Noise Resistant Graph Ranking for Improved Web Image Search

Wei Liu[†] Yu-Gang Jiang[†] Jiebo Luo[‡] Shih-Fu Chang[†]
[†]Electrical Engineering Department, Columbia University, New York, NY, USA
{wliu,yjiang,sfchang}@ee.columbia.edu
[‡]Kodak Research Laboratories, Eastman Kodak Company, Rochester, NY, USA
jiebo.luo@kodak.com

Abstract

In this paper, we exploit a novel ranking mechanism that processes query samples with noisy labels, motivated by the practical application of web image search re-ranking where the originally highest ranked images are usually posed as pseudo queries for subsequent re-ranking. Availing ourselves of the low-frequency spectrum of a neighborhood graph built on the samples, we propose a graph-theoretical framework amenable to noise resistant ranking. The proposed framework consists of two components: spectral filtering and graph-based ranking. The former leverages sparse bases, progressively selected from a pool of smooth eigenvectors of the graph Laplacian, to reconstruct the noisy label vector associated with the query sample set and accordingly filter out the query samples with less authentic positive labels. The latter applies a canonical graph ranking algorithm with respect to the filtered query sample set. Quantitative image re-ranking experiments carried out on two public web image databases bear out that our re-ranking approach compares favorably with the state-of-the-arts and improves web image search engines by a large margin though we harvest the noisy queries from the top-ranked images returned by these search engines.

1. Introduction

Nowadays, image search has become an indispensable feature of all popular web search engines such as Google, Yahoo! and Bing as well as most photo sharing websites such as Flickr, Photobucket and ImageShack. However, these image search engines are not sufficiently effective because images are too complicated to handle. On account of the remarkable success of text retrieval for searching web pages, most of the search engines still return images solely based on the associated or surrounding text on the web pages containing the images, and visual content analysis is rarely explored so far.

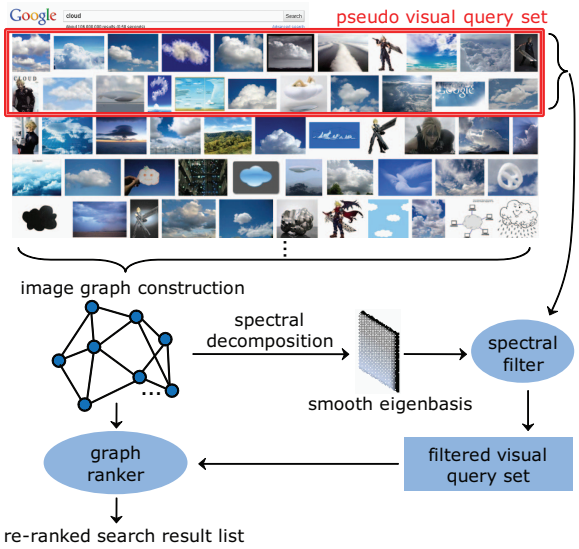

Figure 1. The flowchart of the proposed noise resistant graph ranking framework for web image search re-ranking. Our approach takes the noisy search results returned by a web image search engine as pseudo visual queries, and uses a spectral filter to produce a cleaner set of visual queries which are then cast into a graph ranker to output the re-ranked (better) search results.

Clearly, text-based image search yields unsatisfactory search results due to the difficulty in automatically associating the content of images with text. Thus, researchers started to design algorithms incorporating visual features of images to improve the search results of text-based image search engines. The problem, generally referred to as *web image search re-ranking*, studies mechanisms for re-ranking the images returned by web search engines so that the visually relevant images can appear prominently higher. Some recent works along this direction [1][14][2][7][6][10] have used the visual features for re-ranking web images, while [3][18][11][15] used hybrid textual+visual features.

In the meantime, a surge of efforts have been made in theory and algorithms for graph-based learning, especially in spectral clustering [17] and semi-supervised learn-

ing [23]. Recently, researchers found that the data graph, behaving as an informative platform, can also be utilized to rank practical data including text [24], images [24][13], and videos [12]. One obvious advantage that graph-based methods bear is that the data graph is capable of reflecting the intrinsic manifold structure which is collectively hidden in data such as images and tends to capture the underlying semantics of data. Therefore, performing ranking on image graphs is very suitable for image ranking towards semantics. Inspired by this desirable advantage of ranking on image graphs, we intend to devise a graph-theoretical framework for effective image search re-ranking.

We consider a particular ranking scenario where the query samples are noisy, i.e., they do not all belong to the same class. Such a scenario is pervasively present in the web image search re-ranking problem for which some re-ranking approaches [18][13][21][8] employ a few top-ranked images from web search engines as pseudo visual queries and conduct re-ranking on a larger working set composed of hundreds to thousands of images returned by the search engines. Inevitably, the pseudo visual queries taking pseudo relevance labels (noisy labels) contain the outliers unrelated to user's search intent, as shown in Fig. 1. To this end, we propose a noise resistant graph ranking framework integrating outlier removal and graph ranking. Shown in Fig. 1, our framework first builds a neighborhood graph on the working set to be ranked, then leverages a pool of smooth eigenbases stemming from the normalized graph Laplacian to discover the high-density regions, and lastly selects properly sparse eigenbases to construct a *spectral filter* which automatically filters the low-density outliers out of the noisy query sample set (query set in short). With respect to the filtered query set, a canonical graph ranker, e.g., manifold ranking [24], is used to rank the samples in the working set.

