

## Research Article

# Learn to Rank Images: A Unified Probabilistic Hypergraph Model for Visual Search

**Kaiman Zeng, Nansong Wu, Arman Sargolzaei, and Kang Yen**

*Department of Electrical and Computer Engineering, Florida International University, 10555 West Flagler Street, Miami, FL 33174, USA*

Correspondence should be addressed to Kaiman Zeng; [kzeng001@fiu.edu](mailto:kzeng001@fiu.edu)

Received 5 March 2016; Revised 11 June 2016; Accepted 16 June 2016

Academic Editor: Daniel Zaldivar

Copyright © 2016 Kaiman Zeng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In visual search systems, it is important to address the issue of how to leverage the rich contextual information in a visual computational model to build more robust visual search systems and to better satisfy the user's need and intention. In this paper, we introduced a ranking model by understanding the complex relations within product visual and textual information in visual search systems. To understand their complex relations, we focused on using graph-based paradigms to model the relations among product images, product category labels, and product names and descriptions. We developed a unified probabilistic hypergraph ranking algorithm, which, modeling the correlations among product visual features and textual features, extensively enriches the description of the image. We conducted experiments on the proposed ranking algorithm on a dataset collected from a real e-commerce website. The results of our comparison demonstrate that our proposed algorithm extensively improves the retrieval performance over the visual distance based ranking.

## 1. Introduction

Ranking plays an essential role in a product search system. Given a query, candidate products should be ranked according to their distance to the query. The effectiveness of the product search system is evaluated by its ranked search results, for example, in the form of precision or recall. In addition, the efficiency of a system is evaluated by its running time of a query. The best scenario is that the system returns a series of relevant products at the top of retrieved results. However, in certain cases, even if a system finds the particular relevant product, it is still considered as ineffective for the reason that the retrieved product is not present in the top list but is buried in a number of irrelevant results. In order to compensate for this rank inversion issue, the automated learning techniques and the skills of users are utilized to improve the representation of the query product.

A natural extension of such add-value process is to request users to label the returned results as relevant or irrelevant, which is called relevance feedback (RF). However, in reality users are neither willing to initialize a query by labeling retrieval metadata and samples nor willing to

give feedback of the retrieved results, since these methods make the retrieval procedure inconvenient. Therefore, the insufficient user-labeled images undermine the prospect of supervised learning methods in the visual content-based image retrieval (CBIR) field. A promising and relatively unexplored research direction is to exploit transductive or semisupervised learning, among which graph-based methods [1–4] have demonstrated their effectiveness in image retrieval and therefore received increasing attention. In the graph-based methods, a graph is built on the image dataset and each image is considered as a vertex in the graph. An edge and its weight are defined between two images according to a certain relationship definition. For example, the edge can be defined as the images visual similarity. The weight is formulated by the visual distance between any two image vertices. Then, the ranking can be formulated as a random walk on the graph [1, 5], or an optimization problem [2]. However, these graphs, created in pairs, cannot sufficiently show the relations among images. Hypergraph is introduced to the CBIR field. Hypergraph is a generalization of a simple graph. In a hypergraph, an edge, called hyperedge, can connect any number of vertices; it is a nonempty set of vertices.

A probabilistic hypergraph is proposed in [3] for image retrieval ranking. The hyperedge is formed by a centroid image and its  $k$ -nearest neighbors based on their visual similarity. Gao et al. [4] propose a hypergraph learning algorithm for the social image search, where the weight of hyperedges, representing the impact of different tags and visual words, is automatically learned with a set of pseudopositive images. Their research [3, 4] proves the effectiveness of hypergraph learning in solving ranking problems. However, they fail to establish correlations between images visual content and textual content. They both built the hypergraph model on images visual content or textual content independently. As a result, the search must start with a user assigned keyword. In order to solve this issue and minimize user's efforts in query, we take the product rich textual information into the visual model of image retrieval and propose a novel hypergraph-based transductive algorithm for ranking of visual product search. We design a unified probabilistic hypergraph to model multiple types of features of the products and explore the implicit relations among various visual and textual features.

