Multimedia Systems
DOI 10.1007/s00530-012-0267-z

REGULAR PAPER

Video recommendation over multiple information sources

Xiaojian Zhao · Jin Yuan · Meng Wang · Guangda Li · Richang Hong · Zhoujun Li · Tat-Seng Chua

Abstract Video recommendation is an important tool to help people access interesting videos. In this paper, we propose a universal scheme to integrate rich information for personalized video recommendation. Our approach regards video recommendation as a ranking task. First, it generates multiple ranking lists by exploring different information sources. In particular, one novel source user's relationship strength is inferred through the online social network and applied to recommend videos. Second, based on multiple ranking lists, a multi-task rank aggregation approach is proposed to integrate these ranking lists to generate a final result for video recommendation. It is shown that our scheme is flexible that can easily incorporate other methods by adding their generated ranking lists into our multi-task rank aggregation approach. We conduct experiments on a large dataset with 76 users and more than 11,000 videos. The experimental results demonstrate the feasibility and effectiveness of our approach.

Keywords Video recommendation · Rich information · Online social network · Multi-task rank aggregation

X. Zhao · Z. Li
State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China
e-mail: zhaoxj01@gmail.com

Z. Li
e-mail: lizj@buaa.edu.cn

J. Yuan · G. Li · T.-S. Chua
School of Computing, National University of Singapore, Singapore 117590, Singapore
e-mail: yuanjin@comp.nus.edu.sg

G. Li
e-mail: guangda10@gmail.com

T.-S. Chua
e-mail: chuats@comp.nus.edu.sg

M. Wang (✉) · R. Hong
School of Computer and Information, Hefei University of Technology, Hefei 230009, Anhui, China
e-mail: eric.mengwang@gmail.com

R. Hong
e-mail: hongrc.hfut@gmail.com

1 Introduction

With the rapid advances in storage devices, networks, and compression techniques, video data from different domains are growing at an explosive rate. For example, the most popular video sharing site, YouTube,[1] hosts over 150,000,000 videos [10]. Bewildered by the vast quantity of videos, users often wish to find some interesting video contents. Therefore, automatic video recommendation becomes highly desired to tackle this information-overload problem by presenting a list of potentially interesting videos to the users.

Several efforts have been dedicated to video recommendation by exploring different information sources. For example, the recommendation services on commercial video websites, such as YouTube, Yahoo! Video,[2] and Bing Videos,[3] are usually built based on the textual information of videos or users' profiles (only for registered users) [31]. Generally, the typical video recommendation approaches either analyze the interest relationship of users by mining the viewing history [3, 8, 19, 20, 32] or utilize

[1] http://www.youtube.com.

[2] http://video.yahoo.com.

[3] http://www.bing.com/videos/browse.

Published online: 10 June 2012

the users' profile [28]. There lacks a unified scheme that is able to integrate multiple information sources for video recommendation.

In this paper, we propose a personalized video recommendation scheme by integrating rich information. Figure 1 demonstrates the schematic illustration of the approach. In our approach, we regard video recommendation as a ranking task, i.e., to rank interesting videos as high as possible for a specific user. The approach first independently explores a variety of information sources, such as users' profile, user' viewing history and the video title, tags, etc., to generate multiple ranking lists of recommended videos. Then a multi-task rank aggregation approach is proposed to fuse these ranking lists to generate the final recommended videos. In particular, we propose to recommend videos based on user's social relationship, where the relationship strengths between users are inferred through online social network. We summarize the main contributions of this paper as follows:

1. To the best of our knowledge, this is the first work that integrates rich information for video recommendation. We not only provide a solution but also analyze the effectiveness of different information sources for video recommendation.
2. It is the first work that comprehensively investigates social network in video recommendation. We utilize not only the relationships between users but also their strengths in different activity domains.
3. We propose a multi-task rank aggregation algorithm that is able to accomplish user-adaptive rank aggregation.

Part of the approach has been published in our previous papers [48, 49]. Compared with the works in [48, 49], we have three enhancements: (1) we have added the investigation of social network with relationship strength analysis; (2) we have added popularity-based ranking; and (3) more discussion and analyzes are provided.

The rest of this paper is organized as follows: In Sect. 2, we briefly review the related work on video recommendation. Section 3 introduces the generation of ranking lists with multiple information sources. In Sect. 4, we propose the multi-task rank algorithm. Experiments are provided in Sect. 5. Finally, we conclude the paper in Sect. 6.

2 Related work

Research on video recommendation mainly focuses on three typical approaches, namely, collaborative filtering (CF), content-based filtering (CBF), and hybrid filtering (HF) that combines the aforementioned two approaches in a single framework [1].

The CF approaches compare a user's ratings of videos with those of hundreds of others, find people who share similar preferences, and then recommend videos that are interesting for those people with similar preferences [34]. For example, Setten et al. [37] proposed to use different social filtering methods to predict user interest based on other users' information and designed a combination of prediction techniques. Hill et al. [17] designed an email interface to collect data on a virtual community functions (i.e. person, item, rating, correlation among persons) from Oct. 1993 to May 1994. Then, given a user, the system recommends unseen videos rated by a group of similar persons. Baluja et al. [3] built a user-video graph which represents the co-view information among different users and its recommendation is performed by a graph propagation in which the label of each note is obtained from its neighbors.

CF is probably the most successful technique in the design of video recommendation systems [16]. But the technology has several well-known limitations. The performance of CF is strongly limited by the sparsity of data as a result of the following: (1) the huge number of videos far beyond user's ability to evaluate even a small fraction of them; and (2) users do not have the initiative to rate the viewed videos [35]. Meanwhile, the problems of recommending videos to a new user (called new user problem) as well as recommending the videos that have not been rated (called new item problem) cannot be ignored also.

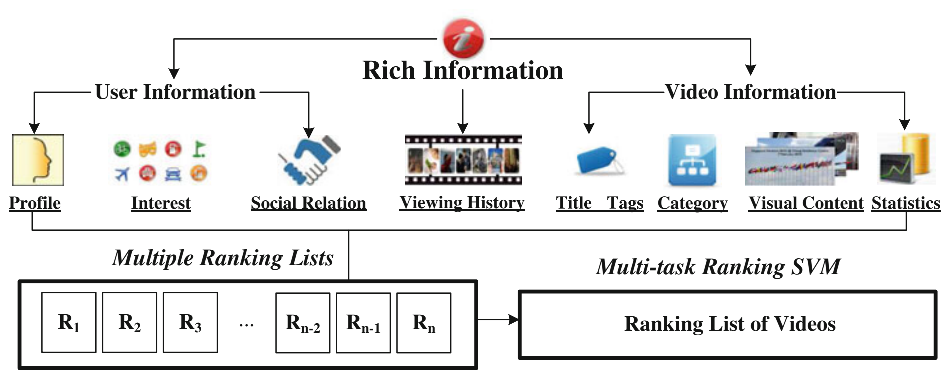

Fig. 1 The schematic illustration of the proposed video recommendation scheme that explores multiple information sources

To overcome the shortage of the CF approach, CBF was proposed to recommend videos based on user's viewing history. For example, Mei et al. [30, 32] presented a novel contextual video recommendation system called *VideoReach* based on multimodal content relevance and user feedback. An online video was represented by different modalities (i.e., visual and audio track, as well as text words). The recommended videos are relevant to current viewing in terms of multimodal relevance. Irie et al. [24] introduced degree-of-edit (DoE) ranking to estimate how much a consumer-generated video (CGV) is edited as a ranking measure for CGV recommendation. Although the CBF approaches are quite effective for uniquely characterizing each user with rich information and able to avoid the new item problem, they neglect the fact that different users may share similar interest.

