iRIN: Image Retrieval in Image-Rich Information Networks*

Xin Jin
University of Illinois at Urbana-Champaign
xinjin3@illinois.edu

Jiebo Luo
Kodak Research Laboratories
Eastman Kodak Company
jiebo.luo@kodak.com

Jie Yu
Kodak Research Laboratories
Eastman Kodak Company
jie.yu@kodak.com

Gang Wang
University of Illinois at Urbana-Champaign
gwang6@illinois.edu

Dhiraj Joshi
Kodak Research Laboratories
Eastman Kodak Company
dhiraj.joshi@kodak.com

Jiawei Han
University of Illinois at Urbana-Champaign
hanj@cs.uiuc.edu

ABSTRACT

In this demo, we present a system called iRIN designed for performing image retrieval in image-rich information networks. We first introduce MoK-SimRank to significantly improve the speed of SimRank, one of the most popular algorithms for computing node similarity in information networks. Next, we propose an algorithm called SimLearn to (1) extend MoK-SimRank to heterogeneous image-rich information network, and (2) account for both link-based and content-based similarities by seamlessly integrating reinforcement learning with feature learning.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**]: Image Processing and Computer Vision

General Terms

Algorithms, Experimentation

Keywords

Image Retrieval, Information Network, Ranking

1. INTRODUCTION

In this paper, we study the problem of performing image retrieval in image-rich information network. Social image sharing websites, such as Flickr and Facebook, have billions of user submitted images and the users interact with the images by social annotations and interest groups, thus forming image-rich information networks. Take Flickr as an example, images are tagged by the users and image owners contribute images to topic groups, forming an information network, as shown in Figure 1. Figure 2 shows another network of Amazon products with product images.

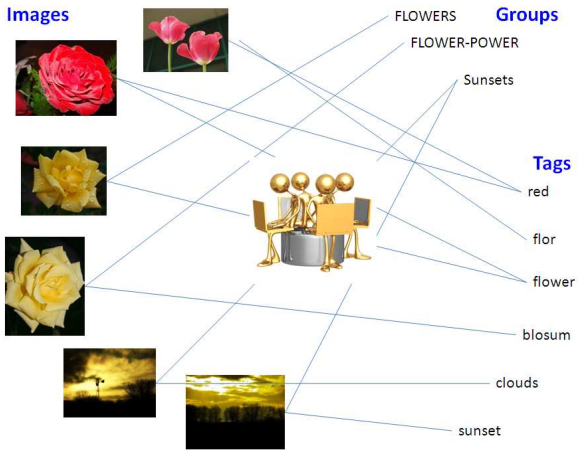

Figure 1: Flickr image information network.

Searching images in such large information networks is very useful but also challenging, for example, user annotations are noisy, incomplete and there are many similar interests or product groups. For keyword based retrieval, we need to find similar annotations to avoid missing relevant images. WordNet does not work for such noisy terms, while Google Distance is too general. For content based image retrieval, traditional methods [2] are only based on image features (or the surrounding text) and do not consider the *network structure*. In addition, these tasks are traditionally treated separately. However, by forming an image information network, we can solve them simultaneously within a general framework.

Definition. We model a heterogeneous image information network as a graph $G = (V, E)$ with vertices/nodes V and edges E. An edge is created if there exists a link between two nodes. Each image node is associated with $F \in R^D$, a D-dimension image feature. In this paper, we consider a graph

*This research was supported by Kodak and also sponsored in part by the U.S. National Science Foundation under grants IIS-08-42769 and IIS-09-05215, and by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (NS-CTA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

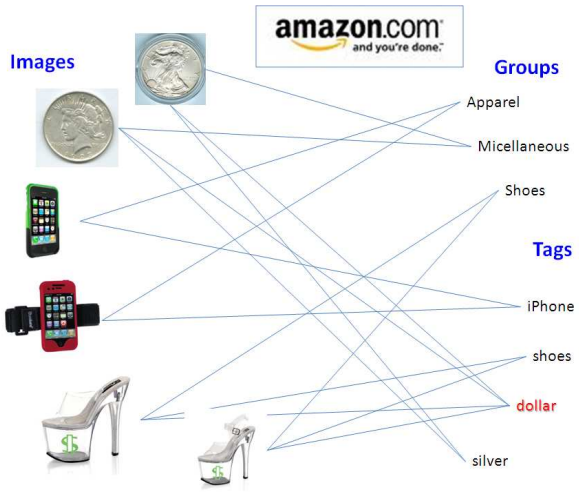

Figure 2: Amazon image information network.

with three types of nodes (images V_I, groups V_G and tags V_T).

