Image Retrieval via Probabilistic Hypergraph Ranking

Yuchi Huang Qingshan Liu Shaoting Zhang Dimitris N. Metaxas
Rutgers University
617 Bowser Road, Piscataway, NJ 08854

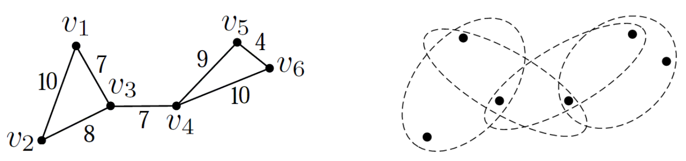
Abstract

In this paper, we propose a new transductive learning framework for image retrieval, in which images are taken as vertices in a weighted hypergraph and the task of image search is formulated as the problem of hypergraph ranking*. Based on the similarity matrix computed from various feature descriptors, we take each image as a 'centroid' vertex and form a hyperedge by a centroid and its k-nearest neighbors. To further exploit the correlation information among images, we propose a* probabilistic hypergraph*, which assigns each vertex v_i to a hyperedge e_j in a probabilistic way. In the incidence structure of a probabilistic hypergraph, we describe both the higher order grouping information and the affinity relationship between vertices within each hyperedge. After feedback images are provided, our retrieval system ranks image labels by a transductive inference approach, which tends to assign the same label to vertices that share many incidental hyperedges, with the constraints that predicted labels of feedback images should be similar to their initial labels. We compare the proposed method to several other methods and its effectiveness is demonstrated by extensive experiments on Corel5K, the Scene dataset and Caltech 101.*

1. Introduction

In content-based image retrieval (CBIR) visual information instead of keywords is used to search images in large image databases. Typically in a CBIR system a query image is provided by the user and the closest images are returned according to a decision rule. In order to learn a better representation of the query concept, a lot of CBIR frameworks make use of an online learning technique called relevance feedback (RF) [24] [15]: users are asked to label images in the returned results as 'relevant' and/or 'not relevant', and then the search procedure is repeated with the new information. Previous work on relevance feedback often aims at learning discriminative models to classify the relevant and irrelevant images, such as, RF methods based on support vector machines (SVM) [29], decision trees [22], boosting [28], Bayesian classifiers [9], and graph-cut [25]. Because the user-labeled images are far from sufficient for supervised learning methods in a CBIR system, recent work in this category attempts to apply transductive or semi-supervised learning to image retrieval. For example, [16] presents an active learning framework, in which a fusion of semi-supervised techniques (based on Gaussian fields and harmonic functions [36]) and SVM are comprised. In [14] and [13], a pairwise graph based manifold ranking algorithm [33] is adopted to build an image retrieval system. Cai et al. put forward semi-supervised discriminant analysis [7] and active subspace learning [6] to relevance feedback based image retrieval. The common ground of [25], [16], [14] and [7] is that they all use a pairwise graph (for simplicity, we denote the pairwise graph as a simple graph) to model relationship between images. In a simple graph both labeled and unlabeled images are taken as vertices; two similar images are connected by an edge and the edge weight is computed as image-to-image affinities. Depending on the affinity relationship of a simple graph, semi-supervised learning techniques could be utilized to boost the image retrieval performance.

Actually, it is not complete to represent the relations among images only by pairwise simple graphs. It should be helpful to take account of the relationship not only between two vertices, but also among three or more vertices containing local grouping information. Such a representation with higher order relationships is a generalization of a simple graph called a *hypergraph*. In a hypergraph a set of vertices is defined as a weighted hyperedge; the magnitude of the hyperedge weight indicates to what extent the vertices in a hyperedge belong to the same cluster [1]. The initial use of hypergraph partitioning algorithms occurs in the field of VLSI (Very-Large-Scale Integration) design and synthesis [3]. In [2], the hypergraph idea is first introduced to the field of computer vision, but it needs to be transferred to a pairwise graph by 'clique average' to solve the clustering problems. [32] presents a hypergraph based image matching problem represented as a convex optimization problem. [26] proposes to use a hypergraph to the problem of multi-label learning. In [17] a hypergraph cut algorithm [34] is adopted to solve the unsupervised video ob-



	e_1	e_2	e_3	e_4	e_5	e_6
v_1	1.0	0.28	0.41	0	0	0
v_2	0.28	1.0	0	0	0	0
v_3	0.41	0.36	1.0	0.41	0	0
v_4	0	0	0.41	1.0	0.32	0.28
v_5	0	0	0	0.32	1.0	0.60
v_6	0	0	0	0	0.60	1.0

Figure 1. Left: A simple graph of six points in 2-D space. Pairwise distances ($Dis(i,j)$) between v_i and its 2 nearest neighbors are marked on the corresponding edges. Middle: A hypergraph is built, in which each vertex and its 2 nearest neighbors form a hyperedge. Right: The H matrix of the probability hypergraph shown above. The entry (v_i, e_j) is set to the affinity $A(j,i)$ if a hyperedge e_j contains v_i, or 0 otherwise. Here $A(i,j) = \exp(-\frac{Dis(i,j)}{\bar{D}})$, where $\bar{D}$ is the average distance.

ject segmentation problem. Tian et al. introduce a semi-supervised learning method named *HyperPrior* [27] to classify gene expression data using biological knowledge as constraints.