The contributions of this paper are summarized in three aspects as follows. First, hypergraph is proposed to represent a commercial product image dataset. The relations between visual and textual features of these images have been explored. Second, a new product retrieval framework for the product search is designed. Third, a novel strategy of starting a query is created. Since relations between visual features and textual features have been established in a specific unified probabilistic hypergraph, problems that lack user-labeled query keywords can be solved via transductive inference on the hypergraph.

The paper is organized as the following outline. In Section 2, we have a literature review of recent ranking techniques. Section 3 discusses the design of the proposed unified hypergraph ranking algorithm. In Section 4, several retrieval experiments are conducted on an apparel dataset and compared with conventional CBIR ranking methods. Finally, we conclude with the proposed ranking scheme and discuss future works in Section 5.

## 2. Related Work

Ranking and hypergraph learning are the two research fields related to our work. These two topics have received intensive attention in information retrieval and machine learning. The conventional image ranking is developed from textual retrieval. The ranking model is defined based on the bag of words, for example, Best Match 25 [6], the Vector Space Model [7], and the Language Modeling for Information Retrieval [8]. Another type of ranking model is based on hyperlink analysis, such as Hyperlink-Induced Topic Search [9], PageRank [10], and its variations [11–13].

In the CBIR systems, the ranking is commonly obtained from the similarity measure of adopted visual features. One type of similarity measures is calculated from Minkowski distance, City-block distance, infinity distance, and cosine distance. They are usually called Minkowski and standard

measures. Statistical measure, for example, Pearson correlation coefficient and Chi-square dissimilarity, is another type of similarity or dissimilarity measure methods. The third type of similarity measures is divergence measure, which includes Kullback-Leibler divergence, Jeffrey divergence, Kolmogorov-Smirnov divergence, and Cramer-von Mises divergence. There are some other measures, such as Earth Mover's distance and diffusion distance [14, 15].

The learning to rank model has gained increasing attention in recent years, utilizing machine learning algorithms to optimize the ranking function by tuning some of the parameters and incorporating relevance features [16, 17]. Manifold ranking [18], a graph-based semisupervised learning method, ranks the data through exploiting their intrinsic manifold structure. Manifold ranking was firstly applied to CBIR in [19] and significantly improved image retrieval performance. Liu et al. [20] proposed a graph-based approach for tag ranking, by which a tag graph was built to mine the correlations among tags, and the relevance scores were obtained through a random walk over the similarity graph. These researches demonstrated the effectiveness of graph-based semisupervised learning techniques in solving different ranking problems. However, they are inadequate for the relations in images via pairwise graphs solely. It would be of great benefit to take into consideration the relationship among 3 or more vertices. Such a model capturing higher-order relations is called hypergraph. In a hypergraph, a nonempty set of vertices is defined as a weighted or unweighted hyperedge; the magnitude of the weight represents the degree that the vertices in the hyperedge belong to the same cluster. Agarwal et al. [21] firstly introduced hypergraph to computer vision and proposed a clique averaging graph approximation scheme to solve the clustering problems. Literature [22] formulated the probabilistic interpretation based image-matching problem as the hypergraph convex optimization and achieved a global optimum of the matching criteria. However, they set up three restrictions that are the same degree of all hyperedges, the same number of vertices in two graphs, and a complete match. Sun et al. [23] employed the hypergraph to capture the correlations among different labels for multilabel classification. The proposed hypergraph learning formulation showed the effectiveness on large-scale benchmark datasets, and its approximate least squares formulation maintained efficiency, as well as competitive classification performance. One shortage of their work is that they limited the target applications to linear models and thus did not have a general performance evaluation on other multilabel applications, such as the kernel-induced multilabels. In [24] the spatiotemporal relationship among different patches is captured by the hypergraph structure, and the video object segmentation is modeled as hypergraph partition. Furthermore, weights are added on important hyperedges. The experimental results have shown good segmentation performance on nature scenes. In the case that there are several different types of vertices or hyperedges, the hypergraph is called unified hypergraph. L. Li and T. Li [25] proposed a unified hypergraph model for the personalized news recommendation where users and multiple news entities are involved as different types of vertices, and their

implicit correlations are captured. The recommendation is modeled as a hypergraph ranking problem. The hypergraph learning algorithms have demonstrated their capability of capturing complex high-order relations. Their applications in image retrieval are also promising [3, 4].

