Online Video Recommendation in Sharing Community

Xiangmin Zhou
School of Computer Science
and Info Tech.
RMIT University, Australia
xiangmin.zhou@rmit.edu.au

Lei Chen
Department of Computer
Science and Engineering,
Hong Kong University of
Science and Technology,
Hong Kong, China
leichen@cs.ust.hk

Yanchun Zhang
Centre for Applied Informatics,
Victoria University, Australia
School of Computer Science,
Fudan University, China
yanchun.zhang@vu.edu.au

Longbing Cao
Advanced Analytics Institute
University of Technology,
Sydney, Australia
longbing.cao@uts.edu.au

Guangyan Huang
School of Information
Technology
Deakin University, Australia
guangyan.huang@deakin.edu.au

Chen Wang
Digital Productivity Flagship
CSIRO, Sydney Australia
chen.wang@csiro.au

ABSTRACT

The creation of sharing communities has resulted in the astonishing increasing of digital videos, and their wide applications in the domains such as entertainment, online news broadcasting etc. The improvement of these applications relies on effective solutions for social user access to video data. This fact has driven the recent research interest in social recommendation in shared communities. Although certain effort has been put into video recommendation in shared communities, the contextual information on social users has not been well exploited for effective recommendation. In this paper, we propose an approach based on the content and social information of videos for the recommendation in sharing communities. Specifically, we first exploit a robust video cuboid signature together with the Earth Mover's Distance to capture the content relevance of videos. Then, we propose to identify the social relevance of clips using the set of users belonging to a video. We fuse the content relevance and social relevance to identify the relevant videos for recommendation. Following that, we propose a novel scheme called sub-community-based approximation together with a hash-based optimization for improving the efficiency of our solution. Finally, we propose an algorithm for efficiently maintaining the social updates in dynamic shared communities. The extensive experiments are conducted to prove the high effectiveness and efficiency of our proposed video recommendation approach.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Design, Experimentation, Performance

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGMOD'15, May 31–June 4, 2015, Melbourne, Victoria, Australia.
Copyright © 2015 ACM 978-1-4503-2758-9/15/05 ...$15.00.
http://DOI string/url.

Keywords

Online Video Recommendation, Social Similarity

1. INTRODUCTION

The popularity of online video shared communities has created an opportunity to share media information at an unprecedented scale. Users access to these communities, operate and comment on videos for various purposes such as entertainment, online news broadcasting and advertisement. In sharing communities, the behavior of people is highly affected by the recommendations from the system. According to a white paper released in February 2012 by Unruly Media, viewers enjoy online videos they discover from a recommendation more than ones they discover through browsing, which reports 65% of viewers who watched a recommended video enjoyed it, representing a 14% increase from 57% who enjoyed a video found through browsing [3]. Therefore, an appropriate recommendation is a promising way of increasing the viewing rate to specific media data, enhancing the effect of online news broadcasting and advertisement. Although many existing video sharing communities, such as Youtube, Netflix, MySpace and Google Video, have provided recommendation services, the recommendations on relevant videos are produced based on social texts. The content and social connections have not been fully exploited in the video recommendation so far.

In this paper, we focus on the problem of online video recommendation in shared community like Youtube, where unregistered users can get accesses to social videos. Traditional recommendation systems generate the recommendations by matching the recommended items and the user input profiles, and deliver them to the appropriate users. However, in sharing community, it is very common that unregistered users access to the social videos, where these users' profiles are unavailable. Some users are just beginners for a certain sharing community, and have no user history. Especially, in recent years, some unregistered new users may disable their cookies or use public machines when they get access to videos in sharing communities for the purpose of privacy protection [20]. The recent statistics data show that 19.1% users use the private browsing mode [2]. Therefore, for such cases, traditional recommendation is not workable any more. Unlike traditional recommendation using users' registered information, our video recommendation takes a user clicked video as the input, and returns a list of recommended videos that are relevant to the user's current view. Technically, our video recommendation is highly related to the video search in multimedia database field. From this point of view, techniques for video search can be extended for video recommendation task. But different from the video search which returns the matched clips in content of a video, our video recommendation system returns the relevant videos that include both the matched clips and those relevant but unmatched ones. Thus, directly using video identification methods will lose some relevant videos that are unmatched to the query. Fortunately, although many users may access the video content anonymously as the *subjects* receiving recommendation from the system, many other registered users comment on videos using their real identities, which provides important *social data source* in sharing communities. Exploiting the social information that named users provided is very promising for recommending videos to anonymous social users. Though Yang etc proposed to recommend online videos based on video content and relevance feedback [33], they have not considered the social information of videos in sharing communities.

Motivated by the limitation of traditional video search and recommendation systems, we propose a multiple feature-based video recommendation in sharing communities using the content and social connections. We first exploit a compact video signature over video segments to capture the spatial and temporal content information. The similarity between two signatures is measured to identify the matched videos that are similar in content with the query. Then we capture the social information of each video by extracting the social users who comment on it. The social information of a video reflects the users who are interested in it. The social similarity between two videos is identified by the Jacard similarity between their user sets. The final video recommendation is performed by a late fusion of the content and social features. Our contributions in this paper are summarized as follows.

1. We propose a new framework which exploits multiple content and social features of videos for video recommendation in sharing communities. While the content feature identifies the matched videos, the social connection captures the relevant unmatched clips.

2. We propose a new complementary video matching based on content and social fusion. The new approach well embeds the rich social information into video content, thus appropriate to the application of social video recommendation.

3. We propose a novel sub-community-based approximation relevance scheme (SAR) and a hash-based optimization for improving the efficiency of our framework. With the SAR, a robust user dictionary is constructed by extracting several sub-communities, and each user is mapped to its sub-community number. Accordingly the social context of each video is converted into a vector. The optimization strategy greatly improves the efficiency of social context vectorization.

4. Last but not the least, we propose a novel algorithm which well maintains the social updates in dynamic sharing social communities. We conduct extensive experiments on hundreds hours of real video data to verify the effectiveness and efficiency performance of the proposed solution.

The rest of the paper is organized as follows: Section 2 reviews the related research on social video recommendation. Section 3 describes the framework of the proposed social video recommendation system. We present our proposed Multiple Feature-based video recommendation approach together with our SAR and hash-based schemes over our relevance identification modal, and social updates maintenance algorithm in section 4. The high effectiveness and efficiency of our approach is evaluated in section 5. Finally, we conclude the whole work in section 6.

2. RELATED WORK

In this section, we review the existing research on two problems closely related to our work, including the video recommendation and near duplicate video detection.

2.1 Video Recommendation

Approaches have been proposed for video recommendation. In [29], Setten et al. use prediction strategies for personalized TV recommendation. In [8], Christakou et al. apply content-based and collaborative filtering to predict personalized movie recommendations. In [4], Baluja et al. presented a personalized approach which generates recommendations based on the analysis of the user-video graph for videos from Youtube. In [19], Luo et al. developed a personalized news video recommendation system, which first detects reliable news topics based on the multimodal information, and then integrates the topic network based on contextual relationship with the users' profiles considered for interactive navigation and exploration of news videos. In [17], a news recommendation was proposed by considering the item characteristics including the news content, access patterns, named entities, popularity and recency. Li et al model the news articles based on the contexts of users and articles [16]. Sedhain et al recommend content in Facebook by a social affinity filtering [22]. In [11], authors introduced a news video recommender system that exploits semantic augmentation of news stories, and represents dynamic user profile by capturing users' evolving information. In [9], the video recommendation system for Youtube was discussed. The system produces personalized recommendations based on users' activity on the site. All these approaches focus on the personalized recommendations, which recommend videos matching to users' profiles or interests, while not the item itself. In [39], a content-based framework VideoTopc was proposed for movie recommendation. VideoTopic exploits both visual features and textual features of videos to construct a topic model, which represents the video content and links to its user interests. The video recommendation problem is transformed into finding the videos that have minimal topic distribution difference with user interests. This topic model-based approach naturally links video content and user interests. However, the user interests need to be learnt from the topics of users' browsing history. In [34], a hybrid recommendation approach has been proposed for video recommendation over social networks by considering the user relationship strength and the interest degree of video. For a given user, the recommendation score of a video candidate is decided by the interest degree of the video by the user's friends, and the relationship strengths between the user and his friends. However, this approach requires the information of users who receive recommendations. Practically, most users access to video sharing communities anonymously, where users' profiles are not available for the system. This poses a challenge for personalized recommendations.

To handle the cases where users' profiles are absent, Yang et al. proposed to perform online video recommendation by exploiting textual, visual and audio information [33]. The relevance values from all modalities are fused to improve the results of recommendations by attention fusion function and relevance feedback. However, for the video recommendation in shared social community, the feasibility of this approach has been challenged due to several reasons. First, it does not consider the social connection among users, which is an important characteristic of videos in sharing communities. Thus, the social relevance of videos can not be identified.