In this paper, we propose a hypergraph based transductive algorithm to the field of image retrieval. Based on the similarity matrix computed from various feature descriptors, we take each image as a 'centroid' vertex and form a hyperedge by a centroid and its k-nearest neighbors. To further exploit the correlation information among images, we propose a novel hypergraph model called the *probabilistic hypergraph*, which presents not only whether a vertex v_i belongs to a hyperedge e_j, but also the probability that $v_i \in e_j$. In this way, both the higher order grouping information and the local relationship between vertices within each hyperedge are described in our model. To improve the performance of content-based image retrieval, we adopt the hypergraph-based transductive learning algorithm proposed in [34] to learn beneficial information from both labeled and unlabeled data for image ranking. After feedback images are provided by users or active learning techniques, the hypergraph ranking approach tends to assign the same label to vertices that share many incidental hyperedges, with the constraints that predicted labels of feedback images should be similar to their initial labels. The effectiveness and superiority of the proposed method is demonstrated by extensive experiments on *Corel5K* [12], the Scene dataset [20] and *Caltech-101* [19].

In summary, the contribution of this paper is threefold: (i) we propose a new image retrieval framework based on transductive learning with hypergraph structure, which considerably improves image search performance; (ii) we propose a probabilistic hypergraph model to exploit the structure of the data manifold by considering not only the local grouping information, but also the similarities between vertices in hyperedges; (iii) in this work we conduct an in-depth comparison between simple graph and hypergraph based transductive learning algorithms in the application domain of image retrieval, which is also beneficial to other computer vision and machine learning applications.

2. Probabilistic Hypergraph Model

Let V represent a finite set of vertices and E a family of subsets of V such that $\bigcup_{e \in E} = V$. $G = (V, E, w)$ is called a hypergraph with the vertex set V and the hyperedge set E, and each hyperedge e is assigned a positive weight $w(e)$. A hypergraph can be represented by a $|V| \times |E|$ incidence matrix H_t:

$$h_t(v_i, e_j) = \begin{cases} 1, & \text{if } v_i \in e_j \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The hypergraph model has proven to be beneficial to various clustering/classification tasks [2] [26] [17] [27]. However, the traditional hypergraph structure defined in Equation 1 assigns a vertex v_i to a hyperedge e_j with a binary decision, i.e., $h_t(v_i, e_j)$ equals 1 or 0. In this model, all the vertices in a hyperedge are treated equally; relative affinity between vertices is discarded. This 'truncation' processing leads to the loss of some information, which may be harmful to the hypergraph based applications.

Similar to [8], in this paper we propose a probabilistic hypergraph model to overcome this limitation. Assume that a $|V| \times |V|$ affinity matrix A over V is computed based on some measurement and $A(i,j) \in [0,1]$. We take each vertex as a 'centroid' vertex and form a hyperedge by a centroid and its k-nearest neighbors. That is, the size of a hyperedge in our framework is $k+1$. The incidence matrix H of a probabilistic hypergraph is defined as follows:

$$h(v_i, e_j) = \begin{cases} A(j,i), & \text{if } v_i \in e_j \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

According to this formulation, a vertex v_i is 'softly' assigned to e_j based on the similarity $A(i,j)$ between v_i and v_j, where v_j is the centroid of e_j. A probabilistic hypergraph presents not only the local grouping information, but also the probability that a vertex belongs to a hyperedge. In this way, the correlation information among vertices is more accurately described. Actually, the representation in Equation 1 can be taken as the discretized version of Equation 2. The hyperedge weight $w(e_i)$ is computed as follows:

$$w(e_i) = \sum_{v_j \in e_i} A(i,j). \tag{3}$$

Based on this definition, the 'compact' hyperedge (local group) with higher inner group similarities is assigned a higher weight. For a vertex $v \in V$, its degree is defined to be $d(v) = \sum_{e \in E} w(e)h(v,e)$. For a hyperedge $e \in E$, its degree is defined as $\delta(e) = \sum_{v \in e} h(v,e)$. Let us use D_v,D_e and W to denote the diagonal matrices of the vertex degrees, the hyperedge degrees and the hyperedge weights respectively. Figure 1 shows an example to explain how to construct a probabilistic hypergraph.

3. Hypergraph Ranking Algorithm

The previously proposed algorithms for partitioning a hypergraph can be divided into two categories. The first category aims at constructing a simple graph from the original hypergraph, and then partitioning the vertices by spectral clustering techniques. These methods include clique expansion [37], star expansion [37], Rodriquez's Laplacian [23], etc. The second category of approaches define a hypergraph 'Laplacian' using analogies from the simple graph Laplacian. Representative methods in this category include Bolla's Laplacian [4], Zhou's normalized Laplacian [34], etc. In [1], the above algorithms are analyzed and verified that they are equivalent to each other under specific conditions. For a hypergraph partition problem, the normalized cost function [34] $\Omega(f)$ could be defined as

$$\frac{1}{2}\sum_{e\in E}\sum_{u,v\in e}\frac{w(e)h(u,e)h(v,e)}{\delta(e)}\left(\frac{f(u)}{\sqrt{d(u)}}-\frac{f(v)}{\sqrt{d(v)}}\right)^2, \quad (4)$$

where the vector f is the image labels to be learned. By minimizing this cost function, vertices sharing many incidental hyperedges are guaranteed to obtain similar labels. Defining $\Theta = D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}}$, we can derive Equation 4 as follows:

$$\begin{aligned}\Omega(f) =& \sum_{e\in E}\sum_{u,v\in e}\frac{w(e)h(u,e)h(v,e)}{\delta(e)}\left(\frac{f^2(u)}{d(u)}-\frac{f(u)f(v)}{\sqrt{d(u)d(v)}}\right)\\ =& \sum_{u\in V} f^2(u)\sum_{e\in E}\frac{w(e)h(u,e)}{d(u)}\sum_{v\in V}\frac{h(v,e)}{\delta(e)}\\ &-\sum_{e\in E}\sum_{u,v\in e}\frac{f(u)h(u,e)w(e)h(v,e)f(v)}{\sqrt{d(u)d(v)}\delta(e)}\\ =& f^T(I-\Theta)f, \end{aligned} \quad (5)$$

where I is the identity matrix. Above derivation shows that (i) $\Omega(f,w) = f^T(I-\Theta)f$ if and only if $\sum\limits_{v\in V}\frac{h(v,e)}{\delta(e)} = 1$ and $\sum\limits_{e\in E}\frac{w(e)h(u,e)}{d(u)} = 1$, which is true because of the definition of $\delta(e)$ and $d(u)$ in Section 2; (ii) $\Delta = I - \Theta$ is a positive semi-definite matrix called the hypergraph Laplacian and $\Omega(f) = f^T\Delta f$. The above cost function has the similar formulation to the normalized cost function of a simple graph $G_s = (V_s, E_s)$:

$$\begin{aligned}\Omega_s(f) &= \frac{1}{2}\sum_{v_i,v_j\in V_s} A_s(i,j)\left(\frac{f(i)}{\sqrt{D_{ii}}}-\frac{f(j)}{\sqrt{D_{jj}}}\right)^2\\ &= f^T(I-D^{-\frac{1}{2}}A_sD^{-\frac{1}{2}})f = f^T\Delta_s f, \end{aligned} \quad (6)$$

where D is a diagonal matrix with its (i,i)-element equal to the sum of the i^{th} row of the affinity matrix A_s; $\Delta_s = I - \Theta_s = I - D^{-\frac{1}{2}}A_sD^{-\frac{1}{2}}$ is called the simple graph Laplacian. In an unsupervised framework, Equation 4 and Equation 6 can be optimized by the eigenvector related to the smallest nonzero eigenvalue of Δ and Δ_s [34], respectively.

In the transductive learning setting [34], we define a vector y to introduce the labeling information of feedback images and to assign their initial labels to the corresponding elements of y: $y(v) = \frac{1}{|Pos|}$, if a vertex v is in the positive set Pos, $y(v) = -\frac{1}{|Neg|}$, if it is in the negative set Neg. If v is unlabeled, $y(v) = 0$. To force the assigned labels to approach the initial labeling y, a regularization term is defined as follows:

$$\|f-y\|^2 = \sum_{u\in V}(f(u)-y(u))^2. \quad (7)$$

After the feedback information is introduced, the learning task is to minimize the sum of two cost terms with respect to f [33] [34], which is

$$\Phi(f) = f^T\Delta f + \mu\|f-y\|^2, \quad (8)$$

where $\mu > 0$ is the regularization parameter. Differentiating $\Phi(f)$ with respect to f, we have

$$f = (1-\gamma)(I-\gamma\Theta)^{-1}y, \quad (9)$$

where $\gamma = \frac{1}{1+\mu}$. This is equivalent to solving the linear system $((1+\mu)I - \Theta)f = \mu y$.

For the simple graph, we can simply replace Θ with Θ_s to fulfill the transductive learning. In [14] and [13], this simple graph based transductive reasoning technique is used for image retrieval with relevance feedback. The procedures of the probabilistic hypergraph ranking algorithm and simple graph based manifold ranking algorithm are listed in Algorithm 1 and Algorithm 2.

4. Feature Descriptors and Similarity Measurements

To define the similarity measurement between two images, we utilize the following descriptors: SIFT [21], OpponentSIFT, rgSIFT, C-SIFT, RGB-SIFT [30] and HOG [10] [5]. The first five are appearance-based color

```
Algorithm 1 Probabilistic Hypergraph Ranking
1: Compute similarity matrix $A$ based on various features using Equation 11, where $A(i,j)$ denotes the similarity between the $i^{th}$ and the $j^{th}$ vertices.
2: Construct the probabilistic hypergraph G. For each vertex, based on the similarity matrix $A$, collect its k-nearest neighbors to form a hyperedge.
3: Compute the hypergraph incidence matrix $H$ where $h(v_i, e_j) = A(j,i)$ if $v_i \in e_j$ and $h(v_i, e_j) = 0$ otherwise. The hyperedge weight matrix is computed using Equation 3.
4: Compute the hypergraph Laplacian $\Delta = I - \Theta = I - D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}}$.
5: Given a query vertex and the initial labeling vector $y$, solve the linear system $((1+\mu)I - \Theta) f = \mu y$. Rank all the vertices according to their ranking scores in descending order.
```

```
Algorithm 2 Manifold Ranking
1: Same to Algorithm 1.
2: Construct the simple graph $G_s$. For each vertex, based on the similarity matrix $A$, connect it to its k-nearest neighbors.
3: Compute the simple graph affinity matrix $A_s$ where $A_s(i,j) = A(i,j)$ if the $i^{th}$ and the $j^{th}$ vertices are connected. Let $A_s(i,i) = 0$. Compute the vertex degree matrix $D = \sum_j A_s(i,j)$.
4: Compute the simple graph Laplacian $\Delta_s = I - \Theta_s = I - D^{-\frac{1}{2}} A_s D^{-\frac{1}{2}}$.
5: Same to Algorithm 1, expect that $\Theta$ is replaced with $\Theta_s$.
```