SimRank [3] is one of the most popular algorithms for evaluating object similarity in information networks. It calculates the similarity between objects based on the intuition that *two objects are similar if they are linked by similar objects in the network.*

There are two disadvantages with SimRank: (1) It is expensive to compute and not scalable to large datasets. (2) It measures object similarity solely by link information. However, in image-rich information networks, object similarity can also be estimated by image content feature.

To address the above two problems, we introduce an efficient approach called MoK-SimRank to significantly improve the speed of SimRank, and propose an algorithm called SimLearn to consider both link and content information by seamlessly integrating reinforcement learning with feature learning.

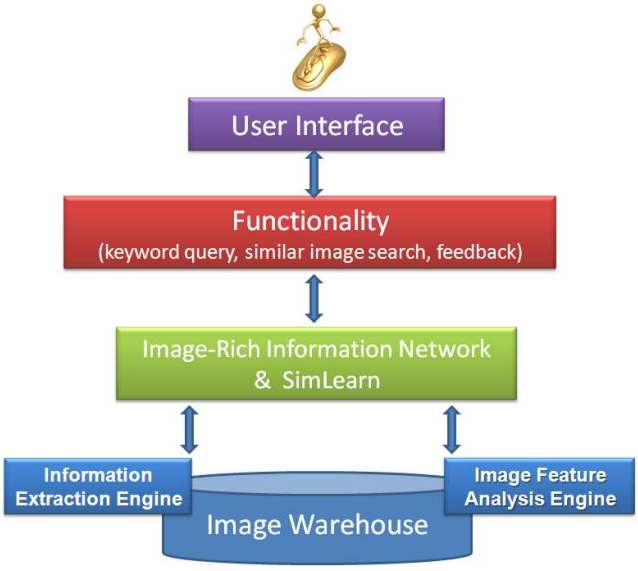

Figure 3: System architecture of iRIN.

2. GENERAL SYSTEM ARCHITECTURE

The iRIN system has a four-layer architecture, as shown in Figure 3. The bottom layer contains an image warehouse, and the information extraction and image feature analysis engines. The lower intermediate layer builds image-rich information network and performs link+content based similarity ranking. The upper intermediate layer is the functional module layer, which implements the major function modules including the ranking information derived from the information network analysis. The top layer contains a user-friendly interface, which interacts with users, responds to their requests, and collects feedback.

3. ALGORITHMS

3.1 MoK-SimRank for Fast Computation

In a homogeneous network, the SimRank similarity score between two objects o and o' is defined as,

$$S(o,o') = \frac{C}{|L(o)||L(o')|}\sum_{a=1}^{|L(o)|}\sum_{b=1}^{|L(o')|} S(L_a(o), L_b(o')) \quad (1)$$

where C is a constant between 0 and 1, $L(o)$ is the set of nodes which link to object o, and $L_i(o)$ is ith object in $L(o)$. To compute SimRank in a network G of N nodes, the space required is $O(N^2)$ to store the similarity scores for all pairs of objects. Let P be the time required to compute Equation 1. The time complexity is $O(N^2P)$ for each iteration.

Among several algorithms proposed for fast SimRank computation, we adopt the pruning approach discussed in [3]. For each object, we choose top $K(K \ll N)$ potentially similar objects as candidates and only compute SimRank for the candidates, thus reduce the time complexity to $O(NKP)$ and space complexity to $O(NK)$. We call this approach K-**SimRank**. This strategy is suitable for image retrieval at large scales where most images are not similar to the query and there is no need to estimate their similarity again and again.

