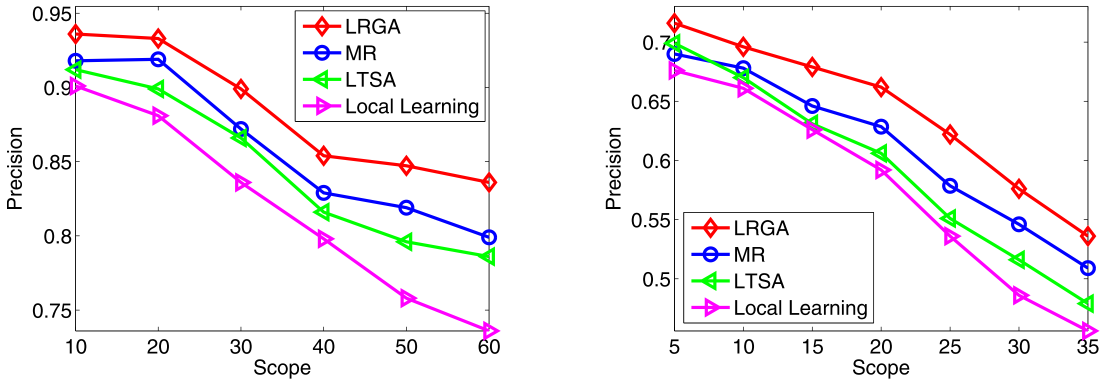
Fig. 3. Data ranking using our LRGA algorithm. This figure shows the performance variation with different parameter λ. The red point is the query. The marker size of each data point is proportional to the ranking score.

the query is used as the ranking score. In Fig. 1, the euclidean distance-based ranking method fails to preserve the Swiss Roll structure. It is because the distance-based ranking algorithm only considers pairwise distance but ignores the whole data distribution. In contrast, the ranking scores decrease smoothly along the Swiss Roll with the LRGA algorithm, demonstrating that the LRGA algorithm is more robust for ranking data that lie on a complicated manifold structure.

We also compare LRGA with Manifold Ranking on a toy problem in terms of the robustness to the parameters. The samples in this toy problem are from two manifolds (i.e., two moons) and the samples from the same class of the query example are expected to be assigned higher ranking scores (i.e., larger marker sizes), compared with the samples from different classes. From Fig. 2, we observe that the ranking results of MR are sensitive to the bandwidth parameter δ in the Gaussian function for this toy. When $\delta = 0.25$, the ranking result is fairly good. However, if we increase δ to 0.5, the ranking results become poor. For these toy data, MR only works when $\delta \in [0.05, 0.3]$. Fig. 3 shows the ranking results using the LRGA algorithm with different values of the parameter λ in (5). As can be seen, the LRGA algorithm works well when λ is within $[1, 10^{16}]$. We observe in the experiment that LRGA is less sensitive to the parameter λ for most of the toy data, compared with manifold ranking [43]. Clearly, ranking algorithms that are not sensitive to parameters are more suitable for real-world retrieval tasks.

5.2 Cross-Media Retrieval Application

We used 2,160 multimedia objects, which are collected from Multimedia Cyclopedia, science, educational and E-business webpages, documentary and educational films, and so on. These multimedia objects are divided into two nonoverlapping groups. The first group is comprised of 2,020 multimedia objects (including 1,000 images, 300 audio records, and 720 text records) from 1,000 multimedia documents. The 1,000 MMDs are from 10 semantic categories and each semantic category contains 100 MMDs. The second group is comprised of 140 multimedia objects, including 100 images and 40 audio records, also from the 10 semantic categories, with each category containing 10 to 18 multimedia objects. The multimedia objects in the first group are all used to construct the Multimedia Correlation Space. We use 500 multimedia objects, including 380 images



(a) Query images using audio records

(b) Query audio records using images

Fig. 4. A comparison of our ranking algorithm LRGA with MR [43], LTSA [41], and Local Learning [32] for cross-media retrieval.

and 120 audio records, as query examples to evaluate the cross-media retrieval performance. Among the 500 test data, 360 multimedia objects (including 280 images and 80 audio records) are selected from the first group to test the cross-media retrieval performance when the query is in the database, and the 140 multimedia objects are from the second group to test the performance when the query example is not in the database. For image objects, three types of color features (color histogram, color moment, and color coherence) and three types of texture features (Tamura coarseness histogram, Tamura directionality, and MSRSAR texture) are used. For audio records, four types of features (RMS energy, Spectral Flux, Rolloff, and Centroid) are used. For text records, we use TF/IDF feature. To obtain the vector representation of each MMD in the MMCS, we need to calculate the distances based on image, audio record, and text record features. In this work, the euclidean distance and the cosine distance are used for images and text records, respectively. We use Dynamic Time Warping to compute the distance for audio records. More information about the above features can be found in [34].

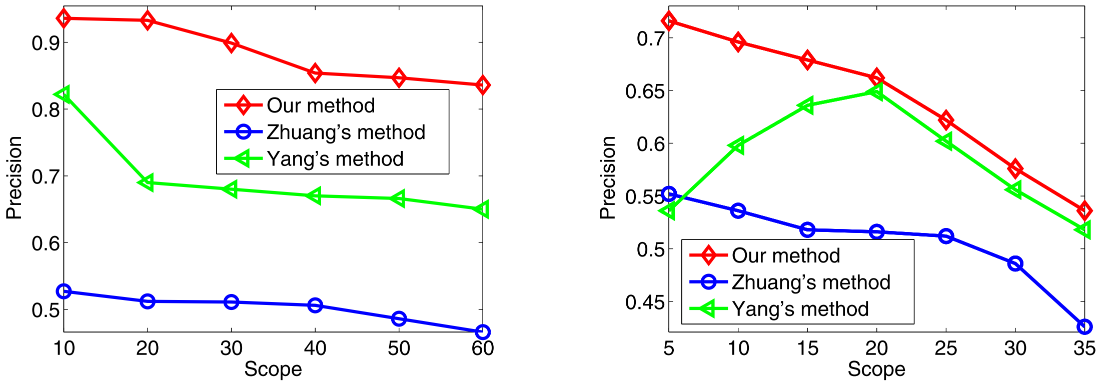
5.2.1 When the Query Is Inside the Database

First, we conduct experiments to test the performances of cross-media retrieval when the query is inside the database. In our experiment, RF is not conducted when the query example is inside the database. Fig. 4a shows the precision of querying images by audio records which are inside the database using different ranking algorithms, and Fig. 4b shows the precision of querying audio records by images which are inside the database. In this experiment, we fix k as 15 for all the ranking algorithms. For MR, we apply the Gaussian function to compute a normalized Laplacian matrix based on the MMD distance defined in [34] for data ranking. Considering that the number of audio records is smaller than that of image objects in our database, the scope of Fig. 4b is smaller than Fig. 4a. From Figs. 4a and 4b, it is clear that our method consistently outperforms the other three methods.

Next, we compare our method with the recent cross-media retrieval methods in [36], [46]. Fig. 5a shows the precision of querying images by audio records which are inside the database. Fig. 5b shows the precision of querying audio records by images which are inside the database. From Figs. 5a and 5b, we observe that our framework consistently outperforms [36], [46].