Second, the existing work [33] focused on designing a multimodal relevance identification method, while the efficiency issue has not been addressed. In this work, we fully exploit the social connections attached to videos, and propose a set of query optimization techniques including the SAR scheme and chained hash-based optimization to overcome the disadvantages of the existing work.

2.2 Near Duplicate Video Detection

Video representation and similarity measure are two basic tasks in near duplicate video detection. Generally, based on the video representation, existing video detection approaches can be put into three categories, global feature-based [6, 13, 12, 24], local feature-based [15, 31, 36, 27], and signature-based [14, 40, 32, 35, 23].

Global feature-based approaches extracted visual features such as color histograms to represent frames, and construct compact representations over these visual features. Typical examples on global feature-based approaches include Video Signature (Visig) [6], Video Triplet (ViTri) [24], bounded coordinate system (BCS) [13] and video distance trajectory (VDT) [12]. In [6], Cheung et al. represent each video as a set of frame samples called Video Signature(ViSig), and perform video matching based on the percentage of similar frames shared by two sequences, which is further estimated by computing the percentage of similar ViSig frame pairs. In [24], Shen et al. estimate the distance between videos by comparing their video clusters and estimating the number of video frames shared by them. In [13], Huang et al. summarize each video as a single representation called bounded coordinate system (BCS) that captures the correlation of it. The video matching is performed by integrating the global difference between two frame sets, the content changing trends and ranges. In [12], Huang et al. describe a video as a series of video distance trajectories, and perform video matching using a weighted edit distance. However, due to the limitation of global visual features, these approaches suffer from severe information loss, leading to low effectiveness of video detection.

Local feature-based approaches extract visual features from a set of local interest points captured in each frame, and improve the efficiency of detection by effective search mechanisms or constructing compact representations over local descriptors. Recent examples on this line include local interest trajectory [15], Hierarchical detection [31], and multiple feature hashing [27]. In [15], Law-To et al. construct trajectories over local interest points of adjacent frames, and use a voting function based on the signal description, the contextual information and the combination of relevant labels for video matching. In [31], Wu et al. proposed a video matching that extracts a number of local descriptors from each keyframe, and compares two videos by calculating the matched local descriptor pairs shared by their keyframes. The search efficiency is improved by using a global feature based filtering before the local descriptor based matching. In [27], Song et al. proposed a multiple feature hashing based on both global and local features for near duplicate video detection. The comparison between two videos is performed over hash codes by approximating the common bits along all dimensions shared by them. In [36], Zhou et al. construct a tensor series over the local descriptors of each video, and perform Hamming based tensor series matching for near duplicate video identification. Using local descriptors, more discriminative information can be captured. However, these approaches either suffer from high computation cost [15, 31], the sensitivity of local descriptors over streams [36], or highly depend on the selected video training dataset [27], thus not robust in sequence detection.

Signature-based approaches extracted compact signatures from video frames or video segments to leverage the effectiveness and efficiency of detection. Typical signature-based approaches include color shift signature [40], cut signature [40], centroid signature [40], ordinal signature [14], sketch representation [32], STF-CE and STR-LBP signatures [23], video cuboid signature [35]. In [40], various types of compact signatures, such as cut length, color shift and centroid signatures, are used for video matching. Here, cut length signature is extracted by detecting the cuts in a video and counting the number of frames between two adjacent cuts. Color shift signature is obtained by detecting the color difference between neighboring frames. Centroid signature describes the shift of the lightest and darkest areas between neighboring frames. In [14], Kim et al. partition each keyframe into a number of small equal size blocks that are represented as ordinal signatures based on the average intensity values of these blocks. The comparison between two videos is performed by calculating the normalized distance of their ordinal signatures. In [32], Yan et al. proposed a frame partition based sketch representation and a set based measure for video matching over streams. In [23], Shang et al. proposed two compact signatures, STF-CE and STR-LBP, which use ordinal relations of keyframe blocks for video matching. While some signatures suffer from severe information loss [40], others are sensitive to the spatial editing of frames [14, 32, 23]. In [35], Zhou et al. proposed a video cuboid signature that captures the spatial and temporal information change of a video segment. The similarity between two video cuboid signatures is decided by the EMD distance between them, and the final video matching is performed by a set based measure with the similarity between each matched signature pair embedded. Video cuobid signatures capture discriminative local information of videos, and the EMD-based signature set measure well handle the spatial and temporal editing of videos, thus more robust for video detection. Therefore, we will apply video cuboid signature model and fuse social information of videos for clip recommendation in sharing communities. The notation used in this paper is listed in Table 1 for easy reference.

Table 1: Notation

Notation	Meaning
Q, V	A video
q_f	The visual feature of a video
q_s	The social connection of a video
Sim	A social video relevance function
S	A list of videos
S_1, S_2	A signature series
D_V	A set of user ids
id_{Vi}	id of a user commenting v.
k	The number of sub-communities
K	The number of top score videos
w	The lightest edge weight
ω	The weight of the social relevance
s	A string
h_i	An intermediate hash value
$e,$	An edge between two users
$\{e_i\}$	An edge set
G_I	A UIG graph
U_i, U_j	Social users
n	Number of returned videos
N	Number of retrieved videos with rating score bigger than 4
γ	The rank of a returned video
$\mathcal{Q}$	The number of queries
$\mathcal{H}$	A chained hash table

3. FRAMEWORK OF OUR SOLUTION

This section describes the framework of our proposed approach for social video recommendation. In our social video recommendation, the input of our system is a user selected video document in a sharing community, which is represented as a pair $Q = (q_f, q_s)$, where q_f denotes its visual feature described as a signature series and q_s is its social connection described as a set ids of users commenting it. Given a social video Q selected by a user, a social video relevance function Sim, the task of video recommendation returns a list of videos S with the best relevance to Q, i.e., for any $V_1 \in S$ and $V_2 \notin S$, $Sim(Q, V_1) \geq Sim(Q, V_2)$ holds. The proposed system framework exploiting multiple features for social video recommendation is shown in Figure 1.

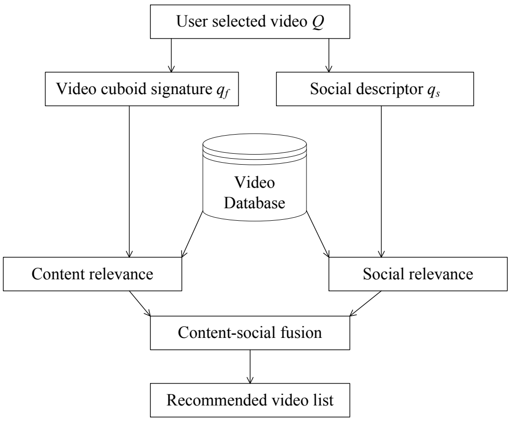


Figure 1: Framework of our social recommendation

Our social video recommendation system consists of three important parts. First, each video is represented using a compact signature to perform the content based relevance identification. Then, the social context of each video is described with a set of social users for social relevance identification. After that, the social relevance and content relevance are fused for the final video recommendation in sharing community. Finally, based on our multi-feature relevance model, we further design a sub-community-based approximation relevance scheme, and a chained hashing scheme to improve the efficiency of our online video content-social based similarity recommendation.

4. MULTIPLE FEATURE-BASED RECOMMENDATION

Social videos are compositions of video sequences and social contexts, each delivers important information on the relevance of a selected clip. Therefore, multiple feature-based relevance is described by the combination of the relevance from all the features. We will detail the content and social relevance in this section.

4.1 Content Relevance

We aim to find the content relevance between a user selected clip and a video in a shared community. Because of the high complexity and huge amount of video data in sharing communities, the task of efficient content relevance identification has been challenging. To overcome the difficulties of this task, we need to find a number of video segments, and describe them compactly. Then, the similarity between two videos is decided based on the representations of their segments. We use the existing techniques, and focus on how to select a suitable approach for our purpose.

We identify content relevance of user selected clips based on video segments. Given a video, we exploit the state-of-the-art shot detection technique proposed in [18] to detect a number of cuts. A series of segments are then obtained by extracting the subsequences between adjacent cuts. These segments can be further represented for video matching using a good representation model. A typical representation is to describe each segment as its keyframe that is described by its global visual features [13, 38], local interest points [31, 37], or compact signatures [14, 40, 32, 35, 12, 36]. However, local interest points based approaches incur high computation cost, while using global visual features, such as color histograms, is neither discriminative enough nor robust to photometric variation and encoding methods. Signature based methods overcome the weakness of local feature-based and global feature-based methods, thus more applicable to content relevance identification.