### 3. Ranking on Unified Probabilistic Hypergraph

In this research, we employ a unified probabilistic hypergraph to represent the relations of commercial product images, its textual descriptions, and its categorization labels. We propose a model for searching and ranking images based on hypergraph learning. Conventional visual search systems sort and search images based on the similarity of their visual content. The idea of this model is to learn the relevance of different product features, images visual feature, textual feature, and the hybrid visual-textual feature, and then combine them with the results of visual similarity based retrieval.

**3.1. Notation and Problem Definition.** Let  $V$  represent a finite set of vertices.  $E$  represents a family of hyperedges on  $V$ , and each hyperedge  $e \in E$  contains a list of vertices that belong to  $V$ . The hypergraph can be denoted as  $G = (V, E, w)$  with a weight function  $w$ . The degree of a hyperedge  $e$  is defined by  $d(e) = |e|$ , that is, the number of vertices in  $e$ . The degree of a vertex  $v$  is defined by  $d(v) = \sum_{e \in E} w(e)$ , where  $w(e)$  is the weight of the hyperedge  $e$ . The hypergraph can be formulated to a vertex-hyperedge incidence matrix  $H \in R^{|V| \times |E|}$ , where each entry  $h(v, e)$  is defined as

$$h(v, e) = \begin{cases} 1 & \text{if } v \in e \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Then we have  $d(v) = \sum_{e \in E} w(e)h(v, e)$ , and  $d(e) = \sum_{v \in V} h(v, e)$ . Let  $D_v$  and  $D_e$  denote the diagonal matrices containing the vertex and hyperedge degrees, respectively, and let  $W$  be the  $|E| \times |E|$  diagonal matrix containing the weights of hyperedges.

Consider a simple example of hypergraph  $G = (V, E)$ , built as shown in Figure 1. Consider  $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}\}$  and  $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ . The incidence matrix  $H$  is defined as

$$H = \begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ v_2 & 1 & 0 & 0 & 0 & 0 & 0 \\ v_3 & 1 & 1 & 0 & 0 & 0 & 1 \\ v_4 & 0 & 0 & 0 & 0 & 0 & 1 \\ v_5 & 0 & 1 & 0 & 0 & 0 & 0 \\ v_6 & 0 & 1 & 1 & 0 & 0 & 0 \\ v_7 & 0 & 1 & 1 & 0 & 0 & 0 \\ v_8 & 0 & 0 & 1 & 0 & 0 & 0 \\ v_9 & 0 & 0 & 0 & 0 & 1 & 0 \\ v_{10} & 0 & 0 & 0 & 1 & 0 & 0 \\ v_{11} & 0 & 0 & 0 & 1 & 0 & 0 \\ v_{12} & 0 & 0 & 0 & 1 & 0 & 0 \end{array} \quad (2)$$

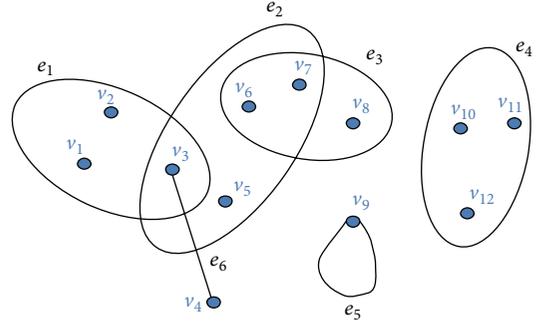


FIGURE 1: An example of a hypergraph.

