


# Robust Multi-view Manifold Ranking for Image Retrieval

Jun Wu<sup>1,2</sup>, Jianbo Yuan<sup>2</sup>, and Jiebo Luo<sup>2</sup>

<sup>1</sup> Beijing Key Lab of Traffic Data Analysis and Mining,  
Beijing Jiaotong University, Beijing 100044, China

<sup>2</sup> Department of Computer Science,  
University of Rochester, Rochester, NY 14627, USA  
{jwu, jyuan10, jluo}@cs.rochester.edu

**Abstract.** Graph-based similarity ranking plays a key role in improving image retrieval performance. Its current trend is to fuse the ranking results from multiple feature sets, including textual feature, visual feature and query log feature, to elevate the retrieval effectiveness. The primary challenge is how to effectively exploit the complementary properties of different features. Another tough issue is the highly noisy features contributed by users, such as textual tags and query logs, which makes the exploration of such complementary properties difficult. This paper proposes a Multi-view Manifold Ranking (M2R) framework, in which multiple graphs built on different features are integrated to simultaneously encode the similarity ranking. To deal with the high noise issue inherent in the user-contributed features, a data cleaning solution based on visual-neighbor voting is embedded into M2R, thus called Robust M2R (RM2R). Experimental results show that the proposed method significantly outperforms the existing approaches, especially when the user-contributed features are highly noisy.

**Keywords:** Image retrieval · Multi-view learning · Manifold ranking · Data cleaning

## 1 Introduction

The emergence of Web 2.0 technology along with the prevalence of mobile devices leads to an explosion of images being uploaded and shared online, which makes image retrieval become an important research topic during the past two decades [25]. In a typical image retrieval system, a search task may be launched by either keywords or examples provided by the user, termed as Query-by-Keyword (QBK) and Query-by-Example (QBE) respectively, and then the system ranks the images in the database according to their similarities to the user's query. However, QBK is always limited by the so-called 'intent gap', i.e. the user

---

J. Wu—This work was performed when the first author was an academic visitor at University of Rochester.

may not describe the visual content of his or her target using proper keywords, while QBE often suffers from the well-known ‘semantic gap’ existing between low-level image pixels captured by machines and high-level semantic concepts perceived by humans.

An effective solution to bridge the gaps is to exploit the users’ feedbacks that could be obtained in either *explicit* [28] or *implicit* manner [8], such that the initial image ranking list would be refined. Some recent studies along this direction [5, 7, 20, 22, 23, 26] are designed for QBK, called image re-ranking, while others serve for QBE including both short-term learning [3, 12, 14, 16, 29] and long-term learning [4, 9, 15, 17, 21]. In the meantime, a surge of efforts have been made for the graph-based similarity ranking, especially in manifold ranking (MR) [27]. By taking the intrinsic geometrical structure into consideration, MR assigns each data point a relative ranking score, instead of an absolute pairwise similarity as traditional ways. The score is treated as a distance metric defined on the data manifold, which is more meaningful to capture the semantics among data points. In addition, the users’ feedbacks are easily exploited by MR in both explicit and implicit ways [18], and previous studies have shown that MR is one of the most successful ranking approaches for the image retrieval with relevance feedback [3, 16, 19].

Despite its success, the regular MR is limited by two major shortcomings when deployed for image retrieval. First, in many search applications, image data are with multiple views, where each view is actually a feature set. For example, images can be represented by their visual information, surrounding text, and users’ query logs. However, the regular MR method cannot effectively integrate multiple views to encode the similarity ranking. Second, the user-contributed views are often completed in a ‘crowdsourcing’ way, such as textual tags and query logs, so the correctness cannot be always guaranteed. Directly exploiting the highly noisy views in MR may degenerate the retrieval performance. In this paper, we propose a novel method named Robust Multi-view Manifold Ranking (RM2R) to address the aforementioned problems. The main contributions can be summarized as follows. First, we extend the regular MR from single-view to multi-view, aiming to exploit the complementary properties of different feature sets. For convenience of discussion in this paper, we focus on the two-view scenario, assuming that only visual features and query logs are available. Furthermore, we develop a data cleaning solution to make the proposed method more robust to the noisy query logs. An empirical study shows encouraging results in comparison to several exiting approaches. In particular, we observe that our RM2R method is quite robust to noisy query logs, even if the noise level reaches 50%.