5.2.2 When Query Example Is Outside the Database

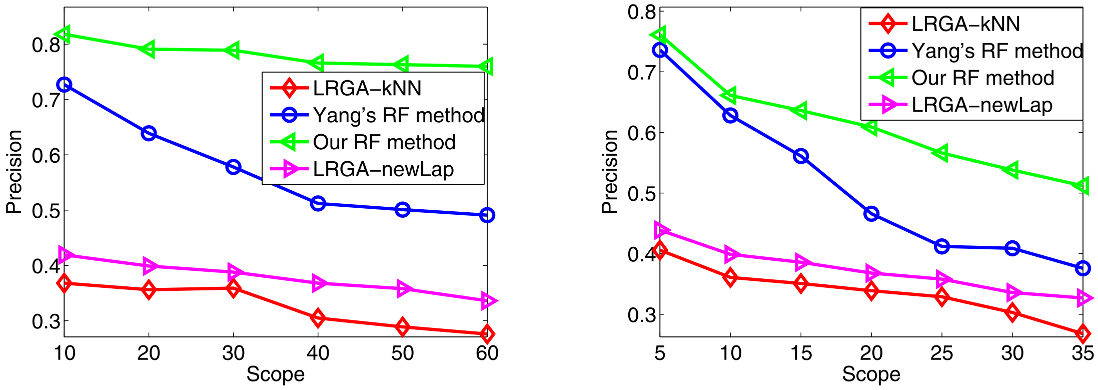
We test the performance of the proposed method when the query example is outside the database, in which we use the same experimental setting as in our initial conference work [34]. In this case, the initial retrieval result may not be good without using relevance feedback. Similarly to the previous work on cross-media retrieval [36], we also make use of short-term RF to learn the user's search intention in our framework. Fig. 6 shows the cross-media retrieval precisions



(a) Query images using audio records

(b) Query audio records using images

Fig. 5. A comparison of our ranking algorithm LRGA with Yang's method [36] and Zhuang's method [46] for cross-media retrieval.



(a) Query images using audio records

(b) Query audio records using images

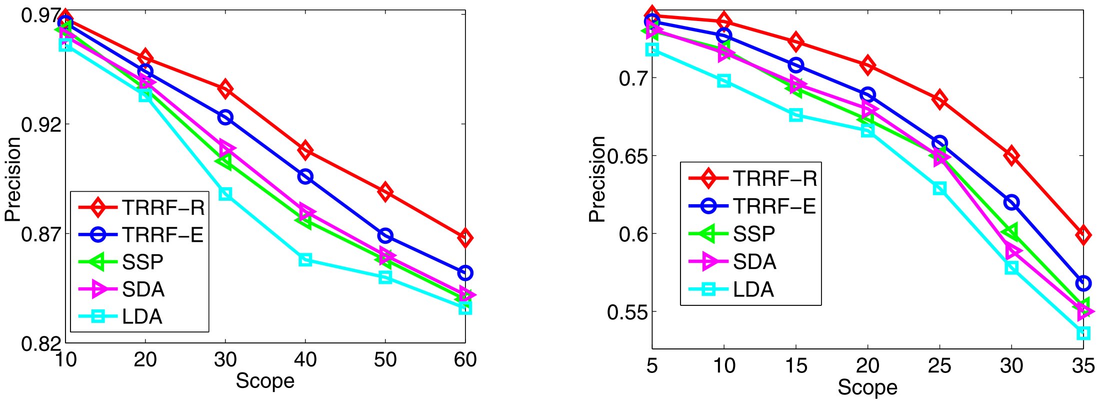
Fig. 6. Comparison of cross-media retrieval methods before conducting short-term RF (i.e., LRGA-kNN and LRGA-newLap) and after performing short-term RF of our method and Yang's RF method in [36]. The query example is outside the database.

using LRGA-kNN and LRGA-newLap *before* conducting RF, as well as the results after performing short-term RF with our method discussed in Section 4 and the algorithm in [36]. We have the following observations: 1) When the query example is outside the database, the cross-media retrieval precision is not as high as that when the query example is inside the database. Compared with LRGA-kNN, LRGA-newLap achieves better retrieval precision, but it is less efficient for large-scale retrieval task because it is time consuming to recalculate the Laplacian matrix for every query data. 2) The cross-media retrieval performance can be significantly improved by using short-term RF. With more information provided by the user, it becomes easier to understand the semantic relationship among different media types. 3) Our short-term RF method outperforms the algorithm proposed in [36].

5.2.3 Long-Term RF in Cross-Media Retrieval

Now, we test the performance of the proposed long-term RF algorithm TRRF, which is used to refine the vector representation of MMDs. We also report the cross-media retrieval precisions using the supervised learning method, Linear Discriminant Analysis [14], and two semi-supervised algorithms, Semi-supervised Discriminant Analysis [6] and Semantics Subspace Projection [37]. Note that SDA and SSP are semi-supervised algorithms that employ the unlabeled database samples when learning the projection matrices. As discussed previously, TRRF can make use of both the label information and pairwise relationship information. Similarly, SSP [37] can take both label information or pairwise relationship information as input. In contrast, both LDA [14] and SDA [6] require label information during the training. Therefore, in this experiment, we ask the users to provide label information during RF. When the information from RF accumulates, the performance of the long-term RF algorithms becomes better and better. In order to demonstrate the effectiveness of our TRRF, we ask users to label five returns during each round of RF. We only perform two rounds of RF for each semantic category. Therefore, for each semantic category, only $\frac{1}{10}$ MMDs are labeled for training. Note that LDA, SDA, and SSP all directly use euclidean distance for data ranking. After performing the long-term relevance feedback method TRRF, a refined multimedia representation can be obtained. We can perform the LRGA ranking algorithm for retrieval based on the refined multimedia representation (denoted as TRRF-R). We can also compute euclidean distance according to the refined multimedia representation for retrieval as well (denoted as TRRF-E). In this section, we not only report the result of TRRF-R but also additionally report the result of TRRF-E.

The results are shown in Fig. 7. In this experiment, we have the following observations:



(a) Query images using audio records

(b) Query audio records using images

Fig. 7. A comparison of different long-term RF algorithms for cross-media retrieval.

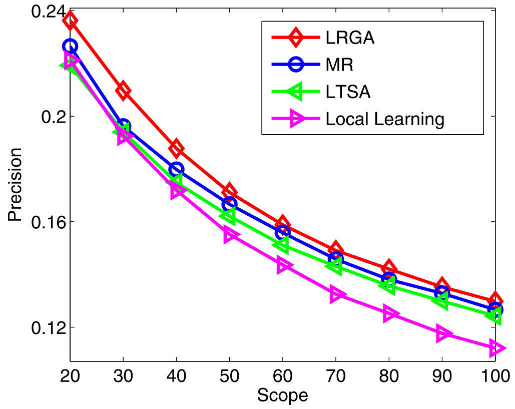


Fig. 8. A comparison of our ranking algorithm LRGA with MR [43], LTSA [41], and Local Learning [32] for content-based image retrieval.

