Personalized Video Recommendation Through Tripartite Graph Propagation*

Bisheng Chen †, Jingdong Wang ‡, Qinghua Huang †, Tao Mei ‡
† South China University of Technology, Guangzhou, P. R. China
‡ Microsoft Research Asia, Beijing, P. R. China
bison.chan@foxmail.com; {jingdw, tmei}@microsoft.com; qhhuang@scut.edu.cn

ABSTRACT

The rapid growth of the number of videos on the Internet provides enormous potential for users to find content of interest to them. Video search, such as Google, Youtube, Bing, is a popular way to help users to find desired videos. However, it is still very challenging to discover new video contents for users. In this paper, we address the problem of providing personalized video suggestions for users. Rather than only exploring the user-video graph that is formulated using the click-through information, we also investigate other two useful graphs, the user-query graph indicating if a user ever issues a query, and the query-video graph indicating if a video appears in the search result of a query. The two graphs act as a bridge to connect users and videos, and have a large potential to improve the recommendation as the queries issued by a user essentially imply his interest. As a result, we reach a tripartite graph over (user, video, query). We develop an iterative propagation scheme over the tripartite graph to compute the preference information of each user. Experimental results on a dataset of $2,893$ users, $23,630$ queries and $55,114$ videos collected during Feb. 1-28, 2011 demonstrate that the proposed method outperforms existing state-of-the-art approaches, co-views [3] and random walks on the user-video bipartite graph [2].

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*video*; H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web-based services*

General Terms

Algorithms, Human Factors, Experimentation.

Keywords

Personalized video recommendation, tripartite graph.

* This work was performed at Microsoft Research Asia.

MM'12, October 29–November 2, 2012, Nara, Japan.

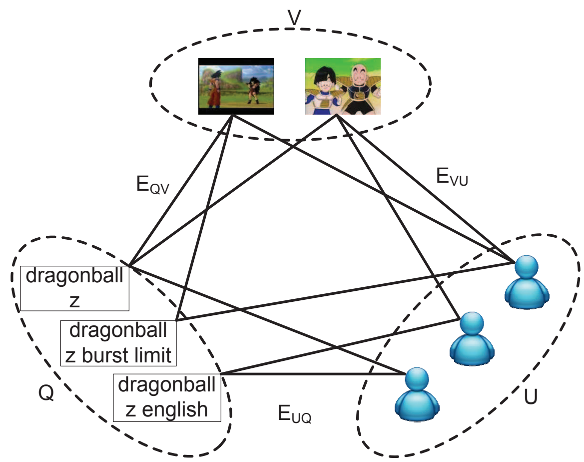

Figure 1: The User-Query-Video tripartite graph.

1. INTRODUCTION

Recently, the videos on the Internet provides have been grown rapidly, and there are a lot of online videos that users can access. Taking Youtube as an example, there are about 569 millions videos, and the number is even still growing rapidly [9]. Video search engines, such as Google, Youtube, Bing, help users to find videos of interest to them.

Video search works satisfactory if a user formulates a good textual query. However, in practice it is not always the case as it is uneasy to describe the search intent using a few textual words. Moveover, it is very useful to discover new contents for users. Therefore, complementary to video search, recommending videos for users, called personalized video recommendation, is another widely-used way, according to some information including the preferences [4, 8] and the click-through information [2, 3].

State-of-the-art personalized video recommendation techniques [2, 3] exploits the click-through information, i.e., recording what videos are clicked/viewed by some user, builds a bipartite graph to connect users and videos, and propagates the preference information over the bipartite graph to suggest videos to users. The key idea behind these techniques is that the videos viewed by a user can imply and hence be used to represent the interest of the user.

Besides the click-through information, there is another kind of important information, what queries have been issued by a user. It is natural that these queries also describe the interest of the users. We propose to explore such information to suggest videos to users and build a user-query graph to record the information. To connect users and videos through queries, we also build a query-video graph to indicate if a video appears in the search result of a query.

