Understanding Video Propagation in Online Social Networks: Measurement, Analysis, and Enhancement

by

Haitao Li

M.Sc., Tsinghua University, 2010

B.Sc., Beihang University, 2006

Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

Doctor of Philosophy

in the
School of Computing Science
Faculty of Applied Sciences

© Haitao Li 2014
SIMON FRASER UNIVERSITY
Spring 2014

All rights reserved.
However, in accordance with the *Copyright Act of Canada*, this work may be reproduced without authorization under the conditions for "Fair Dealing." Therefore, limited reproduction of this work for the purposes of private study, research, criticism, review and news reporting is likely to be in accordance with the law, particularly if cited appropriately.

APPROVAL

Name: Haitao Li

Degree: Doctor of Philosophy

Title of Thesis: Understanding Video Propagation in Online Social Networks: Measurement, Analysis, and Enhancement

Examining Committee:

Dr. Mark Drew
Professor
Chair

Dr. Jiangchuan Liu
Senior Supervisor
Associate Professor

Dr. Qianping Gu
Supervisor
Professor

Dr. Jie Liang
Internal Examiner
Associate Professor

Dr. Zongpeng Li
External Examiner
Associate Professor, University of Calgary

Date Approved: April 24th, 2014

Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the non-exclusive, royalty-free right to include a digital copy of this thesis, project or extended essay[s] and associated supplemental files ("Work") (title[s] below) in Summit, the Institutional Research Repository at SFU. SFU may also make copies of the Work for purposes of a scholarly or research nature; for users of the SFU Library; or in response to a request from another library, or educational institution, on SFU's own behalf or for one of its users. Distribution may be in any form.

The author has further agreed that SFU may keep more than one copy of the Work for purposes of back-up and security; and that SFU may, without changing the content, translate, if technically possible, the Work to any medium or format for the purpose of preserving the Work and facilitating the exercise of SFU's rights under this licence.

It is understood that copying, publication, or public performance of the Work for commercial purposes shall not be allowed without the author's written permission.

While granting the above uses to SFU, the author retains copyright ownership and moral rights in the Work, and may deal with the copyright in the Work in any way consistent with the terms of this licence, including the right to change the Work for subsequent purposes, including editing and publishing the Work in whole or in part, and licensing the content to other parties as the author may desire.

The author represents and warrants that he/she has the right to grant the rights contained in this licence and that the Work does not, to the best of the author's knowledge, infringe upon anyone's copyright. The author has obtained written copyright permission, where required, for the use of any third-party copyrighted material contained in the Work. The author represents and warrants that the Work is his/her own original work and that he/she has not previously assigned or relinquished the rights conferred in this licence.

Simon Fraser University Library
Burnaby, British Columbia, Canada

revised Fall 2013

Abstract

The deep penetration of Online Social Networks (OSNs) has made them major portals for video content sharing recently. It is known that a significant portion of the accesses to video sharing sites (VSSes) are now coming from OSN users. For example, YouTube reported that, as of January 2012, more than 700 tweets per minute containing a YouTube link, and over 500 years' worth of YouTube videos are watched by Facebook users every day. Although the videos shared in OSNs are mostly from VSSes, OSNs provide quite different mouth-to-mouth-like sharing mechanisms, leading to distinctive user access patterns. Yet the unique features of video sharing over OSNs and their impact remain largely unknown.

In this thesis, we conduct a systematic study on the video propagation in OSNs based on large-scale real-world data. Our study unveils the unique characteristics of video requests from OSNs, showing that an OSN can dramatically amplify the skewness of video popularity that 2% most popular videos account for 90% of total views; and video popularity also exhibits much more dynamics with multiple request bursts. We then closely analyze the video propagation process in OSNs with both measurement and modeling, identifying the key influential factors. We further examine the popularity prediction of videos shared in OSNs. We demonstrate that conventional methods largely fail in this new context, and develop a novel propagation-based prediction model. Finally, based on the above studies, we present SNACS (Social Network Aware Cloud Assistance for Video Sharing), which enables OSN operators to cost-effectively enhance the video viewing experience of their users through utilizing content cloud services.

To my family

Every day may not be good, but there is something good in every day.
Enjoy cheerfully what you think is good!

Acknowledgments

First and foremost I want to thank my senior supervisor Dr. Jiangchuan Liu. He has taught me, both consciously and unconsciously, how good research is done. I appreciate all his contributions of time, ideas, and funding to make my PhD experience productive and stimulating. The joy and enthusiasm he has for his research was contagious and motivational for me, even during tough times in the PhD pursuit. I am also thankful for the excellent example he has provided as a successful researcher.

Special thanks to my defense committee: my supervisor Dr. Qianping Gu, internal examiner Dr. Jie Liang, external examiner Dr. Zongpeng Li, and chair Dr. Mark Drew, for their support, guidance and helpful suggestions. Their guidance has served me well and I owe them my heartfelt appreciation.

My time at SFU was made enjoyable in large part due to my lab mates and friends. We discussed papers and experiments together; We played Pingpang and FIFA together; We had memorable trips to Victoria, Whistler and Hope together; And we had wonderful parties in traditional Chinese holidays. I am grateful for time spent with them.

Lastly, I would like to thank my family for all their love and encouragement. For my parents who raised me with a love of science and supported me in all my pursuits. And most of all for my loving, supportive, encouraging, and patient wife Ying whose faithful support during the final stages of this PhD is so appreciated. Thank you.

Haitao Li
Simon Fraser University
April 2014

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

Traditionally, users discover videos on the Web by browsing or searching. Recently, word-of-mouth has emerged as a popular way of discovering videos, particularly over online social network (OSN) sites such as Facebook and Twitter, where users discover video contents following their friends' shares. It has been a key driving force toward the uprise of accesses to video sharing sites (VSSes). YouTube reported that, as of January, 2011 more than 500 tweets per minute containing a YouTube link, and over 150 years worth of YouTube video is watched by Facebook users every day. Till June 2012, the numbers have increased to 700 tweets and 500 years [65]. According to comScore's latest statistics in September 2013 [13], Facebook ranked No.2 in terms of the number of viewers (67 millions), and No.3 in terms of the number of video views (975 millions). Besides Facebook and Twitter, we have seen similar trend around the world. For example, as of May 2011, more than 54 million unique RenRen [45] (the largest Facebook-like OSN in China) users have participated in video viewing and 20 million participated in sharing, generating 12.4 million views, and 1.64 million shares every day [33].

Video propagation in OSNs is based on the friend relationships, and its process can be described as follows. Initially, a user posts a video link from a VSS in an OSN; This link immediately appears on her/his friends' main page as a "News Feed" in chronological order; Meanwhile, this shared video is also listed in the sharer's home page, which lists all her/his ever shared contents. Then her/his friends will probably click the shared video appeared in "News Feed"; or they may regularly visit friends' home pages to watch those shared videos, though this frequency is much lower than the first way. A video can be further propagated if some viewers share the link again.

Although the videos shared in OSNs are from VSSes, OSNs provide quite different video sharing mechanisms. Videos in VSSes are mainly viewed via related videos, their search engines and front pages [4], whereas the videos in OSNs are viewed via friends' direct shares. In VSSes, users can hardly discover niche content, or content that is not properly categorized

or ranked. Instead, a recommendation strategy plays an important role. While in OSNs, each video has a fair opportunity to be propagated along friendship links and the attractiveness of video content itself is the most important factor that determines its popularity. These differences lead to distinguished video popularity distributions and evolutions. Understanding the new characteristics of video sharing behaviors in OSNs can thus provide valuable information to ISPs, CDN providers, video site administrators, and content owners.

Exploring the characteristics of video sharing OSNs however has many challenges. First, privacy protection generally prevents crawling video viewing information as easily in OSNs (e.g., Facebook/RenRen) as in VSSes (e.g., YouTube/Youku). Second, unlike dedicated video sites, OSNs rarely provide rich statistics about shared videos. Finally, given the wide distribution of OSN users, tracing traffic from a small set of network routers/switches can hardly reveal the geographic evolution of video sharing, not to mention the sheer volume of the mixed network traffic to be analyzed. Due to such challenges, the unique features of video sharing over OSNs and their impact remain largely unknown.

1.2 Contributions of this Thesis

In this thesis, we present a comprehensive study on the video propagation in OSNs: from the perspectives of measurement, modeling and system enhancement. The work and contributions of this thesis are summarized as follows:

- We closely collaborate with RenRen, a large-scale Facebook-like OSN in China, to analyze its user access logs spanning over four months, and conduct a long-term and extensive measurement of video sharing in RenRen. Our measurement reveals a number of distinctive features of video requests from OSNs, which noticeably differ from that of conventional videos, including the latest report on YouTube videos with inherent social features. We observe that the OSN amplifies the skewness of video popularity so largely that about 2% most popular videos account for 90% of total views; furthermore, video popularity evolution shows much more dynamics. Our measurements also unveil the video propagation process in OSNs, and user behaviors during the process. These measurement works have motivated our other works such as popularity prediction and system enhancement, and we believe they definitely will trigger much more subsequent studies on this topic.

- To further understand the measured results, we provide modeling analyses. First, we build a simple but effective model-based emulator to generate synthetic video requests from OSNs. Our emulator well captures the observed characteristics in the empirical data, including the video popularity distribution and dynamics. This tool is especially helpful for these studies that need user requests as the input, such as caching strategy

study. We further propose an S^2I^3R model which extends the conventional epidemic models to accommodate diverse types of users and their probabilistic viewing and sharing behaviors. This model is useful in studying the impact of diverse parameters to the video propagation, and thus can be used as a tool to make a preliminary test for system design (e.g., recommendation strategy) before real deployment.

- Popularity prediction, with both technological and economic importance, has been extensively studied for conventional VSSes, where the videos are mainly found via searching, browsing, or related links. Yet the popularity prediction in the OSNs context remains largely unexplored. We present an initial study on the popularity prediction of videos propagated in OSNs along friendship links. We find that typical views-based prediction models are generally ineffective, if not totally fail, especially when predicting the early peaks and later bursts of accesses, which are common during the video propagations in OSNs. To overcome these limits, we develop a novel propagation-based video popularity prediction solution, namely SoVP. Instead of relying solely on the early views for prediction, SoVP considers both the attractiveness of a video and the influence of the underlying propagation structure. The effectiveness of SoVP, particularly for predicting the peaks and bursts, have been validated through our trace-driven experiments.

- Propagated through chains of friends, the coverage of OSN-shared videos can be much broader with stronger micro- and macro-dynamics. Given that the contents are still hosted by external VSSes, such distinct access patterns from OSN users have created significant new challenges to VSSes. To this end, we present SNACS, a cost-effective social network-aware cloud assistance for video sharing. The SNACS module sits between VSSes and an OSN, and is managed by the OSN to improve its users' video access experience using both centralized cloud resources and edges servers. Given the strong dynamics of the access patterns, we are particularly interested in the content management and update strategies in the SNACS' implementation. Motivated by realworld data traces, we show that conventional cache replacement can be quite inefficient in this context. We then develop an optimal offline algorithm with minimized cache misses and replacements. It also motivates an online solution that makes effective use of the video propagation information in the OSN. Our design has been extensively evaluated and its superiority has been validated under diverse network and user configurations.

1.3 Related Works

There have been some related researches on video sharing in OSNs from the perspectives of measurement, modeling and system enhancement. First, some measurement analyses

have been conducted by analyzing proprietary log data provided by RenRen[1] [33, 30, 34, 12] and Tencent Weibo[2] [58, 57], or by analyzing the crawled data of video propagation in Twitter through its official API [47, 55]. Second, based on these measurement results, both epidemic model [12] and branching model [30] have been proposed to capture user behaviors and video propagations in OSNs. Third, incorporating video propagation information, P2P [61], cloud [64, 32], and hybrid [58, 57] paradigms have been proposed to enhance networking distribution of shared videos in OSNs so that users can enjoy more frequent viewing experience.

1.3.1 Information Propagation and Video Sharing

Recently there have been pioneering data-driven analysis of information propagation in different kinds of OSNs, e.g., photos propagation in Flick network [9], likes and fans pages in Facebook [5, 52, 62], links and retweets in Twitter [23, 47, 10, 17, 28, 40, 67], and voting in Digg [28, 51, 29]. Rorigues *et al.* [47] studied the propagation of URL links posted in Twitter, using large data gathered from Twitter. They presented the distribution of height, width, and size of propagation trees. Sun *et al.* [52] studied distribution chains and large-scale cascades across Facebook. Scellato *et al.* [49] focused on the geographic property of social cascades of videos by tracking social cascades of YouTube links over Twitter. Cha *et al.* [8, 9] conducted a large-scale measurement study on the Flickr social network. They found that even popular photos spread slowly through the network. While we found that the videos in an OSN spread much faster. This comparison indicates that different kinds of contents propagate in diverse patterns in OSNs. A very recent work [58] studied the propagation-based social-aware replication strategies for social video contents. They found similar power-law video popularity distribution in another large OSN in China. Instead of making a comprehensive measurement and analysis as we do in our thesis, they focused on the system optimization based on these new traffic patterns.

Comparing with the characteristics of the videos shared in VSSes can provide us more in-depth understanding of the characteristics of the videos shared in OSNs. There are a lot of measurement works on the VSSes videos either by crawling meta-data their websites [7, 11, 16, 14] or tracing traffic from a set of network routers/switches [19, 70]. Cha *et al.* [7] presented an in-depth study of the static popularity distribution of videos in two large-scale VSSes, finding that the video popularity shows a power-law waist with a long truncated tail for huge unpopular videos. Cheng *et al.* [11] also studied the distribution of videos in YouTube, and found similar results. They further presented other statistics of YouTube video files such the length, bitrate, and size. Figueiredo *et al.* [16] found that the popularity growth pattern depends on the choice of the video dataset. Crane *et al.* [14] categorized videos by their popularity evolution patterns into three types: viral videos, quality videos, and junk videos. Gill

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

[2] http://t.qq.com/

et al. [19] and Zink *et al.* [70] both analyzed YouTube video requests from a campus network and observed that the video requests follow a Zipf-like distribution. Our work focuses on similar aspects as pervious works, yet aiming to demonstrate the distinctive characteristics due to the word-of-mouth based sharing mechanism. In particular, we find more skewed popularity distribution, and more complex popularity evolution patterns.

1.3.2 Propagation Modeling and Popularity Prediction in OSNs

To characterize information propagation, a series of epidemic models have been proposed in the literatures [42, 36, 18, 68, 56, 20], among which the SIR (Susceptible-Infectious-Recovered) model is a typical example. Newman *et al.* [42] solved SIR cases in which the time and probability distribution are nonuniform and correlated. Liu *et al.* [36] investigated the SIR model in scale-free and random networks, claiming that a substantial proportion of nodes can never be infected and scale-free networks are more robust against spreads of infection. Ganesh *et al.* [18] further examined the effect of network topologies on the spread of epidemics. These works again have not considered modern social networks, not mention video link propagation. Substantial revisions are needed to apply the SIR model in this new context, as we will show in Chapter 3.

The works on video requests modeling and generation date back to the 1990's, when online video services just became popular. Two early works GISMO [25] and MediSyn [54] modeled the video access patterns in traditional (compared with UGC-based video services) video services. To capture the popularity dynamics of modern user-generated videos, like in YouTube, a model was proposed by Borghol *et al.* [6]. They found in empirical data that the videos' relative popularity stays stable in three different phases, and thus can simply distribute user requests according to three popularity distributions. Nowadays, OSNs are becoming major portals for users to share and view videos. To this end, we provide a novel approach to model these unexplored requests based on video propagation information.

There have also been efforts towards prediction in the OSN context [17, 23, 29]. Galuba *et al.* [17] proposed a propagation model that predicts which users are likely to mention which URLs in Twitter. Hong *et al.* [23] treated the retweets prediction on Twitter as a classification task. They investigated a wide spectrum of features to determine which ones can be successfully used as predictors of popularity. Kooti *et al.* [27] investigated the prediction of emerging social conventions on Twitter. The most close research to ours was conducted by Lerman *et al.* [29]. They predicted popularity of news on Digg, by incorporating aspects of the web site design. They showed that their model-based prediction improves prediction based on simply extrapolating from the early votes. Our work has been inspired by these studies, and differs from theirs in that we focus on video, which, as one of the most information-rich data objects, preserves unique characteristics that are yet to be examined for prediction.