Several HF approaches have been proposed to address the above limitations by integrating the CF and CBF [1]. For example, Burke [4] employed a mixture model which conducts the recommendation based on a linear combination from the content-based prediction and collaborative prediction. In [33], Öztürk and Kesim Cicekli proposed a hybrid video recommendation systems based on a graph algorithm called Adsorption, which is a collaborative filtering algorithm to make recommendation based on users' relationship. Although these models have significant advantages over the early recommendation approaches, the performance is unstable on different problems, and it is also difficult to choose a right hybridization strategy for the problem at hand.

Besides the approaches mentioned earlier, it is widely accepted that social network is a useful tool to recommend items. For example, Jebrin and Williams [26] proposed a new approach to make recommendations based on leaders' credibility in "Follow the Leader" model as Top-N recommender by incorporating social network information into user-based collaborative filtering. In [34], the proposed video recommender can construct a per-user profile as an aggregation of tag clouds of videos viewed by the user and then suggest videos based on the viewing patterns of similar users who were identified according to a similarity function over the users' profiles in online social network. Aiming at modeling recommender systems more accurately and realistically, Ma et al. [29] proposed a novel probabilistic factor analysis framework over social network, which naturally fused the users' tastes and their trusted friends' favors together. In this framework, they coined the term "Social Trust Ensemble" to represent the formulation of the social trust restrictions on the recommender systems.

Different from the above efforts, the approach proposed in this work is a flexible scheme and can easily integrate different methods and information sources. In fact, we can generate multiple ranking lists based on different methods and information sources and then employ the multi-task ranking approach to integrate all the ranking lists.

3 Ranking list generation with rich information

As mentioned earlier, our proposed approach integrates multiple ranking lists with a multi-task method. In this section, we introduce the ranking lists generated by exploring different information [13, 14]. Here we organize them according to the involved user-related information sources, including profile, viewing history, social network, and user-contributed data. In addition, we also generate ranking lists based on collaborative filtering, the most widely applied recommendation approach.

3.1 Profile-based ranking

We download each user's profile information from the Facebook[4] website, including education background, occupation, philosophy, location, interests, etc. An example of the user's profile information is shown in Table 1. Since we can collect the text information of each video, we can generate 3 ranking lists based on user's interests, location, and other information, respectively. Here, the textual information of video includes video's title, description, tags and category, etc. We represent each video v_o as a set of words $\mathcal{W}_o$ and the profile information of each user u_k as a set of words $\mathcal{W}_{u_k}$. The textual similarity between video v_o and user u_k is calculated as:

$$S_p(v_o,u_k)=\frac{1}{|\mathcal{W}_o||\mathcal{W}_{u_k}|}\sum_{w_o\in\mathcal{W}_o,w_{u_k}\in\mathcal{W}_{u_k}}\exp\left(-\frac{\mathrm{NGD}(\mathrm{w_o},\mathrm{w_{u_k}})}{\sigma}\right) \quad (1)$$

where σ is a scaling parameter, and $\mathrm{NGD}(w_o, w_{u_k})$ is the normalized Google distance [7] between the word w_o and w_{u_k}. Google distance is a semantic measure derived from the number of hits returned by the Google search engine for a given set of keywords [7]. Specifically, the normalized Google distance between two search words w_o and word w_{u_k} is defined as

$$\mathrm{NGD}(w_o,w_{u_k})=\frac{\max\{\log f(w_o),\log f(w_{u_k})\}-\log f(w_o,w_{u_k})}{\log M-\min\{\log f(w_o),\log f(w_{u_k})\}} \quad (2)$$

where M is the total number of web pages searched by Google, $f(w_o)$ and $f(w_{u\_k})$ are the numbers of web pages by searching the word w_o and w_{u_k}, respectively, and $f(w_o, w_{u_k})$ is the number of web pages that both w_o and w_{u_k} appear.

[4] http://www.facebook.com.

Table 1 The profile information of an exemplary user

Profile item	Information
Basic Information	
Current city	Singapore, Singapore
Hometown	Tangshan, Hebei, China
Gender	Male
Languages	English, Chinese
Education and work	
High school	CheZhouShan High School ’98
College/University	Beihang University ’08
Employer	CECT
Philosophy	
Religion	Other
Political views	Communist Party of China
Arts and entertainment	
Music	Ren’e Liu, Michael Jackson, John Denver
Books	Rich Dad Poor Dad, Harry Potter, Bible
Movies	Avatar, The Bounty Hunter, Death At A Funeral
Television	CNN, BBC World News, Doug
Activities and interests	
Activities	Jogging, cooking
Interests	Programming, traveling, singing
Sports	Basketball, tennis, swimming, football, cricket, badminton

3.2 History-based ranking

The viewing history information is the list of videos that have been accessed by a user in the past. It reflects the video interests by user; thus we recommend videos according to user’s video viewing history. Here we consider three types of viewing history: recent history (i.e., the most recent video viewed by the user), short-term history (the videos viewed on that day), and long-term history (all the past history). For each type of viewing history, we generate 2 ranking lists based on text-based video similarity and visual content-based video similarity, respectively. As a result, we generate 6 ranking lists based on the user’s history information.

Text-based video similarity As described in Sect. 3.1, the text information of each video includes title, description, tags and category, etc. Here we adopt Bag of Words (BoW) model to represent the text information [23, 42, 45]. The text-based video similarity between v_h and v_o is calculated as follows:

$$S_t(v_h, v_o) = \frac{1}{|\mathcal{W}_h||\mathcal{W}_o|}\sum_{s=1}^{m}\sum_{t=1}^{n}\exp\left(-\frac{\mathrm{NGD}(\mathrm{w_{hs}}, \mathrm{w_{ot}})}{\sigma}\right) \quad (3)$$

where σ is a scaling parameter, and $\mathcal{W}_h = [w_{h1}, w_{h2}, \ldots, w_{hm}]$ is the BoW representation of the text information for h-th video, and the $\mathcal{W}_o = [w_{o1}, w_{o2}, \ldots, w_{on}]$ is the BoW representation of the text information for o-th video, $\mathrm{NGD}(w_{hs}, w_{ot})$ is the normalized Google distance [7] between the words w_{hs} and w_{ot}.

Content-based video similarity To estimate the visual similarity of two videos, we first segment each video into shots and extract a representative key-frame from each shot. We then extract 428-dimensional global visual features for each key frame [18, 47], including 225-D block-wise color moments generated from a 5-by-5 fixed partition of the image [46], 128-D wavelet texture [45], and 75-D edge direction histogram [38, 40–42]. Then the visual similarity between two videos is calculated by averaging the similarities of all key-frame pairs across the two videos [39]:

$$S_v(v_h, v_o) = \frac{1}{|v_h||v_o|}\sum_{\mathbf{x}_i \in v_h, \mathbf{x}_j \in v_o}(1 - \cos(\mathbf{x}_i, \mathbf{x}_j)) \quad (4)$$

where $\mathbf{x}_i$, $\mathbf{x}_j$ are key frames in v_h and v_o, respectively, $|v_h|$, $|v_o|$ represent the key-frame numbers contained in the corresponding videos, and $\cos(\mathbf{x}_i, \mathbf{x}_j)$ is the cosine distance between these two key-frames [41].

