ENHANCING VIDEO RECOMMENDATION FOR YOUTUBE-LIKE SOCIAL MEDIA

by

Xiaoqiang Ma
B.Eng., Huazhong University of Science and Technology, 2010

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Summer 2012

APPROVAL

Name: Xiaoqiang Ma

Degree: Master of Science

Title of Thesis: Enhancing Video Recommendation for YouTube-like Social Media

Examining Committee: Dr. Greg Mori
Chair

Dr. Jiangchuan Liu, Senior Supervisor

Dr. Qianping Gu, Supervisor

Dr. Ivan V. Bajić, SFU Examiner

Date Approved: 23 August 2012

Abstract

YouTube-like video sharing sites (VSSes) have gained increasing popularity in recent years. Meanwhile, Facebook-like online social networks (OSNs), have seen their tremendous success in connecting people with common interest. These two new generation of networked services are now bridged in that many users of OSNs share video contents originating from VSSes with their friends. Through a long-term measurement, we show that friends have higher common interest and their sharing behaviors provide guidance for video recommendation. In this thesis, we take a first step toward learning OSN video sharing patterns for video recommendation. An auto-encoder model is developed to learn the social similarity of different videos. We therefore propose a similarity-based strategy to enhance video recommendation. Evaluation results demonstrate that this strategy can remarkably improve the precision and recall of recommendations, as compared to other widely adopted strategies without social information.

To my parents and my brother

"Success is not final, failure is not fatal: it is the courage to continue that counts."

— Winston Churchill (1874-1965)

Acknowledgments

I would like to express my deepest gratitude to my senior supervisor Dr. Jiangchuan Liu, from whom I learned a lot. His passion towards research has greatly inspired me. His support and encouragement are indispensable to the success of my master study.

I want to thank my supervisor, Dr. Qianping Gu, for reviewing my work and precious suggestions that helped me to improve my thesis. I am grateful to thank Dr. Ivan V. Bajić and Dr. Greg Mori, for serving in my examining committee.

I also want to thank my friends, Haiyang Wang, Feng Wang, Xu Cheng, Haitao Li and Fei Chen, for their kind help during my master study.

My sincerest gratitude goes to my parents and my brother for their love and support. They are always behind me and help me overcome difficulties in my life.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the past several years, the landscape of the Internet has been largely changed. Various kinds of new applications have emerged and attracted a lot of people. Compared with the Internet ten years ago, the networked services in this Web 2.0 era [40] focus more on user experience, user participation and interaction, and rich media such as videos, music, and photos. People are now an active part of the new ecosystem, rather than passively receiving information, as in the past. Optimized search engines can provide more accurate search results, significantly facilitating the acquisition of relevant information. Peer-to-peer systems relieve the burden of servers and utilize the network resources more efficiently. The scope of this thesis is narrowed down to two popular social media services, video sharing and online social networks.

In this chapter, we first introduce the background of social media including video sharing sites (VSSes) and online social networks (OSNs), and recommendation systems. The related works in the fields of social media and recommendation systems are also discussed. Then we present our motivation of this thesis. At last we give the organization of this thesis.

1.1 Background

1.1.1 Social Media

YouTube [12]-like video sharing sites (VSSes) are becoming increasingly popular among Internet users in this Web 2.0 era, thanks to the reduced cost of high speed Internet access and the prevalence of powerful smartphones. In traditional live streaming and VoD (Video on

Demand) services, the videos are provided by the content providers and users can only watch this limited range of videos. The emerging video sharing sites, represented by YouTube, however, enable people themselves to be directors and creators. These sites provide people a free stage to show their own products and a platform for advertising through embedding and sharing in other places. As such, there are countless videos uploaded to VSSes every day. People with any kind of personal interest can always find the videos they like.

This new mode of video services has achieved unprecedented success. The latest statistics [14, 13] show that YouTube has reached 4 billion video views per day and 1 hour of video is uploaded per second; the traffic from mobile devices tripled in 2011 and mobile devices contribute more than 10% of global views. The great success of YouTube has attracted a lot of content generators to upload their videos, partners to collaborate, and companies to advertise commercial products.

Meanwhile, we have witnessed the emergence and success of online social networks (OSNs), as represented by Facebook [4] and Twitter [10]. There are over 900 million active users for Facebook as of May 2012 [5]. Facebook provide users a totally new platform to connect with old friends as well as to make new friends. An important feature of OSNs is that people can share all types of contents, including articles, pictures, music and videos, to their friends. As such, these two new generation of services have been bridged. OSNs can provide users with popular multimedia contents, and thus users are inclined to stay longer on social networks. This benefits OSNs almost as a free lunch that music and videos are stored and maintained by VSSes. On the other hand, VSSes enjoy the boost in video views, which increases the revenue from advertisement. The latest statistics reveal that 40% of all YouTube views come from Facebook [26] and Twitter is also a major sharing platform of YouTube videos [14].

1.1.2 Recommendation Systems

Given the numerous different genres of videos in social media, how to discover the videos of personal interest and recommend them to users is an important task. The first solution is the video search for which users need to input the keywords they are interested in. This is one of the main sources of the video views [54]. Besides the video search, recommendation systems can proactively suggest videos that meet users' potential interest. A good recommendation system can accurately capture personal interest, attracting users to view more videos and thus increasing the chance to make profit through embedded commercial advertisements.

Recommendation systems have been widely used in E-commerce [46] such as eBay [3] and Amazon.com [1]. They provide recommendations based on personalized information so that the recommended items are likely to be of use to a user. A classic way of categorizing recommendation systems is described in [21] and [43]. The six different classes of recommendation approaches are as follows:

- **Collaborative**: The system recommends items based on other similar users' past records [45]. The similarity of two users is calculated based on the similarity of the users' past taste. Collaborative approach is considered to be the most popular and widely adopted technique in recommendation systems.

- **Content-based**: The system makes recommendations based on the user's records in the past [34]. The similarity is calculated on the features of items. For example, if two items are always bought together, or two videos are always watched one by one, by the same user, they would be quite similar to each other. The classification based on keywords and category can further assist to find items with similar content.

- **Demographic**: A demographic recommender provides recommendations according to a demographic profile of the user. For example, search engines may provide search results according to users' location and language. Or suggestions could be customized according to users' gender, age, and other factors.

- **Knowledge-based**: A knowledge-based recommender system suggests items based on users' needs and preferences. For example, people can include such information as vocation, favorite sports, skills, in their detailed account profiles. As such, the related items will be retrieved and recommended.

- **Community-based**: This kind of systems make recommendations by considering the preferences of users' friends. It is shown that people are more likely to adopt the recommendations from their friends than anonymous individuals [42]. In social media, users may have social relations with each other, e.g., first-order friends and second-order friends [52], and customized recommendations can be acquired by analyzing the preferences of these social friends.

- **Hybrid**: The above five approaches can be combined to achieve a hybrid system so that the advantages of some approaches can fix the disadvantages of others.

To evaluate a recommendation system is the next step after its design. There are three types of experiments that are widely used for evaluation [49].

- **Offline experiments**: An offline experiment is conducted on a pre-collected data set of users' activity records. Offline experiments are often adopted since they do not require interactions with real users and thus make our evaluation affordable in terms of cost. The downside of offline experiments is that only a limited range of evaluation metrics can be tested. Another requirement is that user behavior should be relatively stable with the introducing of recommendation systems. Offline experiments are usually conducted at the initial phase to justify the effectiveness of recommendation systems and tune the system parameters.

- **User studies**: User studies are a kind of subjective test, while offline experiments are objective. In a user study, a number of people are recruited to interact with the recommendation system. It can answer a wide range of questions from the system designers. The interaction between the user and the recommendation system, as well as the influence of the recommendations on the user can be tested. The downsides of user studies are also very obvious. First, it is expensive to recruit people to take this kind of studies, especially when the study takes a long time and needs a lot of people. So the experiment should be designed carefully to minimize the cost while the questions should be enough for a comprehensive analysis. Second, the test subjects selected should be representative to prevent the bias towards any specific group of people.

- **Online evaluation**: Online evaluation is performed to explore the interaction between real users and the recommendation system. More importantly, the influence of the system on users can be measured. Online evaluation is usually tested after extensive offline experiments and perhaps a user study so that the new features will not bring much negative user experience.

In our social media context, the goal of recommendation systems is to suggest the videos that are more likely to be viewed. Both OSNs and VSSes utilize recommendation systems, explicitly or implicitly. For non-registered users, VSSes present the most popular, recently added videos and also hot videos in each category on home pages. These are the top ranked items according to all users' activities. Moreover, if the user clicks to watch one video, VSSes will recommend the most related videos, in terms of content classification and viewing pattern. For registered users, personalized recommendations have been developed. For example, YouTube delivers a set of personalized videos to registered users based on their previous activities (watched, favorited, liked) on the YouTube site [27]. The videos related to the personal activities are selected and then ranked according to a linear combination of several metrics (video quality, user specificity and diversification). Finally, the top ranked video candidates are listed as *Recommended* to users.

For OSNs, Facebook has no specific recommendation system yet and it only displays friends' shared videos according to the timeline, shown as "News Feed". If two persons are good friends, and one of them happens to notice the shared video of the other one, s/he is likely to click the sharing link. Unfortunately many a time, the sharing information on OSNs is pushed down to the bottom of home page by other news feed, and thus would be never noticed. As such, an explicit recommendation system is needed to retrieve interesting sharing information which should consider both personal activities and friends' recommendations. For now, there is a first step attempt by YouTube that users can connect their social network accounts (Google Plus[6], Facebook and Twitter) to the YouTube accounts. However, it just retrieves aggregated video sharing information from social networks and simply arranges it according to the timeline, which is far from satisfying individual users.

1.2 Related Work

1.2.1 Social Media

The rapid development of Video Sharing Sites has attracted significant attention from research community. A lot of works [30, 24, 23] have been conducted towards understanding the properties of this new generation of application. Gill *et al.* [30] analyzed the YouTube traffic and presented such characteristics as usage patterns, file properties, popularity, and transfer behavior of YouTube. Cheng *et al.* [24] studied the statistics of YouTube videos and found that these videos are significantly different from conventional streaming videos, in terms of video length, access pattern, growth trend and active life span. It further showed that the links to related videos generated by uploaders' choices render clear small-world characteristics, which indicates that the videos have strong correlations with each other. Cha *et al.* [23] studied the video popularity characteristics of the YouTube videos, especially the popularity lifetime and the relationship between requests and video age. These representative works mainly explored the unique characteristics of the videos in VSSes that are different from those of traditional VoD/streaming videos.