1.3.3 Enhancement of Video Sharing Systems

Works on system enhancement have been conducted from two perspectives: finding interested video content more easily and viewing videos more smoothly. The former one mainly involves video recommendations [60, 37]; the latter one mainly involves video server developments, which is the focus of this section. Incorporating video propagation information, P2P [61], cloud [64] or hybrid [58] paradigms have been proposed to enhance distribution of shared videos in OSNs so that users can enjoy more frequent viewing experience.

Wang *et al.* [61] advocated to utilize social reciprocities among peers for efficient contribution incentive and upload scheduling, to enable efficient social media sharing with low server costs. They designed efficient peer-to-peer mechanisms for social media distribution based on a combination of peers social relationship and historical contribution levels. Wu *et al.* [64] introduced a proactive, online algorithm to scale social media streaming applications for operating in geo-distributed clouds. Specifically, they formulated an optimal content migration and request distribution problem, with long-time and one-shot flavors, respectively. Efficient methods were proposed to solve the one-shot optimization, and a novel $\Delta(t)$-step lookahead mechanism was designed with guarantees to adjust the one-shot optimum to the offline optimum, which is based on solid theoretical analysis. Wang *et al.* [58] proposed a propagation based social-aware replication framework using a hybrid edge-cloud and peer-assisted architecture, namely PSAR, to serve the social video contents. Specifically, they designed three replication indices: a geographic influence index, a content propagation index and a social influence index, which can guide the region selection, bandwidth reservation and cache replacement in the joint edge-cloud and peer-assisted replication.

1.4 Organization of this Thesis

The remainder of the thesis is structured as follows:

- In Chapter 2, we examine the characteristics of video requests from OSNs, comparing them with that in VSSes. In addition, we build a simple but effective model to generate synthetic video requests to make them exhibit similar characteristics with the real trace.
- In Chapter 3, to further understand the underlying reasons behind the characteristics of video requests, we study the video propagation process and user behaviors in OSNs from the perspectives of measurement and modeling.
- In Chapter 4, we present an initial study on the popularity prediction of videos shared in OSNs. Particularly, we confirm the ineffectiveness of conventional methods (e.g., ARIMA, kNN, and MLR) in the OSN context, and propose an effective propagation-based prediction model.

- In Chapter 5, we propose a framework, called SNACS, for OSN operators to cost-effectively enhance their video viewing experience by leveraging content cloud service.
- In Chapter 6, we conclude the thesis, and also discuss some future works.

Chapter 2

Characteristics of Video Requests from OSNs

2.1 Introduction

Although the videos in OSNs are most from VSSes, OSNs provide quite different video sharing mechanisms, leading to distinctive user access patterns. In VSSes, videos are mainly viewed via related videos, their search engines and front pages [69]; whereas in OSNs, videos are viewed via friends' direct shares. In VSSes, users can hardly discover niche content, or content that is not properly categorized or ranked. Instead, recommendation strategies play an important role. In OSNs, each video has a relatively fair opportunity to be propagated along friendship, and both the video quality and the property of the target OSN can affect the video's popularity. Compared with plenty of research on the video requests in VSSes (e.g., YouTube), the characteristics of requests from OSNs have not yet been comprehensively measured at large scales, not to mention video requests modeling and generation.

To unveil the characteristics of video viewing in OSNs, we closely collaborate with RenRen to analyze its server access logs. Starting from March, 2011, we collected the detailed user video viewing and sharing behaviors over four months. Leveraging the proprietary data, we characterize the user requests from the aspects of video popularity[1] distribution and evolution, unveiling a number of distinctive characteristics compared with the video requests directly from VSSes. In particular, we observe that OSNs amplify the skewness of video popularity so largely that about 2% most popular videos account for 90% of total views (compare to 20%-90% in conventional YouTube statistics [7]). We also observe that the video requests distribution exhibits perfect power-law feature except for one hundred most popular videos, where in YouTube, it exhibits a power-law waist with a long truncated tail for huge unpopular videos [7]. For popularity evolution, we find that plenty of very popular videos stay a long term

[1]We use the number of views to denote a video' popularity.

dormancy before a sudden burst in requests. Our dataset can be used to explore the propagations of individual videos, by tracing the viewer-sharer relationships. We can not only see the dynamics of popularity but also what happened there.

To further understand the characteristics observed in the empirical analysis, we build an emulator to model the video viewing and sharing behaviors in OSNs. Our emulator generates user requests that well capture the video popularity distribution and dynamics observed in our empirical data. Using this emulator as a tool, we find that although the top popular videos mostly have large *sharing rate* (sharing rate (ShR_i) is defined as the probability viewers will reshare the video i after viewing), videos with high ShR do not definitely gain large user requests. We also confirm that the large difference of the number of sharers' friends is a major reason for the video popularity dynamics. Our emulator can also be used to synthesize user requests for examining video sharing with assistances from peer-to-peer, content distribution networks, or cloud platforms [58, 59].

2.2 Measurement Methodology

This section introduces our data set, which is also used in other chapters in this thesis.

2.2.1 The RenRen Social Network

Launched in 2005, RenRen is the earliest and so far the largest OSN in China. RenRen can be best characterized as Facebook's Chinese twin, implementing Facebook's features, layout, and a similar user interface. Like Facebook, RenRen's users can post video links from VSSes. Unlike Facebook, RenRen has two unique features that make it an attractive platform for our study. First, while RenRen users have full privacy control over their private profiles, their shared videos are public and thus can be crawled. For example, each individual user has a page that list all shared videos with their statistics, including the number of views and shares within RenRen. Second and perhaps more importantly, RenRen provides certain proprietary information about users' viewing behaviors, as we will be exploring.

Since the shared videos in RenRen are from VSSes [2], many characteristics of video viewing in RenRen are of little difference than that in VSSes, such as object sizes, user activities (e.g., VCR-like stop/fast-forward/rewind/pause), and the bit-rate of streaming objects. Yet OSNs and VSSes provide different sharing mechanisms. While videos in VSSes are mainly viewed via related videos and their searching engines [24], videos in OSNs are viewed via friends' shares. Video sharing in RenRen is based on the friend relationships. Initially, a user posts a video link from a VSS in RenRen; This link immediately appears in her/his friends' main

[2]The videos discussed in this thesis are those linked from third-party VSSes. They do not include private videos that users upload directly in RenRen, which account for a relatively small portion and are usually only shared among direct friends.

page as a "News Feed"‘ in chronological order; Meanwhile, this shared video is also listed on the sharer's home page, which lists all her/his ever shared contents. Then her/his friends will probably click the shared video appeared in "News Feed"; or they may regularly visit friends' home pages to watch those shared videos, though this frequency is much lower than the first way. A video can be further propagated if some viewers share the link again.

2.2.2 Data Set

Our dataset consists of proprietary data provided by RenRen as well as the crawled data from Youku and RenRen websites.

To get the statistics of the videos shared in RenRen, we randomly crawl the information of 25997 shared videos from 10000 individual users. For each video in our data set, we crawl the detailed information, including the sharing time, the URL in the VSS, total shares, and the total views in RenRen. As comparison, we also crawl the statistics of these videos in Youku using the URLs crawled from Renren. We use Youku as the representative VSS in our study, both because almost 80% of shared videos in RenRen are from this site, and also because it enables access to certain valuable detailed information, including the number of their views, likes, dislikes, comments, favorites and external links. Using these data, we can know that for those videos ever shared in OSNs, what percentage of the requests from OSNs among the total requests. We can also analyze whether a popular video in a VSS will be still popular when it is shared in the OSN.

To further understand video spreading in OSNs, we closely collaborate with RenRen, to collect and analyze its video-related user behaviors [3]. Like Facebook, its users primarily interact with information through an aggregated history of their friends' recent activity, called the "News Feed". For video sharing, typically a user may post a video link from a VSS, and the link will appear in its friends' "News Feed". Some friends may click and view the video, and such viewers can then decide whether to re-share the video. If they click the "share" button, the video link will appear in their friends' "News Feed" and hence the video can further propagate.

The data collection process works as follows: when a user clicks a video link shared by her/his friend, a record will be sent to a log server; and the data format is: (*Starting Time*, *Video URL*, *Viewer ID*, *Direct Sharer ID*, *Initial Sharer ID*). We use an example in Fig. 2.1 to illustrate the video propagation and the corresponding log record. Initially at time T_0, user A (denoted as U_A) posted Video x (denoted as V_x) from a VSS, and then a record (T_0, V_x, U_A, U_A, U_A) is sent to log server. Since U_A is the initial user, both direct sharer and initial sharer are itself; At time T_1, U_B viewed V_x through the share link created by U_A, and then U_B further shared V_x after watching it; and then a record (T_1, V_x, U_B, U_A, U_A) is sent to log server. Also as U_A is the initial user, the initial sharer is U_A; At the Time T_2, U_C viewed V_x through the share

[3] To protect user privacy, we translate real User IDs by some hash function, and user IPs are not included in our date set.

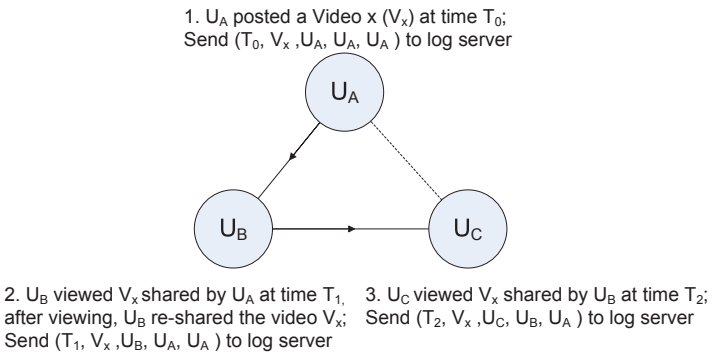


Figure 2.1: Illustration of video propagation and corresponding logs

link created by U_B. A new record (T_1, V_x, U_C, U_B, U_A) is sent to log server. Note that there is a dotted line without any arrow between the friends U_A and U_C, which means although U_A's shared video was exposed in U_C's "News Feed", U_C did not click it maybe because s/he is offline.

Table 2.1: Summary of trace in one-day period

Views	Shares	Users	Videos	New Videos
12,432,708	1,628,852	3,514,461	201,517	71,236

Using (Video URL, Viewer ID), we can extract the number of views of any video in each day. We then use this information to analyze the video popularity evolution patterns, and test views-based prediction models. Using (Video URL, Viewer ID, Direct Sharer ID), we can examine the share-view relationship between two friends. And together with the initial Sharer ID, we can restore a video's propagation process. Such information is useful to analyze the reason underlying the popularity evolution patterns, and inspire the design of our propagation-based prediction model. Our study is based on a one-month trace that began from March 24^{th}, 2011, since we find that most requests of a video are generally cumulated in the first month, and after that the daily requests decline to a very small scale. Table 2.1 presents the statistics in a typical one-day period (March 24^{th}, 2011) during the measurement. Our records covered all video requests in the measurement period. During the one-month period, we recorded about 370 million views and 49 million shares.

2.3 Characteristics of Video Popularity

2.3.1 Popularity Distribution

The Pareto principle (also known as the 80-20 rule) is widely used to describe the skewness in distributions. For example, the analysis of YouTube shows that 10% of the most popular

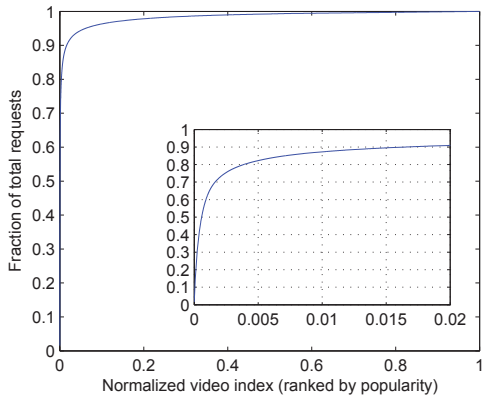


Figure 2.2: Skewness of requests across all videos

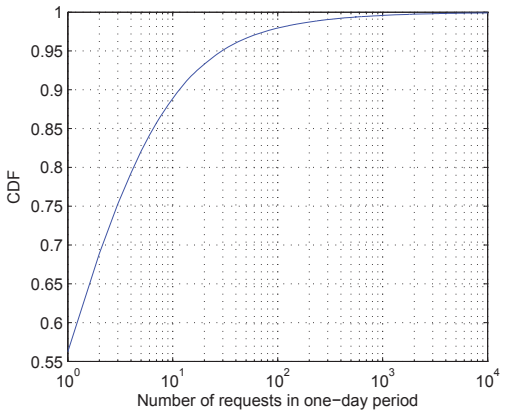


Figure 2.3: Distribution of requests frequency

videos account for 80% of user requests [7]. Then one may wonder whether social-network-based sharing results in a less skewed request distribution across the videos in OSN, since all videos have equal chance to become popular As shown in Fig. 2.2, we can see a counter-intuitive result that 0.4% videos account for more than 80% of requests; the rest 99.6% of the videos, on the other hand, only account for 20% requests (the x-axis of this figure represents the videos sorted from the most popular videos to the least popular ones, with video ranks being normalized between 0 and 1). An intuitive explanation is that the popular videos will become even more popular since the users are more likely to recommend these videos to their friends. The unpopular videos, however, will fade out very soon in the social communities. The difference of video's attraction can be magnified over the propagation process along friend links. As shown in Fig. 2.3, we can observe that 80% of videos only have less than 4 requests and 90% of videos only have less than 10 requests in one-day period. Considering the great number of RenRen user, this result confirms that the social-network-based sharing will result to a more skewed popularity distribution across the videos.

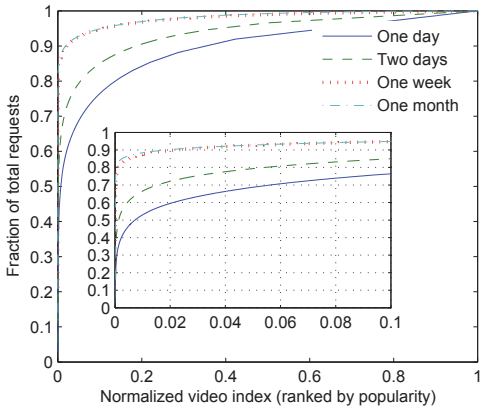


Figure 2.4: Skewness of requests across the videos initially shared in the same day

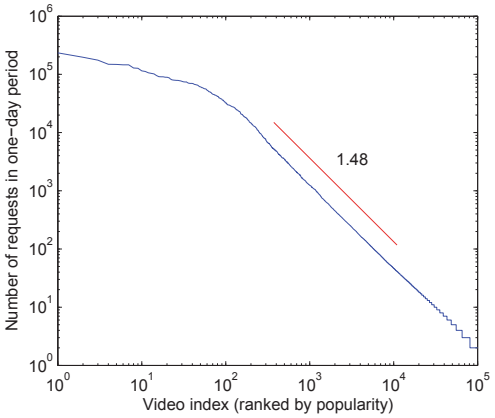


Figure 2.5: Log-log plot of number of requests versus video ranks

To further analyze user requests distribution, we also take a closer look at the videos that are initially shared in the same day (March 24^{th}). Since most users are more interested in newly updated videos, this analysis will avoid the possible bias due to the video age. We count the cumulative requests of those videos within one day, two days, one week and one month separately since they were initially shared in RenRen. We plot the results in Fig. 2.4. Similar to Fig. 2.2, the popularity of those videos also exhibits such a high skewness that the top 2% popular videos account for 90% of total requests. We also notice that the skewness increases as the time-window increases, and becomes converged after one week.