descriptors that are studied and evaluated in [30]. It is verified that their combination has the best performance on various image datasets. HOG (histogram of oriented gradients) is the shape descriptor widely used in object recognition and image categorization. Similar to [30], we extract both the sparse and the dense features for five appearance descriptors to boost image search performance. The sparse features are based on scale-invariant points obtained with the Harris-Laplace point detectors. The dense features are sampled every 6 pixels on multiple scales. For sparse features of each appearance descriptor, we create 1024-bin by *k*–means; for dense features of each appearance descriptor, we create 4096-bin codebooks because each image contains much more dense features than the sparse features. For each sparse (or dense) appearance descriptor, we follow the method in [31] to obtain histograms by soft feature quantization, which was proven to provide remarkable improvement in object recognition [31]. For the HOG descriptor, we discretize gradient orientations into 8 bins to build histograms. For each of above 11 features (5 sparse features + 5 dense features + 1 HOG feature), we use a spatial pyramid matching(SPM) approach [18] to calculate the distances between two images $\{i, j\}$ because of its good performance:

$$Dis(i,j) = \sum_{l=0}^{L} \frac{1}{\alpha^l} \sum_{p=1}^{m(l)} \beta_p^l \chi^2(His_p^l(i), His_p^l(j)). \quad (10)$$

In the above equation, $His_p^l(i)$ and $His_p^l(j)$ are two image's local histograms at the p^{th} position of level l; α and β are two weighting parameters; $\chi^2(\cdot,\cdot)$ are the chi-square distance function used to measure the distance between two histograms. For the sparse and dense appearance features, we follow the setting of [30]: three levels of spatial pyramids are 1×1(whole image, $l = 0$, $m(0) = 1$, $\beta_1^0 = 1$), 1×3(three horizontal bars, $l = 1$, $m(1) = 3$, $\beta_1^1 \sim \beta_3^1 = \frac{1}{3}$), 2×2(image quarters, $l = 2$, $m(2) = 4$, $\beta_1^2 \sim \beta_4^2 = \frac{1}{4}$); $\alpha^0 \sim \alpha^2 = 3$. For the HOG feature, we employ $L = 4$ levels ($l = 0 \sim 3$) of the spatial pyramids as in [5]: 1×1, 2×2, 4×4 and 8×8; $\alpha^0 = 2^L$ and $\alpha^l = 2^{L-l+1}$ for $l = 1, 2, 3$. After all the distance matrices for 11 features are obtained, the similarity matrix A between two images can be computed as follows:

$$A(i,j) = \exp(-\frac{1}{11} \sum_{k=1}^{11} \frac{Dis_k(i,j)}{\overline{D}_k}), \quad (11)$$

where $\overline{D}_k$ is the mean value of elements in the k^{th} distance matrix.

5. Experiments

5.1. Experimental Protocol

In this section, we used SVM and similarity based ranking as the baselines. The similarity based ranking method sorts retrieved image i according to the formula $\frac{1}{|Pos|} \sum_{j \in Pos} A(i,j) - \frac{1}{|Neg|} \sum_{k \in Neg} A(i,k)$, where Pos and Neg denote positive/negative sets of feedback images respectively. We compare the proposed hypergraph ranking frameworks to the simple graph based manifold ranking algorithm [14] [13], and we also evaluate the performances of the probabilistic hypergraph ranking against the hypergraph based ranking. The hypergraph ranking algorithm is the same as Algorithm 1 except for using the binary incidence matrix (where $h_t(v_i, e_j) = 1$ or 0). For the parameter γ in Equation 9, we follow the original work [33] [35] and

	r (scope)	20	40	60	80	100
MR	**P(r)**	0.695 (at $K = 40$)	0.606 (at $K = 40$)	0.537 (at $K = 40$)	0.475 (at $K = 40$)	0.424 (at $K = 40$)
HR	**P(r)**	0.728 (at $K = 40$)	0.644 (at $K = 30$)	0.571 (at $K = 40$)	0.508 (at $K = 40$)	0.450 (at $K = 40$)
PHR	**P(r)**	**0.748** (at $K = 40$)	**0.659** (at $K = 40$)	**0.583** (at $K = 40$)	**0.519** (at $K = 40$)	**0.459** (at $K = 50$)

Table 1. Selection of the hyperedge size and the vertex degree in the simple graph. We list the optimal precisions and corresponding K values at different retrieved image scopes. MR: Manifold Ranking. HR: Hypergraph Ranking. PHR: Probabilistic Hypergraph Ranking. K denotes the hyperedge size and the vertex degree in the simple graph.

fix it as 0.1 for the best performance of both the hypergraph and the simple graph based algorithms. We use the respective optimal hyperedge size or the vertex degree (in the simple graph) in all experiments. Other parameters are directly computed from experimental data. Three general purpose image databases are used in this paper: *Corel5K* [12], the *Scene dataset* [20] and *Caltech-101* [19]. Two measures are employed to evaluate the performance of above five ranking methods: (1) the precision vs. scope curve, (2) the precision vs. recall curve. We use each image in a database as a query example and both measures are averaged over all the queries in this database.