The problem of ranking on the hypergraph is formulated as follows: given a query vector  $y$ , a subset of vertices in the hypergraph  $G = (V, E, w)$ , a ranking score vector  $f$  is produced according to the relevance among vertices in the hypergraph and the query. We define the cost function of  $f$  as follows [4]:

$$\begin{aligned} \Omega(f) &= \frac{1}{2} \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} \sum_{e \in E} \frac{1}{d(e)} \sum_{\{v_i, v_j\} \in e} w(e) \left\| \frac{f_i}{\sqrt{d(v_i)}} - \frac{f_j}{\sqrt{d(v_j)}} \right\|^2 \\ &+ \mu \sum_{i=1}^{|V|} \|f_i - y_i\|^2, \end{aligned} \quad (3)$$

where  $\mu > 0$  is the regulation factor. The first term, known as the normalized hypergraph Laplacian, is a constraint that vertices sharing many incidental hyperedges are supposed to obtain similar ranking scores. The second term is a constraint of the variation between the final ranking score and the initial score.

In order to obtain the optimal solution of the ranking problem, we seek to minimize the cost function:

$$f^* = \arg \min \Omega(f). \quad (4)$$

With the derivations in [4], we can rewrite the cost function as

$$\Omega(f) = f^T (I - \Theta) f + \mu (f - y)^T (f - y), \quad (5)$$

where  $\Theta = D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2}$ . Then the optimal  $f^*$  can be obtained by differentiating  $\Omega(f)$  with respect to  $f$ :

$$\left. \frac{\partial \Omega}{\partial f} \right|_{f=f^*} = (I - \Theta) f^* + \mu (f - y)^* = 0, \quad (6)$$

$$f^* = \left( I - \frac{1}{1 + \mu} \Theta \right)^{-1} y. \quad (7)$$

**3.2. Unified Probabilistic Hypergraph Ranking Model.** In the following we will explain our improved hypergraph formulation for the product retrieval and ranking. In a typical

online shopping system there are three different types of information representing a product. They are product image, product name and description, and product labels, which are discussed in detail in Section 4. With these three types of information we design 6 types of hyperedges. Each image in the product image dataset is considered as a vertex in the unified hypergraph. Let  $X$  denote the product image pool, and  $x_i \in X$  is a particular product image. Let  $F$  denote the visual feature description of the images or, say, visual words, let  $S$  denote the set of product styles, and let  $N$  be the name and description of the product. The unified hypergraph  $G$  that contains 6 different types of hyperedge could represent the following implicit relations:

- (1)  $E^{\text{FSN}}$  (the set of images feature-style-name hyperedges): the products share the same product name, product style, and visual feature word.
- (2)  $E^{\text{FS}}$  (the set of images feature-style hyperedges): the products, which belong to a certain product style, contain the same visual feature word.
- (3)  $E^{\text{FN}}$  (the set of images feature-name hyperedges): the products, containing the same visual feature word, share a common keyword in name.
- (4)  $E^{\text{F}}$  (the set of images visual feature hyperedges): the product images might contain the same visual feature word.
- (5)  $E^{\text{S}}$  (the set of images style hyperedges): the products belong to the same product style.
- (6)  $E^{\text{N}}$  (the set of images name hyperedges): the products have similar keywords in their name and description.

Typically we assign 1 to the weights of these hyperedges. Rather than traditional hypergraph structure, in which an image vertex  $x_i$  is assigned to a hyperedge  $e_j$  in a binary way (i.e.,  $h(x, e)$  is either 1 or 0), we propose a probabilistic hypergraph to describe the relation between vertex and hyperedge. For hyperedge  $E^{\text{F}}$ , each image vertex is treated as a centroid, and the hyperedge is formed by the centroid image and its  $k$ -nearest neighbors. The incidence matrix  $H$  of the probabilistic hypergraph is defined as follows:

$$h(x_i, e_j) = \begin{cases} \text{Sim}(x_j, x_i) & \text{if } x_i \in e_j \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

where  $x_j$  is the centroid of  $e_j$ . In the proposed formulation, a vertex  $x_i$  is softly assigned to a hyperedge  $e_j$  based on the similarity between  $x_j$  and  $x_i$ , which overcomes the limitation of truncation loss with the binary assignment. Besides, we use a parameter to set the desired similarity.