The power-law model has been increasingly used to explain various statistics appearing in the computer science and network systems. A distinguished feature of power-law is a straight line in the log-log plot. To check the power-law pattern of videos in OSN, we first plot the requests against ranks based on all requests in a given day (March 24^{th}) and show the result in Fig. 2.5. We find that except for the top-100 videos, the plot exhibits perfect power-law pattern except for one hundred most popular videos (as a comparison, the video popularity

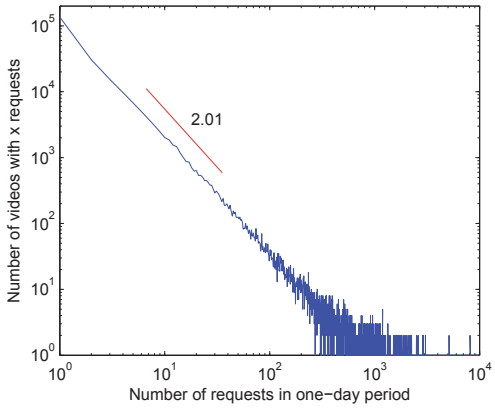


Figure 2.6: Log-log plot of frequency versus number of requests

on YouTube shows a power-law waist with a long truncated tail for huge unpopular videos [7]). The fitted power-law exponents is also shown in the figure. Normally, the power-law distribution arises from a rich-get-richer principle. And this principle seems perfect to explain the popularity distribution of videos shared in OSN. The probability of a request for each video is proportional to the existing shares of this video, because more shares mean that more users can discover this video and thus watch this video. We also show the popularity distribution in another way in Fig. 2.6–a plot of frequency against requests. As expected, it also shows the power-law behavior. In Section 2.4 we design a model to simulate the video propagation process in the OSN and the model generates very similar popularity distribution like that in Fig. 2.5, which indicates that the power-law behavior is the natural shape for videos shared in OSN.

Based on the above analysis, we can see that the social-network-based sharing has changed the pattern of video popularity in the existing video sharing systems. In particular, users' interests are significantly cumulated to a few very popular videos. These videos are widely shared/recommended by many users and become even more popular. The unpopular videos, on the other hand, will fade out very soon in such an environment with "user selection". To better understand this feature, it is thus important to clarify the video popularity evolution in OSN especially for those popular videos.

2.3.2 Popularity Dynamics

Although the videos show similar popularity distribution along the time, we find that their relative positions in the distribution are highly non-stationary. In other words, some current rarely-requested (or low-ranked) videos may become frequently-requested (top-ranked) videos in the near future.

After every 500,000 aggregate requests, we snapshot the number of added views for each video that was initially shared on March 24th. Fig. 2.7 shows scatter plots for the number

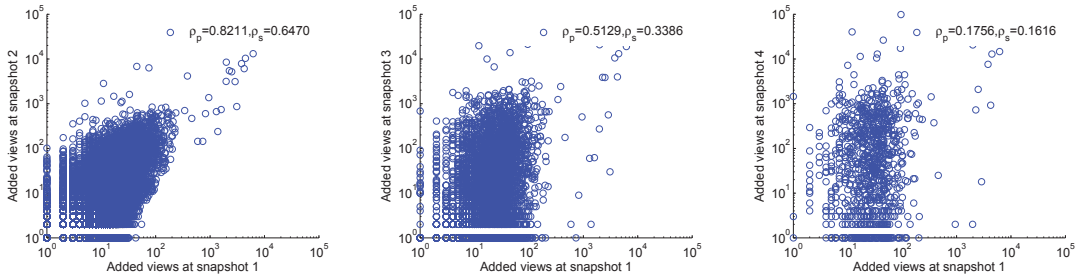


Figure 2.7: The number of added views at snapshot 1 versus snapshots 2, 3, 4

of added views received by a video at snapshot 1 and snapshots 2, 3, 4. It also shows the Pearson correlation coefficient (ρ_p) [46] and Spearman's rank correlation coefficient (ρ_s) [38] [4] between the number of added views at different snapshots. With our notion of added views at a snapshot, this figure illustrates the change in viewing rate between two snapshots. Overall, we observe substantial non-stationarity in the popularity of individual videos. Although the added views of two adjacent snapshots show weak correlation, it is not the case for two non-adjacent snapshots. The correlation declines quickly with the distance of two snapshots. Note that the scatter plots have fewer points for later snapshots owing to the increasing videos that received no views in these snapshots (and hence are not shown on the log-log plots).

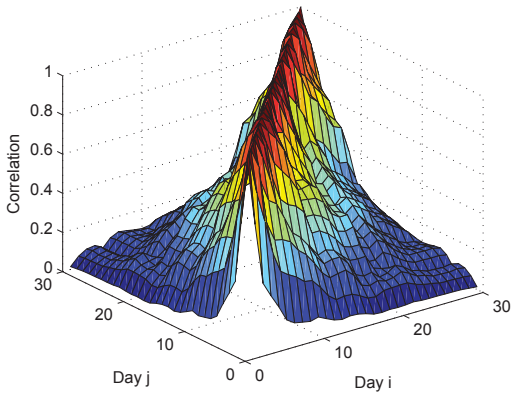


Figure 2.8: Correlation between early and later views

To further explore popularity dynamics, we examine the correlation between early and later views, which is a simple but effective indicator to show whether the number of early views is an effective factor in the prediction of future views. In the span of 30 days, we compute the

[4] ρ_p has been widely used for measuring the strength of linear dependence between two variables, and ρ_s assesses how well the relationship between two variables can be described using a monotonic function. The ranges of both ρ_p and ρ_s are from -1 to 1, where a value greater than 0 indicates positive correlation, and less than 0 indicates negative correlation. The value of 0.8 or more is considered to reflect strong positive correlation [7].

Pearson correlation coefficients [46] in terms of the number of views across the top-2% videos at early and later days and show the result in Fig. 2.8. Both early day and later day vary from 1 to 30. We can see that the correlation is very high when the later day is within 2-3 days of the early day, and becomes very small when the later day is out of this range. This contradicts the conclusion in the previous works that the correlation is still very high even when the later day is tens of days after the early day [7].

2.3.3 Popularity Evolution

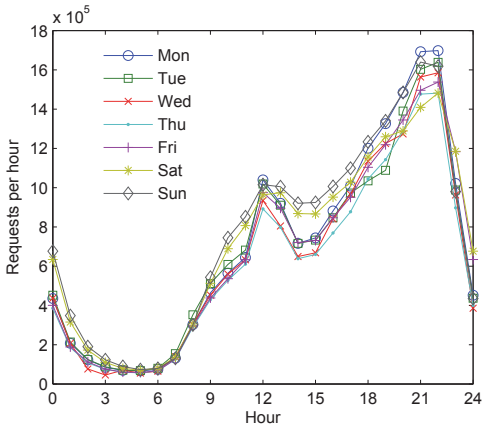


Figure 2.9: Aggregated views of all videos in each hour over one week

In this subsection, we measure the video popularity evolution in OSNs. Like most streaming systems, our measurement shows that video accesses are not distributed uniformly but exhibit diurnal pattern. The diurnal access pattern defines how the number of accesses to a site varies during a given period of time, e.g., a day. To explore such pattern, we count the number of views in each bin of one hour over one week and show the result in Fig. 2.9. First, we find a little more requests during the weekend. Second, there are local peak values during the lunch time, especially in working days; Third, though it is common that the lowest requests appear in the early morning around 6am and highest requests appears appear around 10pm, the large gap between the peak value and the lowest value is out of our expectation. The diurnal access pattern is important for capturing the burst of resource consumption within a given time period. Due to the diurnal access pattern, the inter-arrival times within a given day do not simply follow the exponential distribution. Instead, it is better to be modeled as a nonhomogeneous Poisson process [38], where only the request arrivals within each bin can be modeled by a Poisson process. The request arrival rate for a given bin is computed based on the diurnal pattern specified by the user and the number of accesses within a day determined by system scale.

We also examine the evolution patterns of the videos of different popularity. We randomly select 1000 highly popular videos with more than $10,000$ total requests over three months; 1000

medium popular videos with 400 to 600 requests and 1000 unpopular videos with 10 requests. Fig. 2.10 shows the popularity evolution of these three groups of videos. As we can see, the unpopular videos can only attract some users in the first day of the share. After that, their popularity decreases very quickly to a near-zero level. This means the lifetimes of most unpopular videos are less than one day, and the OSN users will soon lose interest to these videos in their social communities. The medium popular videos, on the other hand, show a very different evolution pattern over time. Although they also achieve the peak value in the first day, the decreasing of popularity is much slower compared to the unpopular videos. For the highly popular videos, their peak values generally arrive after two or three days, and the video popularity will stay at a relative high level for a long time. For example, most of them still have more than 3000 requests after one month. The findings confirm our analysis in the IV.A that the unpopular videos will die out quickly and the popular videos will flourish over a long time. Note that there are many local bursts after the global peak along the evolution of the group of popular videos. This is because some popular videos stay dormant for a relatively long time (e.g., one week to several weeks) before the bursts of requests. These dormant videos are unique due to the unique information propagation in OSNs.

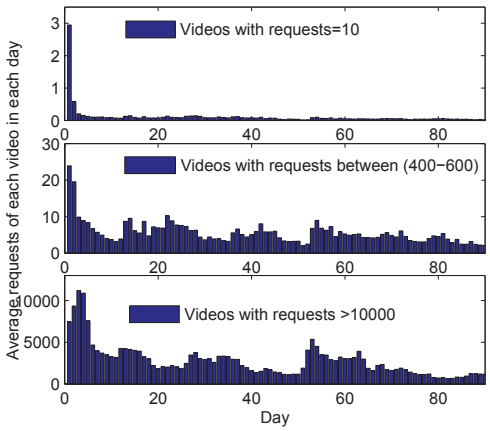


Figure 2.10: Popularity evolutions of groups of different-popularity videos

We then investigate how the popularity of individual videos evolves over time. Here we focus on the popular videos, since only 0.4% most popular videos almost determine the popularity of the whole system. Concretely, we examine the popularity growth of top 100 popular videos from those initially shared on March 24^{th}. All of them attract more than 10000 cumulative views till July 24^{th} and we approximately take this cumulative views as the total views of these videos. We find that the popularity of individual videos evolves differently. We select three representative videos and show how their popularity evolves over ten days in Fig. 2.11.

At a first glance, the three videos show different growth patterns. Video A (*golden video*) is the most popular video in our dataset. It kept an active-growth for a long time and still gained many requests even after one week. Video B (*silver video*) experienced a surge-growth

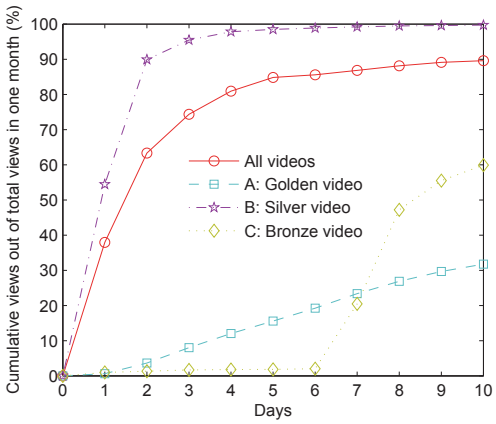


Figure 2.11: Popularity evolution since videos first appear in RenRen

over two days, acquiring 90% accesses, after that it turned to the sluggish state. Video C (*bronze video*) stayed dormant for nearly six days after it was first shared in RenRen; then it experienced a dramatic increase and attracted 40% of total views within 2 days. To explore which growth pattern is dominant, Fig. 2.11 also plots the aggregated trend of all 100 videos in our dataset. We make two key observations about the long-term growth. First, many videos show an active rise in popularity during the first few days after they are shared in RenRen. This is similar to the silver videos. Second, after the first few (e.g., 4) days, most videos, enter a period of steady linear growth. At that point, they have already attracted nearly 80% of requests. A previous measurement [7] explored the popularity evolution of videos in YouTube. But it spends a week for videos in Youku to gain 60% of total requests, which indicates that the popularity decays slowly in VSSes.

To cluster videos in terms of popularity evolution technically, we further use the k-means method, which is very popular in networking field. Given a set of observations (x_1, x_2, , x_n), where each observation is a d-dimensional real vector (y_1, y_2, , y_d), k-means clustering aims to partition the n observations into k sets (k $\leq$ n) S = S_1, S_2, , S_k so as to minimize the within-cluster sum of squares (WCSS):

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x_j \in s_i} \| x_j - \mu_i \|^2 \tag{2.1}$$

where μ_i is the mean of points in S_i.

In our example, y_i is the normalized cumulative views after i days since a video is initially shared. We use 30-days trace as the input in our example, thus here d=30. Our result shows that 40% of these popular videos are golden videos, 35% are silver videos, and surprisingly, up to 25% videos are bronze videos.

2.3.4 Popularity Comparison in OSN and VSS

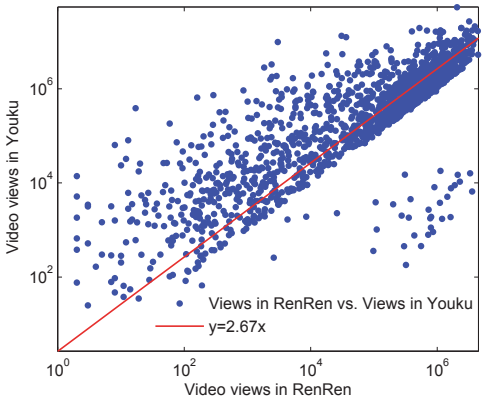


Figure 2.12: Video views in RenRen vs in Youku

We explore whether videos have similar popularity in OSNs and the VSSes. Among all video sharing sites that ever share videos in RenRen, Youku alone accounts for nearly 80% of all shared videos, and the top-5 video sites account for nearly 95%. Fig. 2.12 shows the video views in RenRen against that in Youku. They exhibit a relatively close relationship, which reflects that content itself plays a fundamental role in a video's popularity. The fitted curve suggests about 37% (1/2.67) video views on Youku come from RenRen. Note that we only consider the statistics of those Youku's videos that were ever shared in RenRen. The ratio of the videos in Youku that are ever shared in RenRen is around 11% in March, 2011. The ratio increased to 15% when we measured it in October, 2011. Considering the benefits of this interaction for both OSNs and VSSes, we believe that this ratio will steadily increase in the future.

Table 2.2: Correlation between the video views in RenRen and statistics in Youku

Correlation	Views	Likes	Dislikes	Comments
ρ_p	0.6937	0.2780	0.0188	0.2203
ρ_s	0.8493	0.6801	0.6186	0.6189

To analyze the correlations between the number of video views in RenRen and in Youku more specifically, we leverage two widely used metrics: Pearson correlation coefficient (ρ_p) and Spearman's rank correlation coefficient (ρ_s). The former has been widely used for measuring the strength of linear dependence between two variables, and the latter assesses how well the relationship between two variables can be described using a monotonic function. The ranges of both ρ_p and ρ_s are from -1 to 1, where a value greater than 0 indicates positive correlation, and less than 0 indicates negative correlation.

We can see that the value of ρ_s is 0.84, which is a relatively high positive correlation and confirms the result in Fig. 2.12. As a comparison, ρ_p is 0.69, which is much smaller than ρ_s.

It reflects that the video popularity in RenRen and that in Youku does not have a good linear correlation relationship. To understand what kind of videos are popular in RenRen, we show the correlation coefficients between the video views in RenRen and three other statistics in Table 2.2, including likes, dislikes and comments. We find the number of views in VSSes has the highest correlation. Since a video is generally first uploaded to a VSS and then discovered and shared by some users in OSNs after the video becomes popular, the video popularity in the VSS can be used as an alternative predictor for the potential popular ones when they are first shared in OSNs.

2.4 Synthetic Video Requests Generation

In this section, we provide a synthetic video requests generator by emulating the users' video viewing behaviors in OSNs. As shown in Fig. 4.1, our emulator is designed to assign a sequence of user requests to a set of videos, and the generated requests should capture the video popularity distribution and dynamics observed in the empirical data. Specifically, when a request comes, the emulator needs to decide the probability for each video to be assigned for this request. The probability is not a constant value, but calculated in real time, leveraging historical information of views and shares, as well as the video property such as sharing rate and viewing rate. Leveraging this emulator as a tool, we can analyze above measurement results and various factors that impact the video popularity in OSNs. It can also be used to generate synthesize user requests, which are very helpful for such related researches as video caching and peer-to-peer algorithms. For convenience, Table 4.1 summarizes the major notations in our modeling.