To provide a systematic evaluation, in the first round of relevance feedback 4 positive images and 5 negative images are randomly selected for each query image to form a training set containing 5 positive/ 5 negative examples. In the second and third round, another 5 positive/ 5 negative examples are randomly labeled for training, respectively. In this way a total of 10,20 and 30 images are marked after each of the three relevance feedback cycles. The rest of the images are used as testing data.

Besides the above passive learning setting, we also explore the active learning technique on *Corel5K*. Same setting as in [14], [13], [7] and [6], the ground truth labels of the 10 top ranked examples are used as feedback images after each round of retrieval cycle.

5.2. In-depth Analysis on Corel5K

We choose to conduct an in-depth analysis on *Corel5K* [12] because it is used as the benchmark in [14] and [13] for the manifold ranking method and a lot of other work [11]. Since all 50 categories of *Corel5K* contain the same number of 100 images, the precision-scope curve is used by [14] and [13] as the measurement. Therefore, we choose the precision-scope curve here in order to make a direct comparison with [14] and [13].

Combination of multiple complementary features for image retrieval. As presented in Section 4, we utilize totally 11 features to compute the similarity matrix A. To demonstrate the advantage of combining multiple complementary features, we employ the similarity based ranking method on *Corel5K* using the combined feature and all 11 single features. In this group experiment, only query image is provided and no relevance feedback is performed. As shown in Fig 2, the combined feature outperforms the best single feature (sparse C-SIFT) by $5 \sim 12\%$ for the different retrieval scopes r. All our comparisons are made on the similarity matrix computed with the **same** combined feature.

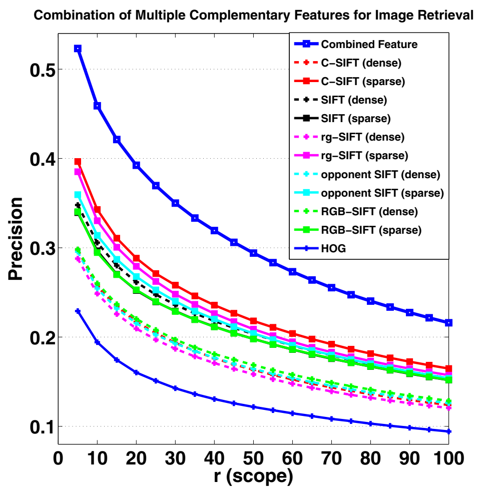


Figure 2. Combination of multiple complementary features for image retrieval. Best viewed in color.

Selection of the hyperedge size and the vertex degree in the simple graph. Intuitively, very small-size hyperedges only contain 'micro-local' grouping information which will not help the global clustering over all the images, and very large-size hyperedges may contain images from different classes and suppress diversity information. Similarly, in order to construct a simple graph, usually every vertex in the graph is connected to its K-nearest neighbors. For the fair comparison, in this work we perform a sweep over all the possible K values of the hyperedge size and the vertex degree in the simple graph to optimize the clustering results. As shown in Table 1, after the first round of relevance feedback (using the passive learning setting), almost all the methods get optimal values at $K = 40$. So we set both the hyperedge size and the vertex degree in the simple graph as 40 in the experiments on *Corel5K*.

Comparison under passive learning setting. As shown in Fig 3, the probabilistic hypergraph ranking outperforms the manifold ranking by $4\% \sim 5\%$ and the traditional hypergraph ranking by $1\% \sim 3\%$ after each round of relevance feedback.

Comparison under active learning setting. As shown in Fig 4, we start the experiment from Round 0, in which

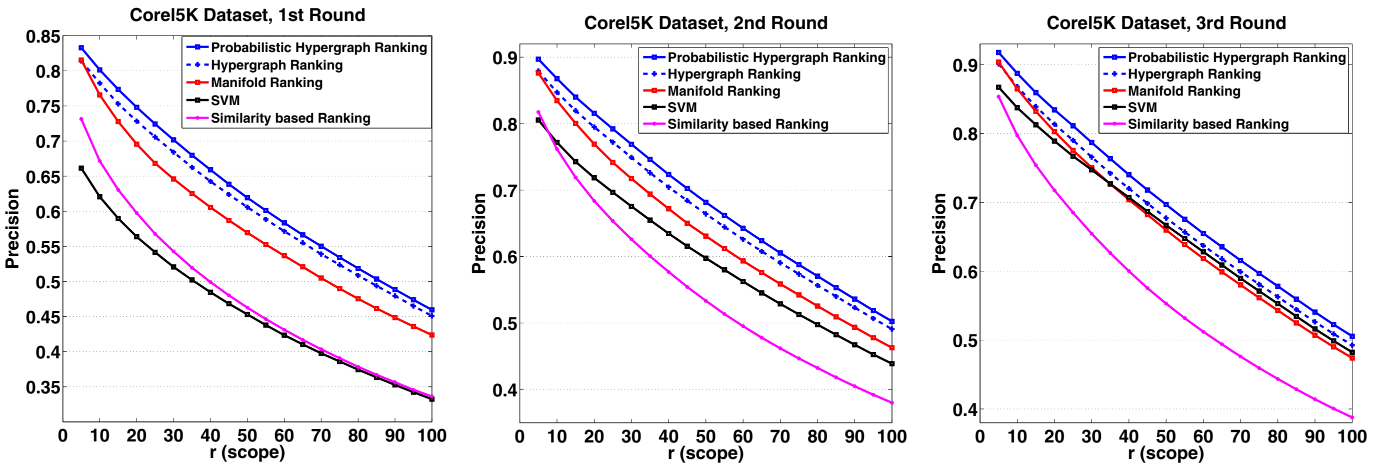


Figure 3. Precision vs. scope curves for *Corel5K*, under the passive learning setting. Best viewed in color.