In the following we start with a brief review of some related works. Then we propose our RM2R method and report the experimental results. Finally we conclude this paper.

## 2 Related Work

Graph-based similarity ranking has been extensively studied in the multimedia retrieval area. Its main idea is to describe the dataset as a graph and then decide the importance of each vertex based on local or global structure drawn from the graph. One canonical graph-based ranking technique is the Manifold Ranking (MR) algorithm [27], and He et al. [3] first applied MR to image retrieval. Its limitations are addressed by latter research efforts. For example, Wang et al. [10] improved the MR accuracy using a k-regular nearest neighbor graph that minimizes the sum of edge weights and balances the edges in the graph as well. Wu et al. [16] proposed a self-immunizing MR algorithm that uses an elastic kNN graph to exploit unlabeled images safely. Xu et al. [19] proposed an efficient MR solution based on scalable graph structure to handle large-scale image datasets.

Multi-view learning concentrates on learning from the data with multiple feature sets. Co-training [1] is probably the most famous representative. It constructs two learners each from one view, and then lets them to provide pseudo-labels for the other learner. Zhou et al. [30] regarded the visual content and surrounding texts of images as two views, and applied co-training to image retrieval. In fact, co-training does not really need the existence of multiple views, and the diversity among the learners is the real essence [13]. A variant of co-training used in image retrieval [29] suggested to train two different rankers based on the same visual feature set, where each ranker identifies unlabeled images with highest confidence to enlarge the training set of the other ranker. Besides co-training, there are many other solutions to fuse multi-view data in literatures, such as multiple kernel learning [26], and multi-graph learning [2, 11, 23], etc.

Our research is also closely related to the collaborative image retrieval (CIR) that aims to combine short-term learning and long-term learning within a unified framework. For example, Yin et al. [21] exploited the long-term experiences fulfilled by different users to select the optimal online ranker from a set of candidates based on reinforcement learning. Hoi et al. [4] regarded the query log as the ‘side information’, and then, taking that as constraints, learned a distance metric from a mixture of labeled and unlabeled images. Su et al. [9] suggested to discover the navigation patterns from query logs, and using the patterns to facilitate new searching tasks. Wu et al. [15, 17] proposed a hybrid similarity measure that preserves both visual and semantic resemblance by incorporating short-term with long-term feedback experiences.

## 3 The Proposed RM2R Method

Our RM2R method is developed based on two intuitions. First, a ‘good’ ranker should be able to exploit the complementary property of different views. Moreover, the ranker should be robust to noisy views. We start from the description of notations.

### 3.1 Preliminaries

For simplicity, assume that we are handling two-view image dataset  $\mathcal{D} = \{X_i = (\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}), i = 1, \dots, n\}$ , where each image instance  $X_i = (\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)})$  is with two views. Concretely, in the first view  $\mathbf{x}_i^{(1)} \in \mathbb{R}^{d_1}$  is a  $d_1$  dimensional visual feature vector of image  $X_i$ , and in the second view  $\mathbf{x}_i^{(2)} \in \{0, 1\}^{d_2}$  is a  $d_2$  dimensional log feature vector recording the clicks made by different users on image  $X_i$ , where ‘1’ means that  $X_i$  is clicked in a certain query session, and ‘0’ otherwise.

To discover the geometrical structure, we build a couple of graphs  $G^{(z)} = (V^{(z)}, E^{(z)}, \mathbf{W}^{(z)})$  on  $\mathcal{D}$ , where  $z \in \{1, 2\}$  is the graph identify. In details,  $V^{(z)}$  is the node set, in which each node corresponds to an image;  $E^{(z)}$  and  $\mathbf{W}^{(z)} \in \mathbb{R}_+^{n \times n}$  are the edge set and the edge weighting matrix respectively; each  $W_{ij}^{(z)}$  represents the weight of edge  $E_{ij}^{(z)}$ . Typically, the weight is defined by a certain similarity measure, and we apply different similarity measures to  $G^{(1)}$  and  $G^{(2)}$  due to the different input spaces.