Table 2.3: Summary of major notations

Notation	Meaning
$\mathcal{T}$	random variable, indicating the inter-arrival time between two requests;
$\mathcal{D}$	random variable, indicating the out-degree of a sharer;
$\mathcal{U}$	random variable, with continuous uniform distribution $U(0,1)$;
M	number of videos in the system;
N	number of the total requests;
V_i	number of views of video i until current time;
S_i	number of shares of video i until current time;
ShR_i	sharing rate(ShR) of video i, indicating the probability users will share the video i after viewing;
ViR_i	viewing rate(ViR) of video i, indicating the probability users will view the video i shared by their friends;
BrF_i	branching factor(BrF) of video i, indicating how many friends view watch it, if one user shares a video i;
P_i	the probability that a request is distributed to the video i; it should be recalculated when a new request arrives;

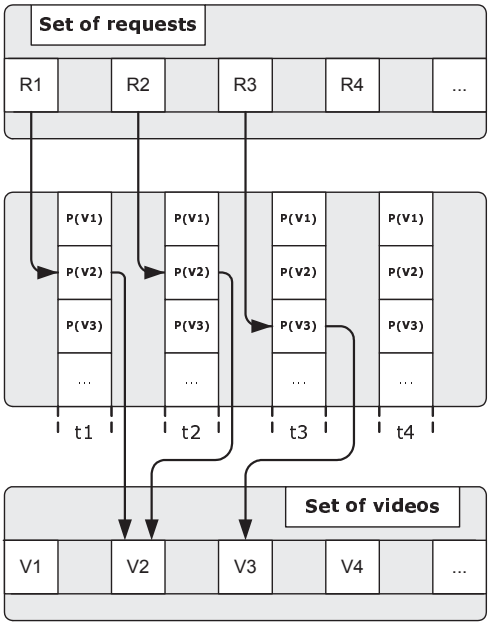


Figure 2.13: Framework of our emulator

2.4.1 Modeling Request Distribution

Note that our work in this chapter only focuses on the effect of the word-of-mouth on the dissemination of content. This word-of-mouth mechanism is widely adopted by a large number of OSNs (e.g., Facebook, Tittwer, Flicker, RenRen) as the basic information dissemination mechanism. It is also a distinctive feature of OSNs from traditional VSSes. Other mechanisms, such as featuring, links between content, and search results, are undoubtedly at play in some OSNs, but studying their impact requires a richer dataset and is beyond the scope of this work.

Now we model how the user requests are distributed across videos (P_i), in order to capture the video popularity distribution and evolution observed in the empirical data. One simple model is distributing requests according to a constant distribution, which is taken in some previous work [25]. This method must assume that the relative popularity of videos maintains stable in a certain time, which is not observed in our empirical data. Another alternative method is using rich-get-richer distribution mechanism[66]. In our case, it is expressed as $P_i = \frac{V_i}{\sum_{j=1}^{M} V_j}$, where M is the number of videos in the system, and V_i is the number of historical views of video i. The probability (P_i) that a new request is assigned to the video i is proportional to its V_i. The most important property of this process is that it generates a distribution following a power law in its tail, as is observed in our empirical dataset. Yet it it can not well reflect the video propagation process in OSNs and hence fail to capture the dynamics of propagation.

Besides viewing history, we try to leverage the video propagation process to provide a more reasonable request distribution mechanism. Our model assumes that users can only find and

view videos shown in the "News Feed" of their homepages. And all these videos are shared by their friends and be pushed to their "News Feed" in a chronological order. Therefore, videos will have more chance to be found and thus be viewed in the future, if they have already been shared by many sharers and at the same time these sharers have plenty of friends. Besides, another two factors are also important: how many of these potential viewers have already watched, and the probability that users will view the video if it appears in their "News Feed". We define E_i as the expected number of requests for all S_i existing shares of the video i.

$$E_i = \sum_{k=1}^{S_i} (D_i^k * ViR_i) \tag{2.2}$$

where S_i is the number of sharers of video i until now; D_i^k is out-degree of the k^{th} sharer of the video i; D_i^k indicates how many users the video i can be exposed to if the k^{th} sharer shares it. Note that similar to V_i and S_i, E_i is a variable changing over time. Accordingly, we get the following rich-get-richer equation:

$$P_i = \frac{E_i - V_i}{\sum_{j=1}^{M} (E_j - V_j)} \tag{2.3}$$

where the value of $E_i - V_i$ reflects the number of expected viewing requests in the future. Larger value of $E_i - V_i$ means more chances to be assigned for the next new request.

2.4.2 Emulator

Based on the above model, Algorithm 1 describes an implementation of our emulator for the video viewing and sharing behaviors in an OSN. It introduces a new request to system after each inter-arrival time ($\mathcal{T}$). We discussed the calculation of the distribution of $\mathcal{T}$ in the Section V. D. For each request, the emulator assigns it to the video i according to the P_i defined in Eq. 2.3. For the chosen video i, the number of its views (V_i) is increased by one. After that, this video should be judged whether to be reshared with the probability ShR_i. If so, the number of shares (S_i) of this video is increased by one, and the expected views (E_i) of this video is increased by $\mathcal{D} * ViR_i$.

In this emulator, the input parameters include the degree distribution of sharers ($\mathcal{D}$), video sharing rate (ShR$_i$), and viewing rate (ViR$_i$) for each video. The emulator distinguishes the attractiveness of videos by assigning different ShR$_i$ and ViR$_i$ to them. Note that it does not distinguish the difference of individual users in the probability of viewing and sharing the same video. The ViR$_i$ and ShR$_i$ are the properties of videos not users. The distribution of $\mathcal{D}$ reflects a topological property of the targeted OSN.

Algorithm 1 Emulator for video viewing and sharing behaviors in an OSN

```
for request = 1 to N do
    generate a inter-arrival time T;
    current time t=t+T;
    a new request arrives, and be assigned to the video i with the probability P_i;
    V_i++;
    extract a random variable U, with continuous uniform distribution U(0,1);
    if U < ShR_i then
        S_i++;
        extract a random variable D;
        for i=1 to D do
            extract a random variable U;
            if U < ViR_i then
                E_i++;
            end if
        end for
    end if
end for
```

2.4.3 Validation

We now validate the efficacy of our emulator in reflecting the video popularity distribution and dynamics by inputting the parameters extracted from real-world trace. For the number of videos and requests, we configure the same values (M=63,591 and N=2,905,276) as those in Fig. 2.5. To get the distribution of ShR, we collect all 12,432,708 views on March 24th and record whether there is a following share behavior after the view. We count the average ShR for each video separately and show the distribution of ShR along with the fitting function in Fig. 2.14. We find the ShR of 90% videos are less than 0.30, with the average value 0.11. Instead of parameterizing ViR and $\mathcal{D}$ separately, the emulator needs only the product of them, which is denoted as the branching factor (BrF). To get the distribution of BrF, we collect all 1628852 shares created on March 24th and count their followed requests over three months. The distribution along with the fitting function are shown in Fig. 2.15. We call the above BrF distribution as the *basic BrF*. To distinguish the values of BrF across different videos, we configure each video with the product of the basic BrF and a random factor that uniformly distributes in (0.5-1.5).

With the above parameters as the input, we first examine the video popularity distribution of the generated user requests. One key observation from the empirical data about the video popularity distribution is the power-law distribution under the plot of video views versus ranks. Another key observation is that the popularity shows high skewness. As shown in Fig. 2.16, we can see the simulation result and real-world data are pretty matched. We also count the skewness of the video popularity distribution, and the simulation result shows that the top-2% videos account for 85% of the total requests, which is very close to our observation (2%-90%). The most popular videos in our simulation are not as popular as that in the empirical data.

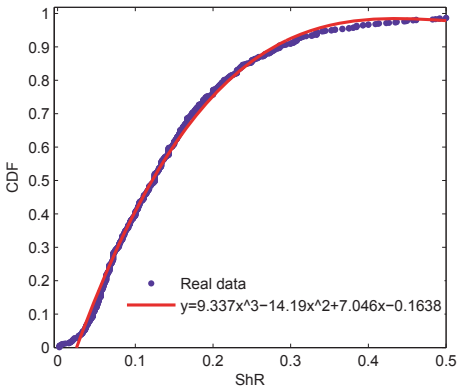


Figure 2.14: Distribution of ShR

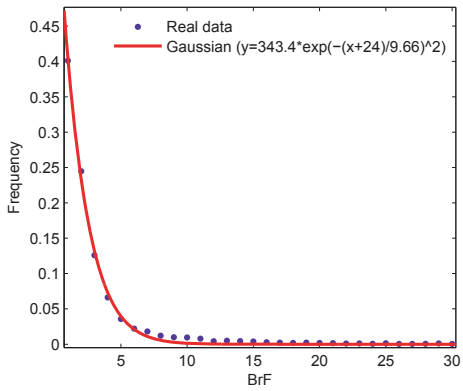


Figure 2.15: Distribution of BrF

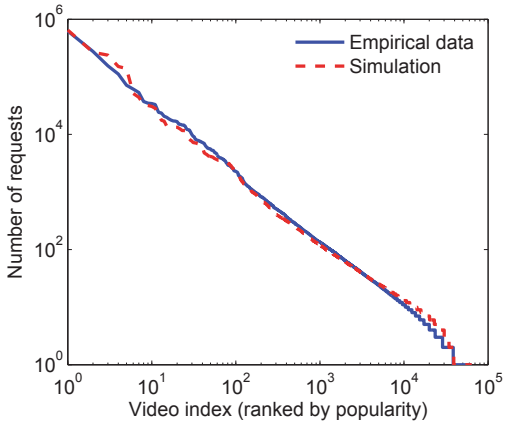


Figure 2.16: Comparison of popularity distribution

In our examined OSN system, a small number (e.g., 120) of popular videos is featured as the most popular videos and are listed on a public page. This behavior can further increase the popularity of the featured videos. We do not include this exogenous factor in our current emulator, considering that it is not a general case in other systems and also does not affect the overall pattern of user requests.

Then, we examine the popularity dynamics. We calculate the Pearson correlation coefficient (ρ_p) and Spearman's rank correlation coefficient (ρ_s) between the numbers of added views at different snapshots, and shown the results in Table 2.4. Overall, the coefficients between the simulation result and the empirical data are very close. A closer look will find that our emulator produces less dynamic. This is because our simulation simply configures each video with a constant ShR that never changes over time. In fact, the ShR of different videos change over time with diverse patterns, which can also affect the video popularity dynamics. Considering the complexity of ShR evolution pattern yet much less importance to the popularity dynamics, we do not model the evolution of ShR in the current emulator and leave it for our future work.

In summary, since the generated requests are the consequence of the emulation algorithm and the input parameters derived from empirical data, rather than directly fitted from the empirical data, the above tests validate the efficiency of our emulator in reflecting the video popularity distribution and dynamics.

Table 2.4: Correlation coefficients between the added views at different snapshots

(ρ_p, ρ_s)	S1 vs S2	S2 vs S3	S3 vs S4	S1 vs S3	S1 vs S4
simulation	(0.845,0.722)	(0.710,0.473)	(0.774,0.430)	(0.623,0.383)	(0.183,0.175)
empirical	(0.821,0.647)	(0.708,0.462)	(0.773,0.428)	(0.512,0.338)	(0.175,0.161)

2.4.4 Analysis

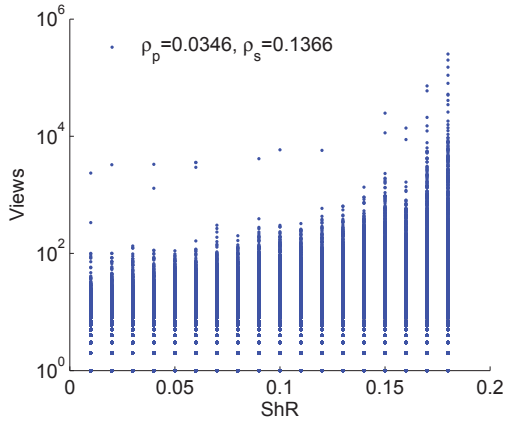


Figure 2.17: Impact of ShR

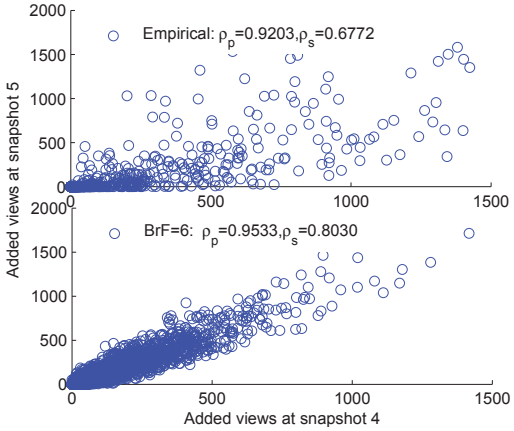


Figure 2.18: Impact of BrF

Based on the verified emulator, we first analyze the impact of ShR value to a video's popularity. Fig. 2.17 shows the scatter plots for ShR and views. On one hand, we find the high

ShR does not definitely result in many requests. As shown in this figure, the correlation coefficients between them are very low. On the other hand, almost all frequently-viewed videos have high ShR. For example, 87.8% videos which gain more than one thousand views have ShR with value 0.17 or 0.18. It indicates the popularity of a video shared in OSN exhibits much randomness and unpredictability, for example owing to the randomness of friends' number of sharers.

We then analyze the impact of BrF to the video popularity dynamics. $\mathcal{D}$ can range from zero to a large number, leading to the dynamics of BrF. We assume that the dynamic of BrF is regarded as one main reason for the popularity dynamic. To illustrate this, we set the BrF with a constant value (e.g., 6) for all 63,591 videos and show the scatter plot of added views in snapshot 4 versus snapshot 5 in Fig. 2.18. For comparison, Fig. 2.18 also presents the result with BrF extracted from the empirical data. Here we only choose snapshot 4 and 5 as the example, since other snapshots show the similar results. Both the plot and the correlation coefficient verify our hypothesis that the BrF dynamic is the major reason for the popularity dynamic.

2.4.5 Discussion

Our emulator can be easily extended to include more features, such as the properties of individual request sessions (session duration, video bitrate, VCR-like user interaction, and geographical information). We can also add the new video introduction process. For each request, a new video may be added to the system with the probability $\frac{1}{k}$, as we find that the number of requests and the number of new videos roughly show a linear relationship in our empirical data (k=175). Once a new video i is introduced to system, the emulator configures E_i=$ShR_i * \mathcal{D}$, S_i=1, and V_i=0.

We only consider the impact of dynamics of BrF to the popularity dynamics in this work. The diverse ShR evolution patterns can also impact the video popularity dynamics. We found that modeling these complex patterns is a great challenging work that cannot be well explored in this work, thus the model validation and analysis parts simply configure a video's ShR with a constant value. We would like to make an in-depth study of ShR's evolution and analyze its impact on the popularity dynamics in our future work. Our work also does not analyze the impact of ViR and $\mathcal{D}$, due to the lack of ViR information from real systems. Instead, we only analyze the product of them.

2.5 Summary

In this chapter, we provided the first major stab at characterizing these requests, by analyzing the logs of video viewing and sharing behaviors in a large-scale OSN over several months. Our measurement unveiled both static and temporal characteristics of video requests from OSNs,

highlighting several distinctive features from the requests directly from VSSes. To better understand the characteristics observed in our empirical data, we built an emulator to model video viewing and sharing behaviors in OSNs. Although simple, our emulator well captures the observed characteristics in the empirical data, including the video popularity distribution and dynamics. Leveraging this emulator as the tool, we analyzed the impact of dynamics of branching factor and the videos' sharing rate to the user requests patterns. Future work involves the measurement of geographic locality of user requests, study of the model's applicability for other contents (e.g., articles and pictures) shared in OSNs, and the impact of the rising traffics to the system design such as P2P, CDN and cloud.

Chapter 3

Video Propagation in OSNs

3.1 Introduction

A common video propagation process is like this: Initially, a user shares a video link to an OSN directly from a VSS. Immediately, this user's friends can find this video in their newsfeed, and some of them watch this video. After that, some portion of these viewers will share this video and can recommend it to their friends. We assume a user never views or shares a video more than once [1]. Some popular videos are generally brought to an OSN by multiple users. Therefore, the propagation structure of a video becomes a forest, which consists of multiple rooted trees. Each node is a user who ever shared or watched this video. The roots of these trees are the users who brought the video to the OSN. If user A watched a video shared by user B, then a direct edge (from A to B) is added to the tree.

To specify this process, we give the following definitions like that in the work [20]. We call the users in the root of a propagation tree $initiators$. These users are the ones who independently shared the video directly from VSSes. We call the users who re-shared the video $spreaders$. We call the users who watched the shared video $viewers$. Since spreaders generally watched the video before re-shared it, most of them are also viewers. The definition of $viewers$ is different from that in [20]. In their model, the $viewers$ are exclusive of spreaders. We define a video's $popularity$ as the number of its $viewers$. We define the $BranchingFactor(BrF)$ as the number of $viewers$ directly follow a $spreader$. We define the $ShareRate(ShR)$ as the ratio of the $viewers$ that re-share the video after watching it.