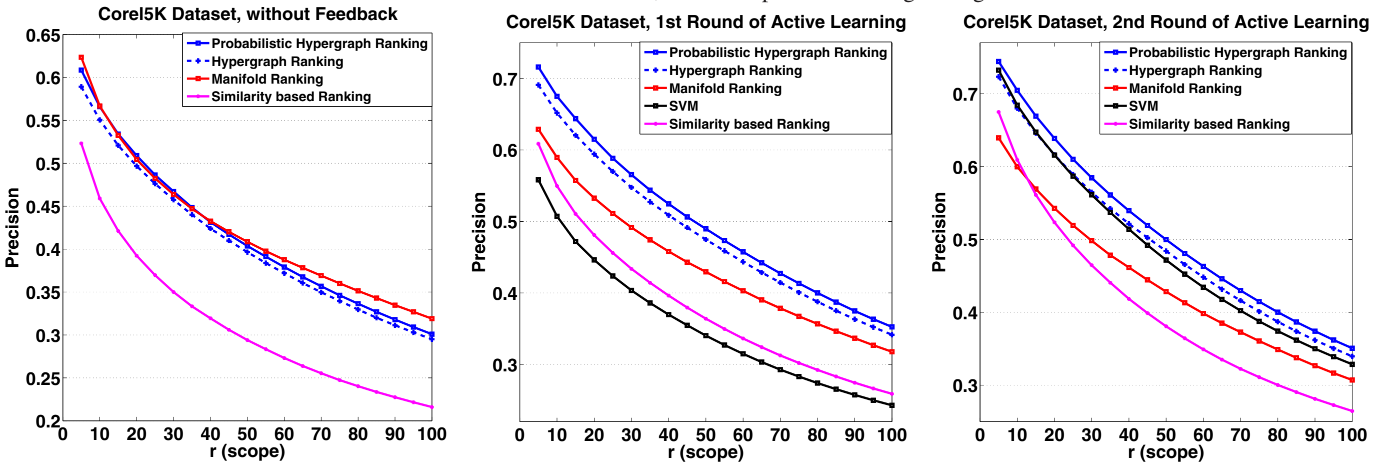


Figure 4. Precision vs. scope curves for *Corel5K*, under the active learning setting. Best viewed in color.

only the query image is used for retrieval. Although the probabilistic hypergraph ranking achieved similar precision to the manifold ranking and the hypergraph ranking in the Round 0(without feedback), it progressively spaces out the difference in precision after the first round and the second round. At the end of the second round, it outperforms the manifold ranking by $4\% \sim 10\%$ and the traditional hypergraph ranking by $1\% \sim 2.5\%$ on different retrieval scope. Another observation is that the manifold ranking provides much less increase on precisions at the end of the second round. For example, at $r = 20$ the precision of the manifold ranking increases from 50.4% to 54.3%, while the precision of the probabilistic hypergraph ranking increases from 50.8% to 63.9%.

Our method outperforms the manifold ranking results in [13] and [14] by approximately $8\% \sim 20\%$ under the similar setting.

Computation cost. The most time consuming parts in both Algorithm 1 and Algorithm 2 are to solve the 5000×5000 linear system in the 5th steps, which have the same time complexity. On a desktop with Intel 2.4GHz Core2-Quad CPU and 8GB RAM, our Matlab code, without code optimization, takes 31.1 and 26.4 seconds to complete 5000 queries for Algorithm 1 and Algorithm 2, respectively. Thus, the computational cost of the hypergraph ranking is similar to that of the simple graph-based manifold ranking.

5.3. Results on the Scene Dataset and Caltech-101

The *Scene dataset* [20] consists of 4485 gray-level images which are categorized into 15 groups. It is also important to mention that we only use 3 features for gray-level images (sparse SIFT, dense SIFT and HOG) to compute the similarity matrix in this experiment; the optimal hyperedge size is $K = 90$ and the optimal vertex degree of the simple graph is $K = 330$. Since every category of the *Scene dataset* contains different number of images, we choose the precision-recall curves as a more rigorous measurement for the Scene dataset. As shown in Fig 6, the probabilistic hypergraph ranking outperforms the manifold ranking by $5\% \sim 7\%$ on Precision for $Recall < 0.8$, after each round of feedback using the passive learning setting; the probabilistic hypergraph ranking is slightly better than the hypergraph ranking. In addition, we also show the per-class comparison on precisions (Fig 5) at $r = 100$ after the 1st round. Our method exceeds the manifold ranking in 13 classes (out of the total 15 classes).