1. Cross-media retrieval precision is significantly improved after using long-term RF based on only two rounds of history RF information for each category.
2. Compared with LDA, the semi-supervised learning methods SDA and SSP can achieve better retrieval performances because of the utilization of the unlabeled database samples.
3. TRRF-E outperforms LDA, SDA, and SSP due to the following two aspects: First, the objective function in TRRF is formulated as the trace ratio form, but LDA, SDA, and SSP are based on the ratio trace criterion. Therefore, TRRF can obtain a better refined multimedia vector representation. The observation that trace ratio criterion is better than ratio trace criterion is consistent with the finding of the existing work in [31]. Second, our proposed regularizer can effectively utilize the manifold structure of both labeled and unlabeled multimedia data.
4. TRRF-R achieves the best performance because LRGA ranking algorithm is a better choice for data ranking compared with distance-based ranking.

5.3 Image Retrieval Application

In the same setup as in the recent image retrieval works [6], [11], [37], [39], we use the Corel image data set in this experiment. Since many concepts have very few images, we only choose 50 concepts that contain the largest number of samples. Specifically, we use the subset in [11], which contains 4,999 images. Again, we extract three types of color features (color histogram, color moment, and color coherence) and three types of texture features (Tamura coarseness histogram, Tamura directionality, and MSRSAR texture). We divide the 4,999 images into two nonoverlapping subsets. The first subset of 4,799 images is used to compute the Laplacian matrix for data ranking, and the second subset contains the remaining 200 images. We select 400 images as the query examples, including 200 images chosen from the first subset and all 200 images from the second subset, which are used to test the image retrieval performance when the query is inside the database and when the query is outside the database, respectively.

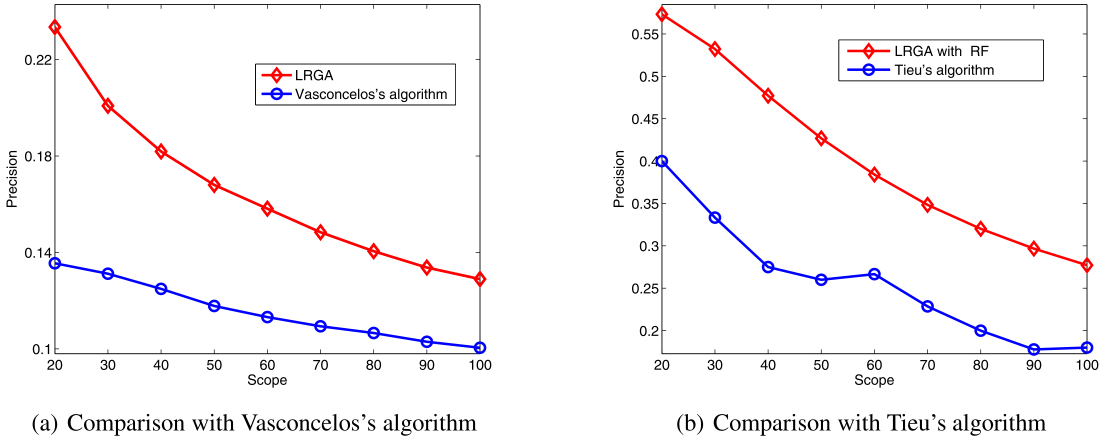
5.3.1 When Query Example Is Inside the Database

We first compare our algorithm with different data ranking algorithms, including MR [43], LTSA [41], and Local Learning [32], when the query example is inside the database. In this experiment, we set k as 15 for all of the four algorithms. From Fig. 8, we observe that our algorithm consistently outperforms the other three algorithms.

In Fig. 9, we compare our algorithm with two state-of-the-art image retrieval methods [28], [29]. Note that the method in [28] is a relevance feedback algorithm, and a few positive and negative samples need to be marked by the user in the relevance feedback stage to learn an effective feature for image retrieval [28]. To fairly compare our LRGA (referred to as LRGA with RF in Fig. 9b) with [28], we use 10 randomly selected positive database examples for both algorithms. Moreover, we provide another 10 negative examples for Tieu's algorithm [28]. Figs. 9a and 9b show that LRGA outperforms both methods. It is also worth mentioning that LRGA is much faster than Tieu's algorithm.

5.3.2 When Query Example Is Outside the Database

In Fig. 10, we compare LRGA-kNN and LRGA-newLap when the query example is outside the database. We also report the retrieval result after using our proposed short-term relevance feedback algorithm discussed in Section 4, in



(a) Comparison with Vasconcelos's algorithm

(b) Comparison with Tieu's algorithm

Fig. 9. A comparison of our algorithm with Vasconcelos's algorithm [29] and Tieu's algorithm [28] for image retrieval. Note that, in (a), we only use one query each time and, in (b), we submit 10 queries for LRGA with RF and 20 queries for [28] each time.

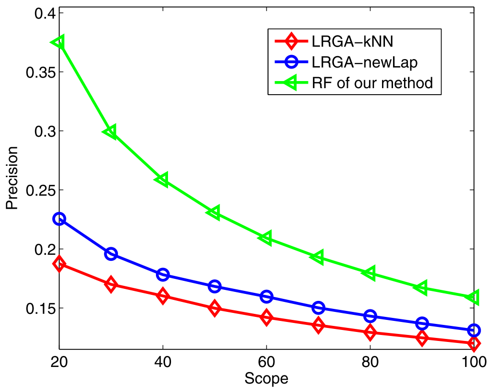


Fig. 10. Comparison of image retrieval methods before conducting short-term RF (i.e., LRGA-kNN and LRGA-newLap) and after performing short-term RF with our method. The query image is outside the database.

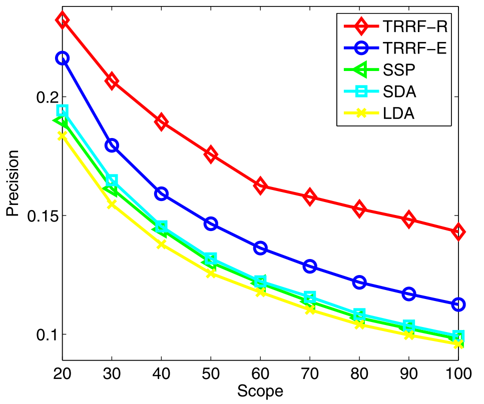


Fig. 11. A comparison of different long-term RF algorithms for image retrieval.

which five positive examples are marked by the user. Similarly as in the cross-media retrieval task, we have the following two observations: 1) LRGA-newLap outperforms LRGA-kNN, but it is unsuitable for large-scale image retrieval task because it is time consuming to recalculate the Laplacian matrix for every query image; and 2) the performance can be significantly improved after using our short-term relevance feedback algorithm. The system can better understand the users' search intention when more positive examples are provided.

5.3.3 *Long-Term RF in Image Retrieval*

We compare our long-term RF method TRRF with the existing RF algorithms for image retrieval, including Linear Discriminant Analysis [14], Semi-supervised Discriminant Analysis [6], and Semantic Subspace Projection [37]. Similarly as in cross-media retrieval, 10 percent of the database images are labeled by different users for all of the algorithms in this experiment.