Conventional video signatures include the ordinal signature, color shift signature, cut signature, local intensity shift signature and video cuboid signature etc. Among them, the ordinal signature is not robust to the frame editing in videos, while the global transformation of videos is well handled by it. The color shift signature is robust to different video transformation and frame editing operations, but not discriminative enough. Though the local intensity shift signature takes the advantages of both ordinal signature and color shift signature, it can not handle the content shift within frames. Video cuboid signature is a more advanced signature comparing with other signature representation models. It captures the spatial and temporal information of videos, and is robust to the content shift with the support of earth mover's distance over signatures. Considering the advantages of video cuboid signatures, we apply it to this work for the content relevance identification in social video recommendation. A video cuboid signature is constructed over a number of temporally consecutive keyframes, and consists of video cuboids that constitute spatially and temporally adjacent pixels. Given a video $q-gram$ consisting of q keyframes, its video cuboids are generated by first dividing each keyframe into a fixed number of equal-size blocks, and then merging the spatially adjacent similar blocks in a reference keyframe. Based on the variable-size blocks in the reference frame, video cuboids are produced by grouping the temporally adjacent blocks, and each is described as a pair (v, μ), where v is the average intensity change between temporally adjacent blocks and μ denotes its weight indicating the block size. To simplify the video cuboid signature, we use bigrams and each v is a single value.

The similarity between video cuboid signatures is measured by the EMD that permits the comparison between two signatures consisting of different numbers of video cuboids. The EMD of two signatures measures the minimal amount of work necessary to transform one signature into another. Formally, the EMD between two video cuboid signatures can be computed as follows [35].

Definition 1. Given two video cuboid signatures $C_1 = \{(v_{1i}, \mu_{1i})\}$ and $C_2 = \{(v_{2j}, \mu_{2j})\}$, $\forall 1 \leq i \leq |C_1| : \mu_{1i} > 0$ and $\forall 1 \leq j \leq |C_2| : \mu_{2j} > 0$ of normalized total mass $\sum_{i=1}^{|C_1|} \mu_{1i} = \sum_{j=1}^{|C_2|} \mu_{2j} = 1$, and c_{ij} the cost to transform a video cuboid unit $v_{1i} \in C_1$ to $v_{2j} \in C_2$, The EMD between them is defined as a minimization over all possible flows $F = [f_{ij}]$ under positivity constraints CPos, source constraints CSource and target constraints CTarget:

$$EMD_c(C_1, C_2) = \min_F \{ \sum_{i=1}^{|C_1|} \sum_{j=1}^{|C_2|} c_{ij} f_{ij} | Constraints \} \qquad (1)$$

with $Constraints = CPos \wedge CSource \wedge CTarget$:

$$\begin{array}{l} CPos : \forall 1 \leq i \leq |C_1|, 1 \leq j \leq |C_2| : f_{ij} \geq 0 \\ CSource : \forall 1 \leq i \leq |C_1| : \Sigma_{j=1}^{|C_2|} f_{ij} = \mu_{1i} \\ CTarget : \forall 1 \leq j \leq |C_2| : \Sigma_{i=1}^{|C_1|} f_{ij} = \mu_{2j} \end{array} \qquad (2)$$

where c_{ij} defines how much dissimilarity one unit of flow from video cuboid v_{1i} to v_{2j} induces, f_{ij} is the fraction of flow units between v_{1i} to v_{2j} that minimizes the total dissimilarity, $CPos$ rules out negative flow, $CSource$ ensures that the total flow from cluster (v_{1i}, μ_{1i}) of C_1 is equal to (v_{2j}, μ_{2j}) and $CTarget$ restricts the total flow to (v_{2j}, μ_{2j}) of C_2. The similarity between C_1 and C_2 is derived from the EMD between them, and is computed by [35].

$$SimC(C_1, C_2) = \frac{1}{1 + EMD_c(C_1, C_2)} \tag{3}$$

Given two signature series, S_1 and S_2, the similarity between them is computed by an extended Jaccard similarity which incorporates the similarity between matched video cuboid signatures.

$$\kappa J(S_1, S_2) = \frac{\sum_{C_i \in S_1, C_j \in S_2} SimC(C_i, C_j)}{|S_1 \cup S_2|} \tag{4}$$

Video cuboid representation model has several advantages when processing short video segments. It captures the spatial and temporal information of a video segment locally by the intensity change of frame blocks over time. Moreover, since the intensity changes over time are invariant to global video transformations and the EMD based measure is robust to frame editing, video cuboid signature model is effective for processing our video segments with various transformation and editing operations. Different from global visual features that totally ignore the local information of frames, video cuboid signatures are extracted from keyframe blocks locally, thus more differentiable. Meanwhile, unlike local interest points that consider local information at a finer level, video cuboid signatures capture the local information of video segments at a coarser level, thus more robust to the feature variations over streams. Therefore, we select video cuboid signature together with its EMD based signature measure for video segment matching. Applying video cuboid signature model on the video segments, we can transform each video into a signature sequence for matching. To improve the efficiency of video signature sequence matching, we also apply the locality sensitive hashing based optimization strategy used in [35] and exploit LSB-index structure which is a B^+-tree-based hash index proposed in [28] for Z-order values of hash keys, to reduce the number of EMD-based signature measures.

4.2 Social Relevance

In this section, we first propose our social modeling together with the optimization schemes including the SAR and chained hashing schemes, for improving the social relevance identification cost. We then discuss the maintenance of sub-communities under social updates in dynamic environment.

4.2.1 Social Modeling

Using video cuboid signature model, the content similarity between videos can be captured to find the matched ones in content, identifying their content relevance. However, video recommendation applications concerns not only the content relevance of clips. There are still some videos which are relevant to the user selected clip, but unmatched to it in content, thus not searchable using content relevance. This has raised new challenge for effective social video recommendation. Fortunately, the interaction of social users on video data in shared communities provides rich social information that can be exploited to identify the relevant but unmatched clips from the aspect of social relevance.

Intuitively, when registered social users review some videos in shared communities, they tend to comment on those interesting ones to them. Meanwhile, a user is usually interested in certain types of media data. While users commenting to a single video have common interests to some extent, two videos commented by a group of common users are usually relevant to each other from some aspects. Thus, exploiting the social interaction of users on videos is a promising way of solving the social relevance problem in video sharing communities. We describe the social user interaction on a video as a social descriptor. Given a video V, its social descriptor is constructed by obtaining a set including its owner user and those users commenting it. The social descriptor of the video V is represented as a set of user ids, i.e. $D_V = \{id_{Vi}\}$, where id_{Vi} is the id of a user commenting V. We identify the social relevance between two videos based on the Jaccard similarity coefficient. Suppose that D_V and D_Q are the social descriptors of videos V and Q respectively, their social relevance is computed by the common users shared by their descriptors, which is defined as:

$$sJ = \frac{|D_V \cap D_Q|}{|D_V \cup D_Q|} \tag{5}$$

The social relevance reflects the number of common users interested in two compared videos. Practically, in video sharing communities, the number of comments to a single video is usually very large, especially for some popular videos. When using sJ measure for social relevance calculation, we encounter two challenges. First, the computation complexity of the measure is quadratic to the number of elements in two compared social descriptors. Second, the number of elements for comparison to the compared videos is usually very large, usually several hundreds to tens thousands, thus the computation cost for social relevance identification becomes prohibitively expensive. Next, we will estimate the sJ calculation by proposing two schemes, the SAR and chained hashing index, for fast social relevance identification, and discuss on how we handle social updates in dynamic environment.

4.2.2 Complexity Reduction

As discussed previously, in real applications, it is vital to reduce the complexity of social relevance identification for online video recommendation. We propose a scheme called sub-community-based approximation relevance (SAR) to reduce the computation complexity and the number of elements for comparison. The basic idea of SAR is to transform the user comparison in sJ calculation into the comparison over a sub-community level. SAR approximates the social relevance computation by three major steps: *sub-community extraction*, *social descriptor vectorization*, and *social relevance approximation*. The social users are first grouped into a small number of sub-communities based on their interests on videos in the whole collection. Then the social descriptor of each video is converted into a vector of user histogram over a small number of sub-communities. Finally the sJ between two descriptors is approximated based on the difference of their user histograms.