In this paper, we build an integrated tripartite graph shown in Figure 1 (user, video), (video, query) and (query, user) by combining the three graphs together. Here, the subgraphs (video, query) and (query, user) provide a new path from users to videos and essentially suggest another way to compute the preference of a user to videos. We propose an iterative message propagation scheme to update the preference by using the three relations. The propagation iteratively updates the preferences of users to videos according to the (query, user) graph and the association of queries with videos, the relations among videos according to the (user, video) graph and the preferences of users to videos, and the association of queries with videos according to the (video, query) graph and the the relations among videos. Finally, the preferences of users to videos are adopted to recommend videos for users.

The remainder of this paper is organized as follows. Section 2 provides a brief review of related work. Section 3 presents the proposed personalized video recommendation algorithm. Section 4 shows experiments and Section 5 concludes the paper.

2. RELATED WORK

Recommendations of non-text contents inspire the study in video domain [1]. The Netflix price challenge, aiming to use previous ratings given by users to improve the DVDs recommendation, attracts extensive research attention [5]. The task is to mine the users' patterns from their past behaviors (ratings). The use of behaviors information opens a door for recommendations [7]. We classify the previous research on video recommendation according to the use of information.

The research related to our work are the ones that only used the click-through information. Davidson *et al.* presented the recommendation system for Youtube based on association rule mined from click-through information [3]. Baluja *et al.* modeled the users' view history using a user-video bipartite graph and studied the users' viewing patterns [2]. They used a label propagation procedure to create a video recommendation system that did not rely on the analysis of video contents. Compared to these works, query and click-through information can give us more abundant associations between users and videos.

There are some works based on not only click-through information but also video contents. Mei *et al.* presented an online video recommendation system using multimodal relevance and click-through information [4, 8]. They expressed the multimodal relevance between two video documents as the combination of textual, visual, and aural relevance and used the attention fusion function to combine the relevance scores. The weights during fusion procedure were adjusted with users' feedback. This system works well. However, the computation of video contents analysis is expensive.

Other studies include the used of metadata like social relationship and tags. Zhao *et al.* presented a recommendation system based on the relationship strength between users in different domain of social network [10]. Park *et al.* presented an online video recommendation system based on tag-cloud aggregation in which the profile of a user's interests were represented by the global tag cloud of the user's previous watched videos [6].

3. APPROACH

The goal is to proposed a personalized video recommendation system that can handle the enormous number of online videos efficiently and effectively. The analysis of video contents may not be a good idea since it will cost expensive computations. Mining click-through information has been proved to be practicable [2, 3]. However, for a more effective recommendation system, the query information, which can enrich the connections between users and videos, should be taken into consider. It can be helpful for generating new paths between users and videos by using queries as intermediaries.

Algorithm 1: Tripartite Propagation

Input: $U,\ Q,\ V,\ E_{UQ},\ E_{QV},\ E_{VU},\ T,\ \alpha$
Initialize $V = I$
for $t \leftarrow 1$ ***to*** T **do**
  $Q = E_{QV} \cdot V$
  normalize Q
  $U = E_{UQ} \cdot Q$
  normalize U
  $V = E_{VU} \cdot U$
  normalize V
  modify $V = (1-\alpha)V + \alpha I$
end
Output: U

3.1 Tripartite Graph

The query and click behaviours of users on search engine form a principal data-sources for recommendation. These behaviours can be modeled as a tripartite graph. A tripartite graph is a graph with its nodes decomposed into three disjoint sets. The nodes in the same set are not adjacent to each other. As shown in Figure 1, the query and click behaviours are modeled into User-Query-Video tripartite graph. The nodes are decomposed into user U, query Q and video V sets according to their roles. The behaviours connect nodes between different sets. For example, a user u issues a query q on search engine. This action generates a edge E_{uq} between user u and query q. Then he/she clicks a video v. This will generate two edges, the first one E_{vu} connects video v and user u, and the second one E_{qv} connects query q and video v.