For graph  $G^{(1)}$ , its nodes are real-valued vectors, so the similarity between  $\mathbf{x}_i^{(1)}$  and  $\mathbf{x}_j^{(1)}$  is defined by the Gaussian kernel

$$W_{ij}^{(1)} = \exp\left(-\frac{d^2(\mathbf{x}_i^{(1)}, \mathbf{x}_j^{(1)})}{\sigma^2}\right) \quad (1)$$

where  $d(\mathbf{a}, \mathbf{b})$  is a distance metric between two vectors  $\mathbf{a}$  and  $\mathbf{b}$  (suggested by [3], L1 distance is considered), and  $\sigma$  is the bandwidth parameter that can be tuned by local scaling technique, the effectiveness of which has been verified in the clustering [24] and ranking [16] tasks.

For  $G^{(2)}$ , its nodes are binary vectors, so the similarity between  $\mathbf{x}_i^{(2)}$  and  $\mathbf{x}_j^{(2)}$  is defined by the Jacquard coefficient

$$W_{ij}^{(2)} = \frac{|\mathbf{x}_i^{(2)} \cap \mathbf{x}_j^{(2)}|}{|\mathbf{x}_i^{(2)} \cup \mathbf{x}_j^{(2)}|} \quad (2)$$

where  $|\bullet|$  denotes the size of a set.

To exploit the complementary property between  $G^{(1)}$  and  $G^{(2)}$ , we will extend the regular MR method [27] from single-view to multi-view in the next subsection.

### 3.2 M2R: Multi-view Manifold Ranking

Let  $\mathbf{r} : \mathcal{D} \rightarrow \mathbb{R}$  be a ranking function that assigns to each image instance  $X_i$  a ranking score  $r_i$ . We also define a label vector  $\mathbf{y} = [y_1, \dots, y_n]^T$  to collect the user’s online feedbacks, where  $y_i = 1$  if  $X_i$  is labeled as positive,  $y_i = -1$  if  $X_i$  is labeled as negative, and  $y_i = 0$  otherwise. The cost function associated with  $\mathbf{r}$  is defined to be

$$\begin{aligned} \mathcal{Q}(\mathbf{r}) &= \frac{\lambda}{2} \sum_{i,j=1}^n W_{ij}^{(1)} \left( \frac{r_i}{\sqrt{D_{ii}^{(1)}}} - \frac{r_j}{\sqrt{D_{jj}^{(1)}}} \right)^2 \\ &+ \frac{1-\lambda}{2} \sum_{i,j=1}^n W_{ij}^{(2)} \left( \frac{r_i}{\sqrt{D_{ii}^{(2)}}} - \frac{r_j}{\sqrt{D_{jj}^{(2)}}} \right)^2 + \frac{\mu}{2} \|\mathbf{r} - \mathbf{y}\|^2 \end{aligned}$$

where  $0 < \lambda < 1$  is a parameter to adjust the weight between visual view and log view,  $\mu > 0$  is a regularization parameter, and  $\mathbf{D}^{(z)}$  ( $z \in \{1, 2\}$ ) is a diagonal matrix with  $D_{ii}^{(z)} = \sum_{j=1}^n W_{ij}^{(z)}$ . The first two terms in the cost function are two smoothness constraints, which make the nearby examples in both visual space and log space having close ranking scores. The third term is a fitting constrain, which makes the ranking result fitting to the label assignment.

By differentiating  $\mathcal{Q}$  with respect to  $\mathbf{r}$ , we have

$$\frac{\partial \mathcal{Q}}{\partial \mathbf{r}} \Big|_{\mathbf{r}=\mathbf{r}^*} = \lambda(\mathbf{r}^* - \mathbf{S}^{(1)}\mathbf{r}^*) + (1-\lambda)(\mathbf{r}^* - \mathbf{S}^{(2)}\mathbf{r}^*) + \mu(\mathbf{r}^* - \mathbf{y}) = 0$$

where  $\mathbf{S}^{(z)}$  is the symmetrical normalization of  $\mathbf{D}^{(z)}$ , i.e.

$$\mathbf{S}^{(z)} = (\mathbf{D}^{(z)})^{1/2} \mathbf{W}^{(z)} (\mathbf{D}^{(z)})^{1/2}. \quad (3)$$

By regrouping, the equation can be transformed into

$$\mathbf{r}^* - \frac{\lambda}{1+\mu} \mathbf{S}^{(1)} \mathbf{r}^* - \frac{1-\lambda}{1+\mu} \mathbf{S}^{(2)} \mathbf{r}^* - \frac{\mu}{1+\mu} \mathbf{y} = 0.$$