The remainder of this paper is organized as follows. In Section 2, we review recent works on web image search re-ranking. After that, we introduce the background knowledge about graph ranking in Section 3, and present our graph ranking framework exclusively designed for the purpose of robust ranking with noisy queries in Section 4. Superior image search performance on two public web image databases is shown in Section 5 and conclusions are drawn in Section 6.

2. Related Work

We roughly divide the existing works on web image search re-ranking into four categories: clustering-based methods, topic models, supervised methods, and graph-based methods.

The clustering-based methods [1][14][2] apply clustering algorithms such as mean-shift, K-means, and K-medoids on the images returned by search engines to seek significant clusters. The images are then re-ranked based on their distances to the cluster centers. This family of methods introduce some uncertain factors such as how to merge clusters, how to drop small clusters, and how to give weights to clusters for distance computation.

The topic models [7][6][10] employ pLSA [6] or LDA [10] to learn the topics which are latent in the images returned by search engines and reveal the visual target of user's textual query. Re-ranking images is then done on the basis of the dominating topics using the topic membership proportions of every image. This family of methods tend to be more suited to object-like text queries, and some of them [7][6] require separate cross-validation sets to determine the dominating topics. Besides, training topic models is nontrivial.

The supervised methods [3][18][11][15] usually acquire binary textual features indicating the occurrence of query terms in the surrounding text on web pages and image metadata (e.g., image tags) of the searched images. Such textual features can be used together with visual features to train a classifier like SVM [18] or logistic regression [15] that predicts the relevance label of each image with respect to the text query. However, these supervised methods train query-specific [3][11] or query-relative [18][15] models that may suffer from the overfitting issue when applied to unseen text queries. Our experiments show that the overfitting issue affects the methods in [18][15].

Our approach to addressing image search re-ranking falls into the fourth category of graph-based methods [12][13][21] by taking into account the sensible idea of ranking on image graphs. Our core contribution that will be discussed in Section 4 is to formulate a spectral filter for effective outlier removal in the noisy query set. In comparison, [12][13] did not consider outlier removal and directly used the noisy query set. [21] handled outliers but did not remove them in a very effective way, as will be validated in our experiments.

3. Background: Ranking on Graphs

In this section, we briefly discuss two popular and theoretically related graph ranking methods: *Personalized PageRank* (PPageRank) [9][13] and *Manifold Ranking* (MRank) [24]. PPageRank tailors graph ranking towards user's interest, which is a query-biased version of the well-known PageRank algorithm in essence. In contrast, MRank explicitly brings the query-biased order to data samples by making use of the inherent manifold structure.

Without loss of generality, suppose that we have n data points $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_q, \mathbf{x}_{q+1}, \cdots, \mathbf{x}_n\} \subset \mathbb{R}^d$, where the first q points are the query samples and the rest ones are the samples to be ranked. The target of either PPageRank or MRank is to rank the samples based on their relevances to the query samples via a neighborhood graph

$G = (V, E, W)$. V is a vertex set composed of n vertices representing n raw data points, $E \subseteq V \times V$ is an edge set connecting nearby data points, and $W \in \mathbb{R}^{n \times n}$ is a matrix measuring the strength of edges. For convenience, we assume the graph is connected, which is satisfied if an adequate number of edges are included in E. The (sparse) weight matrix of the widely used kNN graph is defined by

$$W_{ij} = \begin{cases} \exp(-\frac{d(\mathbf{x}_i, \mathbf{x}_j)^2}{\sigma^2}), & i \in N_j \text{ or } j \in N_i \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $d(,)$ is a distance measure in $\mathbb{R}^d$, e.g., Euclidean distance, $N_i \subset [1:n]$ consists of k indexes of k nearest neighbors of $\mathbf{x}_i$ in $\mathcal{X}$, and $\sigma > 0$ is the bandwidth parameter. Let $D = \text{diag}(W\mathbf{1})$ be a degree matrix whose diagonal elements are $D_{ii} = \sum_{j=1}^{n} W_{ij}$.

For the query-based ranking problem, we define a query indicator vector (query vector in short) $\mathbf{y} \in \mathbb{R}^n$ with $y_i = 1$ if $i \in Q$ ($Q = [1:q]$ denotes the query index set) and $y_i = 0$ otherwise. The q query samples are supposed to come from the same class, taking the ground-truth label 1.