Some other works take efforts to study the human factors in social media propagation. The human factors come from several sources. For example, the viewing activities lead to different popularity and life time of different videos; the sharing behavior that users can share the videos from VSSes on personal websites, blogs, and OSNs such as Facebook and Twitter; the response, such as featuring, like/dislike and comments, conducted by viewers. The characterization of video interactions through video responses is discussed in [18]. The effect of user sharing behavior is explored in [25]. External links that reference the video object outside the YouTube websites provide an important way of advertising the videos especially at their early stage. Rad *et al.* [16] studied the video propagation and popularity in YouTube based on two types of connections, i.e., friends and followers. Given that now a substantial portion of video views come from online social networks, the authors in [37] presented a detailed measurement study and analysis on video sharing in online social network. They also developed an effective model to simulate the video propagation process in online social networks. The authors in [52] used YouTube as a case study to explore to impact of social network structure on content propagation. Different from previous works that focused on the video properties in the stand-alone VSS environment, these works explore how the social relations influence the video propagation and popularity. They found that users' interest in the video and the duration of the interest are crucial factors to video's popularity. As a result, human factors play a more important role than the video content in the video propagation in OSNs. Besides videos, Guo *et al.* [31] also analyzed such kinds of user generated content in OSNs as blog, bookmark sharing, and question-answering, and they found that users' posting behavior has strong daily and weekly patterns and follows stretched exponential distributions, which is similar to the video sharing behavior.

1.2.2 Recommendation Systems

There are many existing studies on the design of online video recommendation system. Baluja *et al.* [17] presented a novel method based on the random walks of the entire user-video graph to provide personalized video suggestions for users. Zhou *et al.* [54] studied the effect of YouTube related video recommendation system on video views. They found that this recommendation is the main source of views for many YouTube videos and it helps to improve the diversity of video views. Hence, further enhancement to the current recommendation system is highly desirable. Davidson *et al.* [27] introduced the customized recommendation system of YouTube that takes personal activity into consideration. Park *et al.* [41] proposed a framework for recommending online videos by constructing user profiles as an aggregate of tag clouds and generating recommendations according to similar viewing patterns. Different from most random-walk models such as [17] and [51], the study from Zhu *et al.* [55] is one of the first papers in the machine learning literature to consider the problem of semi-supervised classifier design for recommendation. The difference between these works and ours is that they mainly focus on the recommendations in a stand-alone environment. The social relations of users are not considered. On the contrary, we have obtained measurement data from both video sharing site (Youku) and online social network (RenRen). In this thesis, videos' popularity and users' social relationship are considered jointly for designing a better recommendation system.

Many studies have also focused on the content recommendation in social network sites. For example, Bogers *et al.* [20] considered recommendation for social bookmarking websites. They just tested different fusion of classic recommendation approached introduced in Chapter 1 without considering social relations. Said *et al.* [44] discussed how social relations can affect user behaviors and similarities. They evaluated the impact of social graphs on user similarities in movie tastes and advocated to use social relations for recommendation. But they did not propose a detailed recommendation system. Pessemier *et al.* [28] developed a tag cloud based recommendation system for user-generated content which exploits social relations. This work actually considers the impact of other users in the YouTube-like social media, while our work considered the users in OSNs and distinguishes friends and non-friends. The most relevant works to ours are [35, 32, 39, 38]. All these works considered to utilize the social relations to calculate the rating and regarded friendship as a kind of trust. Yet our work emphasizes the impact of social relations on the propagations and popularity of videos. Our system recommends the similar videos in terms of the popularity distribution across social circles. Not only to increase the possibility that the recommended videos to be viewed, we also want to assist the propagation of videos.

1.3 Motivation

In this thesis, we for the first time combine the measurements from OSNs and VSSes together and explore the possible gain of a video recommendation system based on social relations.

To better understand the interaction between OSNs and VSSes, we closely cooperated with RenRen [9], the largest OSN in China, to analyze its server access logs. We also accordingly crawled the related data from Youku [11], the largest VSS in China. Starting from March 24th, 2011, both the video viewing and sharing behaviors are recorded over three months. Based on this measurement, we observe that social friends are more likely to have common interests and their sharing behaviors provide guidance to enhance video recommendation. A *social circle* [53] consisting of an OSN user and his/her social friends will manifest a collective preference over a set of videos and thus videos' popularity will be discrepant in different social circles. We examine the video similarity in terms of popularity distribution across social circles and develop an auto-encoder model to learn this similarity. A customized video recommendation system is further proposed to recommend the most similar videos based on viewers history video lists. Based on the collected data set, we conduct offline experiments to evaluate our system.

1.4 Thesis Organization

The rest of this thesis is organized as follows. In Chapter 2 we describe our measurement methodology and present our measurement results, which reveal the opportunity to design a social network based video recommendation system. In Chapter 3, we discuss how to train an autoencoder using our data set and select an appropriate model. We compute the video similarity based on the obtained autoencoder and propose an enhanced recommendation system in Chapter 4. The system is evaluated in Chapter 5. Finally, we give further discussion and conclude this paper in Chapter 6.

Chapter 2

Measurement Study

In this chapter, we first describe our measurement methodology. We then clarify why viewers' social relationship can provide reasonable and meaningful results for video recommendation.

2.1 Measurement Methodology

Our data set is crawled from the websites of RenRen and Youku and also contains some proprietary information provided by RenRen. We use six different computers to conduct crawling and then combine the data together. Video sharing in RenRen is based on the friend relationships. Initially, some users (as initiators) share a video link from a VSS to RenRen. This link immediately appears in their friends' main page as a "News Feed" in chronological order. Then the friends of these initiators will probably click the shared video link that appeared in their "News Feed". A video can be further propagated only if some viewers re-share the link. Such spreading method is also adopted in other systems, such as Facebook and Twitter. For each video in our data set, we get such information as the sharing time, original URL in VSS, total shares and total views in RenRen. We also obtain the statistics of these videos in Youku. Youku is selected since it contributes almost 80% of the total shared videos in RenRen. When a user in RenRen starts to view a video shared by her/his friend, a viewing record will be sent to the log server. The format of one item of log is as follows:

$starting\ time, viewer\ ID, video\ URL, direct\ sharer\ ID, original\ sharer\ ID$

where *starting time* is the time when the viewer ($user_A$) watched the video (as specified by the video URL) shared by the direct sharer ($user_B$). This video was first shared in RenRen by the original sharer or initiator ($user_C$).

In a typical day, there are over 10 million items of logs. The data set includes over 3 million viewers from RenRen and about $200,000$ videos from Youku. It captures the video sharing and watching behaviors among socially related viewers. Specifically, we choose the data of one day (March 24th, 2011) for analysis. There are $14,753,242$ items of logs in total. There are $201,553$ different URLs, namely the video sources and 3,514,460 viewers. For the ease of data processing, we use a unique number as the *video ID* instead of directly using the URL. The average video popularity (the times that a video has been viewed) is 73.2 and the average viewer activity (the number of videos that a viewer has viewed) is 4.2. To show their distribution, we plot the number of views per video and number of views per viewer as a function of the rank of the video and the viewer in Figure 2.1 and Figure 2.2, respectively.

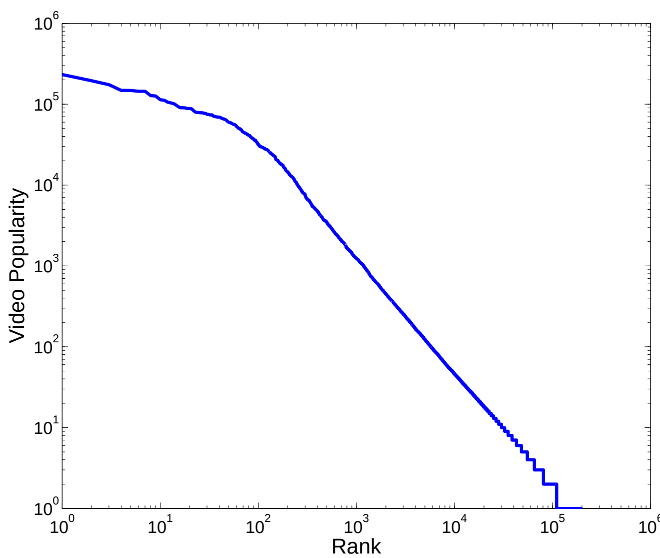

Figure 2.1: Videos rank ordered by popularity

We can see that both plots have a long tail on the log-log scale, however, they do not follow a Zipf distribution, which should be a straight line on a log-log scale [15]. The Pareto principle [8] that roughly 80% of the effects come from 20% of the causes, which is also known as the 80–20 rule, is widely used to describe the distribution with skewness. Previous works have shown that 10% of the most popular videos account for nearly 80%

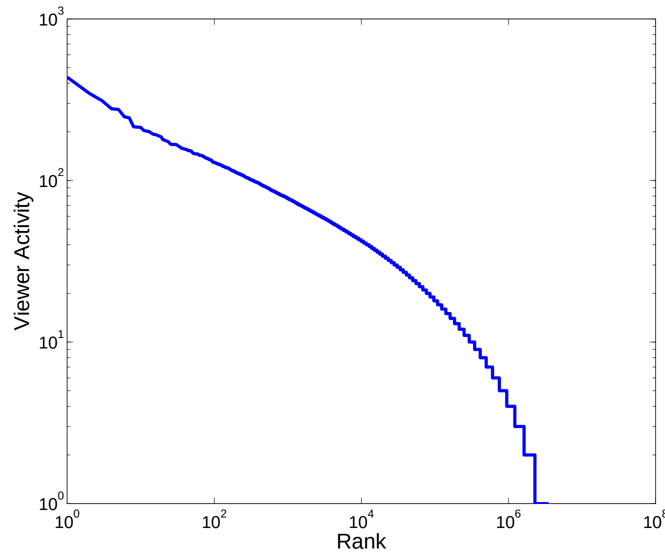

Figure 2.2: Viewers rank ordered by activity

of total views in YouTube [22, 23] and 30% of the most popular videos contribute 80% of total views in Youku [36]. We observe the similar scenario in our data set yet with some difference. The top-0.5% popular videos account more than 80% of total views and the top-1.65% popular videos account for 90% of total views. We can see that the skewness of video popularity is much higher in OSNs than in VSSes. For attractive videos, when someone shares them to her/his friends, more of the friends are likely to view them. These viewers will further share these videos again to their own friends with a very high probability. For unattractive videos, there are fewer direct share users and also fewer indirect share users. This difference in propagation for attractive and unattractive videos would be accumulated through cascading along the friend links. Hence, attractive videos receive more requests and the unattractive videos fade out quickly after very few cascade steps.