Sub-Community Extraction We propose a graph partition approach to discover a number of sub-communities related to the social users of the video collection. A user interest graph (UIG) is first constructed over all users of a video collection. It consists of a set of nodes that represent the social users of a video collection and are linked by a number of weighted edges. The weight of an edge linking two users denotes the number of common interested videos shared by them. Given a collection of 8 videos, $V_1, ..., V_8$, and 5 users, $u_1, ..., u_5$, suppose that we have their interest relationship as follows: $(u_1, < V_1, V_3, V_8 >)$, $(u_2, < V_3, V_8 >)$, $(u_3, < V_2, V_4, V_5 >)$, $(u_4, < V_1, V_4, V_5 >)$, $(u_5, < V_4, V_5, V_6, V_7 >)$, Figure 2 shows the UIG of the users of this collection.

Once the user interest graph is constructed over a set of social users, we can discover a number of sub-communities by graph partition. There are several existing algorithms, such as balanced min-cut and spectral clustering [30], for clustering graph nodes. The spectral clustering is a more advanced algorithm comparing with balanced min-cut clustering. However, the spectral clustering does

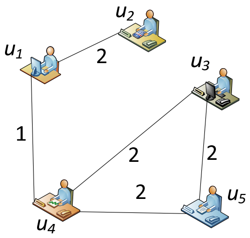


Figure 2: An example of user interest graph

```
Procedure SubgraphExtraction.
input:  $G_I$ - UIG graph        $k$ - Number of sub-communities
output: $S_\mathcal{G}$ - Set of sub-communities
1. Extract components disconnected with each other, put them in $S_\mathcal{G}$.
2. Let $p(G_I)$ be the number of connected components in $G_I$
3. while $p(G_I) < k$
4.     $e$ ←FindLightestEdge($S_\mathcal{G}$)
5.     remove $e$ from $S_\mathcal{G}$
6.     Let $U_i$ and $U_j$ be two users connecting $e$
7.     if $U_i$ and $U_j$ are disconnected
8.         $p(G_I) \leftarrow p(G_I) + 1$
9. Put all connected components of $G_I$ into $S_\mathcal{G}$
10. return $S_\mathcal{G}$
```

Figure 3: Extracting subgraphs.

not perform very well, because of the information loss in dimensionality reduction over very large number of social users. We propose a new approach for sub-community extraction. Figure 3 shows the detailed algorithm for extracting k sub-communities from UIG. Given a UIG graph, and the number of sub-communities k, our algorithm performs in two steps. In the first step, we extract the originally separated components, each corresponds to a sub-community, and put these sub-communities from the original UIG graph into the sub-community set (line 1). In the second step, we recursively remove the edges with lowest weights from the user interest graph until k disconnected subgraphs are obtained (lines 2-9). As the distribution of user connections varies in the user interest graph, we permit the sub-communities to be of different sizes, so the users in a sub-community can be highly similar and high effectiveness can be achieved in recommendation. The set containing k subgraphs is finally returned after the extraction process stops (line 10). Each subgraph is a connected component in the whole user interest graph, and corresponds to a sub-community over the whole user space. We set the subgraph number k as a parameter for evaluation, which balances the sub-community size given a social dataset and fixes the dimensionality in social descriptor vectorization.

We evaluate the superiority of our algorithm over the best practice, the spectral clustering, using a standard metric called *Silhouette Coefficient*, where a bigger value indicates a better overall clustering result [10]. We randomly select 2000 video samples from our whole dataset, and cluster their users using our subgraph extraction approach and spectral clustering algorithm. We measure the average *Silhouette Coefficient* of the partition results produced by our approach and that produced by spectral clustering. The average *Silhouette Coefficient* of our results is 0.498, while that of spectral clustering is only 0.242. This has proved that our algorithm produces better clustering results.

Social Descriptor Vectorization After extracting k sub-communities by graph partition, we map the whole user space into a k-dimensional sub-community space. Users in different sub-communities are stored in a dictionary for later social descriptor vectorization. Using a k-dimensional dictionary, a social descriptor of n users $< u_1, ..., u_n >$ can be converted into a k-dimensional vector $< d_1, ..., d_k >$ by simply counting the number of users in each sub-community.

Social Relevance Approximation The social relevance can be approximated by comparing the social descriptor vectors of different videos. Given two videos Q and V, suppose that their social descriptor vectors are $< d_{Q1}, ...d_{Qk} >$ and $d_{V1}, ..., d_{Vk}$ respectively, the social relevance between them can be approximated as follows.

$$\tilde{s}J = \frac{\sum_{i=1}^{k} \min(d_{Qi}, d_{Vi})}{\sum_{i=1}^{k} \max(d_{Qi}, d_{Vi})} \qquad (6)$$

Using social relevance approximation, the computation complexity of social relevance identification is linear.

4.2.3 Social Relevance Optimization

We use hash table to organize the sub-communities for improving the mapping from a social user to its sub-community, thus further improve the efficiency of social relevance identification. To effectively index the social users in the sub-communities using hash, we need to create a hash structure and a class of hash functions that map each hash key into a hash code. We use chained hash tables to organize the social users because of its simplicity and flexibility on the element number in them. For hashing function selection, since the class of *shift-add-xor* string hashing functions satisfy four important properties including uniformity, universality, applicability and efficiency, and has been proved to be appropriate for practical applications [21], we use it for mapping social users to hash codes. Let $s = c_1, ...c_m$ be a string of m characters, v a seed and h_i an intermediate hash value after examination of i characters. The components in the class of *shift-add-xor* are defined as:

$$\begin{array}{rcll} init(v) & = & v & (a) \\ step(i, h, c) & = & h \bigoplus (\mathcal{L}_L(h) + \mathcal{R}_R(h) + c) & (b) \\ final(h, v) & = & h || T & (c) \end{array} \qquad (7)$$

Here, $\mathcal{L}_L(h)$ denotes the left-shift of value h by L bits, $\mathcal{R}_R(h)$ is the right-shift of value h by L bits. Given a social user name, its hash code is computed by first generating an initial hash code using the equation 7 (a), then recursively computing the intermediate hash code over its first i characters using the equation 7 (b), and finally yielding the modulo value of the hash code over its m characters. For a given video collection, the social users can be organized as a chained hash table containing a list of hash buckets. Each element of the hash table is a triad formed as $< key, cno, nextptr >$, where key denotes the social user name, cno refers to the sub-community id of the key, and $nextptr$ is the pointer to the next element having the same hash code. Given a user in a collection, we first generate its hash code, by which its hash bucket is located. The triad of the user is then inserted at the head of this appropriate bucket. A chained hash table is formed by inserting the triads of all the social users in the data collection. Figure 4 shows a chained hash table of our social data collection.

Given a user selected social video, we first capture all the users interested in it. For each social user, we map it to a sub-community id by first mapping it to a hash bucket, and then comparing its user name with all users in this bucket. Using our chained hashing index structure, the mapping from a social user name to its sub-community id is quickly performed.

We estimate the time complexity for social descriptor vectorization based on approximation with the support of our hash structure. The time cost for the vectorization of a social descriptor is decided by the number of its social users, and the number of string comparison for users with hash collisions. Given a social descriptor containing n social users, suppose the average number of collisions for each social user is η, the cost of each string comparison is a constant β. Then the time complexity for vectorization is $n * \eta * \beta$. We analyze the space complexity of our optimization strategies. The space

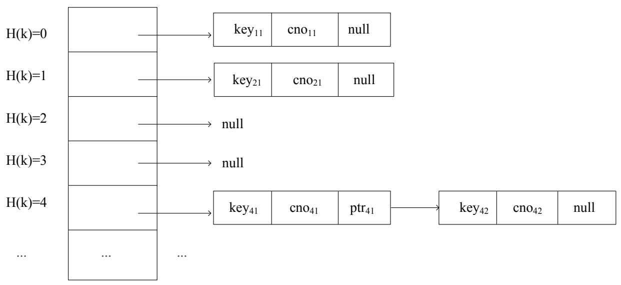


Figure 4: The chained hash table structure

complexity for our social descriptor vectorization and optimization is determined by the size of video database and the number of social users in the database. Suppose that the database size is C_d, and the number of social users in database is C_u, let k be the number of sub-communities, then the space required to store the hash-tree and social descriptor vectors would be $O(C_d * k + C_u)$.

4.2.4 Social Updates Maintenance

This section discuss how to deal with the dynamic issues in video sharing communities. Practically, sharing communities are highly dynamic and social connections change frequently. Considering our social relevance model, the social connections between users are often updated as well. On the one hand, when new comments come, new user connections are built. On the other hand, as the interests of people may change over time, users in a sub-community may not comment the same types of videos any more after a time period. Accordingly, existing user connections may become invalid. We are concerned about how sub-communities are updated to reflect the most recent user connections, how the chained hash table is updated according to the sub-community changes, and how the social descriptor of a video is updated.