Let  $\alpha = \lambda/(1+\mu)$ ,  $\beta = (1-\lambda)/(1+\mu)$  and  $\gamma = \mu/(1+\mu)$ , and then we have

$$(\mathbf{I} - \alpha \mathbf{S}^{(1)} - \beta \mathbf{S}^{(2)}) \mathbf{r}^* = \gamma \mathbf{y}.$$

Note that  $\alpha + \beta + \gamma = 1$ . Since  $(\mathbf{I} - \alpha \mathbf{S}^{(1)} - \beta \mathbf{S}^{(2)})$  is invertible, we have

$$\mathbf{r}^* = \gamma (\mathbf{I} - \alpha \mathbf{S}^{(1)} - \beta \mathbf{S}^{(2)})^{-1} \mathbf{y}. \quad (4)$$

We can directly use the above closed form solution to compute the ranking scores of examples. However, in large scale problems, we prefer to use the iteration solution

$$\mathbf{r}(t+1) = (\alpha \mathbf{S}^{(1)} + \beta \mathbf{S}^{(2)}) \mathbf{r}(t) + \gamma \mathbf{y}. \quad (5)$$

It is easy to prove that Eq. (5) converges to Eq. (4).

*Proof.* Suppose the sequence  $\{\mathbf{r}(t)\}$  converges to  $\mathbf{r}^*$ . Substituting  $\mathbf{r}^*$  for  $\mathbf{r}(t+1)$  and  $\mathbf{r}(t)$  in the equation. We have  $\mathbf{r}^* = (\alpha \mathbf{S}^{(1)} + \beta \mathbf{S}^{(2)}) \mathbf{r}^* + \gamma \mathbf{y}$  that can be transformed into  $(\mathbf{I} - \alpha \mathbf{S}^{(1)} - \beta \mathbf{S}^{(2)}) \mathbf{r}^* = \gamma \mathbf{y}$ . Since  $(\mathbf{I} - \alpha \mathbf{S}^{(1)} - \beta \mathbf{S}^{(2)})$  is invertible, we have  $\mathbf{r}^* = \gamma (\mathbf{I} - \alpha \mathbf{S}^{(1)} - \beta \mathbf{S}^{(2)})^{-1} \mathbf{y}$ .

Given the fact that different users may have different opinions on judging the same image, the inherent noise issue of query logs is inevitable. Hence we will study the Query Log Cleaning (QLC) solution in the next subsection.

### 3.3 Query Log Cleaning via Neighbor Voting

Inspired by neighbor voting [6], our QLC solution is based on the intuition that if a user clicks a group of visually similar images, his or her clicking information is likely to reflect the objective aspects of visual content. This intuition suggests that, given an image, the confidence of a click made on it can be estimated from how its visual neighbors are judged (clicked or not) in the same session, i.e. accumulating the votes from its visual neighbors.

To facilitate voting, we encode each query session to a unique code (e.g. using the session id), and then all sessions can be viewed as a code book. Then, given an image  $X_i = (\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)})$ , all clicks it received can be represented by a bag-of-code  $\mathcal{C}_i = \{c_j, j = 1, \dots, l_i\}$ , where each code  $c_j \in \mathcal{C}_i$  is actually the index of each non-zero element in the log vector  $\mathbf{x}_i^{(2)}$  and there are  $l_i$  codes in total. Given an image  $X_i = (\mathbf{x}_i^{(1)}, \mathcal{C}_i)$ , we define a voting function  $f(\mathbf{x}_i^{(1)}, c_j)$  to measure the relevance of each code  $c_j \in \mathcal{C}_i$  to the visual object  $\mathbf{x}_i^{(1)}$

$$f(\mathbf{x}_i^{(1)}, c_j) = \sum_{m=1}^K \delta(\mathbf{x}_m, c_j) \quad (6)$$

where  $\mathbf{x}_m$  ( $m = 1, \dots, K$ ) denote the  $K$  visually nearest neighbors of  $\mathbf{x}_i^{(1)}$ . The binary function  $\delta(\mathbf{x}_m, c_j) = 1$  if the visual neighbor  $\mathbf{x}_m$  also has the code  $c_j$ , otherwise  $\delta(\mathbf{x}_m, c_j) = 0$ .