In K-SimRank, the time complexity of P is $O(|L_o||L_{o'}|log(K))$, where $log(K)$ is the complexity to decide whether $L_j(o')$ is a candidate of object $L_i(o)$. We propose a more efficient approach. Denote $T(c)$ as the set of top K similar candidates of object c. Among $L(o)$ and $L(o')$, denote L_{max} as the one that has bigger cardinality and L_{min} as the smaller one. Our method works as follows,

1. Starting with L_{min}, for each object $c \in L_{min}$, sum up the following scores;
2. If $k < |L_{max}|$, for each object $d \in T(c)$, search within L_{max} to decide whether $d \in L_{max}$. If yes, return the score; otherwise, return 0;
3. Otherwise, for each object $d \in L_{max}$, search within $T(c)$ to decide whether $d \in T(c)$. If yes, return the score; otherwise, return 0;

The above procedure reduces the time complexity of F to $O(|L_{min}|Klog(|L_{max}|))$ (when $K < |L_{max}|$) or $O(|L_{min}||L_{max}|log(K))$ (when $K > |L_{max}|$), which is the optimal combination with the minimum cost achieved by automatically choosing the minimum optimal order of computation. We call our approach **MoK-SimRank** (minimum order K-SimRank).

3.2 SimLearn

We propose algorithm SimLearn to (1) extend MoK-SimRank to heterogeneous image-rich information networks, and (2) consider both link-based and content-based similarity by seamlessly integrating reinforcement learning with feature learning.

3.2.1 Link-based Semantic Similarity

In an image-rich information network, similar images are likely to link to similar groups and tags, so we define the link-based semantic similarity between images i and j as follows,

$$S_{m+1}(i,j) = \alpha_I S_m^G(i,j) + \beta_I S_m^T(i,j) \tag{2}$$

$$S_m^G(i,j) = \frac{C}{|L^G(i)||L^G(j)|} \sum_{a=1}^{|L^G(i)|} \sum_{b=1}^{|L^G(j)|} S_m(L_a^G(i), L_b^G(j)) \tag{3}$$

$$S_m^T(i,j) = \frac{C}{|L^T(i)||L^T(j)|} \sum_{a=1}^{|L^T(i)|} \sum_{b=1}^{|L^T(j)|} S_m(L_a^T(i), L_b^T(j)) \tag{4}$$

where $L^G(i)$ is set of groups image i links to, $L^T(i)$ is set of tags it links to.

Similarly, we can define the group and tag similarity,

$$S_{m+1}(g,g') = \alpha_G S_m^I(g,g') + \beta_G S_m^T(g,g') \tag{5}$$

$$S_{m+1}(t,t') = \alpha_T S_m^I(t,t') + \beta_T S_m^G(t,t') \tag{6}$$

The group similarity is computed via the similarity of the images and tags they link to, and the tag similarity is calculated via the similarity of the images and groups they link to.

3.2.2 Weighted Content-based Similarity

In addition to link information, image similarity can also be estimated from image content features, such as color/edge histogram, CEDD, GIST, texture, shape and bag-of-word SIFT histogram. An image can be represented as a point in a D-dimension feature space, which consists of either a single type of feature or a combination of multiple types of features.

Instead of directly using the feature vector to compute similarity, many studies in recent years have demonstrated, both empirically and theoretically, that a learned metric can significantly improve the performance of classification and clustering [4]. The reason is that the feature dimensions are not equally important for evaluating image similarity. By identifying the subspace that is most relevant to the semantic meaning of images, we can achieve better performance.

Given a feature weighting vector W and the χ^2 test statistic distance [1] (which shows good performance compared with cosine and L2 measure for image similarity), we define the weighted content similarity between images i and j as:

$$C_{ij}^W \equiv 1 - \frac{1}{2} \sum_{d=1}^{D} \frac{(w^d f_i^d - w^d f_j^d)^2}{w^d f_i^d + w^d f_j^d} = 1 - \sum_{d=1}^{D} w^d x_{ij}^d \tag{7}$$

3.2.3 Feature Weight Learning

To build a bridge between the content and semantics, we learn a weighting vector $W \in R^D$ for the feature space to make the content similarity C_{ij}^W somehow similar to the semantic link similarity S_{ij}. We optimize the following objective function to find W,

$$L(W,c,g) = \sum_{i=1}^{N} \sum_{j=1}^{K} L_{ij} = \sum_{i=1}^{N} \sum_{j=1}^{K} (C_{ij}^W - (cS_{ij} + g))^2 \Phi_{ij} \tag{8}$$

where N is the number of images and K is the number of candidates. S_{ij} is computed from Equation 2. C_{ij}^W and S_{ij} may have different scale and shift, so we introduce parameters c and g to automatically estimate them.