The skewness, however, is much lower for viewer activity. 1.65% of the most active viewers only account for 36.99% of total views. This is reasonable since for URLs, they could be viewed by many viewers simultaneously all day. While most viewers watch one video at the same time and they only spend limited time on watching videos. Further, there is no cascading effect on viewer activity. We can observe the huge difference in terms of average views (73.2 per video vs. 4.2 per viewer). We can also observe an obvious turning point that distinguishes the popular and unpopular videos around the video rank of 100 in

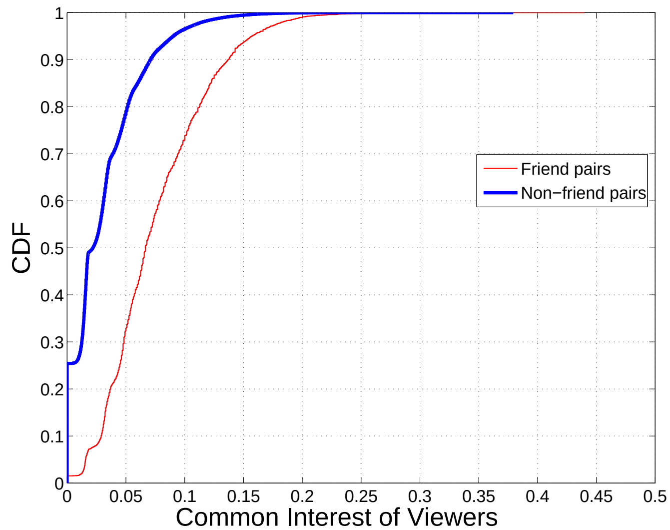

Figure 2.3: CDF of common interest of 10, 000 random viewers.

Figure 2.1, while we cannot find such turning point in Figure 2.2.

2.2 Common Interest

We start from examining the videos of common interest of different viewers. We use $C = \{c_1, c_2, ..., c_m\}$ to denote the set of m viewers. For each viewer $c_i \in C$, we use H_{c_i} to denote the set of the videos in her/his history video list. The common interest of two viewers, for example c_1 and c_2, is thus defined as

$$CI(c_1, c_2) = \frac{|(H_{c_1} \cap H_{c_2})|}{|(H_{c_1} \cup H_{c_2})|} \tag{2.1}$$

Figure 2.3 presents the cumulative distribution function (CDF) of common interest of 10, 000 randomly selected friend/non-friend viewers for one day duration. We can see that 28% non-friend viewers have no common interest, and 80% of them have watched less than 5% videos in common. On the other hand, less than 0.8% friend viewers have no common interest and only 30% of them have watched less than 5% videos in common. It is easy to see that friend viewers have higher common interest compared with non-friend viewers.

To better understand such a feature in long-term measurements, we also conduct a $90-day$ statistic for a number of pairs of friend viewers and compare the results to randomly

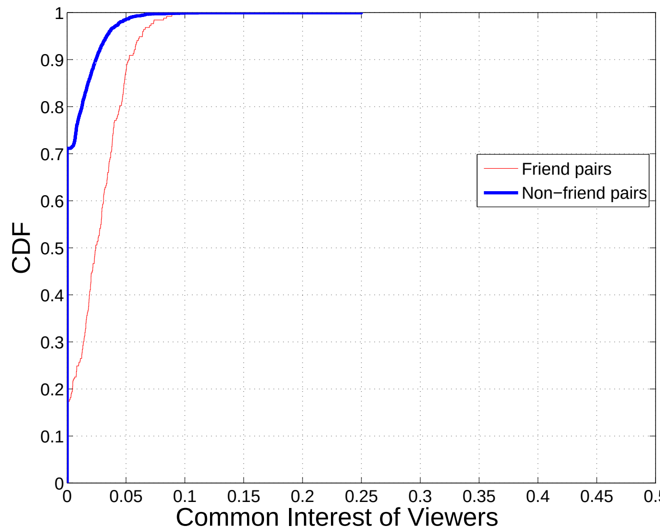

Figure 2.4: CDF of common interest for 90-day measurement

selected pairs of non-friend viewers. As shown in Figure 2.4, we can see that 71% non-friend viewers have no common interested videos. The friend viewers, on the other hand, are more likely to have common interest where only 18% of them never watched a common video in 90 days. Moreover, the average common interest of friend pairs and non-friend pairs is 7.57% and 3.02% respectively in this figure. It is also worth noting that the common interest in Figure 2.4 is lower than that in Figure 2.3. The reason is that for long-term measurements, even a slight difference of interest can leads to an accumulative impact on the common interest.

2.3 Social Similarity of Videos

Besides videos' popularity distribution across individual viewers, we further consider it across different social circles. A social circle will generally manifest a collective preference over a set of videos since the friends have common interests in videos. The relationship of videos and their sharing information can be represented by a matrix R. Each element $R(i, j) = k$, $k \in \mathbb{Z}$ indicates that the video j is viewed by k friends of user i. This value reflects the popularity distribution in the social circle of user i. This matrix can be inferred from our viewing/sharing logs.

For illustration, we use the data set of one day duration on March 24, 2011 that consists of over 12 million items of logs after deleting duplicate ones (viewers may click and watch the same video multiple times, and this behavior is considered as one log entry). We call this data set as raw data set. We extract the videos' ID and the corresponding sharing information from the data set. We filter out the users who shared very few videos, say less than 5); we also filter out the users who share too many, for example more than 1000, videos in one day[1]. This results in $263,115$ log entries including $1,596$ share users and $11,980$ videos (We call this data set $Dataset_A$). A visual illustration of partial R is shown in Figure 2.5.

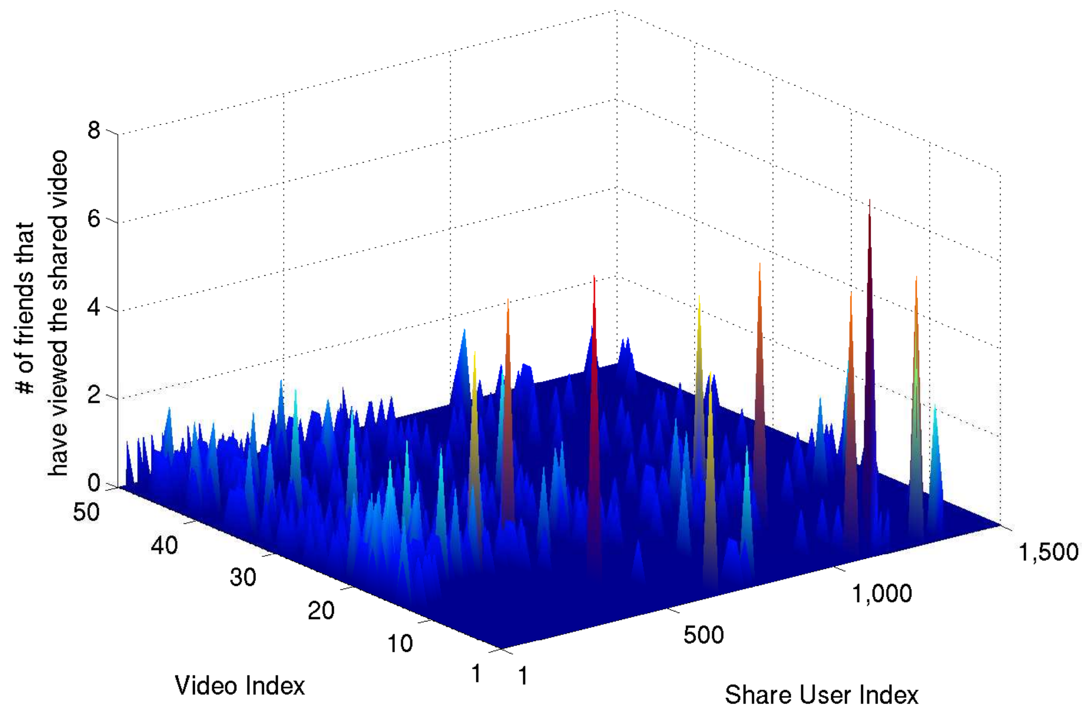

Figure 2.5: Visual illustration of R.

We can see clear video preferences across different social circles in this figure. For example, the friends of user 1 to 500 are not interested in video 1 to 10. The friends of user

[1]The RenRen engineers are also working on the behaviors of these highly active share users to see if they are some "OSN bots" on the user clients. However, the detailed discussion of this problem is beyond the scope of this thesis.

25 to 30, on the other hand, are quite interested in these videos. These different preferences lead to diverse popularity distributions of different videos. As we can see from Figure 2.5, some videos have very similar sharing popularity distribution across all the social circles (for example, video 18 and 21). We define *social similarity* of two videos based on their popularity distribution across social circles. A general definition is

$$m(i,j) = \frac{1}{\|d_i(s_1,...,s_n) - d_j(s_1,...,s_n)\|_1 + D} \tag{2.2}$$

where $d_{i(j)}(s_1,...,s_n)$ denotes video $i(j)$'s popularity distribution across social circles s_1 to s_n, and $\|\cdot\|_1$ denotes the L_1-norm, or the sum of the absolute values of vector entries; D is a smoothness parameter that controls the range of similarity values. When two videos are exactly the same in terms of their social similarity, the similarity value is $\frac{1}{D}$ (it will be infinity without D). Hence, we have $m(i,j) \in [0, \frac{1}{D}], \forall i, j$. We can see that this social similarity can be hardly obtained in practice for real systems due to extremely high dimension of the raw data. The dynamic viewers' preference further increases the difficulty in accurately computing the value. Fortunately, the auto-encoder model, which performs dimensionality reduction, offers a powerful tool to address these problems [33]; it learns a compressed representation for the raw data (matrix R) and extracts a low dimension set of useful features, as will be detailed in the next chapter.

Chapter 3

Learning Model

In this chapter, we first give a brief introduction to autoencoder and discuss how to train the autoencoder using our data set. We then evaluate autoencoder models with different parameter settings and select an appropiate model for learning.

3.1 Autoencoder

To obtain the input of the autoencoder, we first normalize the relation matrix R into a set of binary video profiles. This includes performing standardization over the columns and then the rows of R, and thresholding the resulting matrix at 0. We use R' to denote the normalized matrix. We adopt the neural network notation and refer to every video profile (every column of R') as an input data point v of N_v dimensions, and every share user corresponds to the ith element in the profile as a binary visible unit v_i.