3.2 Characteristics of Video Propagation

By measurement analysis, this section explores the video propagation structure and investigates what factors could potentially affect the video spreading process.

[1] This assumption is reasonable, since in our dataset a user views the same video more than once with less than 0.1% probability.

3.2.1 Propagation Structure of a Popular Video

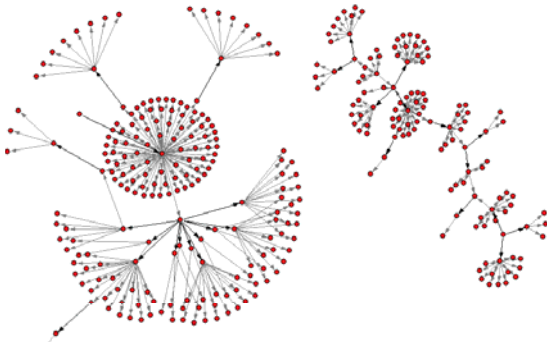


Figure 3.1: Propagation illustration of one video with two initiators

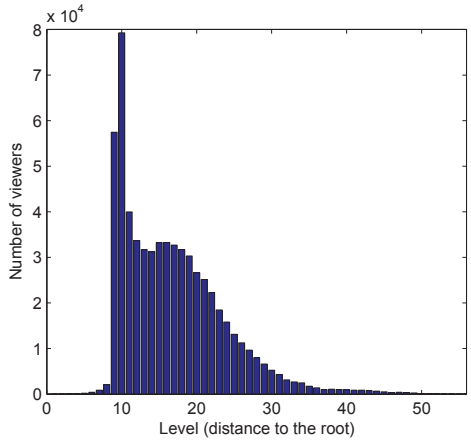


Figure 3.2: Viewers evolution along the level of the tree

This subsection presents a detailed propagation structure of a popular video, which consists of 1022 initiators, 153,185 spreaders, and 995,707 viewers over one month propagation. By studying this representative video, we will get some important conclusions, which motivate our models. An illustration of its propagation structure over several hours is shown in Fig. 3.1. We choose two among its 1022 trees. They exhibit unique propagation patterns. We can observe some super spreaders in the left tree, who are followed by many viewers. The super spreaders and especially the ones that appeared in the early spread stage generally play an important role in the further explosion of the tree. In fact, the left tree has finally become the largest tree among these 1022 trees. We can also observe that a large portion of viewers does not share this video.

Then we show some quantitative results of its biggest component, which consists of 96,220 spreaders, and 625,493 viewers. Fig. 3.2 shows the number of viewers at each level. Fig. 3.3 shows the BrF evolution along the level of the tree. Fig. 3.4 shows the ShR evolution along

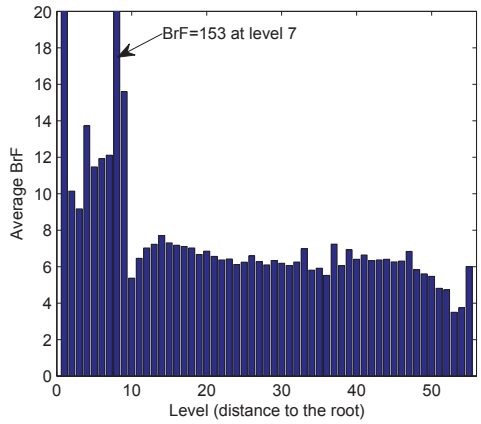


Figure 3.3: BrF evolution along the level of the tree

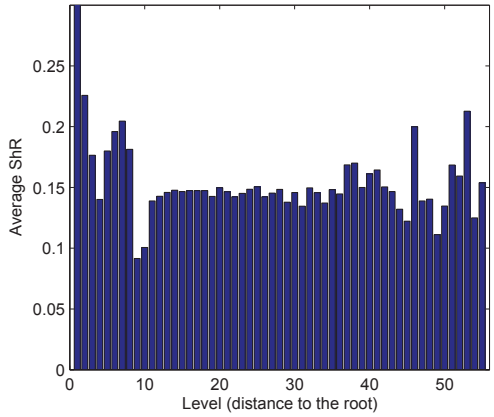


Figure 3.4: ShR evolution along the level of the tree

the tree. The most important observation from this example is that BrF and ShR is level-independent: except for a few exceptional values, the values of BrF and ShR are less correlated with their distance to the root. In fact, this finding is also observed in most other video spreading trees in our data set. This is important for our model, since we can simply set the same BrF distribution for all spreaders and probability of ShR for all viewers, regardless of its distance to the root. Actually, at the beginning of our research, we wonder that the BrF and ShR should decrease along the level of the tree, because the distance of a viewer to the root of the tree may reflect the viewing time. Yet we find two facts which can explain this wonder. First, the relationship between the distance of a viewer to the root and its viewing time is very weak. Second, within a short time, the attraction (reflected by the ShR) of a video has no obvious change.

We also notice that there are some exceptional value in these figures. For example, the viewers suddenly increases to a dramatic high value at level 8 in Fig. 3.2. The average BrF is

153 at level 7 in Fig. 3.3. The BrF is obviously larger at first 7 levels than latter levels. These exceptional values can be explained by one fact that some spreaders have extreme large BrF. We here list the largest five BrF in this tree: 53946, 53532, 1443, 410, 384. There are super nodes (such as movie stars, and public figures) in RenRen, some of them have even more than one million fans. Once they share a video, they get very large BrF and thus accelerate the video spread process in an unusual way. The large spreading trees of a video often have super spreaders at the early stage of propagation. While those small trees generally do not reach to these super spreaders before their propagations die out.

3.2.2 Influencing Factors

In this subsection, we further give some statistics across videos and analyze influencing factors to the video popularity. We first analyze initiators. A video can be shared in OSNs only if some users of OSNs initiate the share directly from VSSes. Intuitively, the number of initiators should have a positive relationship with video's popularity. Surprisingly, the Pearson's coefficient between them is only 0.189. It suggests that the number of initiators does not reflect or affect the video popularity. There are two possible reasons for this measurement result. Fig. 3.5 shows the distribution of initiators for individual videos. Since most videos have an extremely small number of initiators, this parameter is unsuitable to predict a video's popularity in OSNs. We can find that 90% of videos have less than 10 initiators. In addition to the nature of video's attraction, the number of initiators is also affected by the characteristics of original websites, such as the scale and convenience of share button.

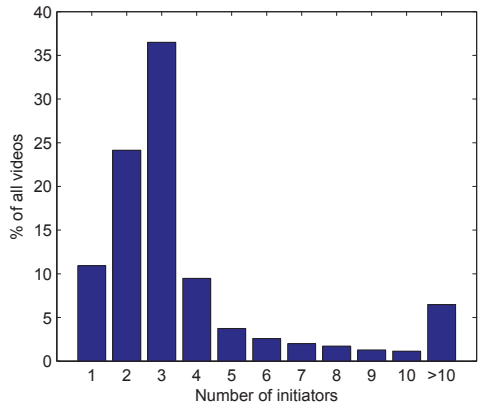


Figure 3.5: Distribution of the number of initiators for individual videos

We also find the number of followed viewers varies a lot for each initiator of the same video. And it is not uncommon that an initial share A is no later than another initial share B of the same video, but B has much more followers (e.g., 547) than A (e.g., 57). It suggests that the spread of a video shows obvious stochastic properties. A further observation shows that generally

the number of viewers followed each initial share of a popular video is either extremely large or small. For example, the numbers of viewers following each initiator of our representative video are 3, 10, 11, 12, 17, 63, 77, 2904, 12130, and 13896 respectively. It suggests that once a shared video from VSSes can survive during the first several rounds of the propagation, it could be followed by a large number of viewers; Otherwise, it will extinct quickly.

We now consider the $BranchingFactor(BrF)$, which is an important metric that reflects the probability that users would like to view friends' shared videos. Fig. 3.6 shows the distribution of BrF for each video. The most frequent values appear around 6 and 7. A few videos have branching factor even more than 25. The work of [24] showed that one user has an average of 78.7 friends in an OSN. Hence the probability a user will view friends' shared videos is around 10% on average.

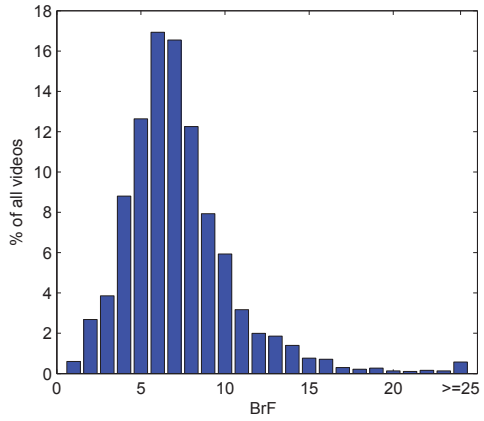


Figure 3.6: Distribution of the average BrF for individual videos

It may be expected that a popular video can attract more total shares, and more viewers for each share. While there is indeed a high correlation (ρ_p=0.91) between the shares and the views of a video, the correlation coefficient between the views and the branching factor is surprisingly low (ρ_p=-0.0014). The average branching factor of the top 10% most popular videos is 6.3, which is indeed much smaller than the average value 6.9. One possible explanation is that an extremely popular video will be shared by many users. Therefore, many friends of a user will likely share the same video, but the user generally watches a video once. In addition, even those users with a very small number of friends also share the extremely popular videos.

We next examine the $ShareRate(ShR)$, which reflects the possibility of share after a view. Compared with BrF, ShR is a better metric to reflect users' interests in a video, since BrF is also affected by the number of a user's friends. Fig. 3.7 shows the cumulative distribution of ShR. We find the ShR of 90% videos are less than 0.3. The average of ShR of all videos in our dataset is 0.13. This low value needs to be further analyzed. If it is a design problem, there are much improvement space. Again, we calculate the relationship coefficient between a video's ShR and its popularity. Small values of ρ_p (0.009202), and ρ_s (0.1357) indicate no

obvious relationship between a video's ShR and its popularity.

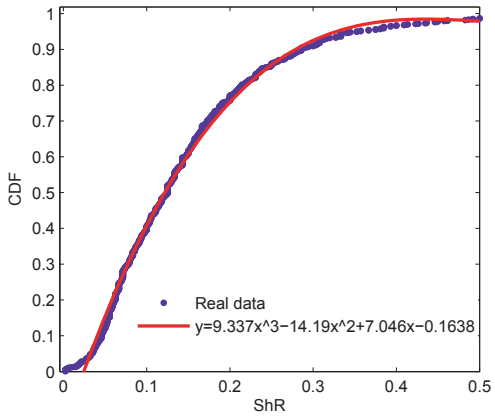


Figure 3.7: Distribution of ShR for individual videos

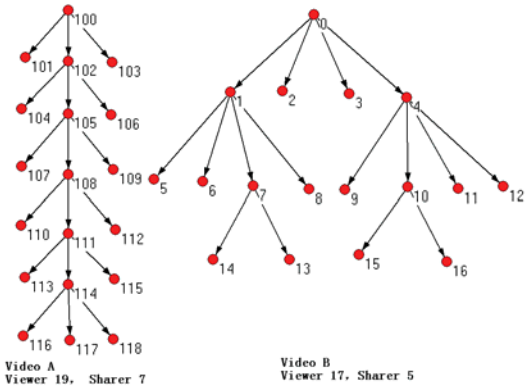


Figure 3.8: Propagation structures of two demonstrated videos

In the above measurements, we have shown that a video's popularity has no obvious correlation with the average value of BrF and ShR, and only has a weak relationship with the number of initiators. Then we suspect the duration of users' interest in a video should be a more important factor determines the video popularity. Although the average value of BrF and ShR can reflect average users' interest in a video to some extent, it ignores a fact that only long-lasted popular videos can cumulate large views. A simple example to illustrate this phenomenon is shown in Fig. 3.8. It consists of two trees, illustrating the propagation of video A and video B. Obviously, A has a smaller BrF ($19/7 \approx$2.7) than B (17/5=3.4), but attracts more viewers. From this example, we can see that a temporary larger BrF can accelerate the video spread in a short time, but it will die out quickly as the users' interest in the video declines. Thus, the evolution of users' interest in videos as well as their average statistic values are the fundamental factors determining a video's popularity. We confirm this assumption by measuring the relationship between the video popularity and the height of the propagation tree. Only

those videos that keep high attractions to users can continue propagating and produce propagation trees with a large height. The coefficients are ρ_p=0.564, and ρ_s=0.856 respectively, which are relatively high values.

Table 3.1 summarizes the relationship between the five parameters and video popularity. A direct finding is that the video popularity has no obvious relationship with the average value of BrF and ShR, weak relationship with the number of initiators, and strong linear relationship with the number of shares and the height of the video propagation tree. The number of shares and views are determined by users' interest in videos as well as the duration that the interest persists.

Table 3.1: Correlation between five factors and a video's popularity

Correlation	Initiators	BrF	ShR	Height	Shares
ρ_p	0.1895	-0.0014	0.0092	0.564	0.9138
ρ_s	0.2342	-0.1453	0.1357	0.856	0.9730

3.3 Characteristics of User Behaviors

This section presents characteristics of user behaviors in initiating, viewing and sharing, and their temporal properties.

3.3.1 Initiating, Viewing, and Sharing

We start by examining the initiators, each of which triggers the first share of a video. From the dataset, we extract 827 thousand initiating records. While this number is not small, it is only 6.5% out of the 12.8 million sharing records. This indeed reflects the pervasiveness and power of video share propagation. The rank distribution of the initiators (in terms of the number of initiated videos) is plotted in Fig. 3.9. Without surprise, it is long-tail scale-free, suggesting that most users initiate few videos, but a few *active users* have initiated a remarkable number of videos. The most active user indeed has initiated over two thousand videos in one week.

The Zipf's law is usually used to fit the long-tail distributions, and the probability distribution function (PDF) $y = \frac{C}{x^a}$ is a straight line in logarithmic scale. Our data cannot be simply fitted by one Zipf's line: the data after top-10 appear to be a straight line, but the top-10 data clearly differ from the rest. Yet they can be roughly fitted by another Zipf's line. The distinction implies the existence of two possible types of users with different initiating behavior.

We compute the ratio of the number of viewed videos and that of the received videos from friends, defined as the *reception rate* for each user. Since the number of received videos is estimated and the accuracy would be affected if the sample size is small, we have removed those users that receive less than 4 videos to obtain a more representative result, and the cumulative distribution function (CDF) of the reception rate is shown as the blue line in Fig. 3.10. This curve

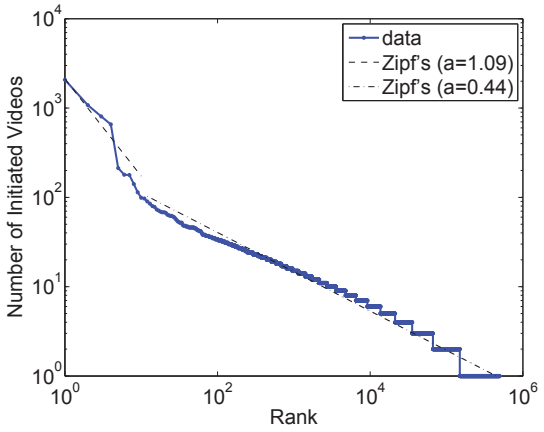


Figure 3.9: Number of initiated videos against rank

can be well fitted by the CDF of a Generalized Pareto Distribution (GPD) $y = 1-(1+\xi \cdot \frac{x-\mu}{\sigma})^{-\frac{1}{\xi}}$. On average, we find that a user watches 16% of videos shared by its friends.

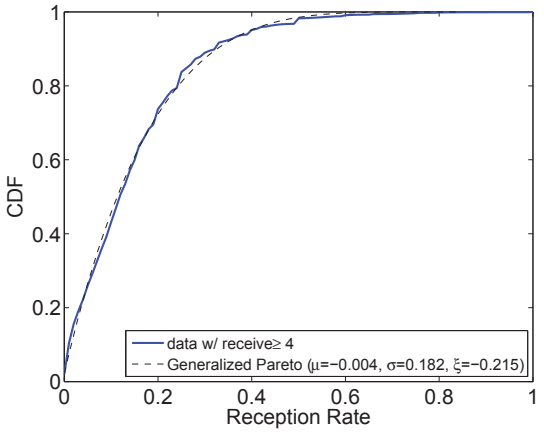


Figure 3.10: CDF of reception rate

We next examine the user behavior on sharing videos after watching, a key step toward propagation. The distribution of the number of each user's shares is again scale-free, clearly indicating that there are some extremely active users sharing a great number of videos, and most of the users only share a small number of videos. To understand how a user reacts upon finishing watching a video, that is, whether or not to further spread the video, we calculate the ratio of the number of shared videos against the number of viewed videos, defined as the *share rate*, for each user. In the calculation, we include the users that have not shared any videos but have watched at least two videos. For the cases where the number of shares is greater than that of views, the share rate is defined as 1. The CDF of share rate is shown as blue solid line in Fig. 3.11. Similar to the reception rate, the share rate can be well fitted by a Generalized

Pareto Distribution. The average value of share rate is 13%.