From Fig. 11, we have the following observations and conclusions:

1. SDA and SSP outperform LDA due to the utilization of the unlabeled data.
2. Using the same distance-based ranking method, TRRF-E is better than SDA and SSP. There are two possible explanations. First, it is more effective to formulate the objective function in the trace ratio form for long-term RF when compared with the ratio trace form. Second, the regularizer proposed in this work can more effectively exploit the manifold structure of both labeled and unlabeled data.
3. TRRF-R achieves the best results, which further demonstrates that the LRGA ranking algorithm outperforms the distance-based ranking algorithms which are widely used in many existing image retrieval systems [6], [28], [37].
4. If we directly use euclidean distance for image retrieval, the performance can be significantly improved by using TRRF. On the other hand, if we use the LRGA ranking algorithm for image retrieval, TRRF cannot improve the retrieval performance much when the scope is small. As the scope increases, the performance improvement of TRRF becomes obvious. The explanation could be that given that the LRGA algorithm is already effective for image retrieval, it is difficult to further improve the performance by using RF.
5. Note that many existing systems use euclidean distance as the distance metric for multimedia data management. It is particularly suitable to use our TRRF algorithm in those systems to infer a more accurate vector representation for multimedia data classification, clustering, indexing, etc.

5.4 Three-Dimensional Motion/Pose Data Retrieval Application

Motion or pose retrieval is very important in many animation systems. As motion data lack annotations, the animators usually need to manually search for motion data of similar motions or poses from the database. Considering the high frame rate of HumanEva data set and the repetitive nature of motions [1], we downsample 10,000 3D motion/pose data from Human/Eva motion data set in this experiment. The 3D motion/pose data are from two different subjects. For each subject, there are five types of motion, including walking, running, jumping, and modern dancing. Therefore, there are $2 \times 5 = 10$ semantic categories in total.

A 3D pose is encoded as a collection of joint coordinates in 3D space and there are 16 joints in the HumanEva data set. Therefore, each 3D motion/pose data is represented by a $16 \times 3 = 48$-dimensional feature vector. We also divide the 10,000 3D motion/pose data into two nonoverlapping subsets. The first subset of 9,800 3D motion/pose samples is used to compute the Laplacian matrix for data ranking, and the second subset contains the remaining 200 3D motion/pose samples. We also choose 400 3D motion/pose samples as the query examples to test the retrieval performance. Among them, the 200 selected 3D motion/pose samples from the first subset are used to test the performance when the query is inside the database, and all 200 3D motion/pose samples from the second subset are used to test the performance when query is outside the database.

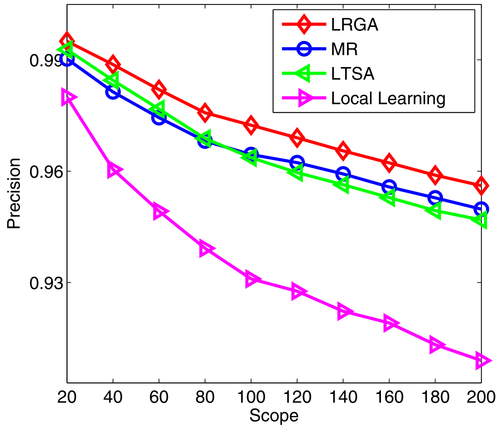


Fig. 12. A comparison of our ranking algorithm LRGA with MR [43], LTSA [41], and Local Learning [32] for 3D motion/pose data retrieval.

5.4.1 When Query Example Is Inside the Database

Fig. 12 shows the 3D motion/pose data retrieval performance by different ranking algorithms when the query example is inside the database. For LRGA, MR, LL, and LTSA, we set k as 10. Because there are 1,000 samples in each semantic category, we set the scope to 200, which is larger than that in the image retrieval application. Fig. 12 shows that our ranking algorithm LRGA achieves the best result for 3D motion/pose data retrieval, again demonstrating that LRGA can better exploit the manifold structure of multimedia data for ranking in various applications. We also observe that the retrieval precisions for the 3D motion/pose data retrieval application are higher when compared with image retrieval application. It is because the features of 3D motion/pose data are more effective for representing the pose semantics than the low-level image visual features for representing image semantics.

5.4.2 When Query Example Is Outside the Database

Fig. 13 shows the 3D motion/pose data retrieval precisions when the query examples are outside the database. Again, we report the results using LRGA-kNN and LRGA-newLap before conducting RF and after performing short-term RF with our method. It is interesting to note that the results from LRGA-kNN and LRGA-newLap are almost similar. The possible explanation is that the features for 3D motion/pose data are more effective than the visual features or audio record features used in the other two applications. In this case, the k nearest neighbors of the query example, which are used in LRGA-kNN to initialize y before conducting ranking, are often with the same semantics as the query example. We also observe that the retrieval performance can be further improved by using our short-term RF algorithm discussed in Section 4 because more information is provided.

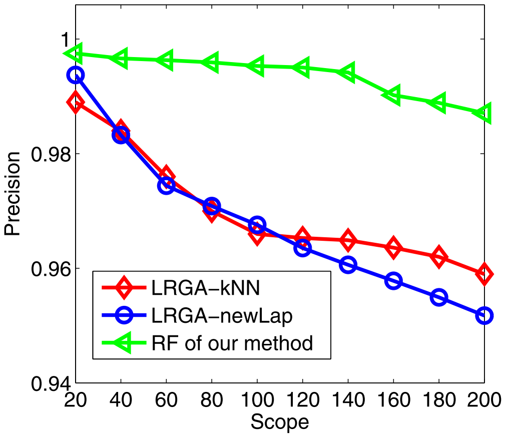


Fig. 13. Three-dimensional motion/pose data retrieval performance using LRGA-kNN, LRGA-newLap, and our proposed short-term RF method. The query example is outside the database.

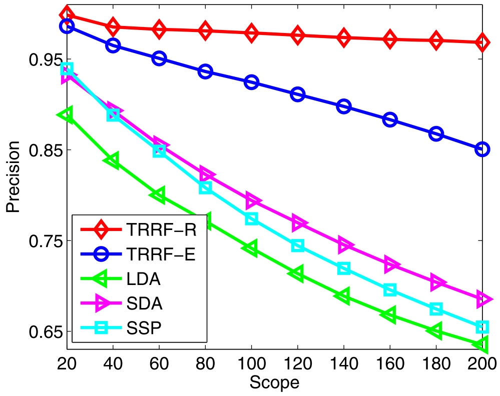


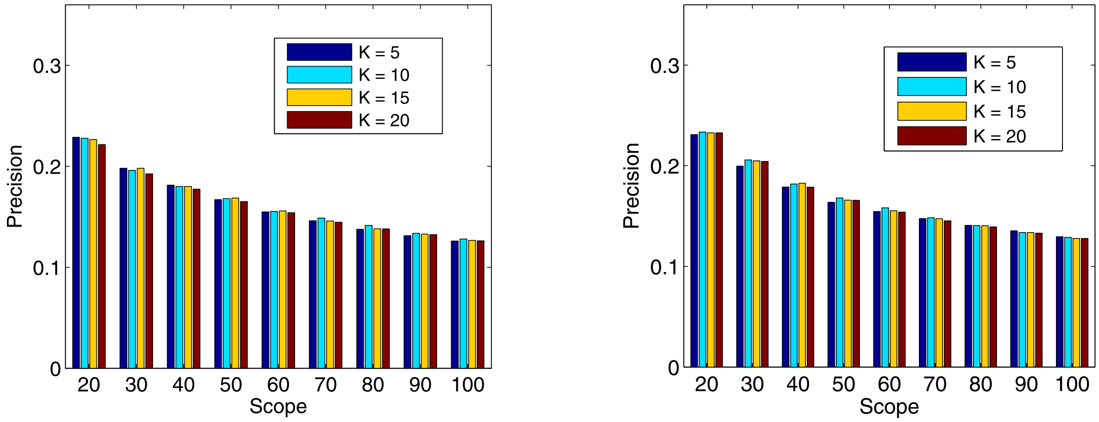
Fig. 14. A comparison of different long-term RF algorithms for 3D motion/pose data retrieval.