The PPageRank algorithm defines a random walk on G using the stochastic transition matrix $P = \alpha D^{-1}W + (1-\alpha)\mathbf{1}\mathbf{y}^\top/q$ ($0 < \alpha < 1$ is the bias parameter), and the resulting ranking scores constitute a stationary probability distribution $\pi \in \mathbb{R}^n$ over V decided by the linear system $\pi = P^\top \pi$ from which PPageRank solves

$$\pi = \frac{1-\alpha}{q}(I - \alpha W D^{-1})^{-1}\mathbf{y}. \quad (2)$$

The MRank algorithm defines an iterative label diffusion process as follow

$$f(t+1) = \alpha D^{-1/2} W D^{-1/2} f(t) + (1-\alpha)\mathbf{y}, \quad (3)$$

in which t is the time stamp and $f(t) \in \mathbb{R}^n$ receives the diffused labels at time t. The solution of MRank at convergence is given by

$$f = (1-\alpha)(I - \alpha D^{-1/2} W D^{-1/2})^{-1}\mathbf{y}. \quad (4)$$

Interestingly, when replacing WD^{-1} with $D^{-1/2}WD^{-1/2}$ and ignoring the constant factors, PPageRank yields the same analytic solution as MRank. Therefore, we unify PPageRank and MRank into the single solution:

$$\mathbf{f} = (I - \alpha S)^{-1}\mathbf{y}, \quad (5)$$

where $\mathbf{f}$ saves the final rank scores, and matrix $S \in \mathbb{R}^{n \times n}$ reveals the ranking type. $S = WD^{-1}$ represents PPageRank whereas $S = D^{-1/2}WD^{-1/2}$ corresponds to MRank. Actually, the ranking performance of PPageRank and MRank is very comparable, as will be shown in the later experiments.

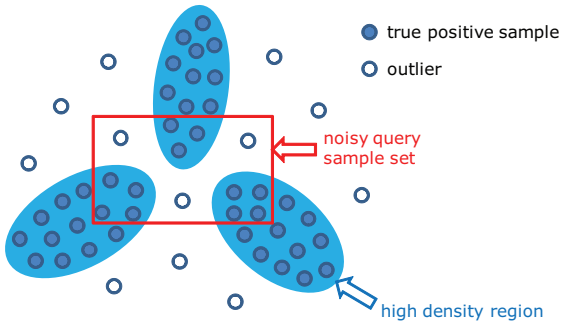

Figure 2. The multi-region assumption for outlier removal.

4. Noise Resistant Graph Ranking

In this section, we address a particular ranking scenario where the query sample set needed by a graph ranker is noisy. Under this setting, directly applying an existing graph ranker (PPageRank or MRank) is very likely to fail. Once the amount of outliers, i.e., those images irrelevant to user's interest, exceeds the amount of the relevant ones in the query set, a graph ranker will end up ranking many irrelevant samples higher. To attain the goal of noise resistant graph ranking, we propose a spectral filter for automatic outlier removal.

4.1. Spectral Filter

As a noisy query set may ruin graph ranking, it is necessary to eliminate the outliers as much as possible and feed a cleaner query set to graph rankers. The previous method *Label Diagnosis* (LabelDiag) in [21] removed an asserted outlier and simultaneously added an asserted positive sample in each iteration of a greedy gradient search algorithm, and it simply set the number of iterations to the half of the query set size. This method is very likely to bring in more outliers owing to sample addition. Given that adding samples is risky, we are only concerned about removing outliers. We desire an outlier filter which can remove the outliers thoroughly and result in a filtered query set that is precise.

As mentioned in [8], the spectrum of the graph Laplacian $L = D - W$ has the potential to suppress the noisy labels in semi-supervised learning. Likewise, we believe that graph spectrum also has potential in handling the noisy labels of query samples in ranking. The spectrum of the kNN graph $G = (V, E, W)$ built in Section 3 is a set of eigenvalue, eigenvector pairs $\{(\lambda_j, \mathbf{u}_j)\}_{j=1}^n$ of the normalized graph Laplacian $\mathcal{L} = D^{-1/2}LD^{-1/2}$, in which the eigenvalues are sorted in a nondecreasing order such that $\mathbf{u}_1$ represents the lowest frequency eigenvector and $\mathbf{u}_n$ represents the highest frequency eigenvector. When the graph is connected, only one eigenvalue is zero, that is, $\lambda_1 = 0$.

[19] pointed out that when the data points have formed clusters, each high density region implicitly corresponds to some low-frequency (smooth) eigenvector which takes rela-

tively large absolute values for points in the region (cluster) and whose values are close to zero elsewhere. Note that we exclude the first eigenvector $\mathbf{u}_1$ because it is nearly constant and does not form clusters. We would assume that *the true positive samples in the query set reside in multiple high-density regions*. Note that this "multi-region" assumption makes sense for web image re-ranking since the in-class images matching user's textual query may fall in a few categories due to polysemy. Fig. 2 gives a schematic illustration of the multi-region assumption.

We only consider m smooth eigenvectors $\mathbf{u}_2, \cdots, \mathbf{u}_{m+1}$ associated with m lowest eigenvalues wrapped in $\Lambda = \mathrm{diag}(\lambda_2, \cdots, \lambda_{m+1})$ to explore the high density regions. Let us inspect the noisy label vector $\mathbf{y}_q = \mathbf{1}$ that is defined on the noisy query set Q and contains the first q entries in vector $\mathbf{y}$. The desired outlier filter aims at refining $\mathbf{y}_q$ through pruning its unreliable 1 entries and then producing a smaller yet cleaner query set $\tilde{Q}$ through picking up the remained 1 entries. Ideally, the exact label vector $\mathbf{y}_q^*$ takes 1 for the true positive samples whereas 0 for the outliers. Within the label value space $\mathbb{R}^q$, we seek m smooth *eigenbases* $\mathbf{u}_{q,2}, \cdots, \mathbf{u}_{q,m+1}$ each of which originates from upper q entries in each eigenvector $\mathbf{u}_j$. According to the multi-region assumption, $\mathbf{y}_q^*$ should nearly lie in some subspace spanned by sparse eigenbases out of $\mathbf{u}_{q,2}, \cdots, \mathbf{u}_{q,m+1}$ (wrap them in the matrix $U_q = [\mathbf{u}_{q,2}, \cdots, \mathbf{u}_{q,m+1}] \in \mathbb{R}^{q \times m}$). Consequently, we formulate a *Spectral Filter* (SpecFilter) via sparse eigenbase fitting, that is, reconstructing the noisy label vector $\mathbf{y}_q$ with sparse eigenbases:

$$\min_{\mathbf{a} \in \mathbb{R}^m} \ \|U_q \mathbf{a} - \mathbf{y}_q\|^2 + \rho \|\mathbf{a}\|_1 + \gamma \mathbf{a}^\top \Lambda \mathbf{a}, \tag{6}$$

where $\mathbf{a}$ is the sparse coefficient vector, $\|\mathbf{a}\|_1 = \sum_{j=1}^m |a_j|$ is the ℓ_1-norm encouraging sparsity of $\mathbf{a}$, and $\rho > 0, \gamma > 0$ are two regularization parameters. Note that the last term in eq. (6) is actually a weighted ℓ_2-norm since $\mathbf{a}^\top \Lambda \mathbf{a} = \sum_{j=1}^m \lambda_{j+1} a_j^2$, which imposes that the smoother eigenbases with smaller λ_j are preferred in reconstruction of $\mathbf{y}_q$.

The nature of eq. (6) is a sparse linear regression model *Lasso* [20] augmented by weighted ℓ_2-regularization. To control the sparsity of $\mathbf{a}$ more conveniently, we convert eq. (6) to the following equivalent problem as done for Lasso:

$$\begin{aligned} \min_{\mathbf{a} \in \mathbb{R}^m} \ & \mathcal{J}(\mathbf{a}, \mathbf{y}_q) = \|U_q \mathbf{a} - \mathbf{y}_q\|^2 + \gamma \mathbf{a}^\top \Lambda \mathbf{a} \\ \text{s.t.} \ & \|\mathbf{a}\|_1 \le z \end{aligned} \tag{7}$$

where the sparsity level parameter $z > 0$ maps to the parameter ρ in one-to-one correspondence. Eq. (7) is a convex optimization problem and can thus be solved accurately by the first-order optimization method *Projected Gradient Descent* [4]. To exploit this method, we must handle the ℓ_1 constraint in eq. (7) via the ℓ_1-ball projection operator

$$\mathbb{B}_z(\mathbf{a}) = \arg \min_{\|\mathbf{b}\|_1 \le z} \|\mathbf{b} - \mathbf{a}\| \tag{8}$$

which has been implemented in $O(m \log m)$ [5]. We describe it in Algorithm 1. By leveraging the ℓ_1-ball projection operator, the iterative updating rule applied in projected gradient descent is

$$\mathbf{a}(t+1) = \mathbb{B}_z \left(\mathbf{a}(t) - \beta_t \nabla_{\mathbf{a}} \mathcal{J}(\mathbf{a}(t), \mathbf{y}_q) \right), \tag{9}$$

where t denotes the time stamp, $\beta_t > 0$ denotes the appropriate step size, and $\nabla_{\mathbf{a}} \mathcal{J}(\mathbf{a}, \mathbf{y}_q)$ denotes the gradient of the cost function $\mathcal{J}$ with respect to $\mathbf{a}$. To expedite the projected gradient method, we offer it a good starting point $\mathbf{a}(0) = (U_q^\top U_q + \gamma \Lambda)^{-1} U_q^\top \mathbf{y}_q$ that is the globally optimal solution to the unconstrained counterpart of eq. (7). Because the dimension of the solution space m is very low in practice (no more than 40 in our all experiments), the projected gradient method converges fast (no more than 100 iterations throughout our experiments).

Algorithm 1 ℓ_1-Ball Projection $\mathbb{B}_z()$

Input: A vector $\mathbf{a} \in \mathbb{R}^m$.
If $\sum_{j=1}^m |a_j| \le z$, $\mathbf{a}' = \mathbf{a}$;
else
  sort $|\mathbf{a}|$ into $\mathbf{v}$ such that $v_1 \ge v_2 \ge \cdots \ge v_m$,
  find $r = \max\{j \in [1:m] : v_j - \frac{1}{j}(\sum_{j'=1}^j v_{j'} - z) > 0\}$,
  compute $\theta = \frac{1}{r}(\sum_{j=1}^r v_j - z)$ and form $\mathbf{a}' = [a'_1, \cdots, a'_m]^\top$
such that $a'_j = \mathrm{sign}(a_j) \cdot \max\{|a_j| - \theta, 0\}$ for $j \in [1:m]$.
Output: A vector $\mathbf{a}' \in \mathbb{R}^m$.