After neighbor voting, each click recorded in query logs is associated with a votes. The higher votes a click received, the more confident it is. Therefore, the clicks with low votes (less than a threshold  $T$ ) can be removed. We apply the ‘Three Standard Deviations’ principle to set the threshold, i.e.  $T = \mu_v - 1.5\sigma_v$ , where  $\mu_v$  and  $\sigma_v$  are the mean and standard deviation of the voted results, respectively.

### 3.4 RM2R-Based Image Retrieval

In this section, we make a brief summary of applying RM2R to image retrieval, described in Algorithm 1. Note that the affinity matrixes can be calculated offline, and therefore RM2R can be quite efficient in processing online queries.

## 4 Experiments

### 4.1 Experimental Setup

We employ the ‘10K Images’ dataset<sup>1</sup> which is publicly available on the web to make our experiments reproducible. The images are from 100 semantic categories, with 100 images per category. Three kinds of visual features are extracted to represent the images, including a 64-dimensional color histogram,

<sup>1</sup> <http://www.datatang.com/data/44353>. The dataset was firstly used in [15].

---

**Algorithm 1.** RM2R-Based Image Retrieval

---

**Input:**Two-View Image dataset:  $\mathcal{D} = \{X_i = (\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}), i = 1, \dots, n\}$ ;Query example:  $\mathbf{q}$ ;Parameters:  $\alpha, \beta, \gamma, K$ **Output:**Rank-Score Vector:  $\mathbf{r}$ 

- 1: Search the  $K$  nearest neighbors for each image based on visual feature set, and connect two images with an edge if they are neighbors (i.e. construct a visual KNN graph);
  - 2: Clean query log set based on visual KNN graph by Eq. (6);
  - 3: Calculate affinity matrix  $\mathbf{W}^{(1)}$  based on visual KNN graph by Eq.(1);
  - 4: Calculate affinity matrix  $\mathbf{W}^{(2)}$  based on cleaned query log set by Eq. (2);
  - 5: Calculate normalized affinity matrixes  $\mathbf{S}^{(1)}$  and  $\mathbf{S}^{(2)}$  by Eq. (3);
  - 6: Initialize label vector  $\mathbf{y}$  according to  $\mathbf{q}$ ;
  - 7: **repeat**
  - 8:   Calculate rank  $\mathbf{r}$  based on  $\mathbf{S}^{(1)}$ ,  $\mathbf{S}^{(2)}$  and  $\mathbf{y}$  by Eq. (4) or Eq. (5);
  - 9:   Update  $\mathbf{y}$  according to user's feedback;
  - 10: **until** user is satisfied with retrieval result
- 

an 18-dimensional wavelet-based texture and a 5-dimensional edge direction histogram [15].

The log dataset consists of 1000 query sessions which are simulated based on the ground truth of image dataset. The average number of clicks in each query session is 20. Further, to evaluate the robustness of our method, three noised log datasets are used in experiments and the noise levels are 10 %, 30 % and 50 %, respectively.

Our RM2R method is compared with three existing approaches which are introduced as follows.

- Self-Tuning Manifold Ranking (STMR): the regular MR method [3] with self-tuned bandwidth parameter [24].
- Co-training based Ranking (CoR): a multi-view learning based image ranking method [29].
- Hybrid Similarity learning (HySim): a CIR method combining short-term learning with long-term learning [15].

In addition, M2R is also included in comparisons to study whether the QLC solution used by RM2R is effective or not.

A query set with 200 image examples is randomly selected to evaluate the average performance of all compared methods. For each query, only one round of feedback is performed (the top ten returned images are labeled), since users have no patience to do more. We use the precision-recall-graph (PR-graph), Mean Average Precision (MAP) and the precision at top  $N$  retrieval results (P@N) for performance evaluation.

## 4.2 Performance Evaluation

At first, we conduct model selection for our M2R and RM2R methods, and there are four parameters in total:  $\alpha$ ,  $\beta$ ,  $\gamma$  ( $\alpha + \beta + \gamma = 1$ ) used in Eq. (5) and  $K$  used in Eq. (6). For convenience,  $\gamma$  is fixed at 0.01, consistent with the previous experiences [3, 19, 27]. Then, we have  $\beta = 0.99 - \alpha$  and thus only need to tune two parameters  $\alpha$  and  $K$ . We first investigate the impact of  $\alpha$  by only evaluating the performance of M2R. Figure 1(a) shows the performance of M2R over different  $\alpha$  values, which is evaluated on the log datasets with different noise levels. We observe that it would be desirable to choose smaller  $\alpha$  values when the noise level of log data is low, and larger ones when the noise level is high. To avoid bias, we set this parameter as the value that the best performance of M2R is achieved on the middle noise level (30%), i.e.  $\alpha = 0.5$ . Also, the performance derived by this parameter value is not too good or too bad, regardless of the noise level is high or low. Further, we evaluate  $K$  for the final results of RM2R on the log dataset with 30% noise, depicted by Fig. 1(b). Fortunately, our RM2R method is not sensitive to  $K$ , and we fixed it at 20.