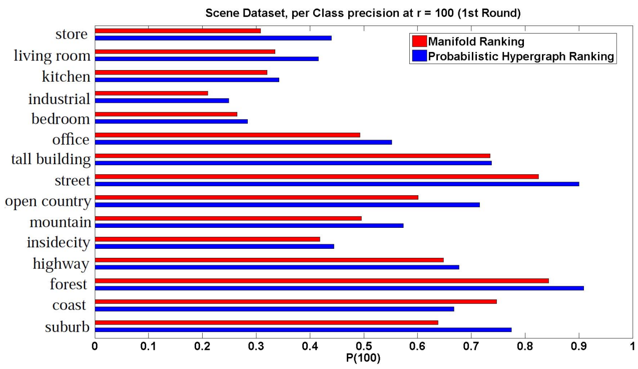


Figure 5. Per-class precisions for *Scene dataset* at $r = 100$ after the 1st round. Best viewed in color.

To demonstrate the scalability of our algorithm, we also conduct a comparison on *Caltech-101* [19] which contains 9146 images grouped into 101 distinct object classes and a background class. For *Caltech-101*, both the optimal hyperedge size and the optimal vertex degree of the simple graph are $K = 40$. The precision-recall curves are shown in Fig 7, in which we can observe the advantage of the probabilistic hypergraph ranking on both the hypergraph ranking and the manifold ranking.

Above analysis confirms our proposed method from two aspects: (1) by considering the local grouping information, both hypergraph models can better approximate relevance between the labeled data and unlabled images than the simple graph based model; (2) probabilistic incidence matrix H is more suitable for defining the relationship between vertices in a hyperedge.

6. Conclusion

We introduced a transductive learning framework for content-based image retrieval, in which a novel graph structure – probabilistic hypergraph is used to represent the relevance relationship among the vertices (images). Based on the similarity matrix computed from complementary image features, we take each image as a ‘centroid’ vertex and form a hyperedge by a centroid and its k-nearest neighbors. We adopt a probabilistic incidence structure to describe the local grouping information and the probability that a vertex belongs to a hyperedge. In this way, the task of image retrieval with relevance feedback is converted to a transductive learning problem which can be solved by the hypergraph ranking algorithm. The effectiveness of the proposed method is demonstrated by extensive experimentation on three general purpose image databases.

References

[1] S. Agarwal, K. Branson, and S. Belongie. Higher order learning with graphs. In *ICML ’06*.

[2] S. Agarwal, J. Lim, L. Zelnik Manor, P. Perona, D. Kriegman, and S. Belongie. Beyond pairwise clustering. In *CVPR’05*.

[3] C. J. Alpert and A. B. Kahng. Recent directions in netlist partitioning: A survey. *Integration: The VLSI Journal*, 19:1–81, 1995.

[4] M. Bolla. Spectra, Euclidean representations and clustering of hypergraphs. In *Discrete Mathematics*, 1993.

[5] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR ’07*.

[6] D. Cai, X. He, and J. Han. Active subspace learning. In *ICCV’09*.

[7] D. Cai, X. He, and J. Han. Semi-supervised discriminant analysis. In *ICCV’07*.

[8] S.-M. Chen. Interval-valued fuzzy hypergraph and fuzzy partition. *IEEE Trans. on Systems, Man and Cybernetics - Part B Cybernetics*, 27, 1997.

[9] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. The Bayesian image retrieval system, Pichunter: Theory, implementation and psychophysical experiments. *IEEE transactions on image processing*, 9:20–37, 2000.

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR’05*.

[11] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):1–60, April 2008.

[12] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary.

[13] J. He, M. Li, H. Zhang, H. Tong, and C. Zhang. Generalized manifold-ranking-based image retrieval. *IEEE transaction on Image Processing*, 15(10):3170–3177, October 2006.

[14] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang. Manifold-ranking based image retrieval. In *ACM MULTIMEDIA ’04*.

[15] X. He, W.-Y. Ma, O. King, M. Li, and H. Zhang. Learning and inferring a semantic space from user’s relevance feedback for image retrieval. In *ACM MULTIMEDIA ’02*.

[16] S. C. H. Hoi and M. R. Lyu. A semi-supervised active learning framework for image retrieval. In *CVPR’05*.

[17] Y. Huang, Q. Liu, and D. Metaxas. Video object segmentation by hypergraph cut. CVPR’09.

[18] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR’06*.

[19] F. Li, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 2007.

[20] F.-F. Li and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR’05*.

[21] D. Lowe. Object recognition from local scale-invariant features. In *ICCV’09*.