Now we are ready to filter out the outliers by pruning the reconstructed vector $U_q \mathbf{a}$ with $\mathbf{a}$ being the optimal solution to eq. (7). We simply obtain the denoised label vector $\tilde{\mathbf{y}}_q = rnd(U_q \mathbf{a})$ by the rounding function $rnd : \mathbb{R}^q \to \{1, 0\}^q$ defined as follows

$$(rnd(\mathbf{v}))_i = \begin{cases} 1, & v_i \ge \delta \cdot \max\{\mathbf{v}\} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where the parameter $0 < \delta < 1$ is properly chosen to prune the low-valued entries in $U_q \mathbf{a}$ which correspond to the unreliable 1 entries (i.e., noisy labels) in $\mathbf{y}_q$. In order to achieve thorough outlier removal, we deploy SpecFilter in a consecutive mode. Specifically, we generate a sequence of $\{\mathbf{y}_q^j\}$ via successive filtering and seek the convergent one as the ultimate denoised label vector $\tilde{\mathbf{y}}_q$. Setting $\mathbf{y}_q^0 = \mathbf{y}_q$, we launch a successive alternating updating process:
given $\mathbf{y}_q^j$, update $\mathbf{a}^{j+1}$ by

$$\mathbf{a}^{j+1} = \arg \min_{\|\mathbf{a}\|_1 \le z} \mathcal{J}(\mathbf{a}, \mathbf{y}_q^j), \ j = 0, 1, 2, \cdots; \tag{11}$$

and given $\mathbf{a}^{j+1}$, update $\mathbf{y}_q^{j+1}$ by

$$\mathbf{y}_q^{j+1} = rnd(U_q \mathbf{a}^{j+1}), \ j = 0, 1, 2, \cdots. \tag{12}$$

Algorithm 2 Spectral Filter (SpecFilter)

Input: A noisy label vector $\mathbf{y}_q \in \mathbb{R}^q$ and model parameters $\gamma, z > 0, 0 < \delta < 1$.
Set $\beta = 0.5, \eta = 0.01, \epsilon = 10^{-4}$;
define functions $\mathcal{J}(\mathbf{a}, \mathbf{y}_q) = \|U_q\mathbf{a} - \mathbf{y}_q\|^2 + \gamma\mathbf{a}^\top\Lambda\mathbf{a}$ and $\nabla_{\mathbf{a}}\mathcal{J}(\mathbf{a}, \mathbf{y}_q) = 2(U_q^\top U_q + \gamma\Lambda)\mathbf{a} - 2U_q^\top\mathbf{y}_q$;
initialize $\mathbf{y}_q^0 = \mathbf{y}_q$ and $j = 0$;
repeat
  initialize $\mathbf{a}(0) = (U_q^\top U_q + \gamma\Lambda)^{-1}U_q^\top\mathbf{y}_q^j$,
  for $t = 0, \cdots$ **do**
    $\mathbf{a}(t+1) := \mathbb{B}_z\left(\mathbf{a}(t) - \beta_t\nabla_{\mathbf{a}}\mathcal{J}(\mathbf{a}(t), \mathbf{y}_q^j)\right)$
    where $\beta_t = \beta^\omega$ such that ω is the smallest nonnegative integer satisfying $\mathcal{J}(\mathbf{a}(t+1), \mathbf{y}_q^j) - \mathcal{J}(\mathbf{a}(t), \mathbf{y}_q^j) \leq \eta\left(\nabla_{\mathbf{a}}\mathcal{J}(\mathbf{a}(t), \mathbf{y}_q^j)\right)^\top(\mathbf{a}(t+1) - \mathbf{a}(t))$,
    if $|\mathcal{J}(\mathbf{a}(t+1), \mathbf{y}_q^j) - \mathcal{J}(\mathbf{a}(t), \mathbf{y}_q^j)| < \epsilon$ **then**
      $\mathbf{a}^{j+1} := \mathbf{a}(t+1)$ and break,
    end if
  end for
  $\mathbf{y}_q^{j+1} := rnd(U_q\mathbf{a}^{j+1})$,
  $j := j + 1$,
until $\mathbf{y}_q^j$ converges;
Output: The denoised label vector $\tilde{\mathbf{y}}_q = \mathbf{y}_q^j$.

We detail the proposed SpecFilter encompassing solving the sparse eigenbase fitting problem eq. (7) and successive filtering eq. (11)(12) in Algorithm 2. In our experiments, Algorithm 2 achieves a convergent denoised label vector $\tilde{\mathbf{y}}_q$ in about 10 iterations, i.e., $\tilde{\mathbf{y}}_q \cong \mathbf{y}_q^{10}$.

We remark SpecFilter for summary.

Remarks: 1) The idea of sparse eigenbase fitting exploited by SpecFilter implements the multi-region assumption, and the sparsity of eigenbases naturally unveils multiple high-density regions which the true positive query samples belong to. 2) SpecFilter removes the low-density outliers that correspond to the low-valued entries in $\mathbf{y}_q$'s reconstructed version $U_q\mathbf{a}$. Hence, we are aware which and how many outliers should be removed from the noisy query set.

4.2. Algorithmic Framework

So far, we can integrate the proposed spectral filter into a noise resistant graph ranking framework which ranks the samples with respect to the query samples of noisy labels. We outline this algorithmic framework, termed SpecFilter+MRank, in below.