3.2 Tripartite Propagation

The basic idea of the proposed iterative message propagation scheme, Tripartite Propagation, is to recommend a list of videos that are reachable from a user node on the User-Query-Video tripartite graph. Inspired by [2] and telecommunication network, the Tripartite Propagation algorithm can be viewed as a process of messages travelling through network. The node in telecommunication network is capable of sending, receiving and forwarding messages. In our study, video nodes in tripartite graph act as the message sources, they send messages representing themselves along the edges to neighbors continually. At the same time, user nodes, query nodes as well as video nodes receive, maintain and forward the messages sent to them. After a few iterations the messages maintained by user nodes are treated as recommended videos to users.

Algorithm 1 gives a formal description of Tripartite Propagation. Mathematically, we use a row vector to denote the messages maintained by each node ($[m_1,\ m_2,\ m_3, ...]$). Each element of the row vector denotes a distinct message. In our setting, each video node v sends a message m_v representing itself. Therefore, the number of distinct messages is equal to the number of video nodes. The dimension of the row vector is also equal to the number of video nodes. The value of each element represents the weight of the corresponding message. The matrix U represents the message vectors maintained by all the user nodes. Each row of matrix U represents the message vector maintained by the corresponding user node. So, the row number of matrix U is equal to the number of user

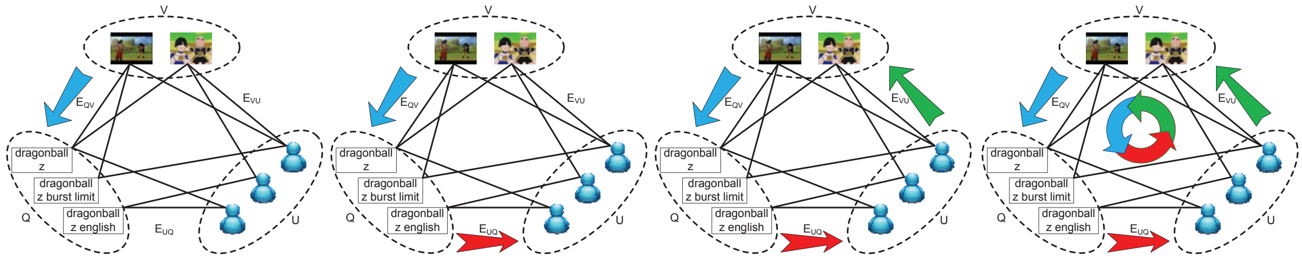

Figure 2: Propagation procedure.

Table 1: example of query log

User ID	67FB0B1FD521400A
Timestamp	2011-02-09 11:09:52.000
Textual query	obama
Clicked video	oPfh6oOV4wyoLw

nodes and the column number of matrix U is equal to the number of distinct messages. The matrix Q and the matrix V represent the message vectors maintained by query nodes and video nodes respectively. They have the similar interpretations with matrix U. The matrix E_{UQ} represents the edges between the set of user nodes and the set of query nodes. The rows of matrix E_{UQ} correspond to the user nodes and the columns of matrix E_{UQ} correspond to the query nodes. Each element in matrix E_{UQ} has a value equal to 1 or 0. An element with value 1 means there is an edge between the corresponding user node and query node, indicating the user issued the query. An element value 0 means there is no edge between the corresponding user node and query node. The matrix E_{QV} denotes the edges between query nodes and video nodes, and the matrix E_{VU} denotes the edges between video nodes and user nodes. They have the similar interpretations with matrix E_{UQ}. I is an identity matrix with the size equal to the number of video nodes. T denotes the iteration number and α denotes a tradeoff parameter. The sensitivity of T and α will be analyzed in experiment section.

Here are some explanations for the algorithm. The initialization step sets each message vector maintained by video nodes to contain only one message ($V = I$). It's a pre-process step that message sources send messages to themselves. Then the messages propagate along the edges between nodes iteratively. Figure 2 shows the propagation procedure. The iteration number T controls how far away messages can spread to. The normalization step in the loop normalizes the weight distribution of message vector maintained by each node to keep a summation equal to 1. The modification step modifies the weight distribution of message vectors of video nodes by setting $V = (1-\alpha)V + \alpha I$. The tradeoff parameter α controls the behaviour of a video node on how to treat the received messages and the message representing itself. This step simulates sending messages continually. Finally, the algorithm outputs the message vectors maintained by user nodes U. Each message vector corresponds to a list of videos and the list of videos are recommended videos for the corresponding user. The order of videos is based on the weight distribution of the corresponding message vector.