Then we adopt a one-layer autoencoder to learn the manifold that will be used to calculate the similarity. Autoencoders are a kind of neural network. They are used to learn a low dimensional hidden manifold from the data, or to extract the features of the original input. A typical autoencoder consists of three or more layers [2] as follows:

1. An input layer. In our problem, the columns of the normalized matrix R' are mapped to the neurons in the input layer.

2. One or many hidden layers whose dimensions are significantly lower than that of the input layer. The hidden layers are often referred to as manifold. In our problem, we

use the manifold of only one hidden layer which is enough. The details of this layer will be discussed later.

3. An output layer. The neurons in this layer have the same meaning as in the input layer.

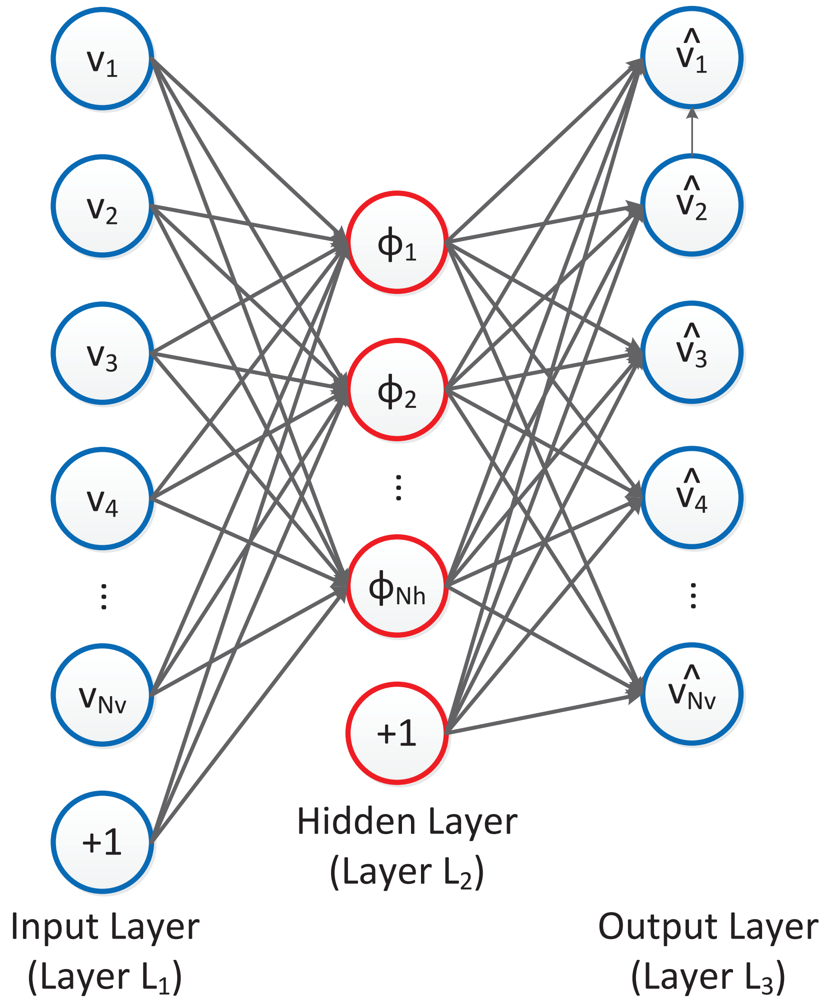

Figure 3.1: Structure of one-layer autoencoder

For better illustration, we present the structure of an autoencoder with one hidden layer in Figure 3.1. The circles labeled "+1" are called bias units. We can see that the graph of autoencoder does not have any directed loops or cycles and thus autoencoder is a kind of feedforward neural network. The manifold considered is in $(0,1)^{N_h}$, where N_h denotes the dimension or the number of hidden units, and each coordinate is denoted by $\phi_j \in (0,1)$. The projection from the input layer to the hidden layer and the projection from the hidden layer to the output layer are both given by sigmoidal transform, defined as follows, respectively:

$$\phi_j = \sigma\left(b_j + \sum_{i=1}^{N_v} w_{ij} v_j\right), \forall j = 1, 2, \dots N_h \tag{3.1}$$

$$\hat{v}_i = \sigma\left(c_i + \sum_{j=1}^{N_h} w_{ij} \phi_i\right), \forall i = 1, 2, \dots N_v \tag{3.2}$$

where $\sigma(x) = 1/(1+\exp(-x))$ is the sigmoid function, and $\hat{v}$ is the reconstruction obtained by projecting the input data v from the input layer to the output layer. This projection is parametrized by $W \in \mathbb{R}^{N_v \times N_h}$, $B \in \mathbb{R}^{N_v}$ and $C \in \mathbb{R}^{N_h}$, where W denotes the weight parameters, and B and C denote the bias terms. We use w, b, c to represent the element. ϕ_j and $\hat{v}_i$ are also called the activation of unit j in the hidden layer and the activation of unit i in the output layer, respectively. Sigmoid function is selected as the activation function in our model (there are other types of activation function such as Gaussian function and step function). We calculate the value of each neuron from lower layer to higher layer and this step is called *forward propagation*. It is worth noting that typically the weight parameter associated with the connection between the units in the input layer and the units in the hidden layer (w_{ij} in Equation 3.1), is different from that associated with the connection between the units in the hidden layer and the units in the output layer (w_{ij} in Equation 3.2). Here in our model, we benefit from the parameter tying technique so that these two sets of parameters are equal. This reduces half number of parameters and works better. The autoencoder seeks to minimize the square reconstruction error $E(W,B,C)$ between v and $\hat{v}$ over the entire training set of T data points. Specifically,

$$E(W,B,C) = \frac{1}{2T} \sum_{t=1}^{T} \sum_{i=1}^{N_v} (\hat{v}_i^{(t)} - v_i^{(t)})^2 \tag{3.3}$$

where the superscript (t) gives the index of the training instance.

The hidden units provide an information bottleneck that prevents the neural network from learning trivial mappings such as the identity. If the number of hidden units, N_h, is too small, the information content through the bottleneck is severely limited and the reconstruction will be poor. On the other hand, as N_h gets larger, the model incurs overfitting and introduces some trivial features. To ensure generalizability, we apply L_2 regularization to encode our preference for the simplest competent model, as follows:

$$E_s(W, B, C) = E(W, B, C) + \frac{\alpha}{2} \sum_{i=1}^{N_v} \sum_{j=1}^{N_h} w_{ij}^2 \tag{3.4}$$

where α is a small constant that trades off model simplicity with accuracy. We perform cross validation to select the best setting for both α and N_h from a discrete set of options. In particular, we train the model by minimizing $E_s(W, B, C)$ of the train set and select the best model based on the reconstruction error $E(W, B, C)$ of the test set. We optimize the reconstruction error on the training set by running 200 iterations of l-BFGS using Mark Schmidt's *minFunc* [48]. *minFunc* is widely used for unconstrained nonlinear optimization (we use sigmoid function as the activation function which is nonlinear). The input of the *minFunc* function is our autoencoder module. In the autoencoder module, we first use the forward propagation method to calculate the values of neurons in the output layer and obtain the reconstruction error by using Equation 3.4. We then use the backpropagation algorithm [19] to calculate the gradients of the weight parameter and update the weight parameter and the bias terms by taking steps proportional to the negative of the gradients. For example, to update W, we have:

$$W^{new} = W^{old} - \gamma \Delta W \tag{3.5}$$

W^{old} and W^{new} are the original W and the updated W, respectively; ΔW is the gradient of W at the value of W^{old}; γ is the learning rate, which is 0.5 in our program. The updating of B and C follows the same method with the same γ.

This gradient descent method [7] is a very popular first-order optimization algorithm to find a local minimum of a function. After running *minFunc*, we obtain the parameters W, B, and C that give minimum reconstruction error. Then we fix the parameters and evaluate the reconstruction error on the test set for the chosen values of α and N_v.

The pseudo code of the antoencoder module is shown in Algorithm 1. The superscript (t) represents the t-th input data point.

We summarize the notations used for the autoencoder in Table 3.1.

Algorithm 1 Autoencoder module $autoencoder(R', W, B, C)$

Input: The normalized matrix R'; Initialized W, B, C;
Output: $E_s(W, B, C)$, ΔW, ΔB, ΔC

```
ΔW = ΔB = ΔC = 0;
for t = 1 to T do
    v^(t) = R'(:, t);
    Compute φ according to Equation 3.1;
    Compute v̂^(t) according to Equation 3.2;
    Compute ΔW^(t), ΔB^(t), ΔC^(t) using backpropagation algorithm;
end for
Compute E_s(W, B, C) according to Equation 3.4;
ΔW = (1/T) Σ_{t=1}^{T} ΔW^(t) + αW;
ΔB = (1/T) Σ_{t=1}^{T} ΔB^(t) + αB;
ΔC = (1/T) Σ_{t=1}^{T} ΔC^(t) + αC;
W = W − γΔW;
B = B − γΔB;
C = C − γΔC;
```

3.2 Model Selection

We choose several typical values for α (0.0001, 0.01 and 1), and N_h from $(10, 20, 50, 100, 200)$. For each pair of parameter setting, we perform 5 trials and in each trial, we randomly split the data into a training set of size $10,000$ and a test set of size $1,980$. We then use the *minFunc* to train the autoencoder using the training data set and evaluate the selected parameter setting through the reconstruction accuracy of the test data set.

The model's sensitivity to the parameters α and N_v is shown in Figure 3.2 and Figure 3.3, respectively. In both figures, we report the average value and the standard deviation as error bars. We can see that the autoencoder yields very high accuracies. The accuracy decreases with α while increases with N_v. We therefore fix the value of α to 0.0001 and the number of hidden units N_h to 50. We do not choose $N_h = 100$ or $N_h = 200$ that has higher test set reconstruction accuracy since it is highly computation intensive yet with trivial performance gain.