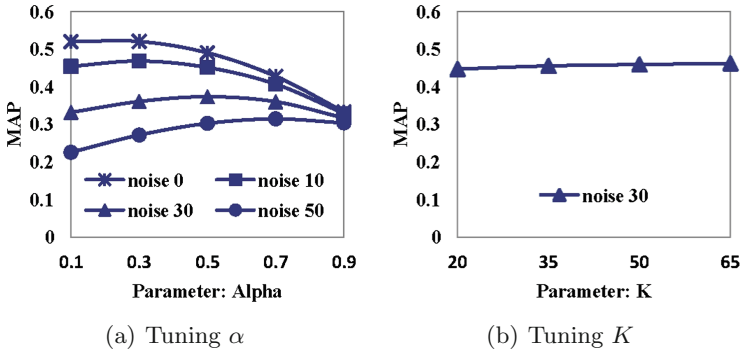
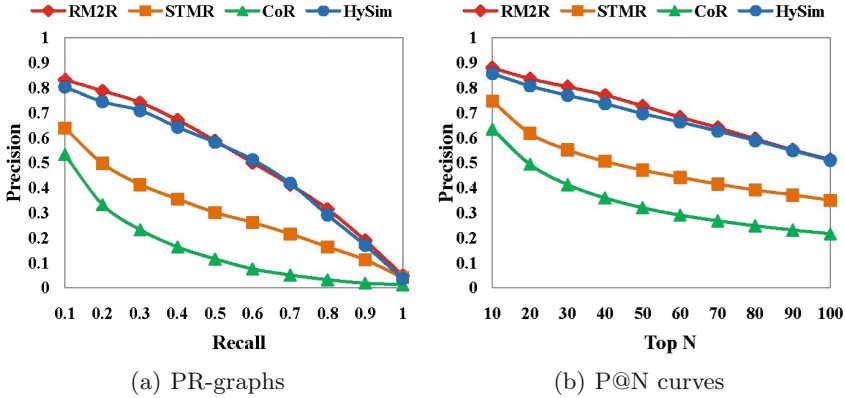


Fig. 1. Tuning parameters for our RM2R method.

Next, with the best parameters setting, we compare our RM2R method with STMR, CoR and HySim on the ideal log dataset (without any noise). Figure 2(a) and, (b) print the PR graphs and the P@N curves respectively. By examining all methods, we observe that the methods using both visual features and query logs (RM2R and HySim) outperform the methods only using visual features (STMR, CoR), which indicates that exploiting the multi-view information is beneficial to the similarity ranking. Further, by comparing STMR with CoR, we find that the performance of STMR is much better than CoR, which indirectly verifies the superiority of the graph-based similarity ranking used by our method in comparison to the pairwise similarity ranking.

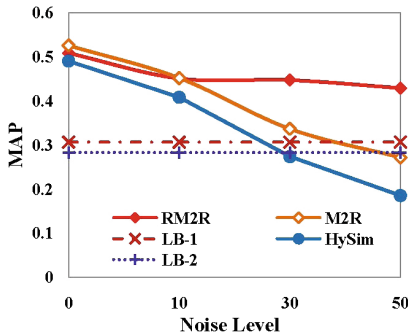
Furthermore, we compare RM2R with M2R and HySim on the noisy log datasets. To analyze the tolerance of an algorithm to noise, we define a performance lower bound for a method first, which refers to the performance of the method evaluated in the single-view case, i.e. no any query logs are used.