We maintain the sub-communities periodically by checking the status of each sub-community and its interactions with other sub-communities. Figure 5 shows the algorithm details for handling the social updates. Given a set of new social connections in the recent time period, our algorithm performs the maintenance mainly in three steps: (1) find recently formed bigger new sub-communities by checking if strong connections have been formed in the new time period, and union the multiple sub-communities if necessary (lines 1-13); (2) find sub-communities which can be split into multiple new sub-communities (lines 14-18); (3) update the index structure (lines 9, 19) and user descriptor vectors (lines 10, 20). We search the hash index, and find the biggest edge weight connecting two sub-communities and that belonging to each single sub-community respectively. Let the lightest edge of the connection in original sub-communities be w. If an edge between two sub-communities is bigger than w, we merge these two sub-communities, and update the chained hash table by replacing the ids of the two original sub-communities with a single new id. For a single sub-community, if the biggest edge between its users over the new social connection set is smaller than w, this sub-community will be further divided into two subgraphs. Once the sub-communities change, the social descriptors of interested videos are updated over the updated dimensions accordingly. As such, the sub-communities are well maintained to better reflect the recent social updates.

4.2.5 Social Update Cost Analysis

We estimate the cost of social update in our proposed algorithm. Let $E = \{e_i\}$ be a set of new connections. Denote the cardinality of the connection set as $|E|$. Suppose that we have the sub-community set undergoing union operation $\{g_{ui}\}$ and the size of its sub-community $|g_{ui}|$, the number of videos to g_{ui} denoted as $\mathcal{N}_{ui}$, the sub-community set undergoing split operation $\{g_{si}\}$ and the size of its sub-community $|g_{si}|$, the number of videos to g_{si} denoted as $\mathcal{N}_{si}$. The social update cost T_{mc} can be estimated as:

```
Procedure SocialUpdatesMaintenance.
input:  $\{e_i\}$ - A set of connections in recent time period
        $S_\mathcal{G}$ - Set of sub-communities
        $w$ - The lightest edge weight in the sub-communities
output: $k$ updated sub-communities
1. Let $U_i$ and $U_j$ be two users connected by $e_i$
2. for each $e_i \in \{e_i\}$,
3.     $SearchIndex(\{e_i\})$
4.     $id_i \leftarrow MapUser2subCommunity(U_i)$
5.     $id_j \leftarrow MapUser2subCommunity(U_j)$
6.     if the weight of $e_i$ is bigger than $w$
7.         if $id_i$ is not equal to $id_j$ /*not in the same sub-community*/
8.             $UnionSubCommunities(id_i, id_j, e_i)$
9.             $UpdateIndex(id_i, id_j)$
10.            $UpdateUserDescriptor(id_j, id_j)$
11.            $SetStatusofSubCommunityAsSplit(id_i)$
12.        else if $id_i$ is equal to $id_j$   /* in the same sub-community*/
13.            $SetStatusofSubCommunityAsSplit(id_i)$
14. while $|S_\mathcal{G}| < k$
15. For all sub-commpunities with Split status
16.     $id_i \leftarrow Getsub-communitywithLightestEdge$
17.     $Splitsub-community(id_i)$
18.     $S_\mathcal{G} \leftarrow PutNewsubCommunities(id_i)$
19.     $UpdateIndex(id_i)$
20.     $UpdateUserDescriptor(id_i, id_j)$
21. return $S_\mathcal{G}$
```

Figure 5: Maintaining social updates.

$$|E|*c_h+\sum_{i=1}^{|\{g_{ui}\}|}(|g_{ui}|*t_1+\mathcal{N}_{ui}*t_2)+\sum_{i=1}^{|\{g_{si}\}|}(|g_{si}|*(t_1+t_3)+\mathcal{N}_{si}*t_2) \qquad (8)$$

where c_h, t_1, t_2, t_3 are constant, which denote the cost used for the mapping from a social user name to its sub-community id, the index update on a sub-community element, the user descriptor update on a dimension, and the element checking in sub-community partition, respectively. Clearly, the cost of social updates maintenance is linear to the cardinality of the new connection set, those of the sub-communities with union and split operations, and the number of videos involved in social updates. In addition, as we adopt incremental updating strategy in our algorithm, the maintenance operation is only performed over the sub-communities involved in update operations and their corresponding videos, which are much smaller comparing with the whole video data collection. Therefore, the cost of social updates can be well controlled.

4.3 Content-Social Fusion

Once we define the content relevance and social relevance of videos, we can fuse them using a good integration function to perform effective recommendation in video sharing communities. Borrowing the idea of search fusion in [26], the relevance fusion can be simply performed by taking the average of the content relevance and social relevance, or retaining the higher relevance score between them. However, while the former approach ignores the importance difference of them in recommendation, the later one completely ignores one of them in relevance identification. As the content relevance and social relevance may contribute to the final recommendation to different extent, we need to find out the best parameter that leads to effective video recommendation.

In this work, we care about the video ranking based on their final relevance in recommendation, while the absolute relevance score of each video is ignored. Given two videos V and Q, let S_V and

Table 2: 5 queries collected from Youtube

Query id	Query description
q_1	youtube
q_2	mariah carey
q_3	miley cyrus
q_4	american idol
q_5	wwe

S_Q be their video cuboid signature series, and D_V and D_Q be their social descriptors, the overall relevance between them is defined as:

$$FJ(V,Q) = (1-\omega)\kappa J(S_V, S_Q) + \omega sJ(D_V, D_Q) \quad (9)$$

where ω is the parameter adjusting the weight of the content relevance and that of social relevance in the final relevance function. FJ fuses the content and social relevance, and takes the difference of their contributions in recommendation. We will discuss the selection of an optimal ω in the experimental study.

4.4 KNN Search

With FJ relevance function, we can identify the top score videos from a given data set, which considers both the video content and social information. To quickly identify the social relevance, we use k inverted files, each of which stores a sub-community id and a list of its corresponding videos $\{v_i\}$. For content relevance, we combine the locality sensitive hashing based technique in [35] and the LSB-index that is a B^+-tree structure in [28], to identify the nearest video cuboids of each signature in a query video. We use the existing index structure for content relevance, and extend it for multiple video cuobids and embed social relevance in KNN search. Specifically, we embed EMD-metric into L_1-norm space like [35], and use LSB-index to index Z-order values of points obtained by hash conversion as in [28]. Meanwhile, we record the video id of each data entry together with its video cuboid signature in our application. Similar to [28], we perform KNN search by continuously finding the next longest common prefix with the query. Unlike the KNN search which processes a single point query in [28], we handle a set of signatures in a query video at the same time to obtain their overall content similarity. Figure 6 shows the detailed algorithm for computing K top score videos.

Given a query video Q, our algorithm performs KNN search by three important steps. (1) Find a list video candidates based on social relevance by vectorizing query social descriptor, obtaining the video candidates related to the query social descriptor and ranking these video candidates (lines 1-3); (2) Obtain a set of video candidates based on the content relevance by searching the next longest common prefix with all query signatures (lines 5-6); (3) Refine the video candidates by first obtaining the most relevant candidate set, and then calculating the overall relevance between the query and each candidate based on FJ function (lines 7-9). The KNN_list is updated if the relevance score of the current candidate is bigger than that of its top K relevant video (line 10). The algorithm recursively performs the operations in steps (2) and (3), until all K top score videos are found (lines 4-11). The final search results are returned to the users (line 12).

5. EXPERIMENTAL EVALUATION

We demonstrate the high effectiveness and efficiency of our proposed content-social based approach for video recommendation in sharing communities.

5.1 Experimental Setup

We conduct the experiments on a 200-hour video collection that is collected by crawling Youtube website based on the recent popular queries [1]. We selected 5 most popular youtube queries listed in Table 2 to retrieve the top favorite videos of each query from Youtube. Following [31], we only kept the short clips with time duration no more than 10 minutes. For each video, we kept its context information including its owner user and the users who comment it as well. Following [33], for each query, we select the top two videos as the source videos and get 10 in total for recommendation.

```
Procedure $K$TopScoreVideoSearch.
input:  $\mathcal{H}$ - A chained hash table     $I$-Inverted file
        $\mathcal{LSB}$ - An LSB-tree           $Q :< Q_f, Q_s >$-A query video
        $K$ - Number of top score videos
output: $KNN\_list$ - A list of $K$ top score videos
1. $\{d_{Qi}\} \leftarrow$ SocialDescriptorVectorization($\mathcal{H}, Q_s$)
2. $\{V_i\} \leftarrow$ GetSocialRelevanceCandidates($I, \{d_{Qi}\}$)
3. $\{V_{ri}\} \leftarrow$ RankRelevanceCandidates($\{V_i\}$)
4. Repeat /*SearchLSB($\mathcal{LSB}, Q_f$)*/
5.     for each $Q_{fi} \in Q_f$, pick the leaf entry, $e_{fi}$, from $\mathcal{LSB}$ having
        the next longest common prefix with $Q_{fi}$
6.     $\{V_{fi}\} \leftarrow$ GetVideoCandidates($\{e_{fi}\}$)
7.     $V_n \leftarrow$ GetNextMostRelevantVideo($\{V_{ri}\}$)
8.     for each $V_i \in \{v_{fi}\} \bigcup \{V_n\}$
9.         ComputeFJ
10.        UpdateKNN_list
11. Until $K$ top score videos are found
12. return $KNN\_list$
```

Figure 6: Computing K top score videos.