Table 3.1: Summary of notations in autoencoder

v	Input data point, $v \in [0,1]^{N_v}$
T	Number of input data points
$\hat{v}$	Output values. $v \in [0,1]^{N_v}$
ϕ	Manifold (hidden units), $phi \in [0,1]^{N_h}$
N_v	Dimension of input/output data points
N_h	Dimension of manifold (number of hidden units)
W	Weight parameter
B, C	Bias term
$\Delta W, \Delta B, \Delta C$	Gradient of weight parameter and bias terms
E	Reconstruction error
E_s	Regularized reconstruction error
α	Weight decay parameter

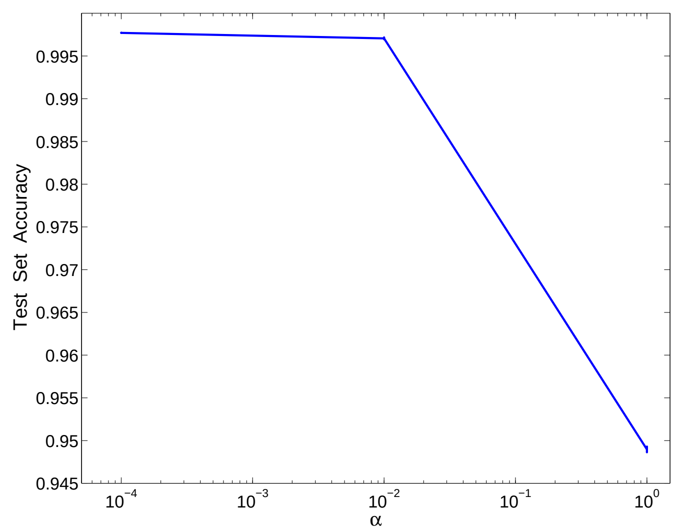

Figure 3.2: The growth of test set accuracy as a function of the weight decay α.

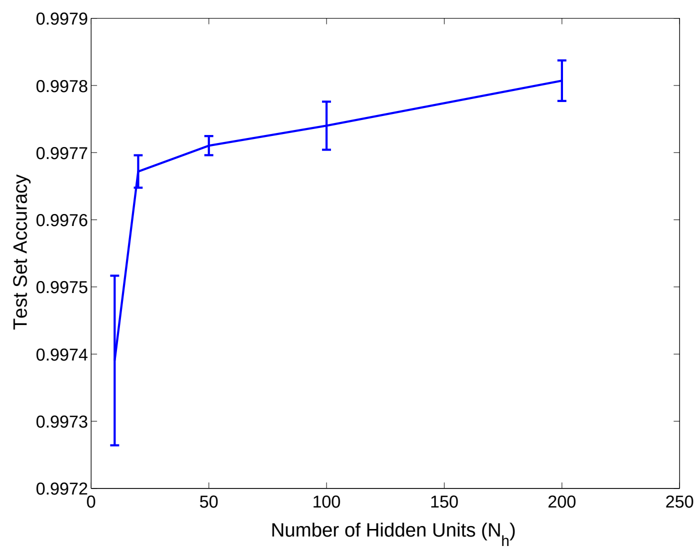

Figure 3.3: The growth of test set accuracy as a function of the number of hidden units N_h.

Chapter 4

Video Recommendation Enhancement

In this chapter, we describe how to utilize the social relations to enhance video recommendations. We first introduce how to compute social similarity of videos based the autoencoder model we have already trained. Then we propose a Social similarity Based Recommendation (SBR) system to recommend similar videos.

4.1 Computing Social Similarity of Videos

After selecting the model of the autoencoder, we compute the social similarity of videos as follows. We run the *minFunc* function again with more iterations to further minimize the reconstruction error. For two different videos i and j, we then use the obtained parameters (W, B, and C) to re-calculate their hidden manifolds respectively and compute their similarity based on the L_2 distance of the hidden manifolds, or formally

$$m(i,j) = \frac{1}{\sqrt{\sum_{k=1}^{N_h}(\phi_k^i - \phi_k^j)^2} + D} \tag{4.1}$$

where $\phi_k^{i(j)}$ is the kth dimension of video $i(j)$'s hidden manifold. The value of the smoothness parameter D is set to be 0.1 and thus $m(i,j) \in [0, 10], \forall i, j$.

The pseudo code to compute the video similarity is shown in Algorithm 2. *MaxIter* refers to the maximum number of iterations and its value is set to be 500 in our program.

Algorithm 2 Computing social similarity of videos

Input: Selected α, N_v; The normalized matrix R'; Initialized W, B, C;
Output: The similarity matrix M;

```
(W, B, C) = minFunc(autoencoder(R', W, B, C), MaxIter);
for t = 1 to T do
    for k = 1 to N_V do
        Compute φ_k^t according to Equation 5.1;
    end for
end for
for i = 1 to T do
    for j = 1 to T do
        Compute m(i, j) according to Equation 6.1;
    end for
end for
```

We obtain a similarity matrix $M \in [0, 10]^{11980\times 11980}$ after computing the similarity of all the video pairs. We choose the videos with indexes from $1,000$ to $1,050$ and illustrate a snapshot of the similarity matrix in Figure 4.1, where the intensity of the squares indicates the similarity value of the corresponding two videos. Darker square means that the corresponding two videos have higher similarity. We can see that there are some noticeable small clusters, which generally include less than 10 videos in this matrix. This observation confirms that some videos are indeed similar in terms of the sharing among social friends. There is also a considerable portion of videos that are nearly different from any other video and thus can be easily excluded from being recommendation candidates.

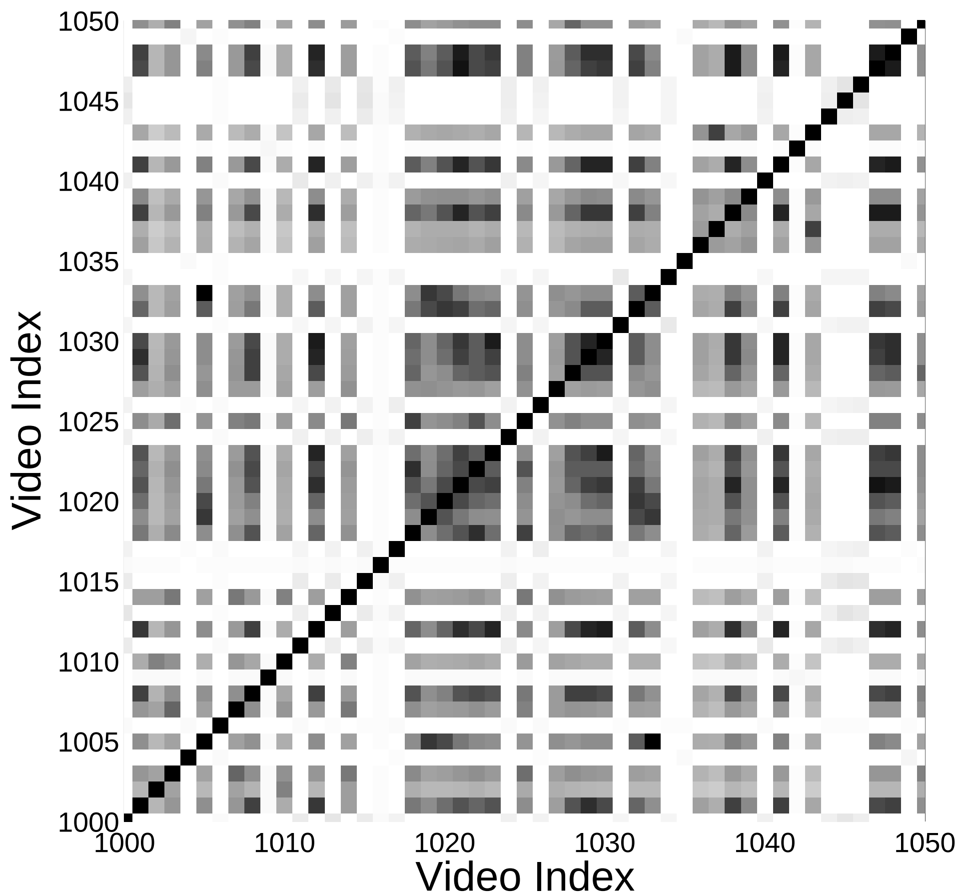

Figure 4.1: Snapshot of similarity matrix.

4.2 Enhanced Video Recommendation System

To compute personalized recommendations we find similar videos association with a viewer's history video list. For an arbitrary viewer, our system selects a number of videos from her/his history video list (denoted as set H) as recommendation seeds ($S = \{s_1, s_2, ..., s_N\}$, where N is the number of seeds). We use *remaining video list* to represent the videos that are not selected as recommendation seeds. For each seed s_i, an equal number (denoted as L) of highest-ranked videos in terms of similarity value are chosen and we use set V_i to represent them. The chosen videos for all seeds are merged together and for each video we compute its average similarity over all the recommendation seeds. We denote the union of these similar videos as set V:

$$V = \bigcup_{i=1}^{N} V_i \tag{4.2}$$

At last a number (denoted as K) of highest-ranked videos in terms of average similarity value are recommended (denoted as set Z) to this viewer. In general, a viewer may have interest in several different topics. We can further impose constraints on the number of videos selected for a single seed, or choose seeds with diverse categories. We will consider this issue in future work. Then we compare Z with the viewer's remaining video list to check which recommended videos are good recommendations, which will be explained in the next chapter. The notations are summarized in Table 4.1. The pseudo code is shown in Algorithm 3.

Table 4.1: Summary of notations in SBR system

H	Set of videos in viewer's history video list
N	Number of recommendation seeds
$S = \{s_1, s_2, ..., s_N\}$	Recommendation seeds
L	Number of videos selected for each seed
V_i	Set of videos selected for s_i
$V = \bigcup_{i=1}^{N} V_i$	Set of videos selected for all seeds
v_i	Videos in V
K	Number of recommended videos
Z	Set of videos recommended for the viewer

Algorithm 3 SBR algorithm for one viewer

Input: Viewer's history video list H; The similarity matrix M;
Output: Recommended videos Z;

```
Select N recommendation seeds from H;
for i = 1 to N do
    Select the top-L similar videos that have not been selected for s_1...s_{i-1} based on M;
end for
V = ⋃_{i=1}^{N} V_i;
for i = 1 to |V| do
    Calculate the average similarity of v_i;
end for
Select K videos with highest average similarity as Z;
```

Chapter 5

Evaluation

In this chapter, we first describe how to obtain the evaluation data set. Then we explain the evaluation metrics and present the evaluation results.

5.1 Evaluation Data set

To obtain the evaluation data set, We first find all the viewers who viewed the videos in $Dataset_A$ (11,980 videos in total) on March 24th, 2011. Then for each viewer, the videos in $Dataset_A$ s/he watched on this day compose her/his history video list. We also filter out the inactive viewers who viewed less than 10 videos on this day. As a result, our data set consists of 808 viewers and their corresponding video lists and we use $Dataset_B$ to denote it. The activity distribution of these viewers is shown is shown in Figure 5.1. We can see that the viewer activities have obvious discrepancy. There are a few viewers that watched more than 50 videos and most viewers watched less than 30 videos. We will examine the influence of viewer activity on the performance of recommendation systems.