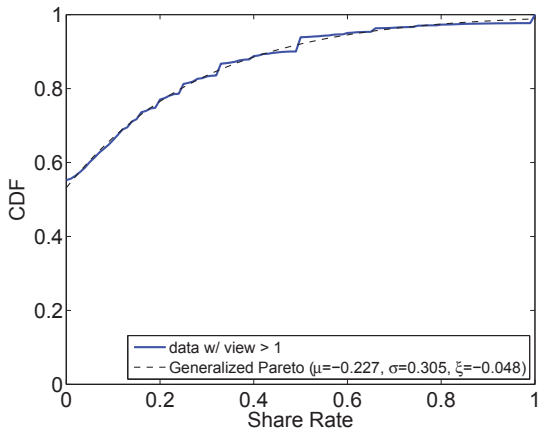


Figure 3.11: CDF of share rate

We notice that there are over half of the users do not share any video. We further examine these users by plotting a CDF of the number of views in Fig. 3.12. Among these users, most of them have only watch a few videos, but we do find the most "selfish" users have watched more than one thousand videos without sharing any. Such *free-riders*, like those in peer-to-peer systems, largely hinder the propagation. Note that, most users generally consume video rather than interacting with it. As a result, for those users who have watched a few number of videos without sharing one, we do not consider them as the free-riders. Therefore, we next propose a simple method to identify free-rider.

It is well known that a power-law distribution usually results in "K/$(100 - K)$" rules, such as 80/20, indicating that a majority of the effects comes from a minority of the causes. We thus utilize this method to identify free-riders. Specifically, we only examine the free-riders with views between 1 and 388[2]. As a result, we find a 94.5/5.5 rule, i.e., 94.5% users have watched less than 5.5% videos, which is $388 \cdot 5.5\% = 22$. We define users that have watched more than 22 videos without sharing one as free-riders, which are around 320 thousand and count for about 3.5% of the all observed users.

3.3.2 Temporal Property

We first check the time span between sharing a video and the actual view of this shared video by the sharer's friends. We examine the sharing records that are created in the first two days, and the corresponding viewing records within 6 days. The reason we do not examine all the sharing records is that, sharing records created in the last day only have less than 24 hours to be watched, leading to unfair comparisons. We define the view from the first friend that

[2]One outlier free-rider has watched 1151 videos, and all others have watched no more than 388.

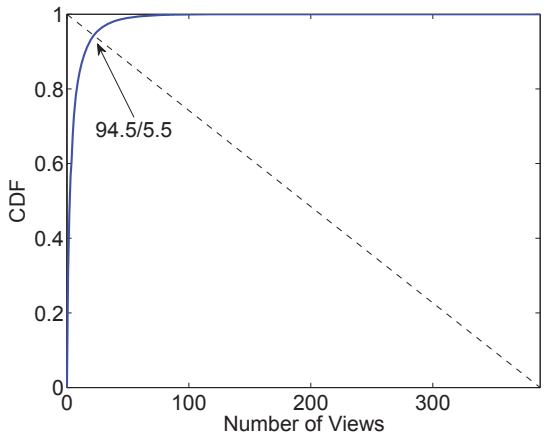


Figure 3.12: CDF of views for free-riders

watches the video as "first view", and if a shared video has not been watched in 6 days, we set the "first view" value to 8640 (minutes of 6 days); all the views by friends are defined as "all view". The respective CDFs of the time spans for both are plotted in Fig. 3.13.

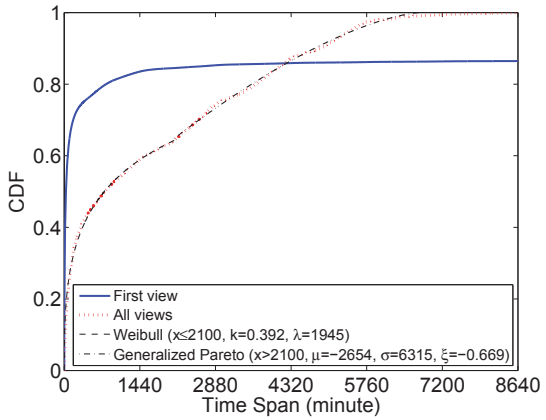


Figure 3.13: CDF of time span from share to view

We observe that 13% of the shared videos will not be watched in 6 days, and for those videos that have been watched, 68% can be watched within one hour. This indicates that videos can quickly propagate to friends in OSNs, exhibiting strong temporal locality. By examining the data of "all view", we find that only 2.6% views appearing after 4 days and less than 1% after 5 days. This implies that the life span of video propagating in OSNs is in general of short durations, and thus our one-week dataset is suitable for the study.

We tried several common distributions to fit the curve (we only fit the "all view" distribution which will be utilized in our model in Section 3.4), but none of them fits well. Therefore, we tried a combined distribution with Weibull ($y = 1 - e^{-(x/\lambda)^k}$) and Generalized Pareto, and obtained

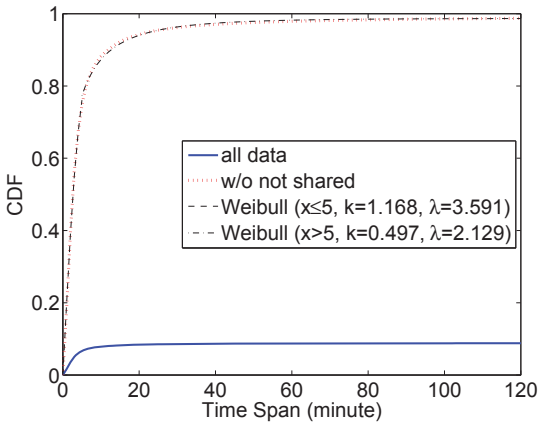


Figure 3.14: CDF of time span from view to share

a good fit.

We then compute the time span between watching a video and sharing it, i.e., how long it takes a user to share a video to friends after clicking to watch it. We find that over 90% of views are not followed by sharing. For the rest of the records, they are clearly affected by the video lengths: 88% of shares are created within 10 minutes after the users starting watching a video, which can be well explained that the videos shared in OSNs are mostly short user-generated-content (UGC) [11]. Fig. 3.14 plots the CDF of the time span within two hours, as well as a combined Weibull CDF fit. Note that, there are only 1.3% of shares occuring after two hours, and thus we only fit the data within two hours. The curve implies that not only the videos are short, but also the users tend to share them right after finishing watching (or even before the finish).

Again, the modeling of time spans from share to view and from view to share facilitate the video sharing propagation model in Section 3.4.

3.4 Modeling Video Propagation in OSNs

In this section, we propose an extended epidemic model to capture the video propagation in OSNs. First, we describe the classical SIR model and extend it to our S^2I^3R model. Then, we validate it based on real trace. Finally, using this S^2I^3R model, we analyze an interesting observation from the measurement.

3.4.1 S^2I^3R model

An epidemic model describes the spread of a contaminative disease through individuals [15]. One of the classical epidemic model is the *SIR (Susceptible-Infectious-Recovered) model*. It

considers a fixed population with three compartments: *Susceptible* (S), *Infectious* (I), and *Recovered* (R). The initial letters also represent the number of people in each compartment at a particular time t, that is, $S(t)$: the number of individuals not yet infected with the disease, or those susceptible to the disease; $I(t)$: the number of individuals who have been infected with the disease and are capable of spreading the disease to those in the susceptible category; $R(t)$: the compartment used for those individuals who have been infected and then recovered from the disease, and those in this category are not able to be infected again or to transmit the infection to others.

In the SIR model, we have the following ordinary differential equations:

$$\begin{cases} \frac{dS(t)}{dt} = -\beta \cdot S(t) \cdot I(t) & (3.1) \\ \frac{dI(t)}{dt} = \beta \cdot S(t) \cdot I(t) - \gamma \cdot I(t) & (3.2) \\ \frac{dR(t)}{dt} = \gamma \cdot I(t) & (3.3) \end{cases}$$

where parameter β is infection rate of the disease, and parameter γ represents the recovery rate.

No direct mapping exists from the classical SIR model for video propagation in an OSN. New compartments and new derivative equations are needed. Six major compartments are introduced:

- Safe (S_1) represents the individuals who are far away from sharers. Initially, all users are Safe expect the friends of the initiator;
- Susceptible (S_2) represents the individuals who have a chance to see the shared video. If an individual shares a video, the shared video will appear in its friends' news feed, and its friends who are on Safe stage become Susceptible;
- Infected (I_1) represents the individuals who are watching the video. Note that individuals at this stage still cannot infect others;
- Immune (I_2) denotes the individuals who choose not to watch the video;
- Infectious (I_3) denotes the individuals who choose to share the video after finishing watching. Only individuals who are at Infectious stage can infect other individuals;
- Recovered (R) denotes the individuals who choose not to share the video.

In the classical SIR model, the transition is time-dependent, that is, at any time, there is a chance that the stage transits to the next one. While for video sharing propagation in OSNs, the transition of the stages depends on decisions at a certain time, e.g, the user needs to choose watch or not, and share or not share. Therefore, we introduce two temporary decision stages

in S^2I^3R: D1 and D2. The user makes watching decision at stage D1, and makes sharing decision at stage D2.

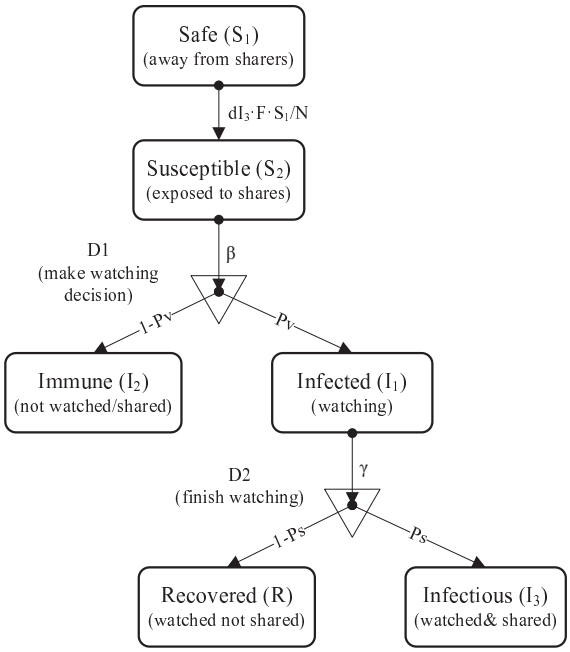


Figure 3.15: S^2I^3R model

The enhanced S^2I^3R (Safe-Susceptible-Infected-Immune-Infectious-Recovered) model is illustrated in Fig. 3.15. For a particular video object, the propagation process is like this: Initially, a user shares this video from an external video sharing site and this initiator becomes Infectious. All other users in the social network are Safe except the friends of the initiator. The shared video appears in the news feed of the initiator's friends and thus they become Susceptible. After a while, these friends log into the social network gradually and decide whether to watch the video (Infected) or not (Immune). For those Infected users, they will usually decide whether to share after watching the video. They become Recovered if they choose not to share. They will become Infectious if they choose to share. Again, these Infectious users will make their friends who are on Safe stage become Susceptible. Note that the case of "not watching but share" is not considered in S^2I^3R. That is because this case accounts for only a small portion (e.g., less than 5%) among all "share" cases. Moreover, omitting this case can let us simplify the model and focus on those more important parameters.

The following derivative equations formally describe the relationships between those compartments:

$$\begin{cases} \frac{dS_1(t)}{dt} = -\frac{dI_3(t)}{dt} \cdot F \cdot \frac{S_1(t)}{N} & (3.4) \\ \frac{dS_2(t)}{dt} = -\frac{dS_1(t)}{dt} - \beta \cdot S_2(t) & (3.5) \\ \frac{dI_1(t)}{dt} = \beta \cdot S_2(t) \cdot P_v - \gamma \cdot I_1(t) & (3.6) \\ \frac{dI_2(t)}{dt} = \beta \cdot S_2(t) \cdot (1 - P_v) & (3.7) \\ \frac{dI_3(t)}{dt} = \gamma \cdot I_1(t) \cdot P_s & (3.8) \\ \frac{dR(t)}{dt} = \gamma \cdot I_1(t) \cdot (1 - P_s) & (3.9) \end{cases}$$

where F is the number of the sharer's friends and N is the number of total users in the system. The transition rate from S to D1 is β, and thus a Susceptible user will spend $1/\beta$ unit time to receive a shared video from a friend. The user then makes a decision whether or not to watch the video. We denote the probability of the user watching the video as P_V. Similarly, we denote the transition rate from I to D2 by γ, and the probability of a user deciding to share the video by P_S.

The S^2I^3R model has four important parameters: β, γ, P_v, and P_s. They can be studied from real traces. Specifically for RenRen system, the cumulative distribution function of $1/\beta$, the time span from share to watch, is well fitted by a combined Weibull and a Generalized Pareto distribution

$$f_{k,\lambda,\mu,\sigma,\xi}(x) = \begin{cases} 1 - e^{-(x/\lambda)^k} & x \leq 2100 \\ 1 - \left(1 + \xi \cdot \frac{x-\mu}{\sigma}\right)^{-\frac{1}{\xi}} & x > 2100 \end{cases}$$

with parameters (k=0.392, λ=1945, μ=-2654, σ=6315, ξ=-0.669). The cumulative distribution function of $1/\gamma$, the time span between watching a video and sharing it, is well fitted by two combined Weibull distributions

$$f_{k_1,\lambda_1,k_2,\lambda_2}(x) = \begin{cases} 1 - e^{-(x/\lambda_1)^{k_1}} & x \leq 5 \\ 1 - e^{-(x/\lambda_2)^{k_2}} & x > 5 \end{cases}$$

with parameters (k_1=1.168, λ_1=3.591, k_2=0.497, λ_2=2.129)

The cumulative distribution functions of both P_v and P_s follow a Generalized Pareto Distribution

$$f_{\mu,\sigma,\xi}(x) = 1 - \left(1 + \xi \cdot \frac{x-\mu}{\sigma}\right)^{-\frac{1}{\xi}}$$

with parameters (μ=-0.004, σ=0.182, ξ=-0.215) and (μ=-0.227, σ=0.305, ξ=-0.048), respectively.

Table 3.2: Validation of S^2I^3R Model

	fitting model	R^2 of fitting model	R^2 of simulation
reception rate	GPD	0.9978	0.9952
share rate	GPD	0.9959	0.9540
time to watch	Weibull + GPD	0.9991	0.9348
time to share	2 Weibulls	0.9989	0.9813

3.4.2 Model Validation

We have simulated the S^2I^3R model multiple times to validate its accuracy. We generate 10000 users participating in 100 video sharing propagations for 8640 minutes (6 days). Specifically, we simulate the propagation of one video each time and run 100 times. For each video propagation, for each minute the simulator checks and updates the state for each of 10000 users according to the derivation equations (3.4) to (3.9). And it runs 8640 rounds for each video propagation. The parameters β, γ, P_v, and P_s are given in Section 3.3. The work [24] provided the distribution of the number of friends in RenRen and we use it in our simulation.

We extract a series of statistics, such as number of received, watched, shared videos for each user, time span from share to watch, and time span from watch to share. We examine these statistics with the real dataset, specifically, we compute R^2, the *coefficient of determination*[3] of the generated data and the real data. We list those goodness of fit, as well as statistical fit names and the corresponding R^2 in Table 3.2. The high values of R^2 (above 0.99) indicates that our model accurately characterizing the user behavior in video propagation.