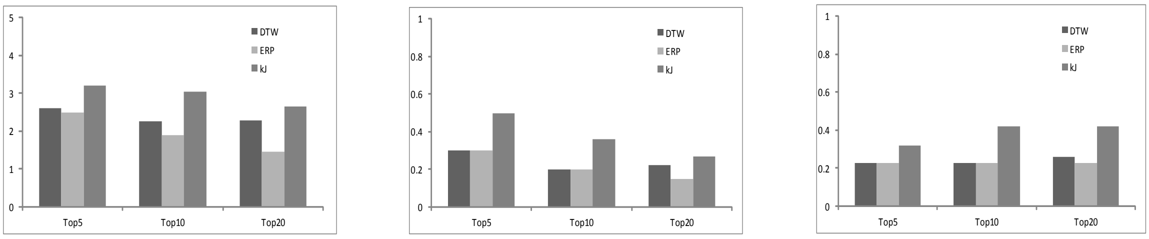
We conducted a subjective user study. 10 evaluators majored in computer science, including eight graduate students and two undergraduate students, participated in the user study. Each individual is given the recommended videos returned by the three approaches in a random order. After viewing these videos, they were asked to give a rating score from 1 to 5 indicating whether the recommended videos are relevant to current source video. Here, higher score indicates more relevance.

5.2 Evaluation Methodology

We evaluate our proposed social video recommendation approach in terms of effectiveness and efficiency. First, we evaluate the effect of different content distance selection, the effect of two parameters, the relevance weighting ω and the number of sub-communities k, to obtain their optimal values. Then the effectiveness and efficiency of our recommendation approach are evaluated using the optimal parameter settings. For each source video, we use the following four schemes to recommend different video lists, and compare our two proposed alternatives,SR and CSF, and two existing video recommendation approaches, AFFRF and CR.

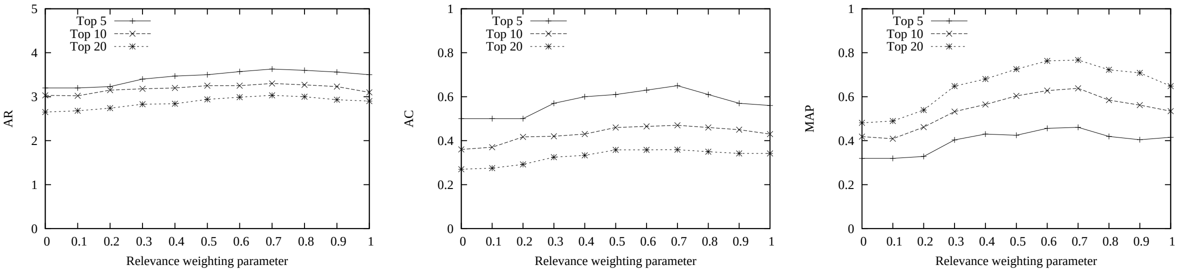
- AFFRF: Using text, visual, aural and relevance feedback for recommendation [33].
- CR: Using content relevance for recommendation [35].
- SR: Using social relevance for recommendation.
- CSF: Using content-social fusion for recommendation.

To evaluate the effectiveness of our recommendation approach, we use three metrics in [33], the average rating score (AR), average accuracy (AC) and mean average precision (MAP) of top 5, 10, and 20 recommended videos, as the measurements. Here, AC is defined as the proportions of videos with the rating bigger than 4 to all recommended videos. The mean average precision is a standard TRECVID metric, which is computed by the mean of non-interpolated average precisions (AP) of all queries [25]. Let n be the number of retrieved videos, r_i the rating score of the i^{th} returned video, N the number of retrieved videos with rating score



(a) AR (b) AC (c) MAP

Figure 7: Effect of content relevance measures



(a) AR (b) AC (c) MAP

Figure 8: Effect of ω

bigger than 4. The formal equations for computing these three metrics are as below.

$$AR = \frac{\sum_{i=0}^{n} r_i}{n} \quad (a) \qquad AC = \frac{N}{n} \quad (b) \qquad (10)$$

$$AP = \sum_{\gamma=1}^{n} (P(\gamma) * rel(\gamma)) \qquad (11)$$

$$MAP = \frac{\sum AP_{\vec{q}=1}^{\mathcal{Q}} AP(\vec{q})}{\mathcal{Q}} \qquad (12)$$

Here,γ is the rank, $rel()$is a binary function on the relevance of a given rank, $P()$ is the precision of the system at a given cut-off rank, $\mathcal{Q}$ is the number of queries.

We evaluate the efficiency of our approach over 200-hour real videos from Youtube in terms of time costs during the recommendation. We compare our SAR scheme, CSF-SAR, and our optimization using SAR and chained hashing, CSF-SAR-H, with the original content-social-based and the content-based recommendations. All the experiments were performed on Window 7 platform with Intel(R) Core(TM) i5-4570S CPU (2.9GHz)and 8GB RAM.

5.3 Effectiveness Evaluation

We first compare three existing content similarity measures for content relevance to select an optimal one as a base of our multi-feature based recommendation. Then we evaluate the effect of parameters by conducting content-social-based video recommendation. After that, we compare our proposed approach with the state-of-art video recommendation approaches by performing the video recommendation over the collected Youtube video dataset. Finally, we prove the scalability of our approach to social updates.

5.3.1 Effect of content relevance measures

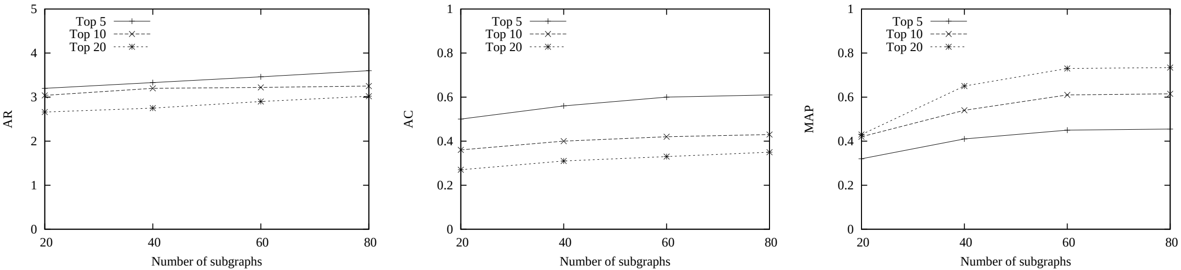
We conduct experiments to test the effectiveness of the system using three typical content similarity measures including (1) ERP [5]; (2) DTW [7]; and (3) κJ [35], for selecting an optimal content relevance measure in our content-social fusion modal. Figures 7 (a)-(c) show the effectiveness comparison of three similarity measures in terms of average rating score, average accuracy and mean average precision respectively.

As we can see, κJ obtains the best effectiveness among three content similarity measures. This is because κJ performs measure in a more flexible manner comparing with ERP and DTW, thus better captures content relevance. For one thing, ERP and DTW perform video similarity measure by considering the temporal order of the whole sequence, similar videos may not be identified because of temporal sequence editing. For another, κJ considers only the temporal order in each video segment, which captures the temporal information of videos. At the same time, as an EMD-based measure, κJ allows the space shift of videos and does not care about the order between different segments, thus the video sequence and frame editing operations can be well handled. Based on the results, we select κJ as an optimal content relevance measure, which is the base of our multi-feature fusion model.

5.3.2 Effect of ω

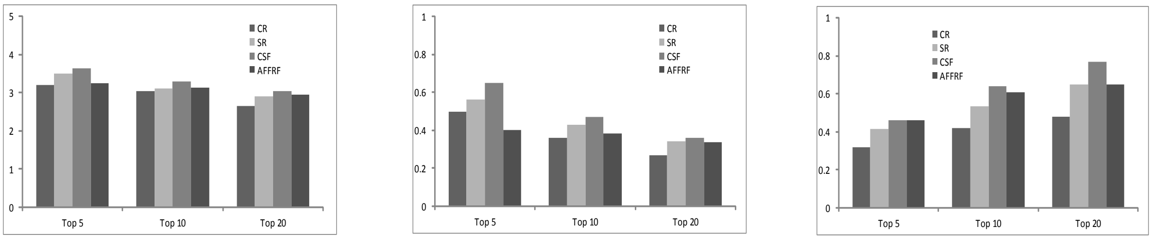
We evaluate the effect of the relevance weighting parameter, ω, on the average rating score, average accuracy and mean average precision using 10 source videos. We vary the value of ω from 0 to 1. For each ω, we recommend top 5, top 10 and top 20 respectively. Figures 8 (a)-(c) show the ω changes on average rating score, average accuracy and mean average precision of our system.