5.2 Evaluation Metrics

We evaluate the performance of our video recommendation system based on two widely used metrics, *precision* and *recall* [50]. In our video recommendation context, for an arbitrary viewer, a *good recommendation* means that the recommended video is in her/his history video list. Precision is the fraction of recommended videos that are good recommendations; recall is the fraction of the videos in video list that are recommended. In some occasions, recall is

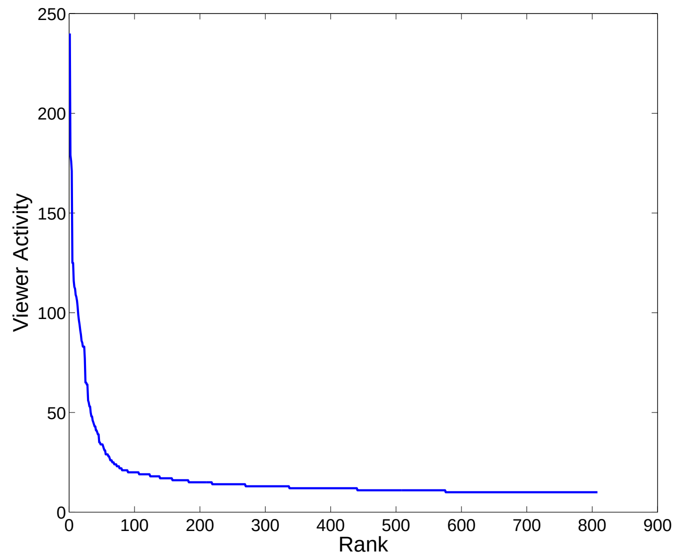

Figure 5.1: Activity distribution of viewers in the evaluation data set

also referred to as the *true-positive rate*. The formal definitions of good recommendation, precision and recall are as follows:

$$\{\text{good recommendations}\} = \{\text{recommended videos}\} \cap \{\text{viewer's history video list}\} \tag{5.1}$$

$$\text{precision} = \frac{|\{\text{good recommendations}\}|}{|\{\text{recommended videos}\}|} \times 100\% \tag{5.2}$$

$$\text{recall} = \frac{|\{\text{good recommendations}\}|}{|\{\text{viewer's history video list}\}|} \times 100\% \tag{5.3}$$

A perfect precision score of 100% means that all the recommended videos are good recommendations. Yet it is possible that some videos in the viewer's video list are not recommended (*false-negative*). A perfect recall score of 100% means that all the videos in the viewer's video list are recommended. Yet it is possible that some recommended videos are not in the viewer's video list (*false-positive*).

Our strategies to evaluate the system performance are slightly different for these two metrics. For precision, we recommend a fixed number of videos for each viewer. Considering that YouTube typically recommends 15 videos for registered users, we use this as a default setting in our system as well as in the systems for comparison. For recall, the number of recommended videos for each viewer is equal to the length of her/his remaining video list.

We also check the *prediction coverage* [29] of our SBR system. It is the percentage of all items that are recommended to users during the evaluation. Larger coverage means that more videos can be recommended to users so that the propagation of these videos could be assisted.

5.3 Evaluation Results

We have compared the performance of our social similarity based recommendation system with two other systems in terms of precision and recall. One is to recommend the most viewed videos (*mostview*) and another is to recommend the most shared videos (*mostshare*). These two methods have been widely used in the existing VSSes, including YouTube and Youku. They are also the basic building blocks of more advanced recommendation systems [27]. The most viewed videos and the most shared videos are both selected from the videos in $Dataset_A$.

5.3.1 Precision

We examine how the number of recommendation seeds and the number of videos chosen for each seed impact the recommendation precision. We first fix the value of N at 1, and choose the value of L from 15, 20, and 25. Then we fix the value of L at 20, and change the value of N to 2 and 3. We use SBR_N_L to represent our recommendation system with the specific setting of parameters.

We run 10 times for each setting of SBR and each time we randomly choose the recommendation seeds. Since *mostview* recommends the same videos for all the viewers, we only run it once and the same with *mostshare*. We report the average precision and the standard deviation as error bars in Figure 5.2.

The results show that our SBR system significantly outperforms the two widely used baseline solutions in terms of the average precision. This verifies that social relations are useful for video recommendation. It is worth noting an appropriate setting of N and L is important to the system performance. Intuitively, larger N and L could provide more videos to choose for recommendation and it would be more likely for the to-be-watched videos to

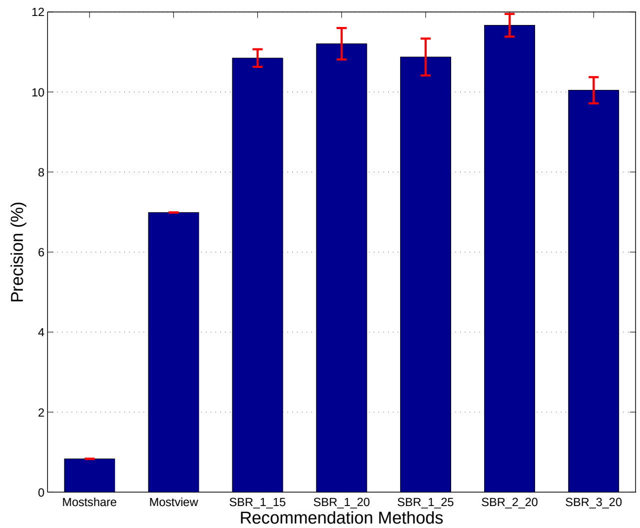

Figure 5.2: Comparison of precision

be included in the recommendation list. However, our results show that a moderate number of video seeds, say 2, and a moderate list size, say 20, can provide higher success. The reason is that we only use a one day data set, and viewers' history lists are not very large. So larger values of N and L would incur overfitting. We also notice that the *mostview* method has a much higher precision than the *mostshare* because there are a lot of videos that are shared by many people but are not as widely watched as the most viewed videos.

We then choose the best pair of parameter settings, $N = 2$ and $L = 20$, and plot the CDF of individual precision of all the viewers for the three methods in Figure 5.3. Again the *mostshare* method is the worst and the precision of more than 90% of the viewers is zero. The remaining two methods both successfully recommend at least one video for about half of the viewers. Compared with the *mostview* method where almost no viewer has a precision higher than 40%, *SBR* can offer a very high precision (40%–100%) to about 10% of the viewers.

We also conduct a one-to-one comparison of *mostview* and *SBR* for each viewer. The statistics are shown in Table 5.1. We use *high* to represent the viewers to whom *SBR* provides higher precision and *low* to represent the viewers to whom *SBR* provides lower precision, both compared with *mostview*. The numbers of these two sets of viewers are

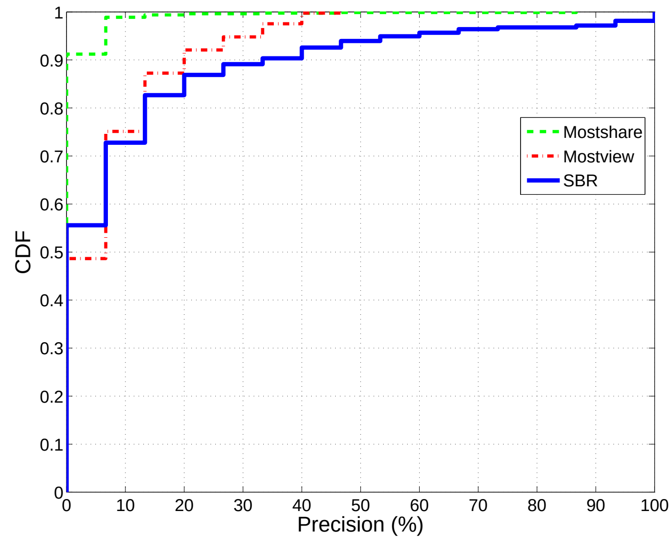

Figure 5.3: Comparison of precision CDF

at the same level. We can see that for the high viewers, *SBR* significantly outperforms *mostview* in average precision and is comparable to *mostview* for the low viewers. The activity distributions of these two sets of viewers are shown in Figure 5.4 and Figure 5.5, respectively. We can see that most of the active viewers benefit from *SBR*, while *mostview* favors the inactive viewers.

Table 5.1: One to one comparison of precision

	Number of viewers	Average precision of mostview	Average precision of SBR
high	252	4.07%	25.55%
low	288	14.19%	13.17%

5.3.2 Recall

The experiment setting to evaluate the recall is a bit different. We vary the number of recommendation seeds N from 1 to 5 and use SBR_N to represent it. The number of videos chosen for each seed could be different for individual viewers. For an arbitrary viewer, it is equal to the length of her/his remaining video list divided by the number of recommendation seeds.

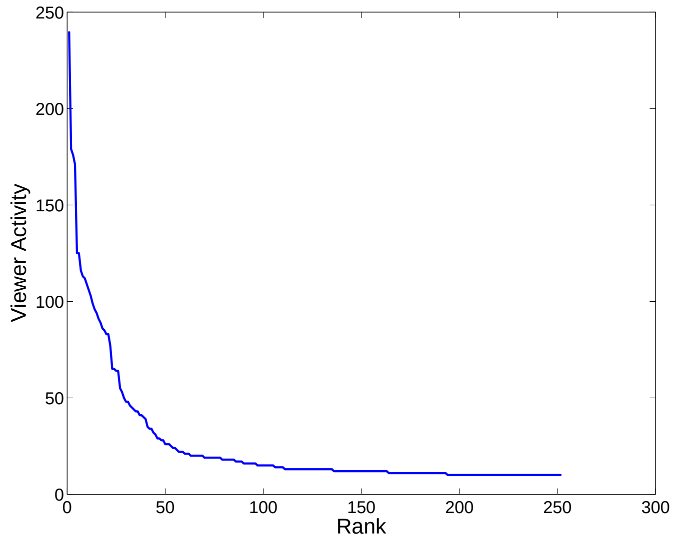

Figure 5.4: Viewer activity of viewers with higher precision

As with evaluating precision, we run 10 times for each setting of SBR and each time we randomly choose the recommendation seeds. We report the average precision and the standard deviation as error bars in Figure 5.6. SBR again significantly outperforms the other two systems. A moderate number of recommendation seeds, say 3, provides the best performance. For this setting, we plot the CDF of individual recall of all the viewers and compare it with $mostview$ and $mostshare$. SBR can offer quite a few viewers very high recall (above 40%), while most viewers experience recall lower than 20% when $mostview$ is used.