3.3 Social network-based ranking

Many recommendation algorithms, such as collaborative filtering, are built by mining the interest relationship of a large number of users [1, 8, 22]. But in fact, a user usually has closer interests with his/her friends than unknown people. For example, when a user wants to choose a movie, it is natural for him/her to turn to friends for suggestions.

Moreover, friends may share interests in different domains such as sport, diet, shopping, etc. This domain information is quite useful for video recommendation. For example, when a user is interested in sport videos, his/her friends with the same interests in the domain "*sport*" are more useful to recommend videos than the other friends. Currently, with the rapid advances of social network, it is possible to collect information on a user's friends as well as explore their video viewing histories. In this work, we investigate to explore social network for video recommendation.

Our approach is as follows: given a user, we denote $\mathcal{V}_i$ as the video set that has been viewed by his/her i-th friend. We integrate all the videos viewed by the friends into a video set $\mathcal{V}$, i.e., $\mathcal{V} = \mathcal{V}_1 \bigcup \mathcal{V}_2 \bigcup \cdots \bigcup \mathcal{V}_r$, where r is the number of the friends. We then generate ranking lists with two methods. For the first method, we set the ranking score 1 to videos in $\mathcal{V}$ and the score 0 to all the other videos. Since we can set $\mathcal{V}_i$ to either the recent one, short-term history, or long-term history of the i-th friend, we can obtain 3 ranking lists with this method. Note that these ranking lists are actually not stable as there are only two scores, 0 and 1, for the videos. But it is not a problem for generating the final recommendation list since our rank aggregation, which will be introduced in the next section, is a score-based fusion approach. However, this method neglects the relationship strength in online social network. For the second method, we still put the videos in $\mathcal{V}$ on top of the ranking list, but the score is determined by the relationship strength in online social network between the user and his/her friends in the specific activity domain which the video v belongs to.

Given the activity domain set $\mathcal{A} = \{A_1, A_2, \ldots, A_L\}$, we assign each video v_h to one of the activity domains A_l before estimating the relationship strength. Based on the observation on our video dataset, we select the top six popular activity domains "*diet*", "*entertainment*", "*shopping*", "*sports*", "*work*", "*tourism*", in addition to the domain "*others*". We represent each video v_h as a set of words $\mathcal{W}_h$ (see Sect. 3.1), and the relatedness degree $\mathrm{Rd}(v_h, A_l)$ between the video v_h and the activity domain A_l is calculated as

$$\mathrm{Rd}(v_h, A_l) = \sum_{w_h \in \mathcal{W}_h} tf_h * \exp\left(-\frac{\mathrm{NGD}(\mathrm{w_h}, \mathrm{A_l})}{\sigma}\right) \tag{5}$$

where σ is a scaling parameter, tf_h is the normalized frequency of the word w_h in $\mathcal{W}_h$, and $\mathrm{NGD}(w_h, A_l)$ is the normalized Google distance [7] between w_h and the domain name A_l. For each video v_h, the domain A_l with the highest relatedness degree is assigned only if $\mathrm{Rd}(v_h, A_l)$ is larger than a threshold; otherwise, this video belongs to "others".

To estimate the relationship strengths in online social network between different users in each activity domain, we build a graphical model (described in Fig. 2) based on two observations:

1. For two users u_i and u_j, given a specific activity domain A_l, the relationship strength $T_l^{(i,j)}$ in this domain is determined by $S^{(i,j)}$, the profile similarity between these two users.
2. The relationship strength $T_l^{(i,j)}$ between u_i and u_j in activity domain A_l impacts their interaction activities in this activity domain (denoted as $D_l^{(i,j)}$).

Furthermore, to increase the accuracy of the graphical model, we introduce an auxiliary variable $Z_l^{(i,j)}$ for each $D_l^{(i,j)}$. The detailed descriptions of these variables in Fig. 2 are as follows:

- $S^{(i,j)} = (s_1^{(ij)}, s_2^{(ij)}, \ldots, s_P^{(ij)})$ is the similarity vector between the users u_i and u_j, where P is the number of attributes in the profile. For the p-th attribute f_p with discrete values, we set $s_p^{(ij)} = 1$ if u_i and u_j have the same values on f_p, and $s_p^{(ij)} = 0$ otherwise. On the other hand, if the values on f_p are continuous, $s_p^{(ij)}$ is determined according to:

$$s_p^{(ij)} = 1 - \frac{|f_p^i - f_p^j|}{\max\limits_{1 \leq k_1, k_2 \leq K} |f_p^{k_1} - f_p^{k_2}|} \tag{6}$$

  where f_p^i represents the value of user u_i on the p-th attribute.
- $T_p^{(ij)}$ is the relationship strength between users u_i and u_j in activity domain A_l.
- $D_l^{(ij)}$ is the strength of interaction activities between users u_i and u_j in activity domain A_l. We measure it based on their related documents in this activity domain, which is calculated as

$$D_l^{(ij)} = \sum_{n=1}^{N} \mathrm{Rd}(d_n, A_l) * ud_{\mathrm{in}} * ud_{jn} \tag{7}$$

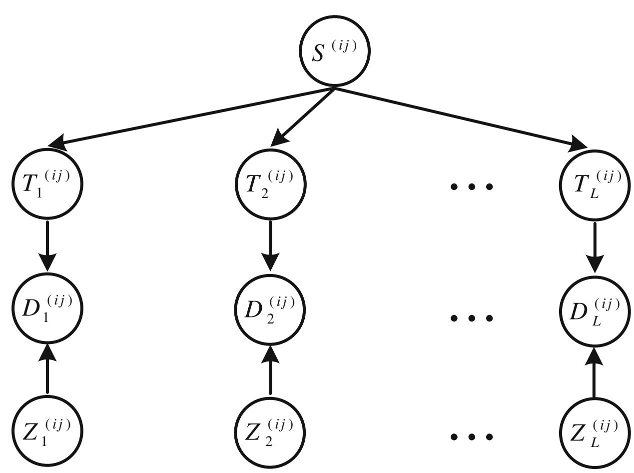

Fig. 2 Graphical model for estimating the relationship strength in a set of activity domains

where d_n is a textual document of the interaction activity (such as messages, news feed, comment, etc. in Facebook), and ud_{in} is an indictor to represent whether the user i is related to the interaction activity n. Here, we set $ud_{\text{in}} = 1$ once the interactive activity n is sponsored or responded by the user i, otherwise, we set $ud_{\text{in}} = 0$. $\mathrm{Rd}(d_n, A_l)$ represents the relatedness degree between the document d_n and activity domain A_l, which can be calculated according to Eq. (5).

– $Z_l^{(ij)}$ is an auxiliary variable. We set $Z_l^{(ij)}$ to 1 in our experiment.