1. Use eq. (1) to build a kNN graph $G = (V, E, W)$ on n samples $\{\mathbf{x}_i\}_{i=1}^n$ of which the first q ones are queries. Compute $S = D^{-1/2}WD^{-1/2}$. Compute $I - D^{-1/2}WD^{-1/2}$ and its low-frequency eigenvectors $U = [\mathbf{u}_2, \cdots, \mathbf{u}_{m+1}]$ corresponding to m lowest eigenvalues $\Lambda = \mathrm{diag}(\lambda_2, \cdots, \lambda_{m+1})$ except $\lambda_1 = 0$.

2. Given the noisy label vector $\mathbf{y}_q = \mathbf{1} \in \mathbb{R}^q$, run SpecFilter (Algorithm 2) using U_q and Λ to output the denoised label vector $\tilde{\mathbf{y}}_q \in \mathbb{R}^q$.

3. Compute the rank score vector $\mathbf{f} = (I - \alpha S)^{-1}\mathbf{y}$ based on the query vector $\mathbf{y} = \begin{bmatrix} \tilde{\mathbf{y}}_q \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^n$.

One can also acquire a slightly different framework SpecFilter+PPageRank by substituting S with WD^{-1}.

5. Experiments

We evaluate the proposed noise resistant ranking framework on two public web image databases: **Fergus** dataset [6] and **INRIA** dataset [15] in which there are 4,091 and 71,478 web images, respectively. Given a text query, each image across the two datasets has an initial ranking score from a web search engine and a ground-truth label indicating whether it is relevant to the query.

For visual feature extraction of web images, we adopt Locality-constrained Linear Coding (LLC) [22] to obtain image representations which have demonstrated state-of-the-art performance. In detail, we use the SIFT descriptors [16] which were computed from 16×16 pixel densely sampled image patches with a stepsize of 8 pixels. The images were all preprocessed into gray scale, and a pre-trained codebook with 1024 bases[1] was used to generate LLC codes. Following the scheme of Spatial Pyramid Matching (SPM) [16], we used 1×1 and 2×2 sub-regions for LLC and max-pooled the LLC codes for each sub-region. Finally, these pooled features from each sub-region were concatenated and ℓ_2-normalized as the final $1024 * 5$-dimensional image feature representation.

As the working set to be re-ranked for each text query across the two datasets has 1000 images at most, we keep a small number of eigenbases to run SpecFilter. Specifically, we use $m = 40$ eigenbases for **Fergus** dataset since its images are more diverse, and $m = 20$ eigenbases for **INRIA** dataset. Accordingly, we set $\gamma = 1, z = 6, \delta = 0.5$ for **Fergus** dataset and $\gamma = 1, z = 3, \delta = 0.5$ for the other. We build a 20NN graph on the working set for each text query with W in eq. (1) defined by the Euclidean distance. We compare six graph-based ranking methods including EigenFunc [8][2], LabelDiag [21], PPageRank [13], MRank [24], and our proposed SpecFilter+PPageRank and SpecFilter+MRank using the same LLC features. To compare them with the existing re-ranking methods, we follow the same evaluation settings and metrics as [6] and [15] on **Fergus** dataset and **INRIA** dataset, respectively.

5.1. Fergus Dataset

On this benchmark, we calculate precision at 15% recall of raw search engine Google and ten re-ranking methods for seven object categories in Table 1. All

[1] Courtesy of http://www.ifp.illinois.edu/ jyang29/LLC.htm

[2] This is a graph-based semi-supervised learning method Eigenfunction. Here we use its one-class version for ranking.

Table 1. Comparisons on **Fergus** dataset. Ranking precision at 15% recall corresponding to seven object categories: airplane, cars (rear), face, guitar, leopard, motorbike, and wrist watch. For each column, two best results are shown in bold.

Precision (%)	airplane	cars rear	face	guitar	leopard	motorbike	wrist watch	**Mean**
Google	50	41	19	31	41	46	70	43
SVM [18]	35	-	-	29	50	63	93	54
LogReg [15]	65	55	72	28	44	49	79	56
TSI-pLSA [6]	57	77	**82**	50	59	72	88	69
LDA [10]	**100**	83	**100**	**91**	**65**	**97**	**100**	**91**
EigenFunc [8]	60	94	47	36	21	48	73	54
LabelDiag [21]	50	54	68	33	42	79	83	58
PPageRank [13]	32	53	64	32	48	79	72	54
MRank [24]	39	53	66	32	50	79	75	56
SpecFilter+PPageRank	80	**94**	75	**61**	47	79	97	76
SpecFilter+MRank	**86**	**100**	75	58	**63**	**79**	**100**	**80**

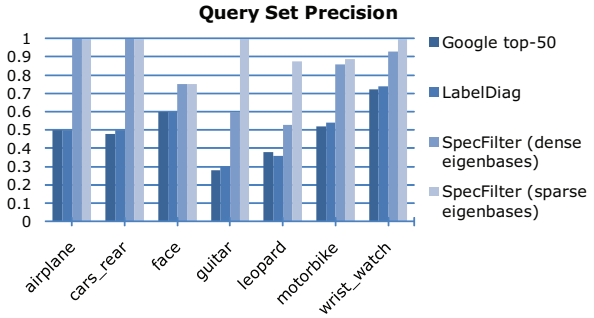

Figure 3. Precision of visual query sets corresponding to seven object categories in **Fergus** dataset. Both LabelDiag and SpecFilter work on initial noisy query sets composed of top-50 images from Google and yield filtered query sets.

of the six graph-based ranking methods work on an initial noisy query set comprised of top-50 images returned by Google for each category. Table 1 exhibits SpecFilter+MRank>SpecFilter+PPageRank>LabelDiag> MRank>PPageRank=EigenFunc in terms of mean precision, which testifies that outlier removal is vital to graph ranking faced with noisy queries. Among the six methods, the proposed SpecFilter+MRank is the best, improving 86% over Google in mean precision. Such an improvement is substantially sharp.