Figure 2 demonstrates an example of constructing such hyperedges. Each vertex and its top 3 similar neighbors form a hyperedge. We set a constraint that only the vertices with a pair similarity larger than 0.4 would be connected

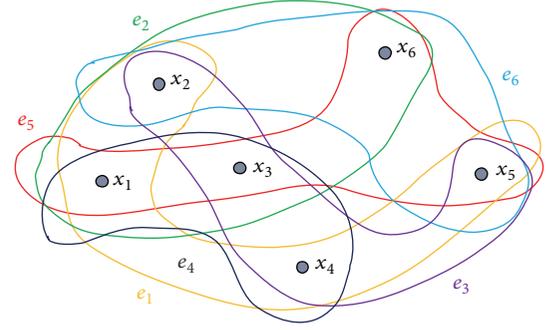


FIGURE 2: An example of probabilistic hypergraph.

into a hyperedge. The incidence matrix  $H$  of the proposed probabilistic hypergraph is

$$H = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ x_1 & 1 & 0.81 & 0 & 0.63 & 0.45 & 0 \\ x_2 & 0.81 & 1 & 0.43 & 0.52 & 0 & 0 \\ x_3 & 0 & 0.43 & 1 & 0.83 & 0.62 & 0 \\ x_4 & 0.63 & 0.52 & 0.83 & 1 & 0 & 0 \\ x_5 & 0.45 & 0 & 0.62 & 0 & 1 & 0.8 \\ x_6 & 0 & 0 & 0 & 0 & 0.8 & 1 \end{matrix} \quad (9)$$

With the hyperedges as designed above we can form the 6 types of unified weight matrix  $W$  and have the vertex-hyperedge incidence matrix  $H$ . The size of both matrices depends on the cardinality of product image dataset involved, and they are all sparse matrices. As a result, the computation of the proposed hypergraph ranking algorithm is fast. It is implemented in two stages: offline training and online ranking. In the offline training stage, we construct the unified hypergraph with matrices  $H$  and  $W$  derived from above. Then based on the matrices, we calculate the vertex degree matrix  $D_v$  and the hyperedge degree matrix  $D_e$ . Finally  $(I - (1/(1 + \mu))\Theta)^{-1}$  can be computed, where  $\Theta = D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2}$ . Note that  $I - (1/(1 + \mu))\Theta$  is invertible, since the hyperedge  $E^{\text{kNN}}$  ensures that  $H$  is full rank. Then the online ranking procedure can be described as follows: firstly build the query vector  $y$ , and secondly compute the ranking score vector  $f^*$ . The elements of the preranked relevant images are set to 1, and the others are 0. The procedures are described in Algorithm 1.

*Algorithm 1* (the unified probabilistic hypergraph ranking algorithm description).

*Input.* The inputs are initial query vector of ranked vertices  $y$ , similarity matrix  $\text{Sim}$ , matrix of textual features  $S$ ,  $N$ , and matrix of visual features  $F$  for all products vertices.

*Output.* The output is vector of optimal ranked product vertices  $f^*$ :

- (1) Construct the vertex-hyperedge incidence matrix  $H_1$  of hyperedge  $E^{\text{F}}$  based on (8).

TABLE 1: A typical product representation in a system.