As we can see, with the increasing of ω, the average rating score, average accuracy and mean average precision of our recommendation increase gradually from 0 to 0.7 to different extent, and reach to their peak values. With the further increasing of ω after 0.7, the effectiveness of our recommendation system drops under all three metrics. This is caused by two reasons. On the one hand, when ω is set to a value between 0 to 0.7, a bigger ω ensures that more information on social user connection is captured, which enhances the effectiveness of our system accordingly. On the other hand, after the peak point 0.7, content information of videos reduces because of the increase of social information. As a group of social users may be interested in videos on multiple unrelated



(a) AR (b) AC (c) MAP

Figure 9: Effect of k



(a) AR (b) AC (c) MAP

Figure 10: Effectiveness comparison

topics, some videos with relevant content are replaced by those irrelevant ones with common social connections. Consequently, an ω bigger than 0.7 weakens the determinativeness of content relevance. Therefore we set ω to 0.7 as its default value to obtain an optimal effectiveness of the recommendation.

5.3.3 *Effect of* k

We evaluate the effect of the subgraph number in our SAR scheme, k, on the effectiveness of the system by applying our content-social recommendation. In this test, the k is varied from 20 to 80, and the optimal ω is applied.

Figures 9 (a)-(c) show the performance change trends of the average rating score, average and mean average precision of our approach. Clearly, with the increasing of the sub-community number in our SAR scheme, the effectiveness of our approach has been improved with the change of k from 20 to 60. The effectiveness on three metrics keeps steady when k is further increased from 60 to 80. This is mainly caused by the information loss of our SAR approximation scheme. When we transform the social user set matching in social relevance computation into the linear compact vector matching, we describe a group of interactive social users as their common community id, which is a user representation over a coarser level. A smaller k value produces coarser social user presentations, leading to more serious social information loss. With the increasing of k value from 20 to 60, the social information loss decreases, thus better effectiveness can be obtained at a bigger k value point. When k is changed from 60 to 80, most removed social connections are redundancy, which does not affect the effectiveness of our approach. Thus, considering a good balance of effectiveness and efficiency, the default value of k in our test is set to 60.

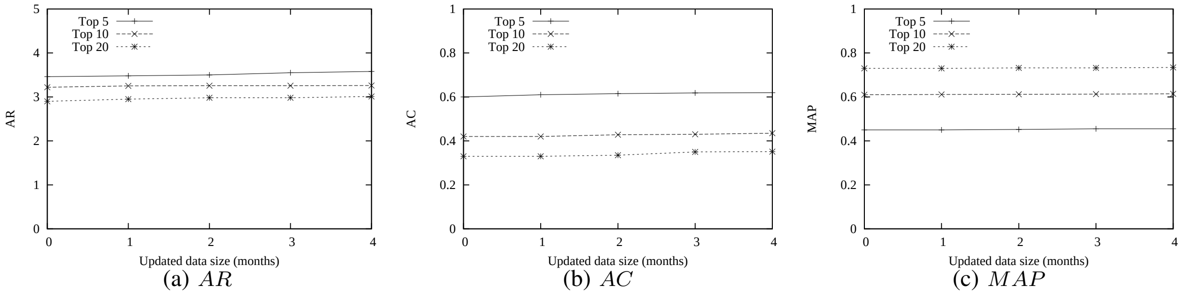
5.3.4 *Comparing Different Recommendations*

We conduct experiments to evaluate the effectiveness of four video recommendation approaches, including two proposed alternatives, SR and CSF, and the existing competitors for video detection and recommendation, CR and AFFRF. For our CSF, we set the parameters, ω and k, to their optimal values. Figures 10 (a)-(c) show the comparison of four approaches in terms of the average rating score, average accuracy, and mean average precision.

As we can see, our content-social based recommendation (CSF) achieves a better performance on all three metrics comparing with the other alternative, the social relevance (SR). This is because the CSF approach exploit the information from both the robust visual content and social user connections, which finds more relevant and rejects more irrelevant videos than other alternatives do in recommendation process. Comparing with two existing recommendation approaches, the content relevance (CR) and the multimodal with relevance feedback, our content-social based recommendation obtains much higher effectiveness because of the fully exploiting of social interaction information and robust video content in our approach. Although AFFRF uses multiple features in their recommendation, our content-social based recommendation still performs better in terms of effectiveness metrics. This is because of two factors. For one thing, videos are user uploaded data in Youtube, and a large portion of them have been edited or undergone different variations. The global video features like color histogram, and aural or text features are not fully reliable, which directly degrades the effectiveness of recommendation. For another thing, the social connections between users are not considered, thus some relevant videos with irrelevant content can not be identified. The high effectiveness of our content-social based recommendation has proved its superiority over other competitors.

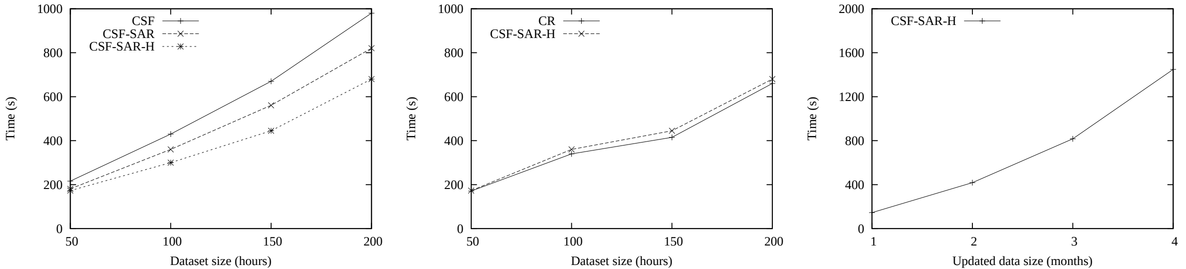
5.3.5 *Effect of Social Updates*

We test the effect of social updates on effectiveness of the video recommendation. We divide the whole social collections into two parts: (1) a test set containing connections in recent 4 months; and (2) a source set containing connections appearing in a year before the 4 months. We measure the effectiveness of our recommendation system under social update operations by fixing our source set and varying the test sets from 1 month to 4 months updates. Figures 11 (a)-(c) show the performance changes with respect to the size of social updates. As we can observe, with more social up-



(a) AR (b) AC (c) MAP

Figure 11: Effect of social updates



(a) $Effect\ of\ optimization$ (b) $Comparing\ recommendations$ (c) $Cost\ of\ updates\ maintenance$

Figure 12: Efficiency evaluation

dates, the effectiveness of our approach remains steady. This has demonstrated the good scalability of our approach over high dynamic social environment.

5.4 Efficiency Evaluation

We evaluate the efficiency of our recommendation approach by first testing the effect of our sub-community-based approximation relevance scheme SAR and the SAR with chained hashing scheme SAR-H, then comparing our approach with the existing content-based recommendation (CR) [35], and finally testing the cost of social updates. Since there is no any strategy proposed for improving the efficiency of AFFRF-based recommendation [33], we omit the efficiency comparison with this approach.

5.4.1 *Effect of Social Relevance Optimization*

We evaluate the effect of our SAR scheme and chained hashing scheme by varying the video dataset size from 50 to 200 hours, and reporting the average time cost of video recommendation using different optimization approach: (1)CSF; (2) CSF-SAR; and (3) CSF-SAR-H, for each dataset size. Figure 12 (a) shows the time cost change trends of three different approaches.

As we can see, CSF-SAR-H performs best, followed by CSF-SAR. The original content-social based recommendation incurs highest time cost. With SAR, the recommendation cost has been reduced significantly comparing with the original content-social based recommendation. This is caused by two factor. For one thing, SAR scheme transforms the original string set to a video into a single vector, which greatly compresses the video representation. For another thing, while the original sJ computation suffers from the high computation complexity, which is exponential to the user set sizes of videos, the time complexity of social information matching under SAR scheme is linear. CSF-SAR-H further reduces the mapping cost of SAR scheme by exploiting a chained hash structure.

5.4.2 *Comparing Different Recommendations*

We compare our content-social based video recommendation with the state-of-the-art technique in terms of the overall time cost. We test the time cost of the recommendation over 50 to 200 hours video datasets. Figure 12 (b) compares our CSF-SAR-H approach with the content based video recommendation (CR) in terms of the time cost used for recommendation.