4. EXPERIMENT

4.1 Dataset

We conducted experiments on query log data collected from a commercial video search engine on US market during Feb. 1-28, 2011. The query log data contains user ID (fully anonymous),

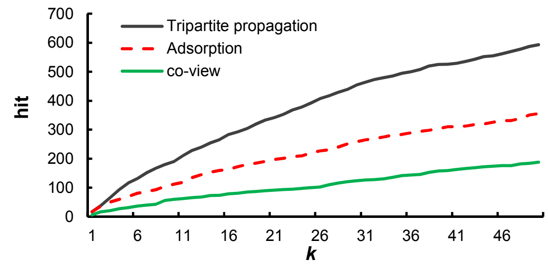

Figure 3: Hit analysis for all the algorithms.

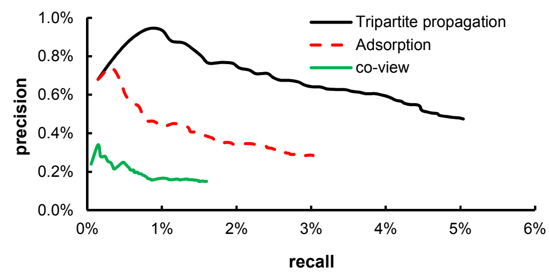

Figure 4: Precision and Recall for all the algorithms.

query timestamp, textual query and clicked video. Table 1 show an example of query log data. About 1.78 million users issued 3.1 million queries and clicked on 1.54 million distinct videos. The raw data was partitioned into training and testing sets according to timestamp. The training set contained data from the first 15 days and the testing set contained data from the remaining. The training set was for generating recommendation lists using our method and other recommendation algorithms. The testing set was used for measuring the effectiveness of these algorithms. Follow the idea of [2], a recommendation of video v to user u was considered successful if user u didn't click video v during the training period but did click video v during testing period.

From the definition of successful recommendation above, we can infer that unless a user u performed some clicks during both training and testing period, otherwise we cannot make recommendation or evaluation for user u. For convenient of evaluation, we sampled the data, restricting the users to have a click number between 10 and 50 during both training and testing periods. Most of the users clicked less than 10 videos during both periods. Therefore, this restriction reduced the user number to a small scale. Finally, we evaluated the recommendation algorithms on a sampled training set with $2,893$ users, $23,630$ queries and $55,114$ videos.

4.2 Evaluations

We compared our method with co-view [3] and Adsorption [2] that only used the click-through information. Each algorithm generated a list of recommended videos for each user. A pre-process

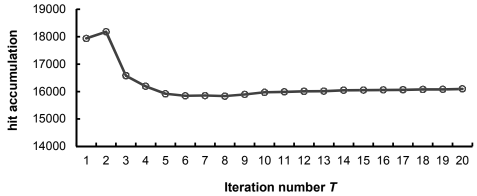

Figure 5: The Hit adcumulation with respect to the iteration number T.

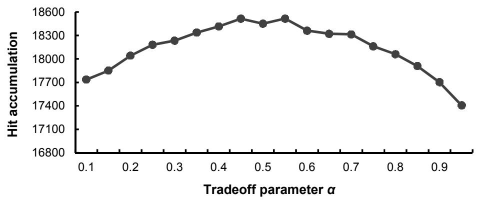

Figure 6: The Hit accumulation with respect to the tradeoff parameter α.

filtering out the videos clicked by users during the training period was performed on all the recommendation lists before evaluations, ensuring that all the recommendations were new for users. Since the clicked number of each user during testing period was limited to 50 in our setting, We took the first 50 videos of all the recommendation lists for comparison. The clicked numbers of users' differ a lot. It's difficult and maybe meaningless to make comparison on user level. So, we treated all the users as a whole and evaluated the algorithms performances on it.