We attribute the success of SpecFilter+MRank to SpecFilter that produces a cleaner query set than the initial noisy query set harvested from Google and the filtered query set achieved by LabelDiag. Additionally, we find that SpecFilter using sparse eigenbases is superior to using dense eigenbases (all m eigenbases). We plot the query set precision of LabelDiag and two versions of SpecFilter in Fig. 3 which discloses that SpecFilter using sparse eigenbases consistently produces the cleanest query set across seven object categories. The precision of the filtered query set of LabelDiag is almost the same as that of Google, so we can say that LabelDiag brings in outliers during sample addition although it filters out some outliers.

We also report the precision of two topic models TSI-pLSA [6] and LDA [10] as well as two supervised models SVM [18] and logistic regression (LogReg) [15] in Table 1. We find out that our approach SpecFilter+MRank surpasses the topic model TSI-pLSA and the two supervised models by a large margin in mean precision. Although the topic model LDA achieves the higher mean precision than our approach (higher precision except on 'cars'), it seems to prefer the object-like text queries since its primary purpose is object detection, not web image re-ranking. In contrast, our re-ranking approach can perform well for diversified text queries, which is verified on the larger dataset **INRIA**.

5.2. INRIA Dataset

We also test our re-ranking approach on the recently released **INRIA** dataset which contains a diversified group of text queries up to 353 including object names 'cloud', 'flag', 'car', celebrity names 'jack black', 'will smith', 'dustin hoffman', and abstract terms 'tennis course', 'golf course', '4x4'. We report the mean average precision (MAP) of the supervised model LogReg and six graph-based ranking methods including ours over the 353 text queries in Table 2. SpecFilter+MRank is still the best among the six graph-based methods when using top-100 images returned by raw search engine as a pseudo visual query set, improving 29% over raw search engine in MAP. Moreover, we list the precision of filtered query sets achieved by LabelDiag and SpecFilter in Table 3, and conclude that SpecFilter using sparse eigenbases yields the cleanest visual query sets consistently.

It is important to note that the supervised model LogReg using both textual and visual features is even inferior to the baselines PPageRank and MRank, which confirms our suspicion that supervised models tend to overfit the training data. In contrast, our re-ranking approach needs neither large training sets nor accurate labels, applying to unseen

text queries adaptively. Remarkably, SpecFilter+MRank using only visual features improves by 10% over LogReg using hybrid textual+visual features in MAP.

Some re-ranked image lists are displayed in Fig. 6, which shows that SpecFilter+MRank outperforms raw search engine and MRank consistently. For the text query 'jack black', the top images ranked by search engine suffer seriously from the issue of *polysemy* as 'jack black' is semantically ambiguous with 'Black Jack card game'. As such, MRank performs worse than search engine because the initial noisy query set contains a lot of outliers from 'Black Jack card game', whereas SpecFilter+MRank is resistant to the outliers owing to SpecFilter. We further study the effect of search engine result quality on our approach by choosing 20 text queries having the best search engine average precision (AP) and 20 text queries having the worst AP. Fig. 4 shows AP for 20 best queries and Fig. 5 shows AP for 20 worst queries, in which SpecFilter+MRank is more robust to the search engine result quality than MRank while MRank works worse for some queries on which search engine works poorly.

Table 2. **INRIA** dataset: mean average precision (MAP) of ranking over 353 text queries with visual query sets of different sizes.

MAP (%)	top-20 images	top-50 images	top-100 images
Search Engine	56.99		
LogReg (textual) [15]	57.00		
LogReg (visual) [15]	64.90		
LogReg (t+v) [15]	67.30		
EigenFunc	48.49	45.38	41.08
LabelDiag	69.51	70.12	69.68
PPageRank	69.31	69.09	68.86
MRank	69.80	69.46	69.04
SpecFilter+PPageRank	71.17	71.35	71.35
SpecFilter+MRank	**72.75**	**73.58**	**73.76**

Table 3. **INRIA** dataset: mean precision of visual query sets over 353 text queries. Both LabelDiag and SpecFilter work on initial noisy query sets from search engine and yield filtered query sets.

Query Set Precision (%)	top-20 images	top-50 images	top-100 images
Search Engine	63.35	56.94	50.91
LabelDiag	63.06	56.82	50.91
SpecFilter (dense eigenbases)	68.54	60.28	51.20
SpecFilter (sparse eigenbases)	**73.51**	**62.83**	**54.30**

6. Conclusions

The conclusions drawn by this paper are three-fold. First, a filtered visual query set produced by the proposed spectral filter, a core component of our graph ranking framework, boosts the performance of existing graph rankers in terms of re-ranked image search results. Second, we find that web image search re-ranking can be addressed using image content alone, and that our re-ranking approach using visual features surpasses supervised re-ranking models relying upon multiple cues including surrounding text and image metadata in addition to visual features [15]. Eventually, compared to other alternative re-ranking approaches, the effect of search engine result quality (e.g., the polysemy issue shown in the last example of Fig. 6) is less significant on our proposed approach.