Φ_{ij} is the confidence or importance of S_{ij},

$$\Phi_{ij} \equiv Conf(S_{ij}) = 1 - e^{-\alpha T} \tag{9}$$

where T is the minimum number of tags for image i or j. The idea is that if the tags of an image are incomplete (0 or very few) and thus cannot fully describe its semantic meaning, the link-based similarity becomes less reliable.

To find (W^*, c^*, g^*) that minimizes the objective function, we first compute the first-order partial derivatives,

$$\frac{\partial L}{\partial w^d} = \sum_{i=1}^{N} \sum_{j=1}^{K} 2\Phi_{ij}(C_{ij}^W - (cS_{ij} + g))(-x_{ij}^d) \tag{10}$$

$$\frac{\partial L}{\partial c} = \sum_{i=1}^{N} \sum_{j=1}^{K} 2\Phi_{ij}(C_{ij}^W - (cS_{ij} + g))(-S_{ij}) \tag{11}$$

$$\frac{\partial L}{\partial g} = \sum_{i=1}^{N} \sum_{j=1}^{K} 2\Phi_{ij}(C_{ij}^W - (cS_{ij} + g))(-1) \tag{12}$$

The variables are estimated by Gradient Decent iteratively,

$$w_{m+1}^d = w_m^d - \gamma_w \frac{\partial L}{\partial w^d}\bigg|_{w^d = w_m^d} \quad (d = 1, ..., D) \tag{13}$$

$$c_{m+1} = c_m - \gamma_c \frac{\partial L}{\partial c}\bigg|_{c=c_m} \tag{14}$$

$$g_{m+1} = g_m - \gamma_g \frac{\partial L}{\partial g}\bigg|_{g=g_m} \tag{15}$$

We initialize the variables as 1.

After feature learning, based on the new feature weighting, update the image similarity as a combination of content-based and link-based similarity.

$$S(i,j) = (1 - \Phi_{ij}) C_{ij}^{W^*} + \Phi_{ij} S_{ij} \tag{16}$$

Note that we could learn a local feature weighting to each image to improve the performance.

3.2.4 SimLearn

Given the newly learned image similarity, update the group and tag similarity. The image similarity is updated based on the new group and tag similarity. The process iterates until it converges or stop criteria satisfied. We call this approach *SimLearn*. Algorithm 1 describes the procedure of SimLearn.

Algorithm 1 SimLearn

1. Initialization;
2. Iterate {
3. For images, $S_{m+1}(i,j) = \alpha_I S_m^G(i,j) + \beta_I S_m^T(i,j)$;
4. Feature learning to update $W = W_{m+1}^*$;
5. Update $S_{m+1}(i,j) = (1 - \Phi_{ij}) C_{ij}^W + \Phi_{ij} S_{m+1}(i,j)$;
6. For groups, $S_{m+1}(g,g') = \alpha_G S_m^I(g,g') + \beta_G S_m^T(g,g')$;
7. For tags, $S_{m+1}(t,t') = \alpha_T S_m^I(t,t') + \beta_T S_m^G(t,t')$;
8. } until converge or stop criteria satisfied.

4. EXPERIMENTS

We build our demo system and perform our experiments based on two datasets **Flickr** and **Amazon**, downloaded from the company API's. We treat the product categories as groups. There are over 14,000 images with 300,000 links for the Flickr data, and over 110,000 images with 1,307,000 links for the Amazon data.

4.1 Time Efficiency

Figure 4 shows the time efficiency of SimRank, *K*-SimRank and MoK-SimRank. SimRank is the slowest, and because it is so slow that we only perform experiments with SimRank for up to 4000 images. *K*-SimRank is significantly faster than SimRank, MoK-SimRank is the fastest.