5.4.3 Long-Term RF in 3D Motion/Pose Data Retrieval

Since we are not aware of any long-term RF methods for 3D motion/pose data retrieval, we still compare TRRF-E and TRRF-R with LDA [14], SDA [6], and SSP [37]. Similarly as before, 10 percent of the database data is the labeled samples marked by different users during long-term RF for all the algorithms. Fig. 14 plots the precisions of different algorithms. Again, both SDA and SSP outperform LDA due to the utilization of the unlabeled data. TRRF-E is also better than SDA and SSP, further demonstrating that it is beneficial to use the trace ratio formulation and utilize the regularizer in (34) to effectively exploit the manifold structure for long-term RF. Finally, TRRF-R achieves the best results, which demonstrate again the effectiveness of our LRGA ranking algorithm.

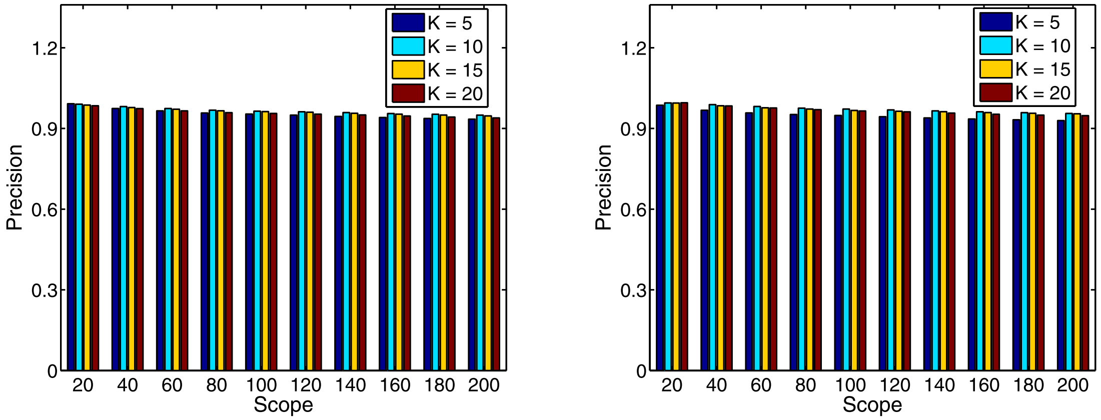
5.5 Parameter Sensitivity

We further evaluate the sensitivity of our framework with respect to its parameters by using the query examples inside the database. Given that we have experimentally demonstrated that LRGA is relatively insensitive to the regularization parameter λ in our initial work [34], here we focus on the evaluation of performance variation with respect to the parameter k, which specifies the size of the neighborhood in the local regression models, using the Corel image data set and the HumanEva data set. We set k to 5, 10, 15, and 20 because k should be set to a small number. Figs. 15a and 15b (Figs. 16a and 16b) show the retrieval precision variations of MR and our LRGA on Corel image data set (HumanEva 3D motion/pose data set) with respect to different k, respectively. As seen in Figs. 15 and 16, while the performances of



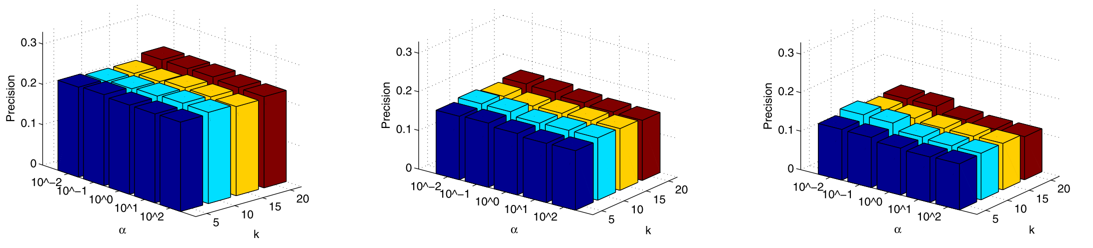
(a) MR (b) LRGA

Fig. 15. Performance variations of MR and LRGA with respect to different k on Corel image data set.



(a) MR (b) LRGA

Fig. 16. Performance variations of MR and LRGA with respect to different k on HumanEva 3D motion/pose data set.



(a) When top 20 resutls are returned (b) When top 50 resutls are returned (c) When top 100 results are returned

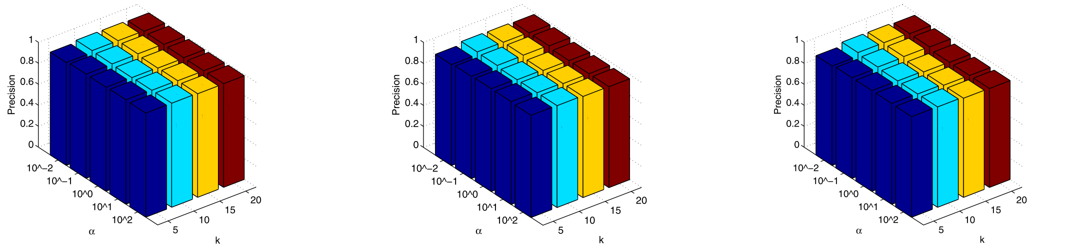
Fig. 17. Performance variations of TRRF-R with respect to α and k on the Corel image database.

MR and LRGA vary when using different k, they are still relatively insensitive to the parameter k when k in a certain range (i.e., $k \in [10, 20]$). We observe similar trends on other databases.

We also test the performance variation of our long-term RF method TRRF-R with respect to two parameters, k and the regularization parameter α in (35), using Corel image data set and HumanEva 3D motion/pose data set. Again, we set k to 5, 10, 15, and 20. We set α to $10^{-2}, 10^{-1}, 10^{0}, 10^{1}$, and 10^{2}. As shown by Figs. 17 and 18, the performances change when using different values for k and α. How to automatically decide the best k and α is still an open problem which will be investigated in our future work.

5.6 Scalability of LRGA

We test the scalability of our proposed LRGA ranking algorithm using the large NUS-WIDE image database [9]. In total, this data set provides 269,648 images and their ground-truth annotations. However, the labels of the images in NUS-WIDE database are not balanced. There are 74,190 positive images for one semantic concept (say ℓ_i), while there are only 60 positive images for another semantic concept (say ℓ_j). If we select the image which is labeled as ℓ_i as the query example, then the precision in the top-ranked returns is potentially high. In contrast, if we use the image which is labeled as ℓ_j as the query example, the search precision may be rather low. Therefore, we preprocess the database to ensure that the numbers of positive images from



(a) When top 20 resutls are returned (b) When top 50 resutls are returned (c) When top 100 resutls are returned

Fig. 18. Performance variations of TRRF-R with respect to α and k on HumanEva 3D motion/pose data set.

different concepts are balanced. We first remove the concepts which have less than 5,000 positive images as well as the concepts which have more than 10,000 images. After removing the images which are associated with multiple labels and the images which are not labeled as one of the remaining concepts, we finally construct a subdatabase with 45,227 images in total.