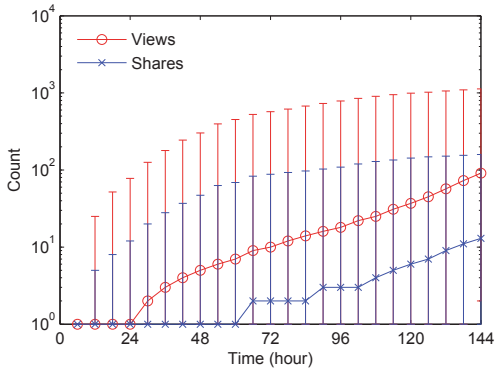


Figure 3.16: Cumulated video views ($I_3 + R$) and video shares (I_3) along time

We next investigate evolutions of the number of users on each stage along the time. We again generate 10000 users participating in 100 video sharing propagations for 8640 minutes

[3] *The coefficient of determination R^2 is a goodness of fit describing how well it fits a set of observations, defined as $1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2}$, where f are generated data or modeled values, y are the real data and $\bar{y}$ is the mean of the real data.*

(6 days). We run the model 100 times under the same system setting (including β, γ, P_v, and P_s), and for each time it simulates a propagation for one video. We calculate the average, maximum and minimum of the cumulated video views ($I_3 + R$) and video shares (I_3) along time in Fig. 3.16. From the figure, we can see that the views and shares are quite diverse for each video even under the same system setting. This result confirms our early measurement results in work [30], in which we found that the number of video views and the number of video shares have very weak correlations with the average share rate (P_s) and reception rate (P_v).

3.4.3 Implications from This Model

In this subsection, we use this model to analyze an interesting measurement finding: the limited propagation range, and evaluate our proposed recommendation strategy which aims to increase the propagation range. The measurement in the work [12] showed that the sizes of most propagation trees are below 100, and even the most popular videos are relatively small as compared with the total number of users in the system. In other words, a vast majority of the cascades vanish quickly and even the top-popular videos do not reach "epidemic" proportions in social networks. This certainly contradicts to the expectation that the shared videos should spread as broadly as possible, not to mention that many of these videos imported into social networks are popular ones in the original video sharing sites.

An underlying reason that limits the spread of videos in the social networks is the mechanism for social contagion [51]. According to the current contagion mechanism, only a video shared by a user's friend will appear in the user's page. The videos watched but not shared by a user's friend will not appear as a news feed for the user. In other words, even if a video is watched by many users, if they do not share the video, the propagation will stop. Unfortunately, according to our statistics in Sections 3.3, an average of 16% users will watch a video shared by a friend, and, among them, only 13% will further share the video. Assume there are n friends for a user. For a sharer, the expected number of his/her friends who will share this video after watching is thus $n{\cdot}0.16{\cdot}0.13$, which we refer to as as the *epidemic index*. For n being less than 48, the epidemic index is smaller than 1; in this case, the sharers will become fewer and fewer, and hence the propagation quickly stops. Furthermore, the measurement in Section 3.3 shows that there exist many free-riders who watch a lot but almost never share any videos, which further confines video propagations.

Since friends number, share rate, and reception rate are intrinsic system properties, which can hardly be tuned, a practical way to boost propagation is to modify the contagion mechanism, in particular, leveraging the users' viewing information. A simple solution is that, once a user watches a video, the link of this video will appear in the news feed of the user's friends as an event of this user. This watching behavior well reflects the popularity of the video among the friends, and can be even more directly than the sharing behavior. Yet it does not preserve the user privacy as the information about all its watched videos are now available among friends.

Therefore, we suggest an anonymous solution: for a user, once a video has been viewed by K friends, the video will appear in this user's news feed as a system suggested news, even if none of the friends have shared the video. A possible system comments with the shared video link might be "K friends have viewed this video" , and there is no need to mention the names of the friends, so that privacy of other users is well preserved. The key issue for this *view-aware* contagion strategy is to set the threshold K. A small K would be more effective for promoting the propagation, but might trigger excessive news feeds. Then we will closely examine its impact through trace-driven experiments.

Now we use S^2I^3R-based simulations to examine the effectiveness of our *view-aware* contagion strategy. We are particularly interested in how the parameter K (the threshold for triggering the news feed that a certain number of friends's watching a video) affects the video views and the number of system suggestions under different settings. Our simulations were run on a RenRen subgraph of 10000 nodes, with an average degree of friends being 17. We fixed P_v to 0.16, but varied P_s and K, respectively.

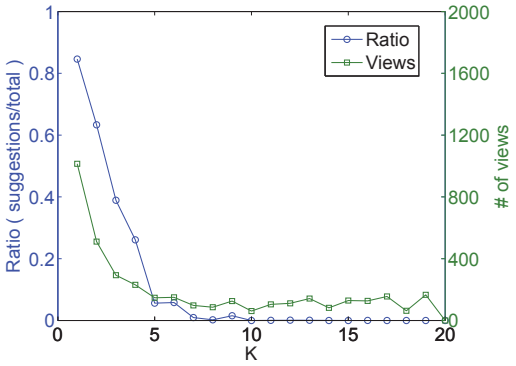


Figure 3.17: Effect of K

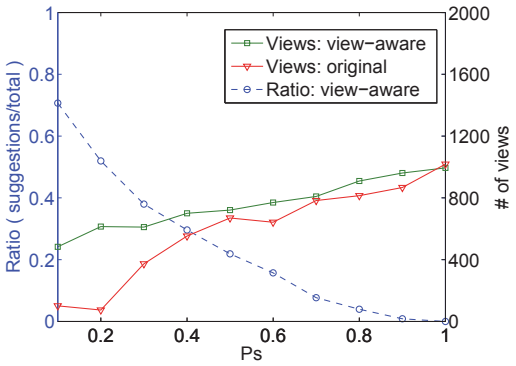


Figure 3.18: Effect of P_s

First, we set P_s as a fixed value 0.2, and let K vary. For each K, we ran simulations ten times and recorded the average values of user views, user shares, and system suggestions.

The results are shown in Fig. 3.17. We can see that the *view-aware* contagion strategy has notably accelerated video propagations. The gain in the number of viewers is remarkable when K is small. On the other hand, small K may increase the ratio of system recommendations to the sum of user shares and system recommendations. Note that the recommendations are generally reasonable, because these suggested videos have been widely viewed by the users' friends and the users are likely to be interested in them. In the RenRen case, we set K as 2 mainly considering its effect on the views increasing. System operators can adjust this value according to their own concerns.

Second, we set K as a fixed value 2, and let P_s vary. Again we ran simulations ten times for each P_s and recorded the average values. The results are shown in Fig. 3.18. We can see that the increase to views promoted by our view-aware contagion strategy is remarkable for small P_s. Since P_s is generally smaller than 15% in real systems, our strategy is practically working.

3.5 Summary

This chapter measured and analyzed the video propagation in OSNs. We first characterized how a popular video is propagated in RenRen, and the effect of potential factors to the propagation size. We then measured user behaviors including initiating, viewing and sharing, and their temporal properties. We further introduced an S^2I^3R Model which extends the conventional epidemic models to accommodate diverse types of users and their probabilistic viewing and sharing behavior. We validated our model and showed that it effectively captures the propagation process of video sharing in social networks. Therefore, the model can serve as a valuable foundation for such applications as workload synthesis, traffic prediction, and resource provision of video servers.

Chapter 4

On Popularity Prediction of Videos Shared in OSNs

4.1 Introduction

Content providers, advertisers, and Web hosts all expect to predict how many view accesses individual videos might generate for a given site. For advertising, the popularity count is tied directly with the ad revenue (see for example the ads shown with YouTube videos); an accurate population prediction thus offers a good revenue (or cost) indication for both YouTube and its content generators. For content-distribution networks, the computation, storage, and bandwidth resources can be well planned with a good prediction of the access patterns [58, 35]. There have been extensive studies on popularity prediction for conventional VSSes, mostly leveraging earlier views of a video as the key predictor [53, 43, 21, 50, 63].

Although the videos shared in OSNs are generally hosted by VSSes, an OSN proactively spreads videos among its users along friendship relations. As such, a video's views are not only determined by the users' interest in it, but also the underlying propagation structure, which generates unique request patterns than that in VSSes. It has been found that the propagation-based video spreading mechanism generates distinguished video popularity distribution [33]. We further find that it would lead to high video popularity dynamics due to the great difference of the numbers of users' friends. As such, even though it is proved that the conventional prediction models perform well in predicting video views in VSSes [53], it is necessary to evaluate their effectiveness in the OSN context and if needed, to develop new tools.

We conduct an initial study on the popularity prediction of videos shared in OSNs. We first test the performance of conventional views-based prediction models. We then propose a novel propagation-based prediction solution. Our contributions are summarized as follows:

- We test the performances of the conventional prediction tools including Autoregressive

Integrated Moving Average (ARIMA) model, Multiple Linear Regression (MLR), and k-Nearest Neighbors (kNN). These models only need the number of early views as the input, and can be easily developed by VSSes without assistances of OSNs. We find that they are generally ineffective, if not totally fail, especially when predicting the early peaks and later bursts of accesses, which are common during video propagations in OSNs.

- We present a novel propagation-based prediction tool, namely SoVP (Social network assisted Video Prediction). SoVP considers both the intrinsic attractiveness of a video and the influence of the underlying propagation structure. The effectiveness of SoVP, particularly for predicting the request bursts, has been validated through our trace-driven experiments.

4.2 Views-based Prediction

One target of this work is to investigate whether the number of future (e.g., one-day ahead) views can be accurately predicted simply based on early views, which can be easily obtained by VSSes so that they can do predictions without assistances of OSNs. To do this, we will examine three conventional prediction models: ARIMA [43], MLR [48], and kNN [41]. To make predictions, they either utilize the early views of the predicted video itself or utilize the similarity of the popularity evolution pattern with early published videos. Here we provide some primary knowledge of these models, and present their performance in Section 4.4.

4.2.1 Autoregressive Integrated Moving Average (ARIMA)

We first examine Autoregressive Integrated Moving Average (ARIMA), one of the most popular time series models for predicting future values of a time series [43, 21]. Given the time series of video popularity in the past several days, it can make fine-grained prediction for the video's future evolution, leveraging the trend, periodicity and autocorrelation exhibited in the history information. ARIMA consists of three parts: an Autoregressive (AR) model, a Moving Average (MA) model and an integrated part. They are applied in the cases where data show evidence of non-stationarity and an initial differencing step (corresponding to the "integrated" part of the model) can be used to remove the non-stationarity. Given a time series Y, an AR model of order p is defined as:

$$Y(t) = \sum_{i=1}^{p} \beta_i Y(t-i) + \epsilon \tag{4.1}$$

where $Y(t)$ is the number of views in the t^{th} day; β_1,..., β_p are the parameters of the model; and ϵ is a white noise error term. An MA model of order q is defined as follows:

$$Y(t) = \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t \tag{4.2}$$

where θ_1,...,θ_q are the parameters of the model and ϵ_t,...,ϵ_1 are again white noise error terms. Combing Eq. 4.1 and 4.2, an ARIMA model of order (p, q) is written as follows:

$$Y(t) = \sum_{i=1}^{p} \beta_i Y(t-i) + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t \tag{4.3}$$

The error terms, ϵ_t, are generally assumed to be Gaussian random variables with zero mean and constant variance.

4.2.2 Multiple Linear Regression (MLR)

A major drawback of ARIMA model is that it needs a relatively long period of historical information for prediction. For our data set, the numbers of views of at least the first 4 days are required to generate the model and thus the initial population evolution for a newly released video cannot be predicted using ARIMA. The high correlation of neighbor days motivates us to use regression models. Multiple Linear Regression (MLR) [48] is widely used to model the relationship between a dependent variable and several explanatory variables. In our scenario, early views are regarded as explanatory variables and used to predict later views, which is shown in Eq. 4.4:

$$Y(t) = \alpha + \sum_{i=t-n}^{t-1} \beta_i Y(i) + \epsilon_t \tag{4.4}$$

where $Y(t)$ is the number of views in the t^{th} day; α is a constant number; β_i is the weight for the i^{th} day; and ϵ_t is the residual value. n is the critical parameter in this model that defines the number of early days used for prediction.

4.2.3 k-Nearest Neighbors Regression (kNN)

kNN regression [41] is also a widely used regression model. It estimates the value of an unknown function at a given point based on the values of its nearest neighbor points. The kNN estimator is defined as the weighted average function value of the nearest neighbors. In our scenario, the views of the videos in the training set are used to predict the views of the videos in the test set, as shown in Eq. 4.5:

$$Y_x(t) = \sum_{x' \in N(x)} \frac{1/d(x,x')}{\sum_{x'' \in N(x)} 1/d(x,x'')} Y_{x'}(t) \tag{4.5}$$

where $Y_x(t)$ is the number of views of video x in the t^{th} day; $N(x)$ is the set of k nearest points to video x in the training set with regard to the views in previous days; $d(\cdot)$ denotes

the distance function; and k is the parameter defining the number of neighbors. We choose Euclidean distance as the distance function. Similar to MLR, we use the early views as the vector to compute the distance between future days. To break ties in neighbor selection, we include all the videos with equal distance since the late views can vary a lot with equal early views, especially when only a short period of early views are considered.

4.3 Propagation-based Prediction

Compared with VSSes, OSNs know much more information about a video beyond the number of its early views, such as viewers, sharers, whether viewers would like to share the video after viewing, whether users would like to view the videos shared by their friends. Yet, how to utilize such information in video popularity prediction is not clear, as the previous work has shown that they have no simple (e.g., linear) relationship with the video popularity [30]. In this section, we propose a novel propagation-based prediction framework to predict video future views in the OSN.

4.3.1 Modeling Video Propagation

Before modeling the video propagation, we first define some notations. For a given video, $V(t)$ and $S(t)$ are defined as the sets of its viewers and sharers by the time t, respectively. We use —$V(t)$— to denote the number in the set $V(t)$, and this notation can also apply to other sets such as $S(t)$. $ShR(t)$ (short for *Sh*aring *R*ate) is the probability that a user will reshare a video after viewing it. $ViR(t)$ (short for *Vi*ewing *R*ate) is the probability that a user will eventually view the video shared by his/her friend. To some extent, both $ShR(t)$ and $ViR(t)$ reflect how interesting the video is. $W(t)$ is the number of sharers' friends by time t who have not yet viewed the video. In other words, $W(t)$ = the number of all sharers' friends - —$V(t)$—. Similar to [22], we assume the $W(t)$ users view the video at a constant rate, which is denoted by λ. $f(S(t))$ is the number of friends of the new sharer exclusive of those friends who viewed the video before the time t. Generally, the average new potential viewers brought by per new sharer will decrease as the increase of the number of sharers in $S(t)$, because most of the new sharer' friends may have already viewed the video from his/her other friends who also shared the video earlier than the new sharer.

Based on the above notations, the propagation process of one video can be described by the following three equations:

$$\begin{cases} \dfrac{d|V(t)|}{dt} = \lambda \cdot W(t) & (4.6) \\ \dfrac{d|S(t)|}{dt} = ShR(t) \cdot \dfrac{d|V(t)|}{dt} & (4.7) \\ \dfrac{dW(t)}{dt} = \dfrac{d|S(t)|}{dt} \cdot f(S(t)) \cdot ViR(t) - \dfrac{d|V(t)|}{dt} & (4.8) \end{cases}$$

where Eq. 4.6 reflects that the increased viewers during the time dt come from the potential viewers $W(t)$, who are going to view the video at a rate of λ. Eq. 4.7 reflects that $ShR(t)$ portion of new viewers (d—V(t)—) will become sharers during the time dt. Based on the previous measurement work [12], here we assume that viewers will immediately share the video after the viewing, otherwise will never share the video. Recalling that we define $W(t)$ as the number of all sharers' friends - —$V(t)$—. Accordingly, the variation of $W(t)$ during time dt (dW(t)) can be expressed as the combination of the growth in the number of potential viewers brought by new sharers ($d|S(t)| \cdot f(S(t)) \cdot ViR(t)$) and the reduction caused by the views during dt ($-d|V(t)|$). This relation is given in Eq. 4.8.

Initially, there is only one sharer (we call it *initiator*), who posted the video from a VSS. Thus, $S(0)$=1, $V(0)$=1, and $W(0)$ is equal to the number of friends of the initiator multiplying $ViR(0)$. There are four parameters that will affect the evolution of $W(t)$: ShR, ViR, $f(S(t))$ and λ. ShR and ViR reflect the characteristics of specific videos to some extent; $f(S(t))$ depends on the friends of the sharers and social topology around them; λ depends on the frequencies users visit the OSN and watch videos. Our prediction framework in the following subsections will introduce how these parameters can be extracted from real trace.