As illustrated in Fig. 2, our graphical model represents the likely causal relationships among all the variables by modeling their conditional dependencies. Based on these dependencies, the joint distribution decomposes as follows:

$$P(T_1^{(ij)}, \ldots, T_L^{(ij)}, D_1^{(ij)}, \ldots, D_L^{(ij)} | \mathbf{S}^{(ij)}, Z_1^{(ij)}, \ldots, Z_L^{(ij)}) = \prod_{l=1}^{L} P(T_l^{(ij)} | \mathbf{S}^{(ij)}) P(D_l^{(ij)} | T_l^{(ij)}, Z_l^{(ij)}) \tag{8}$$

In this work, we adopt the widely used Gaussian distribution [15] to model the conditional probabilities $P(T_l^{(ij)} | \mathbf{S}^{(ij)})$ and $P(D_l^{(ij)} \mid T_l^{(ij)}, Z_l^{(ij)})$, which are expressed as

$$\begin{aligned} P(T_l^{(ij)} | \mathbf{S}^{(ij)}) &= \mathcal{N}(\mathbf{w}_l^T \mathbf{S}^{(ij)}, v) \\ P(D_l^{(ij)} | T_l^{(ij)}, Z_l^{(ij)}) &= \mathcal{N}(\alpha_l T_l^{(ij)} + \beta_l Z_l^{(ij)}, v) \end{aligned} \tag{9}$$

where $\mathbf{w}_l$ is a P-dimensional weight vector to be estimated, α_l, β_l are two coefficients, and v is the variance in Gaussian model, which is configured to be 0.5 in our experiments. To avoid over-fitting, we put L_2 regularizes on parameters $\mathbf{w}_l$ and α_l, β_l, which can be regarded as Gaussian priors:

$$\begin{aligned} P(\mathbf{w}_l) &\propto e^{-\frac{\lambda_1}{2} \mathbf{w}_l^T \mathbf{w}_l} \\ P(\alpha_l, \beta_l) &\propto e^{-\frac{\lambda_2}{2}(\alpha_l^2 + \beta_l^2)} \end{aligned} \tag{10}$$

Among all the variables, $\mathbf{S}^{(ij)}$, $D_l^{(ij)}$, $Z_l^{(ij)}$ are all visible, and $\mathbf{w}_l$, α_l, β_l are to-be-learned parameters. Given the samples of the user pairs $\mathcal{P} = \mathcal{U} \times \mathcal{U}$, the joint probability is expressed according to Eqs. (8–10):

$$\begin{aligned} &\prod_{l=1}^{L} P(\mathcal{P} | \mathbf{w}_l, \alpha_l, \beta_l) P(\mathbf{w}_l) P(\alpha_l, \beta_l) \\ &= \prod_{l=1}^{L} \prod_{(i,j) \in \mathcal{P}} P(D_l^{(ij)} | Z_l^{(ij)}, T_l^{(ij)}, \alpha_l, \beta_l) P(T_l^{(ij)} | \mathbf{S}^{(ij)}, \mathbf{w}_l) \\ &\times P(\mathbf{w}_l) P(\alpha_l, \beta_l) \\ &\propto \prod_{l=1}^{L} \prod_{(i,j) \in \mathcal{P}} e^{-\frac{1}{2v}(\mathbf{w}_l^T \mathbf{S}^{(ij)} - T_l^{(ij)})^2} e^{-\frac{1}{2v}(\alpha_l T_l^{(ij)} + \beta_l Z_l^{(ij)} - D_l^{(ij)})^2} \\ &\times e^{-\frac{\lambda_1}{2} \mathbf{w}_l^T \mathbf{w}_l} e^{-\frac{\lambda_2}{2}(\alpha_l^2 + \beta_l^2)} \end{aligned} \tag{11}$$

Since the joint probabilities of L activity domains in Eq. (11) are independent, we can divide Eq. (11) into L independent joint probabilities and infer the solution for each activity domain separately. In our implementation, we use a gradient-based method [21] to optimize it over the parameters $\mathbf{w}_l^T$, α_l, β_l, and variable $T_l^{(ij)}$. Due to the limited space, the detailed implementation is not presented here.

Given a user u_k, for each video v_h viewed by the user u_t, the recommendation score $R(u_k, v_h)$ is determined by the relationship strength $T_l^{(tk)}$ between u_t and u_k on the special activity domain A_l, which the video v_h belongs to:

$$R(u_k, v_h) = T_l^{(tk)} \mu(v_h, A_l) \tag{12}$$

where $\mu(v_h, A_l)$ is an indicator to represent whether the video v_h belongs to the activity domain A_l. The assignment of v_h to an activity domain could be referred in Eq. (5).

3.4 Popularity-based ranking

In addition to the video's intrinsic information such as title, tags, category and visual characteristic, there usually exist several data contributed by users that can reflect a video's popularity, such as the time of views, and the count of comments. Here, we consider four types of users' contributed data: total views, favorites, rating, and comment count. These data can be collected from most websites. Table 2 lists the data for an example video. We generate four ranking lists based on these data. For each type of data, we generate one ranking list based on its value directly. For "Rating", the ranking list is generated based

Table 2 User-contributed data for a video that reflects its interestingness

Data item	Value	Description
Total views	7,838,561	The number of views
Favorites	76,058	The number of times added as favorite videos by different users
Rating		
Likes	114,028	The positive rating by viewers
Dislikes	2,917	The negative rating by viewers
Comment count	42,551	The number of comments by viewers

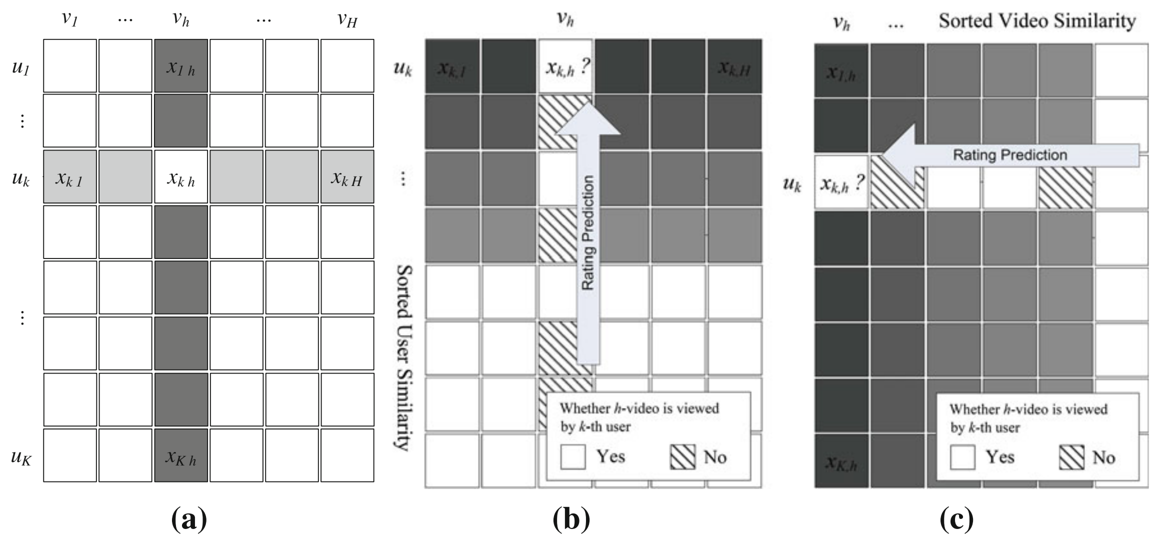

Fig. 3 **a** The user-item matrix. **b** User-based collaborative filtering. **c** Item-based collaborative filtering

on the number of "Likes" minus the number of "Dislikes". Therefore, we generate 4 popularity-based ranking lists.

3.5 Collaborative filtering-based ranking

Collaborative filtering is the most widely adopted approach for video recommendation. It is usually accomplished by mining a user-item matrix (Fig. 3a). For K users and H videos, each element $x_{k,h} = r$ indicates that user u_k rated video v_h by r, where $r \in [0, \ldots, 1]$.

The user-item matrix can be decomposed into row vectors:

$$X = [\mathbf{u}_1, \ldots, \mathbf{u_K}]^T, \quad \mathbf{u_k} = [x_{k,1}, \ldots, x_{k,H}], \quad k = 1, \ldots, K \tag{13}$$

Each row vector $\mathbf{u_k}^T$ corresponds to a user and represents the particular user's video ratings. As discussed below, this decomposition leads to user-based collaborative filtering.