Product image	Product name and short description	Product labels
	(i) BCBGMAXAZRIA Dress, Leyla Sleeveless Keyhole Lace Fit and Flare (ii) Round neck, sleeveless, keyhole front, contrast lace bodice, contrast waistband, drop waist	Occasion: cocktail Type: fit and flare Type: lace Length: short Sleeve length: sleeveless

- (2) Construct the vertex-hyperedge incidence matrix  $H_2$  of hyperedges  $E^S, E^N, E^{FS}, E^{FN},$  and  $E^{FSN}$  based on (1).
- (3) Form the incidence matrix  $H$  by concatenating  $H_1$  to  $H_2$ .
- (4) Calculate the vertex degree matrix  $D_v$  and the hyper-edge degree matrix  $D_e$  using  $d(e_i) = \sum_{v \in V} h(v_j, e_i)$  and  $d(v_i) = \sum_{e \in E} w(e_j)h(v_i, e_j)$ , respectively.
- (5) Compute  $(I - (1/(1 + \mu))\Theta)^{-1}$ , where  $\Theta = D_v^{-1/2}HWD_e^{-1}H^T D_v^{-1/2}$ .
- (6) Compute the optimal  $f^*$  by (7).

#### 4. Experimental Results

In the experiment, we build the unified images hypergraph using different combinations of hyperedges to test the effect of different factors on the ranking performance. We then investigate the performance of different hypergraphs. The superiority of the transductive inference is demonstrated in handling the queries that lack user labels. We use the visual similarity based ranking as a baseline. We compare the different hypergraph-based ranking models with the visual similarity ranking. Also we use the visual similarity ranking score to deduce the preranked score in hypergraph ranking.

For an online shopping system, a product is represented by three types of information, as shown in Table 1: (1) images, which demonstrate the product visually (this usually has several photos taken from different viewpoint); (2) name, which is the name of the product or gives a brief description of the product; (3) labels, which are the textual tags that classify the product into different categories according to the sorting rules. For example, for apparel products, we could have different categories like style, length, sleeve length, occasions, and so forth.

The product image dataset used in the experiment is obtained from a list of prominent brands of women apparel. It contains 3 product categories, 58 brands, and 4210 images. We use different dress categories such as type, length, and sleeve length to form the set of product styles, which contains 7 types, 3 lengths, and 6 sleeve lengths. The product name is the product brand, its style name, and a short description. Here we generate a bag of words to represent it. For visual features, we first extract a color boosted SIFT feature [26], which captures the product color feature and its local patterns, and

then quantize the visual feature descriptors into 65 visual words. For parameters  $k$  and  $\mu$ , we follow the setting in literature [4], where they are empirically set to 100 and 0.001. We choose Normalized Discounted Cumulative Gain (NDCG) [27] to evaluate the ranking performance. NDCG is widely adopted in machine learning approaches to ranking. It is designed for situations of nonbinary notions of relevance. In our research, an experiment participant is asked to judge the relevance of each retrieval result to the query. Each returned image is to be judged on a scale of 0–3 with  $rel = 0$  meaning irrelevant,  $rel = 3$  meaning completely relevant, and  $rel = 1$  and  $rel = 2$  meaning “somewhere in between.” NDCG at position  $k$  is defined as

$$NDCG@k = \frac{\sum_{i=1}^k ((2^{rel_i} - 1) / \log_2(i + 1))}{IDCG}. \quad (10)$$

In our experiment, the relevance is assessed over a subset of the collection that is formed from the top  $k$  results returned by different ranking methods. Human’s judgments are subjective, idiosyncratic, and variable. Thus, we employ kappa statistic [28] to evaluate the degree of agreement between judges:

$$\text{kappa} = \frac{p(A) - p(E)}{1 - p(E)}, \quad (11)$$

where  $p(A)$  is the proportion of the times the judges agreed and  $p(E)$  is the proportion of the times they should agree by chance.

In our proposed method, we integrate 7 different relations and hyperedges into constructing product hypergraph so that it effectively represents the product image dataset. The hypergraph also encloses multiple correlations among different visual words and text features. To evaluate the effectiveness of such a representation in product search, we consider different hypergraph constructions with different hyperedge integration. Figure 3 illustrates the ranking performance in terms of average NDCG at different depths of 10, 20, and 30. It is evident that the hybrid hypergraph (FSN, FN, and FS) outperforms the simple construction of hypergraph (F) and the visual similarity based ranking ( $k$ NN). And the proposed unified hybrid hypergraph FSN achieves the best performance. The reason for this is quite straightforward: high-order correlations among product visual features and its textual labels are well captured in our unified hypergraph model. The representation and description of a product are extensively enhanced in database.