We use 45,000 images to compute the Laplacian matrix for data ranking, among which 200 images are selected as the query examples to test the image retrieval performance when query examples are inside the database. The remaining 227 images are used to test the image retrieval performance when query examples are outside the database. In this work, we use the SIFT feature-based Bag-of-Words (BoW) feature provided in [9] (please refer to [9] for the details). For LRGA, MR, LTSA, and Local Learning, we fix k as 10. Fig. 19 shows the image retrieval performances by using the four ranking algorithms. We set the scope as 200 in Fig. 19 because this database contains much more images than the Corel image database. Again, our ranking algorithm LRGA achieves the best result on this data set. We also compare our ranking algorithm LRGA with Vasconcelos's algorithm [29] for image retrieval using the NUS-WIDE image database. From Fig. 20, we observe that LRGA outperforms [29] again. We do not compare with Tieu's method [28] because it is unsuitable for large-scale image retrieval task.

We test LRGA-kNN for large-scale image retrieval when the query example is outside the database, and the results are shown in Fig. 21. We also report the results from our proposed short-term RF method, for which five images are marked by the user. We do not report the result of LRGA-newLap because it is not feasible to recalculate the Laplacian matrix for every query image in a large-scale image retrieval task. Again, we observe that the retrieval performance can be improved by using short-term RF.

5.7 Effectiveness of Exploiting Unlabeled Data in Long-Term RF

We use the HumanEva 3D motion/pose data set as an example to measure the performance of our long-term RF method TRRF-R by setting $\alpha = 0$ and $\alpha = 1$. In particular, 10 percent of the database data are labeled samples marked by different users. Note that the unlabeled data are not utilized when setting $\alpha = 0$. Fig. 22 clearly shows that it is beneficial to utilize both labeled data and unlabeled data in TRRF-R for long-term RF because we can obtain a better refined multimedia vector representation. This observation is consistent with the findings in previous semi-supervised learning works in [6] and [37].

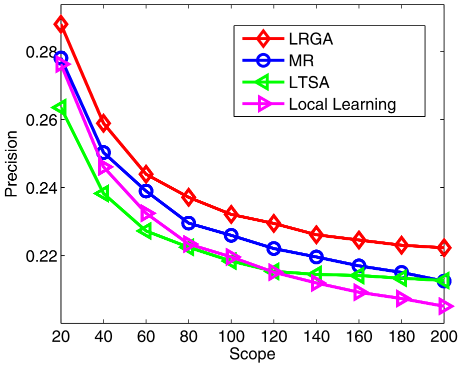


Fig. 19. A comparison of our ranking algorithm LRGA with MR [43], LTSA [41], and Local Learning [32] for image retrieval.

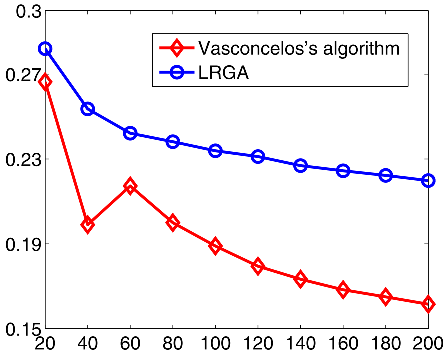


Fig. 20. A comparison of our algorithm with Vasconcelos's algorithm [29] for image retrieval.

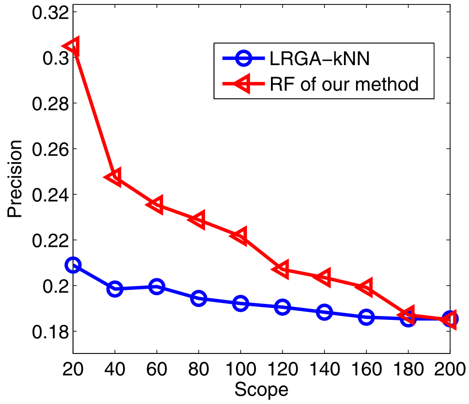


Fig. 21. Image retrieval performance using LRGA-kNN and our proposed short-term RF method. The query example is outside the database.

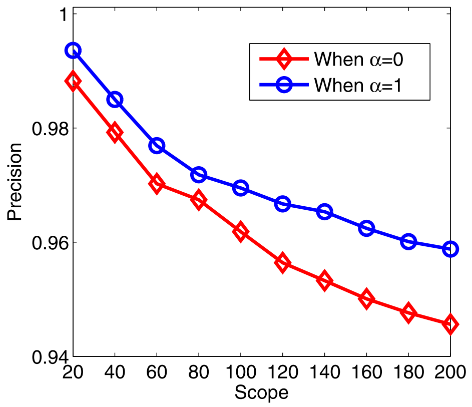


Fig. 22. The effectiveness of exploiting unlabeled data for long-term RF.

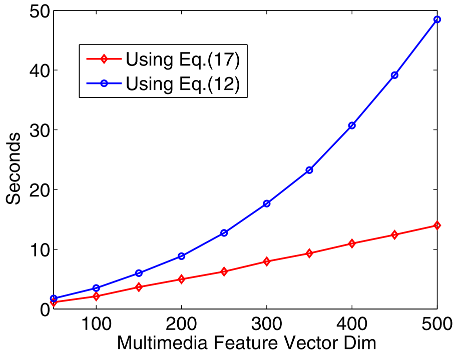


Fig. 23. A comparison of the running time for calculating L with (12) and (17), respectively.

5.8 Computational Speed of the Proposed Algorithms

In Section 2.3, we propose a faster algorithm to compute the Laplacian matrix L defined in (14) for data ranking. Note that for each multimedia datum x_i, we need to compute the matrix L_i using (12) or (17). Fig. 23 compares the running time of using (12) or (17) to compute the Laplacian matrix L. In this experiment, we set k to 10 and use randomly generated vectors with different dimensions, varying from 50-dim to 500-dim with an interval of 50. This setting is reasonable because in real applications, multimedia feature vectors are always of high dimension and the parameter k that specifies the number of nearest neighbors is usually small. For example, the SIFT feature of an image may be of a very high dimension. One thousand vectors were randomly generated, representing 1,000 multimedia data. We repeat the experiment for 50 times and report the average running time for computing the Laplacian matrix L. Fig. 23 shows that it is more efficient to use (17) to compute L, especially when the dimension of multimedia feature vector increases.

To conduct long-term RF, we need to optimize the trace ratio problem in (36). In Fig. 24, we compare the convergence speed of our proposed algorithm and Wang's algorithm [31] for solving the same trace ratio problem. In this experiment, the feature vector dimension is fixed to 500. As shown in Fig. 24, our algorithm converges faster than Wang's algorithm. Specifically, our algorithm converges within five iterations, whereas Wang's algorithm converges after the seventh iteration. We observe a similar trend on other cases as well, which is consistent with the theoretical analysis in Section 3.

6 Conclusions

In this paper, we have proposed a framework for multimedia retrieval. The framework consists of two independent algorithms which are applicable to a broad range of applications including cross-media retrieval, image retrieval, and 3D motion/pose data retrieval. The algorithms proposed in this paper solve two important problems in content-based multimedia retrieval. First, given the multimedia data vector representation and the query example provided by the user, how to rank the multimedia data for retrieval. Second, given the multimedia data and RF information, how to refine the multimedia vector representation in order to better describe the multimedia semantics. The framework is flexible; we can apply only one or both algorithms to multimedia retrieval or we can combine any one algorithm with other existing algorithms for multimedia retrieval.