Alternatively, the matrix can also be represented by its column vectors:

$$X = [\mathbf{v}_1, \ldots, \mathbf{v_H}], \quad \mathbf{v_h} = [x_{1,h}, \ldots, x_{K,h}]^T, \qquad h = 1, \ldots, H \tag{14}$$

where each column vector $\mathbf{v_h}$ corresponds to a specific video's rating by all K users. This representation leads to the item-based recommendation approach.

User-based collaborative filtering User-based collaborative filtering approach first finds the "peers" who have the similar tastes with u_k (e.g., rated the same videos similarly), and then recommends the videos that are most liked by those "peers". As illustrated in Fig. 3b, each "peer" (row vector) is represented as a rating vector, where each element is a rating score on a video. The similarity between two "peers" is calculated by measuring the cosine distance between these two vectors. We then select the top-N "peers" and recommends the videos they like to the user u_k.

Item-based collaborative filtering Item-based approach uses a similar strategy to measure the similarity between videos instead of users. As illustrated in Fig. 3c, the unknown rating of a test video v_h by a test user u_k can be predicted by averaging the ratings of other similar videos rated by him/her. Again, each video (column vector) is sorted and re-indexed according to its similarity towards the test video in the user-item matrix, and rating from more similar videos are weighted stronger [8, 36]. In this paper, we adopt the cosine measure to estimate the item similarity. Like the top-N similar users ("peer"), a set of top-N similar videos toward v_h are selected.

In this paper, we regard the rating value $x_{k,h}$ as an indicator. Here, $x_{k,h} = 1$ means that user u_k has viewed video v_h before, and $x_{k,h} = 0$ otherwise.

We employ the aforementioned two methods, i.e., user-based collaborative filtering and item-based collaborative filtering, to generate two ranking lists and integrate them into our approach.

4 Multi-task rank aggregation

After generating multiple video ranking lists using different information sources, our next task is to aggregate these video lists into an optimized video list such that the top items can be recommended to users. It can be formulated as a rank aggregation problem. Rank aggregation methods usually can be categorized into two approaches, namely, the rank-based approach [9] and score-based approach

[27]. Here we adopt the score-based approach, i.e., we fuse ranking lists according to the ranking scores of each video instead of their ranking positions. Instead of linearly combining the search results, we propose a novel ranking approach, called "multi-task ranking SVM", to simultaneously learn models for multiple users such that the correlation of the users can be explored [2, 6, 11]. In our approach, the multi-task SVM algorithm is employed to explore the correlations among multiple ranking lists, and the ranking SVM algorithm [5, 44] is used to focus on the ranks among samples. Compared with the typical SVM, which uses a set of feature vectors as well as samples' labels to learn a classifier, the ranking SVM uses the difference between two feature vectors as well as the ranking order to learn a ranking classifier. When two samples are provided to the ranking SVM algorithm, the classifier can predict which sample has a higher rank.

We first introduce several notations. Let $\mathbf{v}$ represent a vector of the video v, where each element of $\mathbf{v}$ is a score value of this video in a ranking list. We denote $\mathbf{v}_i \succ \mathbf{v}_j$ and $(\mathbf{v}_i, \mathbf{v}_j) \in \mathcal{R}$, if a video $\mathbf{v}_i$ is ranked higher than a video $\mathbf{v}_j$ in an order $\mathcal{R}$. Otherwise, we denote $(\mathbf{v}_i, \mathbf{v}_j) \notin \mathcal{R}$. We assume for simplicity that $\mathcal{R}$ has a strict order, which means that, for all pair $\mathbf{v}_i$ and $\mathbf{v}_j$ in $\mathcal{R}$, we have either $\mathbf{v}_i \succ \mathbf{v}_j$ or $\mathbf{v}_i \prec \mathbf{v}_j$. Let $\mathcal{R}^*$ be the optimal ranking of video in which the video is ordered perfectly according to the user's preference. A training set for multi-task ranking SVM is a set of partial orders $\mathcal{R}^* \subset \mathcal{R}$, which are the total number of pairwise orderings. The target of the multi-task ranking SVM is to learn a function f_k that satisfies $f_k(\mathbf{v}_i) > f_k(\mathbf{v}_j)$ for all pairs of $(\mathbf{v}_i, \mathbf{v}_j) \in \mathcal{R}$ given the k-th user. For simplicity, we assume $f_k(\mathbf{v}) = \mathbf{w}_k \cdot \mathbf{v}$. We assume that for every user, all $\mathbf{w}_k$ can be defined:

$$\mathbf{w}_k = \mathbf{w}_0 + \Delta\mathbf{w}_k \tag{15}$$

where $\mathbf{w}_0$ can be viewed as a common part of all users and $\Delta\mathbf{w}_k$ indicates the distinct difference of the k-th user. The vectors $\Delta\mathbf{w}_k$ are usually enforced to be *small*, i.e., we assume that the tasks are related in a way that the true models are all close to some model $\mathbf{w}_0$. We then estimate all $\Delta\mathbf{w}_k$ as well as $\mathbf{w}_0$ simultaneously. To this end, we solve the following optimization problem, which is analogous to SVM [11]:

$$\begin{aligned} &\min \sum_{k=1}^{K}\sum_{i=1}^{n_k} \xi_{ki} + \lambda_1 \sum_{k=1}^{K} ||\Delta\mathbf{w}_k||^2 + \lambda_2 ||\mathbf{w}_0||^2 \\ &\text{s.t. } (\mathbf{w}_0 + \Delta\mathbf{w}_k)(\mathbf{v}_{ki}^{(1)} - \mathbf{v}_{ki}^{(2)}) \geq 1 - \xi_{ki} \;\; \xi_{ki} \geq 0 \end{aligned} \tag{16}$$

In the aforementioned equation, n_k denotes the number of training sample pairs for the k-th user, $(\mathbf{v}_{ki}^{(1)}, \mathbf{v}_{ki}^{(2)})$ refers to the k-th training pair for the i-th user, λ_1 and λ_2 are the positive regularization parameters, and the ξ_{ki} are slack variables that measure the error of the final models $\mathbf{w}_k$. We can see that, actually the formulation poses a regularization constraint on the "common part" $\mathbf{w}_0$ and control how much the solutions $\mathbf{w}_k$ differ from each other by controlling the size of the $\Delta\mathbf{w}_k$. If either λ_1 or λ_2 is set to 0, the problem degrades to the conventional ranking SVM algorithm. The difference is that, if $\lambda_1 = 0$, the formulation is to learn a global ranking SVM with all training data; if $\lambda_2 = 0$, the formulation is to learn a ranking SVM for each user independently with the training data for this specific user.