As for the one-to-one comparison, the statistics are shown in Table 5.2. Here high and low have the analogous meaning as in Table 5.1 while based on the comparison of recall. We can draw a similar conclusion that SBR significantly outperforms $mostview$ in average recall for the high viewers and is close to $mostview$ for the low viewers. The activity distributions of these two sets of viewers are shown in Figure 5.8 and Figure 5.9, respectively. We can see that most of the active viewers benefit from SBR, while $mostview$ favors the inactive viewers.

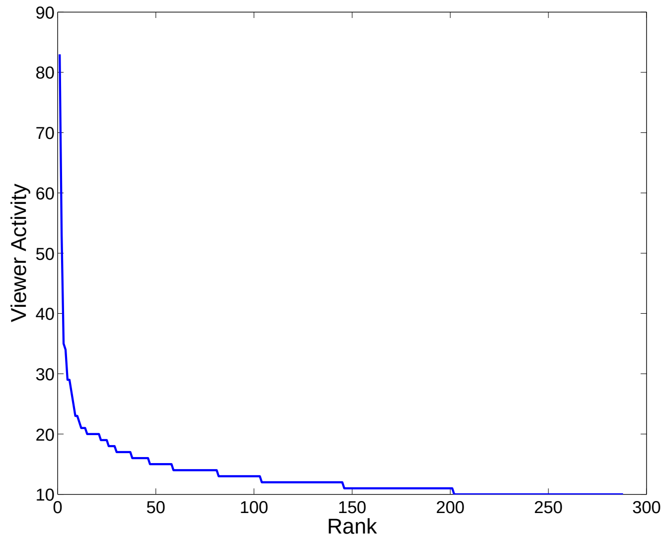

Figure 5.5: Viewer activity of viewers with lower precision

Table 5.2: One to one comparison of recall

	Number of viewers	Average recall of mostview	Average recall of SBR
high	296	4.89%	24.70%
low	261	16.56%	13.36%

5.3.3 Prediction Coverage

In $Dataset_B$, there are 6,233 different videos in the selected 808 viewers' video lists. Both the *mostview* and *mostshare* systems recommend a very limited range of videos, say 15 in the precision experiments and no more than 240[1] in the recall experiments. As a result, a major portion of videos have no chance to be recommended, which is not desirable in recommendation systems. As such, we check the prediction coverage of our SBR system. We also conduct each experiment 10 times and report the average value in Table 5.3. We can see that our SBR system has a good coverage of the videos in both cases.

[1] In fact, the exact number equals the length of the longest video list, which is 240, minus the number of recommendation seeds.

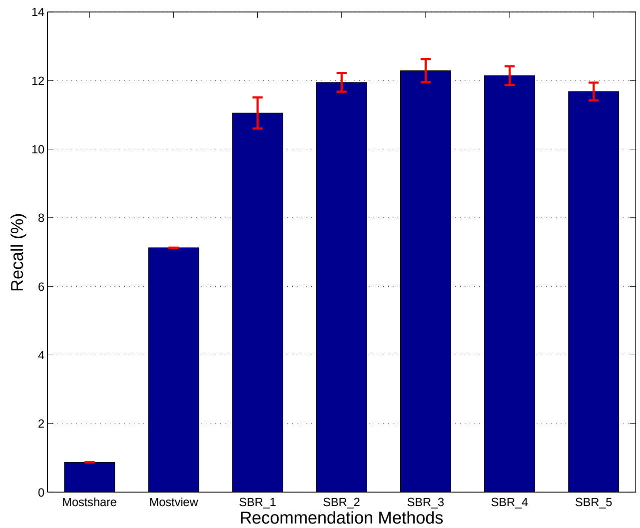

Figure 5.6: Comparison of recall

Table 5.3: Prediction coverage

	Precision	Recall
Average number of recommended videos	3958.7	5332.7
Average prediction coverage	63.51%	85.56%

5.4 Summary

We conducted offline experiments using the existing data set. We compared the three recommendation systems, *SBR, mostview*, and *mostshare*, in terms of precision and recall on $Dataset_B$ that consists of 808 viewers' history video lists. SBR provides the highest average precision and average recall, while *mostshare* provides the lowest. An appropriate setting of system parameters, such as the number of recommendation seeds and the number of videos selected for each seed, plays an important role in the SBR system. The best setting could be different for heterogeneous data sets. Through one-to-one individual comparison, we concluded that SBR performs much better than *mostview* in best cases, and is comparable to *mostview* even in worst cases. We also showed that SBR has a reasonably good prediction coverage. These results are encouraging and verify that social relations can benefit video recommendations.

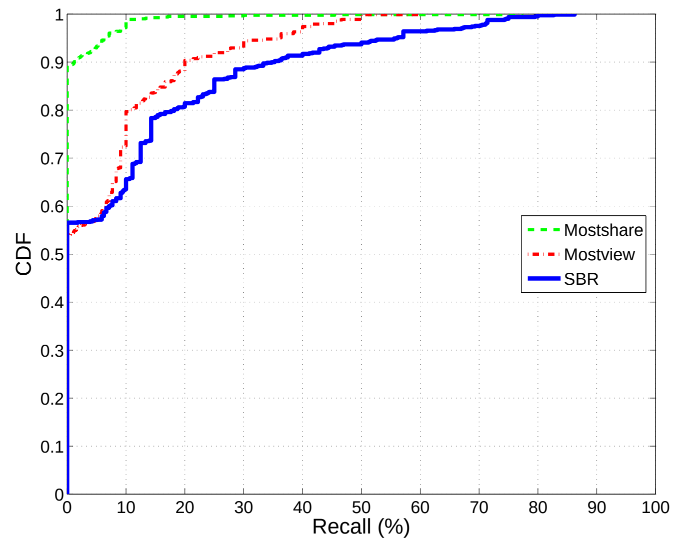

Figure 5.7: Comparison of recall CDF

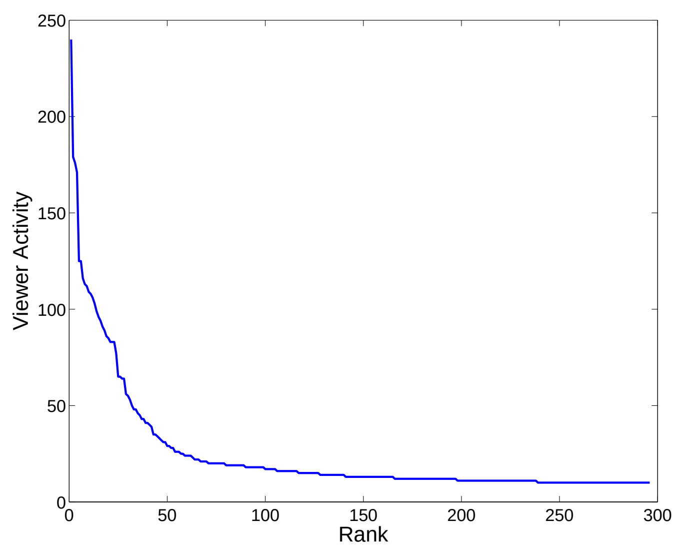

Figure 5.8: Viewer activity of viewers with higher recall

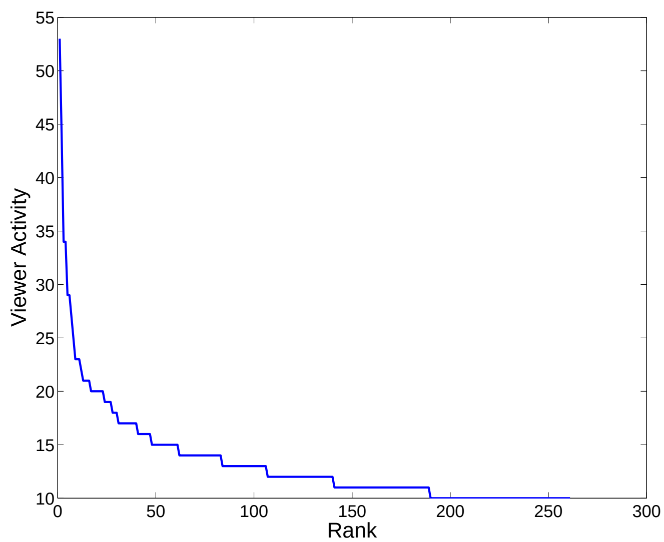

Figure 5.9: Viewer activity of viewers with lower recall

Chapter 6

Discussion and Conclusion

6.1 Discussion

In this section, we give our further discussion on the data set, leaning of similarity, system design, and evaluation. We try to shed light on some possible concerns as well as to point out potential improvements for future work.

6.1.1 Data Set

We extracted three data sets from the one-day raw data. In the first data set, we randomly selected $10,000$ friend/non-friend viewers to clarify that friends have more common interests. We are more interested in the other two data sets: $Dataset_A$ that captures the sharing patterns of $11,980$ videos across $1,596$ share users and $Dataset_B$ that contains the history video lists for 808 viewers who have viewed the selected videos in $Dataset_A$.

For $Dataset_A$, a critical problem is whether the selected share users can represent the social circles in the OSN. It is possible that many of the share users are friends to each other and their social circles overlap a lot. As a result, the relation matrix R unavoidably has information redundancy, which is verified by our autoencoder model that shows that a low dimension encoding incurs very small distortion for the data set with extremely high dimension (50 $vs.$ $1,596$). It is possible to reduce the number of share users if we can divide the social circle accurately. This is very useful since it can speed up the learning of similarity matrix and may further reduce the dimension of the manifold in the autoencoder. To divide the social circles we need to dive into the relationship graph of OSN users, which is a non-trivial task. So we use an approximation which is easy to conduct. Another problem is that by sampling the share users, we also lose the information for a large number of videos ($Dataset_A$ only contains the information of $11,980$ video while there are over $200,000$ videos in total). Through further examination, we find that most of the videos that are missing in $Dataset_A$ have very low popularity. In future work, we will select more share users and thus more videos, especially the popular videos missing in $Dataset_A$.