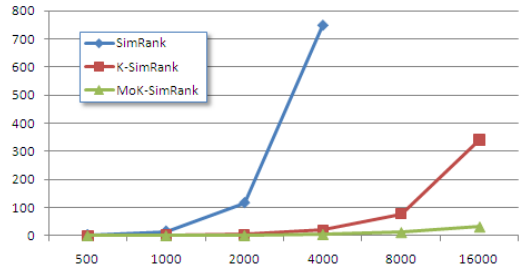

Figure 4: Time performance. X-axis denotes the number of images, Y-axis denotes the running time (in seconds). ($K = 60$)

4.2 SimLearn Result Examples

Our experiments have shown good performance of SimLearn to find similar images, tags and groups. The details are beyond the scope of this demo paper. Figure 5 shows the top 10 most similar products for the query of "thinkpad", using SimRank (1st row), content similarity (2nd row), and RankLearn (3rd row). SimLearn obtains the most relevant matches in both semantics and visual appearances. It correctly finds all the similar groups, for example, the groups similar to group "FLOWERS" are "Flower Photography", "FLOWER-POWER" and "Flower Pictures (NO LIMITS)".

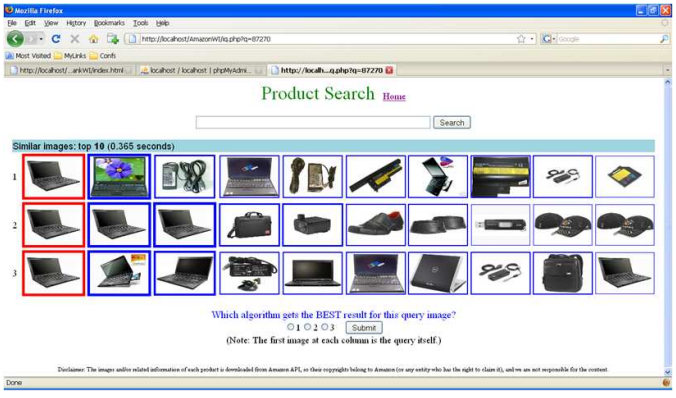

Figure 5: Top 10 results of content similarity, SimRank and SimLearn.

5. INTERFACE AND DEMONSTRATION

Figures 6 and 7 show our demo system for retrieval on Flickr images and Amazon products, respectively. Given a user query, the system returns a list of relevant images/products based on matching the query keywords with the tags according to the tag similarity computed by SimLearn. User can click on an image to find similar images according to the scores also computed by SimLearn.

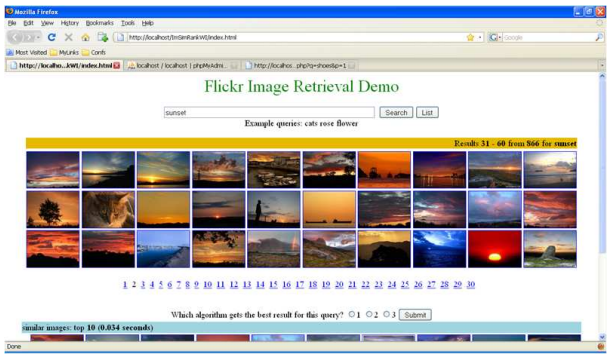

Figure 6: The demo system for Flickr image retrieval.

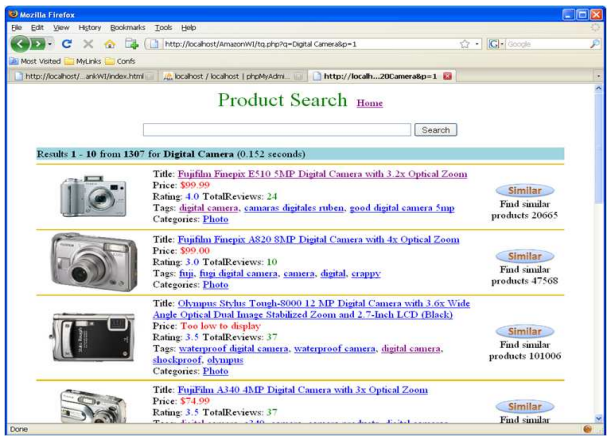

Figure 7: The demo system for Amazon product search.

6. REFERENCES

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

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

[3] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the 8th International Conference on Knowledge discovery and data mining(KDD'02)*, 2002.

[4] L. Yang and A. R. Jin. Contents distance metric learning: A comprehensive survey. Technical report, Department of Computer Science and Engineering, Michigan State University, 2006.