It can be derived that the optimal solution to the optimization problem is

$$\Delta\mathbf{w}_k^* = \frac{T}{2\lambda_1}\sum_{i=1}^{m} \alpha_{ik}(\mathbf{v}_{ik}^{(1)} - \mathbf{v}_{ik}^{(2)}) \tag{17}$$

$$\mathbf{w}_0^* = \frac{\lambda_1}{K\lambda_2}\sum_{k=1}^{K} \Delta\mathbf{w}_k^* \tag{18}$$

where α_{ik} is the nonnegative Lagrange multipliers. The dual formulation for the above problem is:

$$\begin{aligned} &\max_{\alpha_{ik}} \left\{ -\frac{1}{2}\sum_{i=1}^{m}\sum_{s=1}^{K}\sum_{j=1}^{m}\sum_{k=1}^{K} \alpha_{is}\alpha_{jk} K_{sk}(\mathbf{v}_{\mathbf{is}}^{(\mathbf{1})} - \mathbf{v}_{\mathbf{is}}^{(\mathbf{2})}, \mathbf{v}_{\mathbf{jk}}^{(\mathbf{1})} - \mathbf{v}_{\mathbf{jk}}^{(\mathbf{2})}) \right. \\ &\quad \left. + \sum_{i=1}^{m}\sum_{k=1}^{K} \alpha_{it} \right\} \\ &\text{s.t. } 0 \leq \alpha_{ik} \leq T/2\lambda_1 \end{aligned} \tag{19}$$

where

$$K_{sk}(\mathbf{y}, \mathbf{z}) := \left(\frac{\lambda_1}{K\lambda_2} + \delta_{sk}\right)\mathbf{y} \cdot \mathbf{z}, \quad s, k = 1, \ldots, T \tag{20}$$

Accordingly, we can obtain the optimal $\mathbf{w}_0$ and $\Delta\mathbf{w}_k$. By performing rank aggregation with the weight vector $\mathbf{w}_0 + \Delta\mathbf{w}_k$, we can obtain the final ranking list for recommendation.

5 Experiments

5.1 Experimental settings

We conduct experiments on 76 participants with their permissions to download their information on Facebook [12] and YouTube. Online social networks such as Facebook allow users to interact and share content using social links. Take the most popular interaction activity "comment" for example, on Facebook; a user's friends can post comments to the user's wall and these comments appear on the user's wall and can be seen by others who visit the user's profile. So, on Facebook, we downloaded the profiles of these 76 users (including education background, occupation, philosophy, location, interests, etc.), as well as the interaction activities (such as messages, news feed and comments) for each user between Sep. 2010 and Oct. 2010. Totally, there are about 76,000 interaction activities. On Youtube, we collected the viewed videos by each user in a one-month period (from Dec. 2010 to Jan. 2011). The videos and their associated information, including title, description, tags, category and popularity, were collected. As a result, we collected 11,400 videos in total (150 videos/user). Figure 4 shows a set of video examples downloaded from YouTube.

For the interaction activities from Facebook, we randomly divided it into two parts: the training set that containied about 38,000 interaction activities and the testing dataset that included 38,000 interaction activities. The training set is used to learn the parameters in the graph model, and the testing set is used to evaluate the performance of our graph model on users' relationship strength. For YouTube videos dataset, we split the viewed videos into two parts for each user: the training set (5,664 videos), including all the videos, viewed in the first two weeks by each user, and the testing set (5,736 videos) containing all the videos, viewed in the next two weeks. For the training set, we further divided it into two parts: the first part consists the videos viewed in the first week (2,756 videos) and the second part consists the videos viewed in the second week (2,908 videos). For the first part in the training set, we take it as a viewing history since the History-Based Ranking needs history information to conduct video recommendation. For the second part, we use it to train all ranking lists as well as learn a multi-task ranking SVM model. The testing set is used to evaluate the performance of video recommendation.

To evaluate the performance, on Facebook dataset, we adopt a manual labeling procedure to generate ground truth. We ask 76 users to label the relationship strengths. For each user, we provide a list of his friends and an activity domain, and then the user labels his relationship strengths with each of his friends on the scale of "strong", "normal", "weak". When two users label a different relationship strength between them, we will ask them to relabel this relationship strength. After that, we obtain 39,900 relationship strengths on all the domains. On YouTube video dataset, for each user, we simply take the relevant videos for recommendation as those videos have been viewed before by that user. All the other videos are taken as negative samples. It is worth noting that this setting could make the performance underestimated, as the users may also be interested in the other unseen videos. However, it is sufficient to compare the performance among different algorithms.

For performance evaluation metric, we adopt the normalized discounted cumulative gain (NDCG) [25]. Given the ranking list $v_1, \ldots, v_k$, NDCG@k is calculated by

Fig. 4 Several examples of the 11,400 videos used in our experiments

$$\text{NDCG}@k = \frac{1}{Z}\sum_{i=1}^{k}\frac{2^{t(i)}-1}{\log(1+i)} \tag{21}$$

where $t(i)$ is the function that represents the reward given to the video at position i, and Z is a normalization term derived from the top k videos of a perfect ranking. We set $t(i) = 1$ if the i-th result is relevant, 0 otherwise.

5.2 Experimental results

5.2.1 *Evaluation on social network based video recommendation*

In this experiment, we evaluate the performance of social network-based ranking on video recommendation. The parameters of our graph-based method in Sect. 3.3 are set by the 5-fold cross-validation on the training set of Facebook dataset, where we set the scaling parameter $\sigma = 0.5$ (see Eq. 5) and $\lambda_1 = \lambda_2 = 1$ (see Eq. 10).

In the first experiment, we evaluate the effectiveness of our graph-based approach on relationship strength estimation. We compare our approach against the Linear Combination Approach (LCA), where the relationship strength between two users in a specific activity domain is calculated by the linear combination of two sources: the profile similarity of these two users and the interaction activity strength between these two users in that domain. The fuse weights in LCA approach is determined by the 5-fold cross-validation on the training set of Facebook dataset.

Figure 5 shows the performance comparison results. We can see that for each activity domain, our approach which attempts to represent the intrinsic causality of social interaction activities via statistical dependencies outperforms the linear combination method, in which the relationship strength on a specific activity domain is calculated as the linear combination of two users' profile similarity and the interaction activity strength in that domain. This demonstrates the effectiveness and feasibility of our approach. The performance difference is significantly large in the "sports" domain. One possible reason is that "sports" domain contains some representative words such as "basketball", "swimming", and "jogging", to ensure a high classification accuracy on interaction activities, and thus the relationship strength estimation is more accurate.

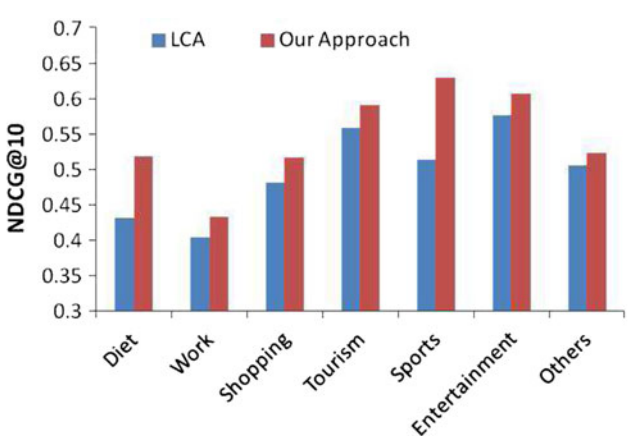

Fig. 5 The performance comparison between our approach and the LCA on various activity domains

In the second experiment, we evaluate the utility of our social network-based ranking approach using or without using activity domain. For the approach without using activity domain, we use our graph model to infer user's relationship strength without discriminating the activity domains. This unique relationship strength between two users is used to recommend videos.

Table 3 illustrates the comparison results measured by average NDCG@10. From the results, we can see that our approach using activity domain outperforms that without using activity domain. The activity domain is quite useful source to recommend videos. For example, when two users u_i, u_j share interest on the domain sport, the sport videos from u_j are better items to be recommended to u_i, as compared with the other videos from u_j.