For $Dataset_B$, a major concern would be about its scale. It only contains 808 viewers, which seems to be not enough to evaluate the recommendation system. Yet it is still representative since we filter out a lot of inactive viewers. It is reasonable to focus on those active viewers who are more likely to be the targets of OSN/VSS monetization. And for the inactive users, there will be very limited selection for the recommendation seeds. It is a tradeoff between user coverage and recommendation accuracy. Further, If we have more videos in $Dataset_A$, we would find more active viewers who have viewed these videos.

To increase the scale of $Dataset_A$ and $Dataset_B$, we can use the raw data of several days, and this can be part of our future work.

6.1.2 Learning of Similarity

For the ease of computation, we normalized the relation matrix to a binary matrix, which would incur some distortion. We could make the value continuous between [0,1] through shifting and scaling, but it will significantly increase the memory overhead and also slow down the training of the autoencoder. We will try different normalization methods (e.g., using continuous value, taking log to deal with the heavy tail problem) and other activation functions in future work.

The dynamics of similarity is another important issue. There are two kinds of dynamics to be considered. One is how to compute the similarity for a new video. This problem has been largely solved by our autoencoder model. Recall that in the training of the autoencoder, we use a part of videos as the training set to compute the weight parameters in the autoencoder, and use the remaining videos as the test set which can be regarded as new videos. When a new video comes, we only need to obtain their sharing distribution, add it as a new column of the relation matrix R and recalculate the similarity matrix using the existing autoencoder. Yet we will encounter the second problem of how the similarity evolves over time. Even for the existing videos, their popularity will change day by day and the similarity of videos could also change. In this situation, we may need to reconstruct the autoencoder from scratch, which is very time consuming. We will examine this issue in our future work.

6.1.3 System Design

First, the selection of recommendation seeds is of paramount importance. Viewers' interest may change, and everyday we have different hot topics. The most recently viewed videos are usually selected as seeds. Further, for a new viewer, there is no recommendation seed, which is the well-known *cold-start* problem [47]; for the viewer who has not viewed videos or even has not logged in the OSN for a long time, the videos in her/his video list are too old to be recommendation seeds. In both situations, we can choose videos from her/his friends' video lists as seeds.

Second, we selected an equal number of videos for each recommendation seed and used the average similarity value over all the recommendation seeds in our enhanced recommendation system. The recommendations could be more successful if we introduce a weight for each seed. This weight can be used to determine the number of videos selected for each seed or to calculate the weighted average similarity. We can obtain such weight based on the popularity of the seed, especially the popularity in the viewer's social circle.

6.1.4 Comparison Baseline

The baseline methods for comparison *mostview* and *mostshare* seem to be naive though they are widely used. However, we are not expecting to replace the more advanced systems, say the one used by YouTube which suggests the most related videos. Instead, SBR can work as a complementary component utilizing the social relations to enhance the existing system. Furthermore, the high prediction coverage of SBR would assist the propagation of videos and improve the recommendation diversity. We will work on how to combine SBR and other recommendation systems in our future work.

6.2 Conclusion

In this thesis, we investigated the possible gain of using social relations to enhance video recommendation. We developed an autoencoder learning model to quantify the social similarity of different videos. The trace-based evaluation results demonstrated that our similarity-based recommendation system can remarkably improve the precision and recall. Our work represents an initial attempt towards this issue. We will conduct more experiments on a larger data set to investigate its adaptation to dynamics. More in-depth studies are expected to further examine the interactions between OSNs and VSSes. We believe that a better understanding of their relationship will facilitate the design of existing video recommendation systems.

Bibliography

[1] Amazon.com. http://www.amazon.com.

[2] Autoencoder. http://en.wikipedia.org/wiki/Autoencoder.

[3] ebay. http://www.ebay.com.

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

[5] Facebooks prospects may rest on trove of data. http://www.nytimes.com/2012/05/15/technology/facebook-needs-to-turn-data-trove-into-investor-gold.html.

[6] Google plus. http://plus.google.com.

[7] Gredient descent. http://en.wikipedia.org/wiki/Gradient_descent.

[8] Pareto principle.

[9] Renren. http://www.renren.com.

[10] Twitter. http://www.twitter.com.

[11] Youku. http://www.youku.com.

[12] Youtube. http://www.youtube.com.

[13] Youtube reaches 4 billion video views per day. http://www.forbes.com/sites/annakupka/2012/01/24/youtube-reaches-4-billion-video-views-per-day/.

[14] Youtube statistics. http://www.youtube.com/t/press_statistics/.

[15] Zipf's law. http://en.wikipedia.org/wiki/Zipf's_law.

[16] Amir Afrasiabi Rad and Morad Benyoucef. Measuring propagation in online social networks: the case of youtube. *Journal of Information Systems Applied Research*, 5(1):26–35, Jan. 2012.

[17] Shumeet Baluja, Rohan Seth, D. Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed 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.

[18] Fabrício Benevenuto, Tiago Rodrigues, Virgilio Almeida, Jussara Almeida, and Keith Ross. Video interactions in online video social networks. *ACM Transactions on Multimedia Computing, Communications and Applications*, 5(4):30:1–30:25, Nov. 2009.

[19] Christopher M Bishop. Pattern recognition and machine learning. pages 242–245. Springer, 2006.

[20] Toine Bogers and Antal Van Den Bosch. Fusing recommendations for social bookmarking web sites. *International Journal of Electronic Commerce*, 15(3):31–72, Apr. 2011.

[21] Robin Burke. The adaptive web. chapter Hybrid web recommender systems, pages 377–408. Springer-Verlag, Berlin, Heidelberg, 2007.

[22] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pages 1–14, 2007.

[23] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. Analyzing the video popularity characteristics of large-scale user generated content systems. *IEEE/ACM Transactions on Networking*, 17(5):1357–1370, Oct. 2009.

[24] Xu Cheng, Cameron Dale, and Jiangchuan Liu. Statistics and social network of youtube videos. In *Proceedings of the 16th IEEE International Workshop on Quality of Service*, IWQoS '08, pages 229–238, 2008.

[25] Xu Cheng, Kunfeng Lai, Dan Wang, and Jiangchuan Liu. Ugc video sharing: measurement and analysis. *Intelligent Multimedia Communication: Techniques and Applications*, 280:367–402, 2010.

[26] Lake Chris. Online video sharing doubles within a year. `http://econsultancy.com/us/blog/7198`.

[27] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. The youtube video recommendation system. In *Proceedings of the 4th ACM Conference on Recommender Systems*, RecSys '10, pages 293–296, 2010.

[28] Toon De Pessemier, Tom Deryckere, and Luc Martens. Context aware recommendations for user-generated content on a social network site. In *Proceedings of the 7th European*

Conference on European Interactive Television Conference, EuroITV '09, pages 133–136, 2009.

[29] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the 4th ACM Conference on Recommender Systems*, RecSys '10, pages 257–260, 2010.

[30] Phillipa Gill, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. Youtube traffic characterization: a view from the edge. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, pages 15–28, 2007.

[31] Lei Guo, Enhua Tan, Songqing Chen, Xiaodong Zhang, and Yihong (Eric) Zhao. Analyzing patterns of user content generation in online social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 369–378, 2009.

[32] Jianming He and Wesley W. Chu. A social network-based recommender system. Technical Report 090014, Computer Science Department, UCLA, 2009.

[33] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[34] Pazzani Michael J. and Billsus Daniel. The adaptive web. chapter Cotent-based recommender systems, pages 325–341. Springer-Verlag, Berlin, Heidelberg, 2007.

[35] Ioannis Konstas, Vassilios Stathopoulos, and Joemon M. Jose. On social networks and collaborative recommendation. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 195–202, 2009.

[36] Haitao Li, Jiangchuan Liu, Ke Xu, and Song Wen. Understanding video propagation in online social networks. In *Proceedings of the 20th IEEE International Workshop on Quality of Service*, IWQoS '12, pages 1–9, Jun. 2012.

[37] Haitao Li, Haiyang Wang, and Jiangchuan Liu. Video sharing in online social network: measurement and analysis. In *ACM NOSSDAV*, 2012.

[38] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 287–296, 2011.

[39] Hao Ma, Tom Chao Zhou, Michael R. Lyu, and Irwin King. Improving recommender systems by incorporating social contextual information. *ACM Transactions on Information Systems*, 29(2):9:1–9:23, Apr. 2011.

[40] Tim O'Reilly. What is web 2.0: design patterns and business models for the next generation of software. http://oreilly.com/web2/archive/what-is-web-20.html.

[41] Jonghun Park, Sang-Jin Lee, Sung-Jun Lee, Kwanho Kim, Beom-Suk Chung, and Yong-Ki Lee. Online video recommendation through tag-cloud aggregation. *Multimedia, IEEE*, 18(1):78–87, Jan. 2011.

[42] R. Sinha Rashmi and Swearingen Kirsten. Comparing recommendations made by online systems and friends. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries '01*, 2001.

[43] Francesco Ricci, Lior Rokach, and Shapira Bracha. Recommender systems handbook. pages 1–35. Springer-Verlag, Berlin, Heidelberg, 2011.

[44] Alan Said, Ernesto W. De Luca, and Sahin Albayrak. How social relationships affect user similarities. In *Proceedings of the ACM IUI'10 Workshop on Social Recommender Systems*, 2010.

[45] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. The adaptive web. chapter Collaborative filtering recommender systems, pages 291–324. Springer-Verlag, Berlin, Heidelberg, 2007.

[46] J. Ben Schafer, Joseph Konstan, and John Riedi. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*, EC '99, pages 158–166, 1999.

[47] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 253–260, 2002.

[48] Mark Schmidt. minfunc. `http://www.di.ens.fr/~mschmidt/Software/minFunc.html`.

[49] Guy Shani and Gunawardana. Evaluating recommender systems. MSR-TR-2009-159, 2009.

[50] Xiaodan Song, Belle L. Tseng, Ching-Yung Lin, and Ming-Ting Sun. Personalized recommendation driven by information flow. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 509–516, 2006.

[51] Martin Szummer and Tommi Jaakkola. Partially labeled classification with markov random walks. In *Advances in Neural Information Processing Systems*, NIPS '01, pages 945–952, 2001.

[52] Hema Yoganarasimhan. Impact of social network structure on content propagation: a study using youtube data. *Quantitative Marketing and Economics*, 10:111:150, 2012.

[53] Elena Zheleva and Lise Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 531–540, 2009.

[54] Renjie Zhou, Samamon Khemmarat, and Lixin Gao. The impact of youtube recommendation system on video views. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, pages 404–410, 2010.

[55] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of 20th International Conference on Machine Learning*, ICML '02, pages 912–919, 2002.