To the above end, an LRGA ranking algorithm is first proposed to rank the multimedia data for retrieval. In contrast to the existing transductive ranking algorithm of manifold ranking, LRGA does not compute the Laplacian matrix directly. Instead, it learns a Laplacian matrix for data ranking via a statistic approach. For each data point, a linear regression model is used to predict the ranking scores for its

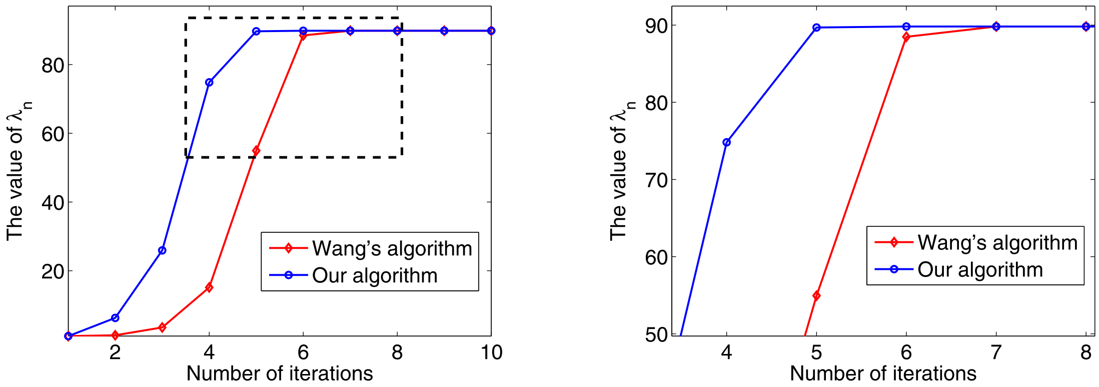


Fig. 24. A comparison of the convergence speed of our algorithm and Wang's algorithm [31] for solving the trace ratio problem. The right figure is the zoomed version of dashed area in the left one.

neighboring points. In order to assign optimal ranking scores to all the data points, we have proposed a unified objective function to globally align all the local regression models. We have developed a short-term RF algorithm to improve the retrieval accuracy when the query example is outside the database. Consequently, the system can better understand the user's search intention so that better multimedia retrieval performance can be achieved.

In order to refine the multimedia vector representation for a higher multimedia retrieval precision, we further develop a long-term RF algorithm. Differently from the short-term RF algorithm that only makes use of the RF information marked by the current user, the long-term RF algorithm proposed in this paper utilizes the RF information from the entire history. Short-term RF can only improve the retrieval precision of current search. Long-term RF can boost the retrieval performance of all the subsequential searches by refining the multimedia vector representation. In our framework, the information from the RF history is first converted into pairwise constraints. Upon that, we propose a new semi-supervised algorithm to obtain the refined multimedia vector representation by simultaneously learning from the multimedia data distribution and the pairwise constraints.

We have applied the proposed framework to several diverse applications, including cross-media retrieval, image retrieval, and 3D motion/pose data retrieval. Comprehensive experiments show that the proposed framework obtains remarkable performance. The proposed LRGA ranking algorithm outperforms the existing transductive ranking algorithm MR. In addition, multimedia retrieval precision can be significantly improved by using the proposed long-term RF algorithm.

Acknowledgments

The authors would like to thank the Associate Editor and the anonymous reviewers for their helpful comments. This work was partially supported by the National Natural Science Foundation of China (No. 90920303), partially supported by the US National Science Foundation (NSF) (IIS-0917072 and IIS-0751185), and partially supported by the Singapore National Research Foundation Interactive Digital Media R&D Program (No.NRF2008IDM-IDM004-018). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the US National Science Foundation (NSF).

References

[1] http://vision.cs.brown.edu/humaneva/, 2011.

[2] X. Bai, X. Yang, L. Latecki, W. Liu, and Z. Tu, "Learning Context-Sensitive Shape Similarity by Graph Transduction," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 5, pp. 861-874, May 2010.

[3] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 711-720, July 1997.

[4] A.D. Bimbo and P. Pala, "Content Based Retrieval of 3D Models," *ACM Trans. Multimedia Computing, Comm. and Applications,* vol. 2, no. 1, pp. 20-43, 2006.

[5] L. Bottou and V. Vapnik, "Local Learning Algorithms," *Neural Computation,* vol. 4, no. 6, pp. 888-900, 1992.

[6] D. Cai, X. He, and J. Han, "Semi-Supervised Discriminant Analysis," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[7] D. Cai, X. He, and J. Han, "Training Linear Discriminant Analysis in Linear Time," *Proc. IEEE Int'l Conf. Data Eng.,* 2008.

[8] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large Scale Online Learning of Image Similarity through Ranking," *Proc. Iberian Conf. Pattern Recognition and Image Analysis,* 2009.

[9] T. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A Real-World Web Image Database from National University of Singapore," *Proc. ACM Int'l Conf. Image and Video Retrieval,* 2009.

[10] F. Chung, *Spectral Graph Theory.* AMS Bookstore, 1997.

[11] P. Duygulu, K. Barnard, J. Freitas, and D. Forsyth, "Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary," *Proc. European Conf. Computer Vision,* 2002.

[12] J. Fan, A. Elmagarmid, X. Zhu, W. Aref, and L. Wu, "Classview: Hierarchical Video Shot Classification, Indexing, and Accessing," *IEEE Trans. Multimedia,* vol. 6, no. 1, pp. 70-86, Feb. 2004.

[13] A. Frome, Y. Singer, F. Sha, and J. Malik, "Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[14] K. Fukunaga, *Introduction to Statistical Pattern Recognition,* second ed. Academic Press, 1991.

[15] J. He, M. Li, H. Zhang, H. Tong, and C. Zhang, "Manifold-Ranking Based Image Retrieval," *Proc. ACM Int'l Conf. Multimedia,* pp. 9-16, 2004.

[16] X. He, W.-Y. Ma, and H.-J. Zhang, "Learning an Image Manifold for Retrieval," *Proc. ACM Int'l Conf. Multimedia,* pp. 17-23, 2004.

[17] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih, "Image Indexing Using Color Correlograms," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* pp. 762-768, 1997.

[18] Y. Jia, F. Nie, and C. Zhang, "Trace Ratio Problem Revisited," *IEEE Trans. Neural Networks,* vol. 20, no. 4, pp. 729-735, Apr. 2009.

[19] A. Langville and C. Meyer, "Survey: Deeper Inside Pagerank," *Internet Math.* vol. 1, no. 3, pp. 335-380, 2003.

[20] M. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content Based Multimedia Information Retrieval: State of the Art and Challenges," *ACM Trans. Multimedia Computing, Comm. and Applications,* vol. 2, no. 1, pp. 1-19, 2006.

[21] N. Maddage, C. Xu, M. Kankanhalli, and X. Shao, "Content Based Music Structure Analysis with Applications to Music Semantics Understanding," *Proc. ACM Int'l Conf. Multimedia,* pp. 112-119, 2004.

[22] M. Müller, T. Röder, and M. Clausen, "Efficient Content Based Retrieval of Motion Capture Data," *ACM Trans. Graphics,* vol. 24, no. 3, pp. 677-685, 2005.