**Fig. 2.** The performance of our RM2R method compared with several existing approaches.

The MAPs of the proposed methods (RM2R and M2R) and HySim evaluated under different noise levels are plotted in Fig. 3, where LB-1 and LB-2 are the performance lower bounds of (R)M2R and HySim, respectively. As expected, the performances of all methods degrade as the noise level grows, but RM2R is more robust to the noise than M2R and HySim. As illustrated, the MAP curve of RM2R is always above its lower bound and degrades gracefully, while the curves of M2R and HySim degrade sharply with the growing noise levels and drop below their lower bounds when the noise level hits 50 % and 30 %, respectively. This observation confirms that our QLC solution is effective and essential to achieving the robust similarity ranking. Note that the performance of RM2R is not always the best. When the noise level is low (10 %), the MAP curves of RM2R and M2R are very similar to each other and even RM2R is worse than M2R when there is no noise in log dataset. It is conjectured that QLC is a lossy data cleaning technique, i.e. a very few examples are mistakenly removed, and



**Fig. 3.** The robustness of the proposed methods compared with the HySim approach.

**Table 1.** Comprehensive comparison.

Method\NoiseLevel	0 %	10 %	30 %	50 %
RM2R	0.508	<b>0.451</b>	<b>0.447</b>	<b>0.429</b>
M2R	<b>0.525</b>	0.450	0.337	0.272
HySim	0.491	0.408	0.274	0.185
STMR	0.307	0.307	0.307	0.307
CoR	0.157	0.157	0.157	0.157

thus its impact is trivial when the noise level is low. With the growing of the noise level, QLC is increasingly helpful to RM2M.

At last, a comprehensive MAP comparison of all compared methods is summarized in Table 1, where the best performance has been boldfaced. We observe that M2R and HySim perform no better, and often worse, than STMR when the noise level is high, which validates the superiority of our QLC solution again. In summary, when there is no noise in log dataset, the performance of our RM2R method is only marginally better than HySim, but it outperforms STMR and CoR significantly. Considering that our RM2R method is much more robust than M2R and HySim when the log dataset is (highly) noisy, the above presented experiments confirm that our RM2R method is the best among the compared methods.

## 5 Conclusions

This paper presents a novel RM2R method that aims at the noise-resistant exploitation of the complementary information hidden in multiple views. We first propose a M2R framework to integrate visual feature with query log feature, and then, based on Neighbor Voting, develop a data cleaning solution to noisy query logs. To our best knowledge, not much has been reported on investigating multi-view learning with noisy data in literatures. Experimental study has validated the superiority of the proposed method in comparison to several existing similarity ranking approaches. In this paper, we focus on the two-view data. However, extending to more views will suffer from the inconvenience of tuning a number of parameters. In the future, we will study the self-tuning solution to multi-view manifold ranking.

**Acknowledgments.** The authors would like to thank the anonymous reviewers for their constructive suggestions. This work was supported in part by the ‘Natural Science Foundation of China’ (61301185, 61370070 and 61300071), the ‘Fundamental Research Funds for the Central Universities’ (2015JBM029), and the ‘Science Foundation of Beijing Jiaotong University’ (2015RC008).