4.2.1 Hit Analysis

In our evaluation, a successful recommendation was called a hit. We counted the hit number of the first k recommendations (k varied from 1 to 50) of each algorithm on the whole testing set. For a user u, the hit number of first k recommendations was the number of videos in the intersection of these k recommendations and the clicked videos of user u during testing period. For an algorithm, the hit number of the first k recommendations was the sum of the hit number of all the users. Figure 3 shows the result of hit analysis for all the algorithms. As we can see, our proposed Tripartite Propagation outperforms all the other algorithms at diferent k. Video recommendation using query information and click-through information can better capture the user behaviours than recommendations using only click-through information.

4.2.2 Precision and Recall

Figure 4 shows the precision and recall curve which is commonly used in information retrieval domain. It demonstrates the superiority of our method in a different way. As mentioned on [2], this evaluation is very conservative. Therefore, the precision values are quite small overall. Precision and Recall borrowed from information retrieval cannot fully reveal the benefits of the methods. Nonetheless, they provide a useful way to compare different algorithms.

4.2.3 Parameter Sensitivity

Figure 5 shows the Hit accumulation with respect to the iteration number T. For each iteration, the Hit accumulation is obtained by summing up the hit number over all the k ($Hit_T = \sum_{k=1}^{50} Hit_k$). Tripartite Propagation achieves the best performance with $T = 2$ and enters the steady-state with the growing of T. Since T controls how far away a message can spread to, this result shows that messages sent from neighbourhood of user nodes better capture users' behaviours. Figure 6 shows the Hit accumulation with respect to the tradeoff parameter α. The best performance is achieved when α has a value near 0.5.

Figure 7: Recommendations for one user.

4.2.4 Example

Figure 7 shows an example of recommended videos for one user. The first line of table shows the user's clicked videos during testing period. The remaining lines show recommended videos generated by all the algorithms. The thumbnails with red background color represent the successful recommendations.

5. CONCLUSIONS

We have presented our personalized video recommendation approach leveraging a tripartite graph, which is generated from users' queries and clicks behaviours, to propagate messages representing videos to user nodes. Experiment results demonstrate that query information is helpful for improving the recommendation. Future work includes determining the parameters automatically. We are also interested in exploring other applications of Tripartite Propagation, such as query suggestion, music suggestion, personalized advertising.

6. REFERENCES

[1] G. Adomavicius, E. Tuzhilin, and E. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. pages 734–749, 2005.

[2] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, M. Aly, and M. Aly. Video suggestion and discovery for youtube: taking random walks through the view graph. In *WWW*, pages 895–904, 2008.

[3] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. V. Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, D. Sampath, and D. Sampath. The youtube video recommendation system. In *RecSys*, pages 293–296, 2010.

[4] T. Mei, B. Yang, X.-S. Hua, and S. Li. Contextual video recommendation by multimodal relevance and user feedback. *ACM Trans on Information Systems*, 29(2):10, 2011.

[5] Netflix. `http://www.netflixprize.com/`.

[6] J. Park, S. Lee, S. J. Lee, K. Kim, B.-S. Chung, Y. K. Lee, and Y. K. Lee. Online video recommendation through tag-cloud aggregation. In *IEEE MultiMedia*, pages 78–87, 2011.

[7] J. Wang, Z. Zhao, J. Zhou, H. Wang, B. Cui, and G. Qi. Recommending flickr groups with social topic model. *Inf. Retr.*, 15(3-4):278–295, 2012.

[8] B. Yang, T. Mei, X.-S. Hua, L. Yang, S.-Q. Yang, M. Li, and M. Li. Online video recommendation based on multimodal fusion and relevance feedback. In *CIVR*, pages 73–80, 2007.

[9] Youtube. `http://www.youtube.com/results?search_query=*&search=search`.

[10] X. Zhao, J. Yuan, R. Hong, M. Wang, Z. Li, T.-S. Chua, and T.-S. Chua. On video recommendation over social network. In *MMM*, pages 149–160, 2012.