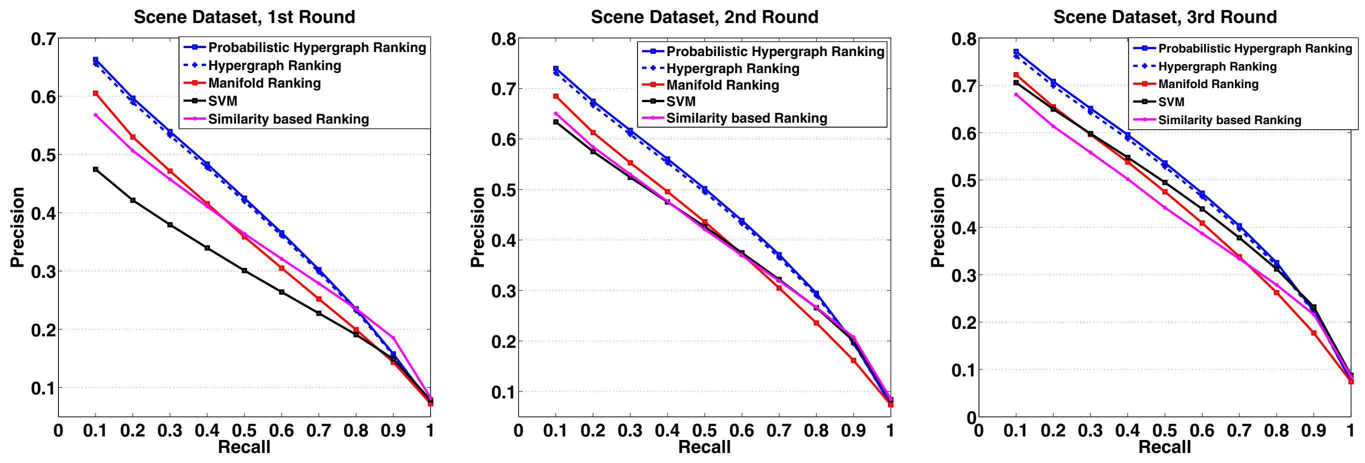


Figure 6. The precision-recall curves for *Scene dataset* under the passive learning setting. Best viewed in color.

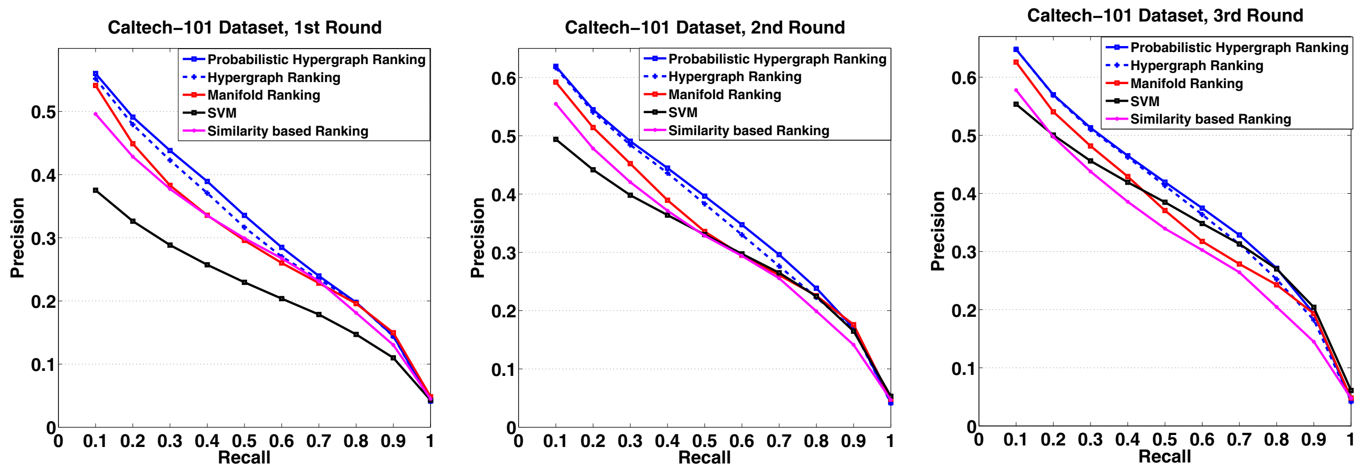


Figure 7. *The precision-recall curves for Caltech-101*. Best viewed in color.

[22] S. MacArthur, C. Brodley, and C. Shyu. Relevance feedback decision trees in content-based image retrieval. In *CBAIVL '00: Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries*, page 68, 2000.

[23] J. Rodríquez. On the laplacian spectrum and walk-regular hypergraphs. In *Linear and Multilinear Algebra*, 2003.

[24] Y. Rui, T. S. Huang, and S.-F. Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10(1):39–62, March 1999.

[25] H. Sahbi, J. Audibert, and R. Keriven. Graph-cut transducers for relevance feedback in content based image retrieval. In *ICCV'07*.

[26] L. Sun, S. Ji, and J. Ye. Hypergraph spectral learning for multi-label classification. In *SIG KDD '08*.

[27] Z. Tian, T. Hwang, and R. Kuang. A hypergraph-based learning algorithm for classifying gene expression and array cgh data with prior knowledge. *Bioinformatics*, July 2009.

[28] K. Tieu and P. Viola. Boosting image retrieval. In *International Journal of Computer Vision*, pages 228–235, 2000.

[29] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *ACM MULTIMEDIA '01*.

[30] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (in press), 2010.

[31] J. van Gemert, J. Geusebroek, C. Veenman, and A. Smeulders. Kernel codebooks for scene categorization. In *ECCV'08*.

[32] R. Zass and A. Shashua. Probabilistic graph and hypergraph matching. In *CVPR'08*.

[33] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schökopf. Learning with local and global consistency. In *NIPS'03*.

[34] D. Zhou, J. Huang, and B. Schökopf. Learning with hypergraphs: Clustering, classification, and embedding. In *NIPS'06*.

[35] D. Zhou, J. Huang, and B. Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *ICML'05*.

[36] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML'03*.

[37] J. Y. Zien, M. D. F. Schlag, and P. K. Chan. Multi-level spectral hypergraph partitioning with arbitrary vertex sizes. In *Proc. ICCAD*, pages 201–204. IEEE Press, 1996.