As we can see, our CSF-SAR-H performs as good as CR in terms of efficiency, although huge amount of social information is embedded in the process of recommendation to improve the effectiveness. This is because CSF-SAR-H adopts SAR scheme and chained hash structure, which greatly reduces the time cost of the social relevance computation and the mapping from social user sets to sub-community id vectors. Comparing with the content relevance computation in CR, the time cost of social relevance computation can be neglected. Thus, our CSF-SAR-H achieves competitive efficiency performance, while the effectiveness is improved greatly as demonstrated in Section 5.3.4. Consequently, our approach greatly improves the overall performance of the recommendation system.

5.4.3 *Efficiency of Social Updates*

We test the cost of social updates using different sizes of updates. We use the social connections to our 200 hours video dataset collected based on the five most popular queries. We divide the whole social collections into two parts: (1) a test set containing connections in recent 4 months (Sept.2014-Dec.2014); and (2) a source set containing connections appearing in a year before the recent 4 months. We fix our source set, and vary the test sets from 1 month to 4 months updates. The time costs of social updates over different time periods are reported in figure 12 (c). As shown in the results, the time cost is only hundreds of seconds for maintaining social updates within 3 months, and about 1500s for 4 months. The results show that we can maintain the social updates efficiently as we adopt incremental maintenance strategy and hash scheme.

6. CONCLUSIONS

In this paper, we study the problem of video recommendation in sharing communities. First, we propose a novel multiple feature-based model for video relevance identification in recommendation.

Then, we propose a sub-community-based approximation relevance scheme for improving the social relevance computation. Finally, we propose to optimize the efficiency of our recommendation by designing a chained hash structure, and performing hash-based mapping from user sets to sub-community id vectors. In addition, the maintenance of sub-communities and our hash structure has been discussed for the dynamic environment with social updates. We have conducted extensive experiments to evaluate our proposed recommendation approach. The experimental results have proved that our proposed approach outperforms the existing methods in terms of the efficacy.

7. ACKNOWLEDGMENTS

The research presented in this paper has been supported via ARC projects DP140100841 and DP130102691, NSFC project 61332013, the Hong Kong SRFDP&RGC ERG Joint Research Scheme M-HKUST602/12, National Grand Fundamental Research 973 Program of China under Grant 2014CB340303 , Microsoft Research Asia Gift Grant and Google Faculty Award 2013.

8. REFERENCES

[1] https://spreadsheets.google.com/spreadsheet/pub?key =0am4ow7xs15awchn6z3hka0o4rzzzmktjskcwsfpnrgc& gid=1.

[2] https://www.elie.net/blog/privacy/19-of-users-use-their-browser-private-mode.

[3] http://www.marketingcharts.com/direct/online-viewers-prefer-socially-recommended-videos-21011.

[4] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly. Video suggestion and discovery for youtube: taking random walks through the view graph. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 895–904, 2008.

[5] L. Chen and R. Ng. On the marriage of edit distance and L_p norms.

[6] S.-C. S. Cheung and A. Zakhor. Efficient video similarity measurement with video signature. *IEEE Trans. Circuits Syst. Video Techn.*, 13(1):59–74, 2003.

[7] C.-Y. Chiu, C.-H. Li, H.-A. Wang, C.-S. Chen, and L.-F. Chien. A time warping based approach for video copy detection.

[8] C. Christakou and A. Stafylopatis. A hybrid movie recommender system based on neural networks. In *ISDA*, pages 500–505, 2005.

[9] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath. The youtube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 293–296. ACM, 2010.

[10] J. Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[11] F. Hopfgartner. Interactive news video recommendation: An example system.

[12] Z. Huang, H. T. Shen, J. Shao, B. Cui, and X. Zhou. Practical online near-duplicate subsequence detection for continuous video streams. *IEEE Transactions on Multimedia*, 12(5):386–398, 2010.

[13] Z. Huang, H. T. Shen, J. Shao, X. Zhou, and B. Cui. Bounded coordinate system indexing for real-time video clip search. *ACM Trans. Inf. Syst.*, 27(3), 2009.

[14] C. Kim and B. Vasudev. Spatiotemporal sequence matching for efficient video copy detection. *IEEE Trans. Circuits Syst. Video Techn.*, 15(1):127–132, 2005.

[15] J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujemaa. Robust voting algorithm based on labels of behavior for video copy detection. In *ACM Multimedia*, pages 835–844, 2006.

[16] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW*, pages 661–670, 2010.

[17] L. Li, D. Wang, T. Li, D. Knox, and B. Padmanabhan. SCENE: a scalable two-stage personalized news recommendation system. In *SIGIR*, pages 125–134, 2011.

[18] Z. Liu, E. Zavesky, D. Gibbon, B. Shahraray, and P. Haffner. At&t research at trecvid 2007. In *TRECVID*, 2007.

[19] H. Luo, J. Fan, and D. A. Keim. Personalized news video recommendation. In *Proceedings of the 16th ACM international conference on Multimedia*, MM '08, pages 1001–1002, 2008.

[20] J. Mayer and J. Mitchell. Third-party web tracking: Policy and technology. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 413–427, 2012.

[21] M. V. Ramakrishna and J. Zobel. Performance in practice of string hashing functions. In *DASFAA*, pages 215–224, 1997.

[22] S. Sedhain, S. Sanner, L. Xie, R. Kidd, K. Tran, and P. Christen. Social affinity filtering: recommendation through fine-grained analysis of user interactions and activities. In *Conference on Online Social Networks*, pages 51–62, 2013.

[23] L. Shang, L. Yang, F. Wang, K.-P. Chan, and X.-S. Hua. Real-time large scale near-duplicate web video retrieval. In *ACM Multimedia*, pages 531–540, 2010.

[24] H. T. Shen, B. C. Ooi, and X. Zhou. Towards effective indexing for very large video sequence database. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA, June 14-16, 2005*, pages 730–741, 2005.

[25] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR*, 2006.

[26] J. R. Smith, A. Jaimes, C.-Y. Lin, M. R. Naphade, A. Natsev, and B. L. Tseng. Interactive search fusion methods for video database retrieval. In *ICIP (1)*, pages 741–744, 2003.

[27] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong. Multiple feature hashing for real-time large scale near-duplicate video retrieval. In *ACM Multimedia*, pages 423–432, 2011.

[28] Y. Tao, K. Yi, C. Sheng, and P. Kalnis. Quality and efficiency in high dimensional nearest neighbor search. In *SIGMOD*, pages 563–576, 2009.

[29] M. van Setten, M. Veenstra, A. Nijholt, and B. van Dijk. Prediction strategies in a tv recommender system - method and experiments. In *ICWI*, pages 203–210, 2003.

[30] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[31] X. Wu, A. G. Hauptmann, and C.-W. Ngo. Practical elimination of near-duplicates from web video search. In *ACM Multimedia*, pages 218–227, 2007.

[32] Y. Yan, B. C. Ooi, and A. Zhou. Continuous content-based copy detection over streaming videos. In *ICDE*, pages 853–862, 2008.

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

[34] X. Zhao, J. Yuan, R. Hong, M. Wang, Z. Li, and T.-S. Chua. On video recommendation over social network. In *Advances in Multimedia Modeling*, volume 7131 of *Lecture Notes in Computer Science*, pages 149–160. Springer Berlin Heidelberg, 2012.

[35] X. Zhou and L. Chen. Monitoring near duplicates over video streams. In *ACM Multimedia*, pages 521–530, 2010.

[36] X. Zhou, L. Chen, and X. Zhou. Structure tensor series-based matching for near-duplicate video retrieval. In *ACM Multimedia*, pages 1057–1060, 2011.

[37] X. Zhou, X. Zhou, L. Chen, A. Bouguettaya, N. Xiao, and J. A. Taylor. An efficient near-duplicate video shot detection method using shot-based interest points. *IEEE Transactions on Multimedia*, 11(5):879–891, 2009.

[38] X. Zhou, X. Zhou, L. Chen, Y. Shu, A. Bouguettaya, and J. A. Taylor. Adaptive subspace symbolization for content-based video detection. *IEEE Trans. Knowl. Data Eng.*, 22(10):1372–1387, 2010.

[39] Q. Zhu, M.-L. Shyu, and H. Wang. Videotopic: Content-based video recommendation using a topic model. In *Multimedia (ISM), 2013 IEEE International Symposium on*, pages 219–222, 2013.

[40] J. Zobel and T. C. Hoad. Detection of video sequences using compact signatures. *ACM Trans. Inf. Syst.*, 24(1):1–50, 2006.