5.2.2 *Evaluation on multiple information sources*

In this experiment, we compare our approach that integrates all the information sources to the approaches that only use one or a part of the information sources. The parameters are set as follows: for the User Information approach (UI) and the History-Based approach (H), we empirically set the scaling parameters $\sigma = 0.5$ in Eqs. (1) and (3). The parameters of the social network-based ranking are set by the 5-fold cross-validation on the training set of Facebook dataset. We use our Multi-Task Rank Aggregation to combine multiple ranking lists, where we set the parameters $\lambda_1 = 1$ and $\lambda_2 = 3$ (see Eq. 16) by the 5-fold cross-validation on the second part of the training set.

Table 4 illustrates the comparison results measured by average NDCG@10. From the results, we draw the following conclusions: First, the text-based recommendation consistently achieves a better performance than the visual content-based recommendation and the popularity-based recommendation. Second, comparing the results on history information, recommendation based on recent viewing history performs better than that on short-term history and long-term history. This is because many users' interests are temporally continuous and thus the recently viewed videos are usually closer to users' interests. Third, among the two approaches based on online social network, the one leveraging the relationship strengths has much better results than the other one which only uses the binary

Table 3 The utility of activity domain in social network-based ranking methods, where R, S, and L indicate the approaches using the recent videos, short-term history videos and long-term history videos, respectively

	R	S	L
SN-based ranking without using activity domain	0.22	0.25	0.24
SN-based ranking using activity domain	0.26	0.31	0.27

relationship. This is because the approach based on binary relationship treats the relationships among all users as equal and neglects an important fact that the close friends may be a more reliable source than the acquaintances in video recommendation. Fourth, the more the information sources are integrated, the higher the performance is achieved. Thus our approach integrating all the information sources achieves the best performance.

5.2.3 *Evaluation on multi-task rank aggregation*

In this experiment we compare our multi-task rank aggregation approach with the following two methods for rank aggregation:

1. Learning a global rank aggregation model for all users, i.e., the weighting vectors of all users are identical.
2. Learning a specific rank aggregation model for each user, i.e., $\mathbf{w}_0$ is removed in Eq. 2.

We denote our approach and these two methods as "Multi-Task", "Global" and "Local", respectively. These two methods both use Ranking SVM, and the only difference is that, the first one learns a model for all users and the second one learns a model for each individual user. As mentioned in Sect. 4, these two methods can also be regarded as setting λ_1 and λ_2 to 0, respectively. For these two methods, the parameters of Ranking SVM are tuned by the 5-fold cross-validation on the second part of the training set.

These three methods are used to fuse the 21 ranking lists introduced in Sect. 3. Figure 6 illustrates the comparison of average NDCG at different depths. We can see that our approach consistently outperforms the other two methods. Figure 7 illustrates the detailed NDCG@10 results for the 76 users. We can see that for most users our approach achieves the best results. It is worth noting that, by adopting the multi-task rank aggregation approach, we partially address the problem of recommendation videos

Table 4 The comparison of exploring different information sources for video recommendation (for the history type, R, S and L indicate the recent video, short-term history and long-term history, respectively)

Ranking method	Textual information			Visual information		
User information (UI)						
Interest	0.34			–		
Location	0.23			–		
Profile	0.22			–		
	R	S	L	R	S	L
History-based (H)	0.36	0.33	0.30	0.29	0.28	0.26
Social network-based (SN)						
Binary relationship						
R	0.20					
S	0.24					
L	0.23					
(0,1] relationship						
R	0.26					
S	0.31					
L	0.27					
Popularity-based (P)						
Total views	0.09					
Favorites	0.08					
Rating	0.10					
Comment count	0.06					
Collaborative filtering-based (CF)						
Item-based	0.22					
User-based	0.23					
UI+SN	0.36					
UI+H+SN	0.38					
UI+H+SN+P	0.39					
All Information	0.42					

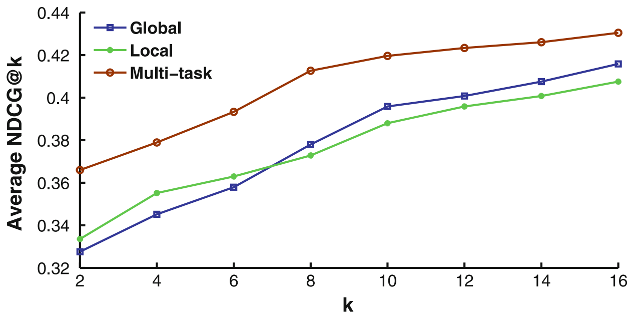

Fig. 6 The performance variation of different ranking strategies in different depth

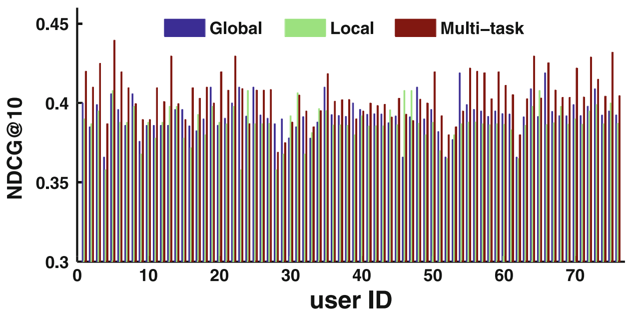

Fig. 7 The performance comparison of the three rank aggregation methods for each user. NDCG@10 is used as the performance evaluation metric here

for new users. That is because our model is built upon multiple information sources, which is still workable when some information sources are unavailable. Besides, we can employ several model adaptation methods to adapt the models of the existing users to a new user [43].

6 Conclusion and further work

In this paper, we proposed a video recommendation scheme that is able to integrate multiple information sources with a multi-task rank aggregation approach. Ranking lists are generated by exploring different information sources, including the education background, occupation, location, interest, past viewing history, social network of users; as well as the title, description, tags, user-contributed data, and visual content of videos. We then fuse the ranking lists with a multi-task learning approach. We conducted experiments on more than 11,000 videos and the results demonstrated the feasibility and effectiveness of our approach. Our scheme is flexible and different ranking methods can be easily integrated as we only need to fuse several ranking lists in the aggregation step. In our future work, we will conduct experiments with more users and videos and we will also investigate the problem of new users with empirical justification. Besides, we will explore automatic methods for determining activity domains that will be used in the social network-based ranking.

Acknowledgments This work was supported by the Innovation Scholarship for Ph.D. students at Beihang University under research grant (YWF-12-RBYJ-012), the National Natural Science Foundation of China (61170189, 60973105), the Fund of the State Key Laboratory of Software Development Environment under Grant No. SKLSDE-2011ZX-03 and the Singapore National Research Foundation & Interactive Digital Media R&D Program Office, MDA under research grant (WBS:R-252-300-001-490). The authors would like to thank the editors and the anonymous reviewers for their valuable comments and remarks on this paper.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)
2. Bakker, B., Heskes, T.: Task clustering and gating for bayesian multitask learning. J. Mach. Learning Res. **4**, 83–99 (2003)
3. Baluja, S., Seth, R., Sivakumar, D., Jing, Y., Yagnik, J., Kumar, S., Ravichandran, D., Aly, M.: Video suggestion and discovery for youtube: taking random walks through the view graph. In: Proceeding of the 17th International Conference on World Wide Web, pp. 895–904 (2008)
4. Burke, R.: Hybrid web recommender systems, pp. 377–408. Lecture Notes in Computer Science (2007)
5. Cao, Y., Xu, J., Liu, T.-Y., Li, H., Huang, Y., Hon, H.-W.: Adapting ranking svm to document retrieval. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 186–193 (2006)
6. Caruana, R.: Multitask learning. Mach. Learn. **28**(1), 41–75 (1997)
7. Cilibrasi, R.L., Vitányi, P.M.: The google similarity distance. IEEE Trans. Knowl. Data Eng. **19**(3), 370–385 (2007)
8. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Trans. Inf. Syst. **22**, 143–177 (2004)
9. Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank aggregation methods for the web. In: Proceedings of the 10th International Conference on World Wide Web, pp. 613–622 (2001)
10. Encyclopedia: http://en.wikipedia.org/wiki/YouTube/. Accessed June 2011
11. Evgeniou, T., Pontil, M.: Regularized multi-task learning. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 109–117 (2004)
12. Facebook facesheet: http://www.facebook.com/press/info.php?statistics. Accessed June 2011
13. Geng, B., Yang, L., Xu, C., Hua, X.: Content-aware ranking for visual search. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3400–3407 (2010)
14. Geng, B., Yang, L., Xu, C., Hua, X.: Ranking model adaptation for domain specific search. IEEE Trans. Knowl. Data Eng. (2010)
15. Goodman, N.: Statistical analysis based on a certain multivariate complex gaussian distribution (an introduction). Ann. Math. Stat. **34**(1), 152–177 (1963)
16. Herlocker, J., Konstan, J., Terveen, L., Riedl, J.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. **22**(1), 5–53 (2004)