In Figure 4, an example of a query is demonstrated, in which the system cannot find the best match at the top 10. With the similarity ranking, a black tuxedo jumpsuit is recognized as dress, pants, and coats, while with the proposed unified probabilistic hypergraph learning ranking, the system provides a series of products with similar styles, which is meaningful for the online shoppers. The reason is that we not only capture the visual feature and textual feature separately, but also model the correlations between them. In this way, improved search results are produced.

Then, we use kappa statistics to assess relevance. Here  $rel = 1$  and  $rel = 2$  are considered as agreed. If two judges agree

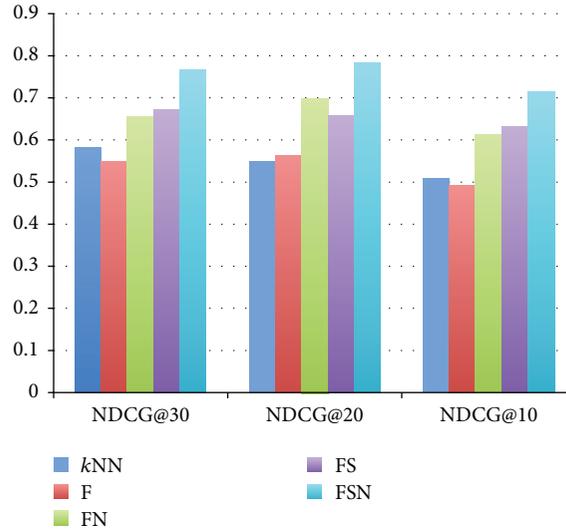


FIGURE 3: Performance comparison of different hypergraph constructions.



FIGURE 4: Top 10 retrieval results with different ranking methods. (a) Visual similarity based ranking. (b) Proposed unified probabilistic hypergraph learning ranking.

on relevance for all results, kappa = 1. If they agree at the probability of agreement by chance, kappa = 0. kappa < 0, if they agree at a rate below random probability. Table 2 gives an example of relevance rating data by two judges on our experiment results. The kappa statistic value between judge 1 and judge 2 is calculated with (11) as 0.7740. The kappa values for all pairs of judges are evaluated in the same way, and their average pairwise kappa value is 0.7483. The level of agreement in our experiments falls in the range of “fair” (0.67–0.8). As a result, the evaluation results of our experiments are proved to be valid.

## 5. Conclusion

In this paper, we address the problem of ranking in product search by image. We focus on integration of various types of product textual information and visual image. We introduce a hypergraph learning approach to the visual product search and propose a more comprehensive and robust ranking model. In this way the supervised classification and

TABLE 2: An example of relevance rating data by two judges.

	Judge 1			Total
	Relevant	Between	Nonrelevant	
Judge 2				
Relevant	132	11	0	143
Between	21	207	21	249
Nonrelevant	0	14	74	88
Total	153	232	95	480

unsupervised visual search are well balanced. Specifically, we construct the hypergraph by combining three types of product information that embed the relevance among textual features and visual images. Experimental results show that the proposed hypergraph learning framework is a promising ranking scheme for product search. In future work we will consider exploring the adaptive feature weight and other hypergraph learning operators.

## Competing Interests

The authors declare that they have no competing interests.