[23] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan, "Trace Ratio Criterion for Feature Selection," *Proc. Nat'l Conf. Artificial Intelligence,* 2008.

[24] S. Roweis and L.K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science,* vol. 290, pp. 2323-2326, 2000.

[25] Y. Rubner, C. Tomasi, and L. Guibas, "A Metric for Distributions with Applications to Image Databases," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 1998.

[26] Y. Rui and T.S. Huang, "Optimizing Learning in Image Retrieval," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2000.

[27] D. Spielman and S. Teng, "Nearly-Linear Time Algorithms for Graph Partitioning, Graph Sparsification, and Solving Linear Systems," *Proc. ACM Symp. Theory of Computing,* pp. 81-90, 2004.

[28] K. Tieu and P. Viola, "Boosting Image Retrieval," *Int'l J. Computer Vision,* vol. 56, nos. 1/2, pp. 17-36, 2004.

[29] N. Vasconcelos and A. Lippman, "A Probabilistic Architecture for Content Based Image Retrieval," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2000.

[30] F. Wang and C. Zhang, "Label Propagation through Linear Neighborhoods," *IEEE Trans. Knowledge and Data Eng.,* vol. 20, no. 1, pp. 55-67, Jan. 2008.

[31] H. Wang, S. Yan, D. Xu, X. Tang, and T. Huang, "Trace Ratio vs. Ratio Trace for Dimensionality Reduction," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2007.

[32] M. Wu and B. Schölkopf, "Transductive Classification via Local Learning Regularization," *Proc. Int'l Conf. Artificial Intelligence and Statistics,* 2007.

[33] Y. Yang, D. Xu, F. Nie, S. Yan, and Y. Zhuang, "Image Clustering Using Local Discriminant Models and Global Integration," *IEEE Trans. Image Processing,* vol. 19, no. 10, pp. 2761-2773, Oct. 2010.

[34] Y. Yang, D. Xu, F. Nie, J. Luo, and Y. Zhuang, "Ranking with Local Regression and Global Alignment for Cross Media Retrieval," *Proc. ACM Int'l Conf. Multimedia,* 2009.

[35] Y. Yang, Y. Zhuang, D. Xu, Y. Pan, D. Tao, and S. Maybank, "Retrieval Based Interactive Cartoon Synthesis via Unsupervised Bi-Distance Metric Learning," *Proc. ACM Int'l Conf. Multimedia,* 2009.

[36] Y. Yang, Y. Zhuang, F. Wu, and Y. Pan, "Harmonizing Hierarchical Manifolds for Multimedia Document Semantics Understanding and Cross-Media Retrieval," *IEEE Trans. Multimedia,* vol. 10, no. 3, pp. 437-446, Apr. 2008.

[37] J. Yu and Q. Tian, "Learning Image Manifolds by Semantic Subspace Projection," *Proc. Ann. ACM Int'l Conf. Multimedia,* 2006.

[38] L. Zhang, F. Lin, and B. Zhang, "Support Vector Machine Learning for Image Retrieval," *Proc. Int'l Conf. Image Processing,* 2001.

[39] L. Zhang, C. Chen, W. Chen, J. Bu, D. Cai, and X. He, "Convex Experimental Design Using Manifold Structure for Image Retrieval," *Proc. ACM Int'l Conf. Multimedia,* 2009.

[40] R. Zhang and Z. Zhang, "Effective Image Retrieval Based on Hidden Concept Discovery in Image Database," *IEEE Trans. Image Processing,* vol. 16, no. 2, pp. 562-572, Feb. 2007.

[41] Z. Zhang and H. Zha, "Nonlinear Dimension Reduction via Local Tangent Space Alignment," *Proc. Int'l Conf. Intelligent Data Eng. and Automated Learning,* pp. 477-481, 2003.

[42] D. Zhou and B. Schölkopf, "A Regularization Framework for Learning from Graph Data," *Proc. ICML Workshop Statistical Relational Learning,* pp. 132-137, 2004.

[43] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, "Ranking on Data Manifolds," *Proc. Advances in Neural Information Processing Systems,* 2003.

[44] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with Local and Global Consistency," *Proc. Advances in Neural Information Processing Systems,* 2003.

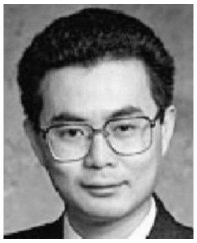
[45] X. Zhu, "Semi-Supervised Learning Literature Survey," technical report, Univ. of Wisconsin, Madison, 2008.

[46] Y. Zhuang, Y. Yang, and F. Wu, "Mining Semantic Correlation of Heterogeneous Multimedia Data for Cross-Media Retrieval," *IEEE Trans. Multimedia,* vol. 10, no. 2, pp. 221-229, Feb. 2008.

Yi Yang received the BS degree from the College of Computer Science, Zhejiang University in 2003, and the PhD degree from the Department of Computer Science at Zhejiang University in 2010. He had been a postdoctoral research fellow at ITEE, The University of Queensland. He is now a postdoctoral research fellow in the School of Computer Science at Carnegie Mellon University. His research interests include machine learning and data mining and their applications to multimedia analysis, information retrieval, and computer vision.

Feiping Nie received the BS degree in computer science from North China University of Water Conservancy and Electric Power, China, in 2000, the MS degree in computer science from Lanzhou University, China, in 2003, and the PhD degree in computer science from Tsinghua University, China, in 2009. Currently, he is a research assistant professor at the University of Texas, Arlington. His research interests include machine learning and its application fields, such as pattern recognition, data mining, computer vision, image processing, and information retrieval.

Dong Xu received the BEng and PhD degrees from the University of Science and Technology of China, in 2001 and 2005, respectively. He is currently an assistant professor at Nanyang Technological University, Singapore. During his PhD studies, he worked with Microsoft Research Asia and The Chinese University of Hong Kong. He also spent one year at Columbia University, New York, as a postdoctoral research scientist. He was the coauthor of a paper that won the Best Student Paper Award from the prestigious IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2010). His research interests include computer vision, statistical learning, and multimedia content analysis.



Jiebo Luo is a senior principal scientist with Kodak Research Laboratories, Rochester, New York. His research interests include image processing, machine learning, computer vision, computational photography, biomedical imaging and informatics, multimedia data mining, and ubiquitous computing. He has authored more than 160 technical papers and holds more than 60 issued US patents. He is the editor-in-chief of the *Journal of Multimedia*. He also serves on the editorial boards of the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Multimedia*, *IEEE Transactions on Circuits and Systems for Video Technology*, *Pattern Recognition*, and *Machine Vision and Applications*. He has been involved in organizing numerous leading technical conferences sponsored by IEEE, ACM, and SPIE. He is a fellow of the SPIE, IEEE, and IAPR.

Yueting Zhuang received the BS, MS, and PhD degrees from Zhejiang University, Hangzhou, China, in 1986, 1989, and 1998, respectively. Currently, he is a professor and PhD supervisor at the College of Computer Science and Technology, Zhejiang University. His research interests include multimedia databases, artificial intelligence, digital library, and video-based animation.

Yunhe Pan is a professor at the College of Computer Science and Technology, Zhejiang University. His research interests include multimedia databases, artificial intelligence, digital library, and computer graphics.