17. Hill, W., Stead, L., Rosenstein, M., Furnas, G.: Recommending and evaluating choices in a virtual community of use. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 194–201 (1995)
18. Hong, R., Wang, M., Xu, M., Yan, S., Chua, T.: Dynamic captioning: video accessibility enhancement for hearing impairment. In: Proceedings of the ACM International Conference on Multimedia, pp. 421–430 (2010)
19. Hopfgartner, F., Vallet, D., Halvey, M., Jose, J.: Search trails using user feedback to improve video search. In: Proceeding of the 16th ACM International Conference on Multimedia, pp. 339–348 (2008)
20. Hu, J., Zeng, H.-J., Li, H., Niu, C., Chen, Z.: Demographic prediction based on user's browsing behavior. In: Proceedings of the 16th International Conference on World Wide Web, pp. 151–160 (2007)
21. Hu, X., Sun, N., Zhang, C., Chua, T.: Exploiting internal and external semantics for the clustering of short texts using world knowledge. In: Proceeding of the 18th ACM Conference on Information and Knowledge Management, pp. 919–928 (2009)
22. Hu, X., Tang, L., Liu, H.: Enhancing accessibility of microblogging messages using semantic knowledge. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 2465–2468. ACM (2011)
23. Hu, X., Liu, H.: Text analytics in social media, pp. 385–414. Mining Text Data (2012)
24. Irie, G., Hidaka, K., Satou, T., Yamasaki, T., Aizawa, K.: A degree-of-edit ranking for consumer generated video retrieval. iN: IEEE International Conference on Multimedia and Expo, pp. 1242 –1245 (2009)
25. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. ACM Trans. Inf. Syst. **20**, 422–446 (2002)
26. Jebrin, A., Williams, M.: Credibility-aware web-based social network recommender: follow the leader. Recommender Systems and the Social Web, p. 1 (2010)
27. Liu, Y.-T., Liu, T.-Y., Qin, T., Ma, Z.-M., Li, H.: Supervised rank aggregation. In: Proceedings of the 16th International Conference on World Wide Web, pp. 481–490 (2007)
28. Luo, H., Fan, J., Keim, D.A.: Personalized news video recommendation. In: Proceeding of the 16th ACM International Conference on Multimedia, pp. 1001–1002 (2008)
29. Ma, H., King, I., Lyu, M.: Learning to recommend with social trust ensemble. In: Proceedings of the 32nd international ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 203–210. ACM (2009)
30. Mei, T., Yang, B., Hua, X.-S., Yang, L., Yang, S.-Q., Li, S.: Videoreach: an online video recommendation system. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 767–768 (2007)
31. Mei, T., Aizawa, K.: Internet multimedia search and mining. In: Video Recommendation. Bentham Science Publisher (2011)
32. Mei, T., Yang, B., Hua, X., Li, S.: Contextual video recommendation by multimodal relevance and user feedback. ACM Trans. Inf. Syst. **29**(2), 10 (2011)
33. Öztürk, G., Kesim Cicekli, N.: A hybrid video recommendation system using a graph-based algorithm. Modern Approaches in Applied Intelligence, pp. 406–415 (2011)
34. Park, J., Lee, S., Lee, S., Kim, K., Chung, B., Lee, Y.: Online video recommendation through tag-cloud aggregation. IEEE MultiMedia, pp. 78–87 (2010)
35. Resnick, P., Kuwabara, K., Zeckhauser, R., Friedman, E.: Reputation systems. Commun. ACM **43**(12), 45–48 (2000)
36. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, pp. 285–295 (2001)
37. van Setten, M., Veenstra, M., Nijholt, A., van Dijk, B.: Prediction strategies in a TV recommender system-method and experiments. In: Proceedings of the Second IADIS International Conference WWW/Internet, pp. 203–210 (2003)
38. Shen, J., Tao, D., Li, X.: Modality mixture projections for semantic video event detection. IEEE Trans. Circuits Syst. Video Technol. **18**(11), 1587–1596 (2008)
39. Wang, M., Hua, X.: Active learning in multimedia annotation and retrieval: a survey. ACM Trans. Intell. Syst. Technol. **2**(2), 10 (2011)
40. Wang, M., Hua, X., Hong, R., Tang, J., Qi, G., Song, Y.: Unified video annotation via multigraph learning. IEEE Trans. Circuits Syst. Video Technol. **19**(5), 733–746 (2009)
41. Wang, M., Hua, X., Tang, J., Hong, R.: Beyond distance measurement: constructing neighborhood similarity for video annotation. IEEE Trans. Multimedia **11**(3), 465–476 (2009)
42. Wang, M., Yang, K., Hua, X., Zhang, H.: Towards a relevant and diverse search of social images. IEEE Trans. Multimedia **12**(8), 829–842 (2010)
43. Yang, J., Hauptmann, A.G.: A framework for classifier adaptation and its applications in concept detection. In: Proceeding of the 1st ACM International Conference on Multimedia Information Retrieval, pp. 467–474 (2008)
44. Yu, H.: SVM selective sampling for ranking with application to data retrieval. In: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 354–363 (2005)
45. Yuan, J., Zha, Z., Zhao, Z., Zhou, X., Chua, T.: Utilizing related samples to learn complex queries in interactive concept-based video search. In: Proceedings of the ACM International Conference on Image and Video Retrieval, pp. 66–73 (2010)
46. Zha, Z., Hua, X., Mei, T., Wang, J., Qi, G., Wang, Z.: Joint multi-label multi-instance learning for image classification. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
47. Zha, Z., Yang, L., Mei, T., Wang, M., Wang, Z.: Visual query suggestion. In: Proceedings of the ACM International Conference on Multimedia, pp. 15–24 (2009)
48. Zhao, X., Li, G., Wang, M., Li, S., Chen, X., Li, Z.: An online video recommendation framework using rich information. In: Proceedings of the 3rd ACM International Conference on Internet Multimedia Computing and Service, pp. 46–50 (2011)
49. Zhao, X., Li, G., Wang, M., Yuan, J., Zha, Z., Li, Z., Chua, T.: Integrating rich information for video recommendation with multi-task rank aggregation. In: Proceedings of the 19th ACM International Conference on Multimedia, pp. 1521–1524 (2011)