## References

- [1] W. H. Hsu, L. S. Kennedy, and S.-F. Chang, "Video search reranking through random walk over document-level context graph," in *Proceedings of the 15th ACM International Conference on Multimedia (MM '07)*, pp. 971–980, September 2007.
- [2] Y. Jing and S. Baluja, "VisualRank: applying pagerank to large-scale image search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1877–1890, 2008.
- [3] Y. Huang, Q. Liu, S. Zhang, and D. N. Metaxas, "Image retrieval via probabilistic hypergraph ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 3376–3383, IEEE, San Francisco, Calif, USA, June 2010.
- [4] Y. Gao, M. Wang, Z.-J. Zha, J. Shen, X. Li, and X. Wu, "Visual-textual joint relevance learning for tag-based social image search," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 363–376, 2013.
- [5] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: from pairwise approach to listwise approach," in *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, pp. 129–136, ACM, Corvallis, Ore, USA, June 2007.
- [6] S. E. Robertson and D. A. Hull, *The TREC-9 Filtering Track Final Report*, 2000.
- [7] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.
- [8] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98)*, pp. 275–281, Melbourne, Australia, August 1998.
- [9] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.
- [10] S. Brin and L. Page, "Reprint of: the anatomy of a large-scale hypertextual web search engine," *Computer Networks*, vol. 56, no. 18, pp. 3825–3833, 2012.
- [11] M. R. Henzinger, "Hyperlink analysis for the web," *IEEE Internet Computing*, vol. 5, no. 1, pp. 45–50, 2001.
- [12] R. Baeza-Yates and E. Davis, *Web Page Ranking Using Link Attributes*, ACM, 2004.
- [13] R. Lempel and S. Moran, "The stochastic approach for link-structure analysis (SALSA) and the TKC effect," *Computer Networks*, vol. 33, no. 1, pp. 387–401, 2000.
- [14] M. Arevalillo-Herráez, J. Domingo, and F. J. Ferri, "Combining similarity measures in content-based image retrieval," *Pattern Recognition Letters*, vol. 29, no. 16, pp. 2174–2181, 2008.
- [15] J. Collins and K. Okada, "A comparative study of similarity measures for content-based medical image retrieval," in *Proceedings of the Cross Language Evaluation Forum Conference (CLEF '12)*, Rome, Italy, September 2012.
- [16] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: from pairwise approach to listwise approach," in *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, pp. 129–136, ACM, June 2007.
- [17] T.-Y. Liu, "Learning to rank for Information retrieval," *Foundations and Trends in Information Retrieval*, vol. 3, no. 3, pp. 225–231, 2009.
- [18] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," *Advances in Neural Information Processing Systems*, vol. 16, no. 16, pp. 321–328, 2004.
- [19] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang, "Generalized manifold-ranking-based image retrieval," *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 3170–3177, 2006.
- [20] D. Liu, X. S. Hua, L. Yang, M. Wang, and H. J. Zhang, "Tag ranking," in *Proceedings of the ACM 18th International Conference on World Wide Web (WWW '09)*, Madrid, Spain, April 2009.
- [21] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, and S. Belongie, "Beyond pairwise clustering," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 838–845, IEEE, San Diego, Calif, USA, June 2005.
- [22] R. Zass and A. Shashua, "Probabilistic graph and hypergraph matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, IEEE, Anchorage, Alaska, USA, June 2008.
- [23] L. Sun, S. Ji, and J. Ye, "Hypergraph spectral learning for multi-label classification," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*, pp. 668–676, ACM, 2008.
- [24] Y. Huang, Q. Liu, and D. Metaxas, "Video object segmentation by hypergraph cut," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 1738–1745, June 2009.
- [25] L. Li and T. Li, "News recommendation via hypergraph learning: encapsulation of user behavior and news content," in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM '13)*, pp. 305–314, ACM, Rome, Italy, February 2013.
- [26] K. Zeng, N. Wu, and K. K. Yen, "A color boosted local feature extraction method for mobile product search," *International Journal on Recent Trends in Engineering and Technology*, vol. 10, no. 2, pp. 78–84, 2014.
- [27] G. Karypis and V. Kumar, "Multilevel k-way hypergraph partitioning," *VLSI Design*, vol. 11, no. 3, pp. 285–300, 2000.
- [28] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, New York, NY, USA, 2009.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