## References

1. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: COLT, pp. 92–100 (1998)
2. Deng, C., Ji, R., Tao, D., Gao, X., Li, X.: Weakly supervised multi-graph learning for robust image reranking. *IEEE Trans. Multimedia* **16**(3), 785–795 (2014)
3. He, J., Li, M., Zhang, H., Tong, H., Zhang, C.S.: Manifold-ranking based image retrieval. In: ACM Multimedia, pp. 9–16 (2004)
4. Hoi, S., Liu, W., Chang, S.F.: Semi-supervised distance metric learning for collaborative image retrieval. In: CVPR, pp. 1–7 (2008)
5. Jain, V., Varma, M.: Learning to re-rank: query-dependent image re-ranking using click data. In: Proceedings of the 20th International Conference on World Wide Web, WWW 2011, pp. 277–286 (2011)
6. Li, X., Snoek, G.C., Worring, M.: Learning social tag relevance by neighbor voting. *IEEE Trans. Multimedia* **11**(7), 1310–1322 (2009)
7. Pan, Y., Yao, T., Mei, T., Li, H., Ngo, C.W., Rui, Y.: Click-through-based cross-view learning for image search. In: SIGIR, pp. 717–726 (2014)
8. Smith, G., Brien, C., Ashman, H.: Evaluating implicit judgments from image search clickthrough data. *J. Am. Soc. Inf. Sci. Technol.* **63**(12), 2451–2462 (2012)
9. Su, J., Huang, W., Yu, P., Tseng, V.: Efficient relevance feedback for content-based image retrieval by mining user navigation patterns. *IEEE Trans. Knowl. Data Eng.* **23**(3), 360–372 (2011)
10. Wang, B., Pan, F., Hu, K.M., Paul, J.C.: Manifold-ranking based retrieval using k-regular nearest neighbor graph. *Pattern Recogn.* **45**(4), 1569–1577 (2012)
11. Wang, M., Li, H., Tao, D., Lu, K., Wu, X.: Multimodal graph-based reranking for web image search. *IEEE Trans. Image Process.* **21**(11), 4649–4661 (2012)
12. Wang, T., Dai, G., Ni, B., Xu, D., Siewe, F.: A distance measure between labeled combinatorial maps. *Comput. Vis. Image Underst.* **116**(12), 1168–1177 (2012)
13. Wang, W., Zhou, Z.-H.: Analyzing co-training style algorithms. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenić, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 454–465. Springer, Heidelberg (2007)
14. Wu, J., Lin, Z., Lu, M.Y.: Asymmetric semi-supervised boosting for SVM active learning in CBIR. In: ACM CIVR, pp. 182–188 (2010)
15. Wu, J., Shen, H., Li, Y.D., Xiao, Z.B., Lu, M.Y., Wang, C.L.: Learning a hybrid similarity measure for image retrieval. *Pattern Recogn.* **46**(11), 2927–2939 (2013)
16. Wu, J., Li, Y., Feng, S., Shen, H.: A self-immunizing manifold ranking for image retrieval. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013, Part II. LNCS, vol. 7819, pp. 426–436. Springer, Heidelberg (2013)
17. Wu, J., Lu, M.Y., Wang, C.L.: Collaborative learning between visual content and hidden semantic for image retrieval. In: ICDM, pp. 1133–1138 (2010)
18. Wu, J., Xiao, Z.B., Wang, H.S., Shen, H.: Learning with both unlabeled data and query logs for image search. *Comput. Electr. Eng.* **40**(3), 964–973 (2014)
19. Xu, B., Bu, J., Chen, C., Wang, C., Cai, D., He, X.: EMR: a efficient manifold ranking model for content-based image retrieval. *IEEE Trans. Knowl. Data Eng.* **27**(1), 102–114 (2014)
20. Yang, X., Zhang, Y., Yao, T., Ngo, C.W., Mei, T.: Click-boosting multi-modality graph-based reranking for image search. *Multimedia Syst.* **21**(2), 217–227 (2015)
21. Yin, P.Y., Bhanu, B., Chang, K.C., Dong, A.: Integrating relevance feedback techniques for image retrieval using reinforcement learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(10), 1536–1551 (2005)

22. Yu, J., Tao, D., Wang, M., Rui, Y.: Learning to rank using user clicks and visual features for image retrieval. *IEEE Trans. Cybern.* **45**(4), 767–779 (2015)
23. Yu, J., Rui, Y., Chen, B.: Exploiting click constraints and multi-view features for image re-ranking. *IEEE Trans. Multimedia* **16**(1), 159–168 (2014)
24. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: *NIPS*, pp. 1601–1608 (2004)
25. Zhang, L., Rui, Y.: Image search: from thousands to billions in 20 years. *ACM Trans. Multimedia Comput. Commun. Appl.* **9**(1s), 36:1–36:20 (2013)
26. Zhang, Y., Yang, X., Mei, T.: Image search reranking with query-dependent click-based relevance feedback. *IEEE Trans. Image Process.* **23**(10), 4448–4459 (2014)
27. Zhou, D., Weston, J., Gretton, A., Bousquet, O., Scholkopf, B.: Ranking on data manifolds. In: *NIPS*, pp. 169–176 (2003)
28. Zhou, X.S., Huang, T.S.: Relevance feedback in image retrieval: a comprehensive review. *Multimedia Syst.* **8**(6), 536–544 (2003)
29. Zhou, Z.H., Chen, K.J., Dai, H.B.: Enhancing relevance feedback in image retrieval using unlabeled data. *ACM Trans. Inf. Syst.* **24**(2), 219–244 (2006)
30. Zhou, Z.H., Zhan, D.C., Yang, Q.: Semi-supervised learning with very few labeled training examples. In: *AAAI*, pp. 675–680 (2007)