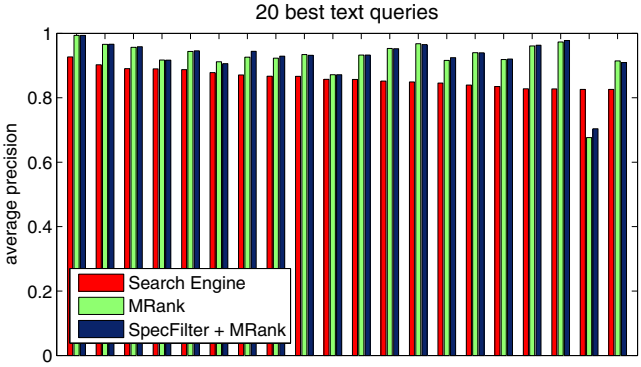

Figure 4. Average precision of ranking for 20 best text queries in **INRIA** dataset. Initial noisy query sets are composed of top-100 images from search engine.

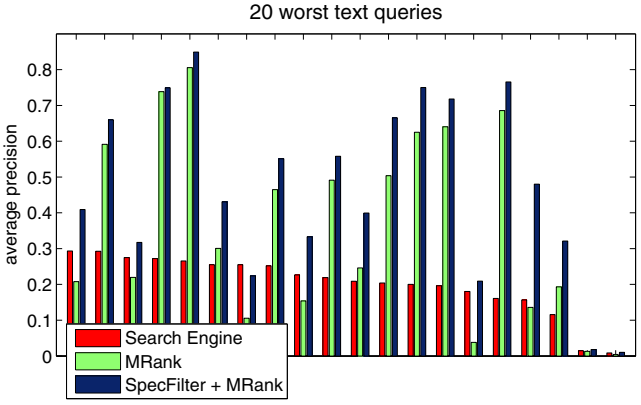

Figure 5. Average precision of ranking for 20 worst text queries in **INRIA** dataset. Initial noisy query sets are composed of top-100 images from search engine.

Acknowledgement

This work has been supported in part by Eastman Kodak Research, NSF (CNS-07-51078), and ONR (N00014-10-1-0242).

References

[1] N. Ben-Haim, B. Babenko, and S. Belongie. Improving web-based image search via content based clustering. In *SLAM Workshop of CVPR*, 2006.

Figure 6. **INRIA** dataset: in each subfigure, the first row shows top-20 images ranked by raw search engine, the second row ranked by MRank, and the third row ranked by SpecFilter+MRank; the initial noisy query set is composed of top-100 images from search engine.

[2] T. Berg and A. Berg. Finding iconic images. In *the 2nd Internet Vision Workshop of CVPR*, 2009.

[3] T. Berg and D. Forsyth. Animals on the web. In *Proc. CVPR*, 2006.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[5] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *Proc. ICML*, 2008.

[6] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *Proc. ICCV*, 2005.

[7] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *Proc. ECCV*, 2004.

[8] R. Fergus, Y. Weiss, and A. Torralba. Semi-supervised learning in gigantic image collections. In *NIPS 22*, 2010.

[9] D. Fogaras, B. Rácz, K. Csalogány, and T. Sarlós. Towards scaling fully personalized pagerank: Algorithms, lower bounds, and experiments. *Internet Mathematics*, 2(3):333–358, 2005.

[10] M. Fritz and B. Schiele. Decomposition, discovery and detection of visual categories using topic models. In *Proc. CVPR*, 2008.

[11] D. Grangier and S. Bengio. A discriminative kernel-based model to rank images from text queries. *IEEE Trans. PAMI*, 30(8):1371–1384, 2008.

[12] W. Hsu, L. Kennedy, and S.-F. Chang. Reranking methods for visual search. *IEEE MultiMedia*, 14(3):14–22, 2007.

[13] Y. Jing and S. Baluja. Visualrank: Applying pagerank to large-scale image search. *IEEE Trans. PAMI*, 30(11):1877–1890, 2008.

[14] L. Kennedy and M. Naaman. Generating diverse and representative image search results for landmarks. In *Proc. WWW*, 2008.

[15] J. Krapac, M. Allan, J. Verbeek, and F. Jurie. Improving web image search results using query-relative classifiers. In *Proc. CVPR*, 2010.

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. CVPR*, 2006.

[17] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS 14*, 2002.

[18] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *Proc. ICCV*, 2007.

[19] T. Shi, M. Belkin, and B. Yu. Data spectroscopy: Eigenspaces of convolution operators and clustering. *The Annals of Statistics*, 37(6B):3960–3984, 2009.

[20] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B*, 58(1):267–288, 1996.

[21] J. Wang, Y.-G. Jiang, and S.-F. Chang. Label diagnosis through self tuning for web image search. In *Proc. CVPR*, 2009.

[22] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proc. CVPR*, 2010.

[23] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS 16*, 2004.

[24] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In *NIPS 16*, 2004.