For ease of exposition, Table 4.1 provides a reference for major notations used in this chapter. Generally, we use upper superscript k (e.g., k in V^k) to denote a video k, and lower subscript i (e.g., i in V_i) to denote a user i. Note that for concise presentation, sometimes we may omit the video superscripts under the premise of no concept confusion (e.g., use $V(t)$ to denote $V^k(t)$ of video k).

4.3.2 Framework of SoVP

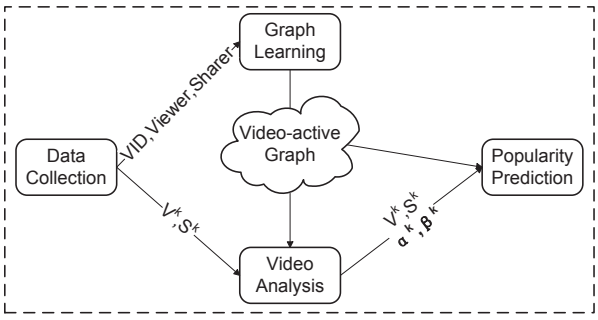


Figure 4.1: Framework of SoVP

The propagation-based prediction architecture, as shown in Fig. 4.1, consists of a data collection module, a graph learning module, a video analysis module, and a popularity prediction module. First, the data collection module collects logs that record user viewing actions. The basic log format is (Video ID, Viewer ID, Sharer ID, Time), the meaning of which is described in Section 2.2. Then the logs are taken as the inputs by the graph learning module and

Table 4.1: Summary of major notations

Notation	Description
F_i	set of the friends of user i;
$V_{i \to j}$	set of videos shared by user i and viewed by user j;
V_i	set of videos viewed by user i;
S_i	set of videos shared by user i;
S_{F_i}	set of videos shared by user i's friends;
ShR_i	the average probability that user i will share the videos that s/he viewed;
$ViR_{i \to j}$	the average probability that user j will view the videos shared by its friend user i;
BrF_i	the average number of friends will view a video shared by user i;
$V^k(t)$	set of viewers of video k until time t;
$S^k(t)$	set of sharers of video k until time t;
v^k_Δ	number of views of video k during period of Δ
$W^k(t)$	number of waiting viewers of video k at time t
α^k	a factor that reflects the normalized ShR of video k;
β^k	a factor that reflects the normalized ViR of video k;
ShR^k	the average probability video k will be shared after being watched;
ViR^k	the average probability video k will be viewed by a friend of a sharer;
ShR^k_i	probability user i will share video k that s/he viewed;
$ViR^k_{i \to j}$	the probability that user j will view the video k shared by its friend user i;
t^k_i	sharing time of video k by sharer i;
λ	the rate of users counted in $W(t)$ who will view video in current time instance;
$\Phi(t)$	the CDF of time (t) between a share and the viewing from the sharers' friends;
$f(S(t))$	the number of potential viewers brought by a new sharer given $S(t)$;

the video analysis module. For the graph learning module, historic user viewing records are used as the input. The graph learning module generates a graph called video-active graph, which records the viewing-sharing relationships between users as well as the statistics of user sharing and viewing actions. The video analysis module takes two kinds of inputs: video information (sharers S^k and viewers V^k) that is gotten directly from the data collection module, and the video-active graph that is generated by the graph learning module. The video analysis module analyzes video attractiveness (α^k,β^k) in the context of the video-active graph. Finally, the popularity prediction module uses both the video-active graph and the video attractiveness to make predictions.

4.3.3 Video-active Graph Learning Module

The topology of an OSN is an important influencing factor in the propagation of videos shared in it. Instead of simply using the original unweighed friend-friend graph, we build a weighted graph called video-active graph. There is a directed edge from user i to user j if the user j ever viewed a video shared by the user i. We assign weights to vertices and edges according to users' viewing and sharing activity. Users show inhomogeneous activity in sharing and

viewing videos. For example, as shown in Fig. 4.2, the power-law distribution indicates that the numbers of videos viewed by each user in one-month period exhibits large skewness.

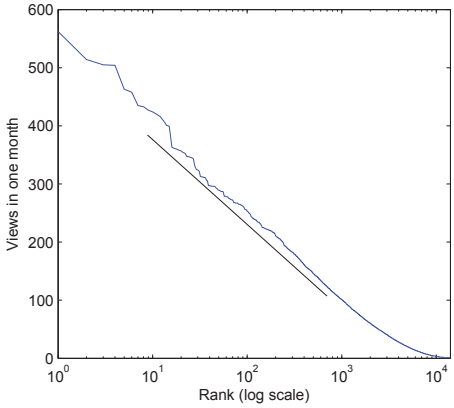


Figure 4.2: Distribution of user views in one month

Fig. 4.3 illustrates the properties of vertices and edges in the video-active graph. The properties of a vertex i include a set of viewed videos (V_i), a set of shared videos (S_i), and sharing rate (ShR_i). The properties of an edge (i, j) include $V_{i\to j}$, which is defined as the set of video viewed by user j and shared by user i, and $ViR_{i\to j}$, which is defined as the ratio that user j has viewed the videos shared by user i. Taking records (Video ID, Viewer ID, Sharer ID) as the input in a chronological order, V_i, S_i, $V_{i\to j}$ can be extracted directly. ShR_i and $ViR_{i\to j}$ can thus be calculated by $ShR_i = \frac{|S_i|}{|V_i|}$, and $ViR_{i\to j} = \frac{|V_{i\to j}|}{|S_i|}$, respectively.

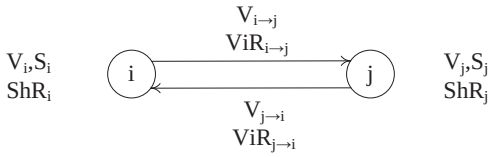


Figure 4.3: Propeties of the video-active graph

In real OSN systems, the video-active graph grows gradually, continuing to bring new vertices and edges especially at their early stage. Statistics of these newly added edges and vertices cannot be measured directly from real trace at such an early stage. The learning process should adapt to this dynamic. For a new friend link created between two users i and j, time is needed for the $ViR_{i\to j}$ be learned from the interaction between the two users. As such, it is necessary to estimate it from the relationships between i, j and their friends F_i, F_j. We denote the estimated value as $\widehat{ViR}_{i\to j}$, and use Eq. 4.9 to calculate its value:

$$\widehat{ViR}_{i\to j} = \frac{|V_j|}{|S_i \cap S_{F_j}|} \tag{4.9}$$

where V_j is the set of videos that are viewed by the user j; S_i is the set of videos that are

shared by the user i; S_{F_j} is the set of videos that are shared by the user j's friends. We take $\widehat{ViR}_{i\to j}$ as the initial value for $ViR_{i\to j}$.

4.3.4 Video Analysis Module

For a given video k, the video analysis module uses the video statistics (V^k, S^k) provided by the data collection module to analyze its attractiveness in the context of the video-active graph. Both ShR and ViR are influenced by the video's attractiveness as well as the characteristics of involved users, so that they are not suitable be used to exactly reflect a video's attractiveness. For example, one video is shared among the users who are very active to share and watch videos, while the other video is shared among the users with less activeness. The two videos may happen to have same ShR and ViR based on the simplest definition. Therefore, to gain real values of a video's attractiveness, the video analysis module should remove the effect of the involved users.

For the video k, the video analysis module calculates two factors ($\alpha^k(t)$ and $\beta^k(t)$) to reflect the normalized video attractiveness. The calculation methods are shown in Eq. 4.10 and 4.11, respectively.

$$\alpha^k(t) = \frac{|V^k(t)|}{\sum_{i\in S^k(t)}(\Phi(t-t_j^k)\cdot\sum_{j\in F_i} ViR_{i\to j})} \tag{4.10}$$

where $\Phi(t)$ is the cumulative distribution function (CDF) of time span between sharing a video and the actual view of this shared video by the sharer's friends. We studied the fitting function in the prior work [12]. It is a combined distribution with Weibull (t$\leq$2100, k=0.392, λ=1945) and Generalized Pareto (x$\geq$2100, μ=-2654, σ=6315, ξ=0.669) [12]. t_j^k is the sharing time of video k by sharer j. $|V^k(t)|$ is the actual number of cumulated viewers of video k by time t. $\sum_{i\in S^k(t)}\sum_{j\in F_i}(ViR_{i\to j}\cdot\Phi(t))$ is the estimated average number of cumulated viewers over all videos. The α of an attractive video is usually bigger than 1.

$$\beta^k(t) = \frac{|S^k(t)|}{\sum_{i\in V^k(t)} ShR_i} \tag{4.11}$$

where $|S^k(t)|$ is the actual number of cumulated sharers of video k by time t. $\sum_{i\in V^k(t)} ShR_i$ is the estimated average number of cumulated sharers over all videos. The β of an attractive video is usually bigger than 1.

When making predictions, we use Eq. 4.12 and Eq. 4.13 to decide whether a user will view or share the video k, respectively. The decisions depend on both the video attractiveness and social context.

$$ViR_{i\to j}^k = \alpha^k(t)\cdot ViR_{i\to j} \tag{4.12}$$

$$ShR_i^k = \beta^k(t)\cdot ShR_i \tag{4.13}$$

4.3.5 Popularity Prediction Module

Based on our propagation model, the popularity prediction module takes the information of both video attractiveness and the video-active graph as the input to make predictions.

We rewrite Eq. 4.6 as Eq. 4.14, which calculates the number of video views during the time Δ (e.g., one day in this work). And v_Δ is what we finally need to calculate to be as the predicted views during the time Δ. According to Eq. 4.14, we need $W(t)$ to calculate v_Δ. We can easily calculate the $W(t)$ at the beginning time of Δ by Eq. 4.15. Then what we also need to do is to infer $W(t)$ during the time Δ.

$$v_\Delta = |V(T+\Delta)| - |V(T)| = \int_T^{T+\Delta} \lambda \cdot W(t)\, dt \tag{4.14}$$

$$W(T) = \sum_{i \in S^k(T)} \sum_{j \in F_i} ViR^k_{i \to j} - |V(T)| \tag{4.15}$$

From Eq. 4.6, 4.7, and 4.8, we get Eq. 4.16.

$$\frac{dW(t)}{dt} = \lambda \cdot W(t) \cdot (ShR(t) \cdot ViR(t) \cdot f(S(t)) - 1) \tag{4.16}$$

We define ω as:

$$\omega = \lambda(ShR(t) \cdot f(S(t)) \cdot ViR(t) - 1) \tag{4.17}$$

Then Eq. 4.16 can be rewritten as Eq. 4.18.

$$\frac{dW(t)}{dt} = \omega \cdot W(t) \tag{4.18}$$

Since in a short period the users' interest in a video will not vary a lot, we assume ω is a constant value from time T to $T+\Delta$, Eq. 4.18 can be further expressed as Eq. 4.19.

$$W(t) \approx \delta \cdot e^{\omega \delta t} \tag{4.19}$$

where δ can be calculated using the initial value of $W(t)$ at time T, as is shown in Eq. 4.15.

Finally, from Eq. 4.14 and 4.19, we get:

$$v_\Delta = |V(T+\Delta)| - |V(T)| \approx \frac{\lambda}{\omega}(e^{\omega\delta(T+\Delta)} - e^{\omega\delta T}) \tag{4.20}$$

where T and $T+\Delta$ are the beginning time and the end time of the day when we need to predict.

4.4 Performance Evaluation

In this section we compare the performances of conventional views-based prediction models with our propagation-based prediction model, SoVP. We first examine their overall performance on a large set of popular videos. We further examine their performances on the three typical popular videos, which can provide a direct illustration about what kind of evolutions may make the conventional prediction models inefficient.

4.4.1 Performance Metrics

We evaluate the efficiency of the prediction models using the metric of Relative Absolute Error (RAE). For the video k on the day t, we have:

$$RAE_k(t) = \frac{|\widehat{N}_k(t) - N_k(t)|}{N_k(t)} \tag{4.21}$$

where $\widehat{N}_k(t)$ is the predicted number of views of video k on the day t, and $N_k(t)$ is the actual number of views. For the average RAE of all testing videos on the day t, we have:

$$RAE(t) = \frac{\sum_k |\widehat{N}_k(t) - N_k(t)|}{\sum_k N_k(t)} \tag{4.22}$$

For the average RAE of all testing videos on all testing days, we have:

$$RAE = \frac{\sum_t \sum_k |\widehat{N}_k(t) - N_k(t)|}{\sum_t \sum_k N_k(t)} \tag{4.23}$$

4.4.2 Prediction Results

As shown in the previous work [33], video popularity distribution exhibits extremely high skewness that top-2% videos account for over 90% views. For the remaining 98% unpopular videos, any of them only received less than 10 views per day on average. Therefore, we take those top-2% popular videos that were initially shared on the same day (March 24^{th}, 2011) as our test set.

First, we need to select proper models for MLR and kNN. We split our data set into a training set that contains the viewing information of 27000 videos, and a test set that contains the viewing information of another 5000 videos. For both MLR and kNN regression, we vary the value of n from 1 to 9; for kNN regression, we also vary the value of k from 1 to 4. We evaluate the performance of each setting on the test data set and the results are shown in Fig. 4.4 and 4.5, respectively. Considering the tradeoff of RAE and complexity, we select $n = 5$ for MLR, and $n = 1$, $k = 3$ for kNN.

Then, we evaluate the overall performance of SoVP as well as the three conventional models with the selected parameters. The average RAE over all test videos for each day is shown in Fig 4.6. Overall, the SoVP has much better prediction performance than other three models.

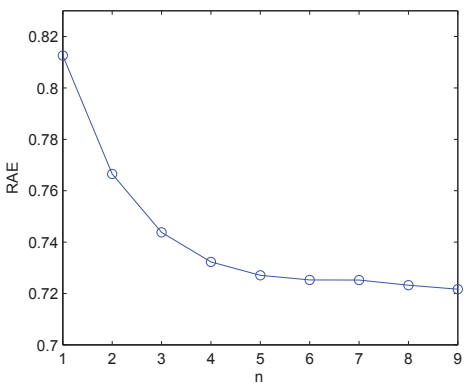


Figure 4.4: Parameter selection for MLR

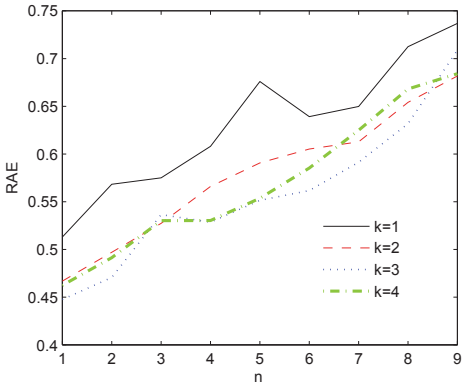


Figure 4.5: Parameter selection for kNN

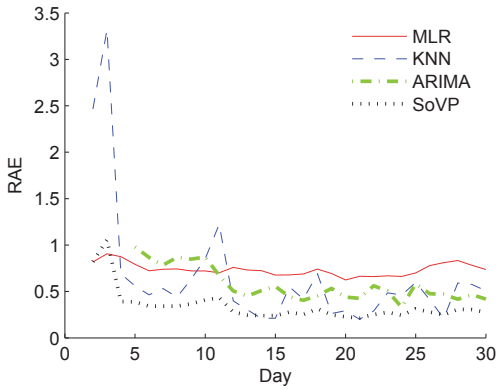


Figure 4.6: Average performance for testing videos

It is worth noting that ARIMA requires several (e.g., 4 in our experiments) days of early views to learn the model, and so its prediction starts from the fifth day. For MLR, $n = 5$ is used starting from the sixth day, and smaller values are used in earlier days (e.g., $n = 1$ for the second day and $n = 2$ for the third day). ARIMA works well in later days, say after 12 days. It can dynamically select the length of historical information used to predict for each day. For MLR, it works better during the first 10 days and its performance is rather stable. kNN shows dynamic performance. For some days it has the most accurate prediction while for others it performs much worse. The reason is that only the number of views during the last day is used and the popularity distribution could change significantly day by day.

Table 4.2: RAE of predictions for the type-1 video

	day 2	day 3	day 4	day 5	day 6
kNN	0.823	0.580	0.765	0.720	0.314
MLR	0.886	0.952	0.907	0.820	0.742
SoVP	0.262	0.247	0.186	0.208	0.157

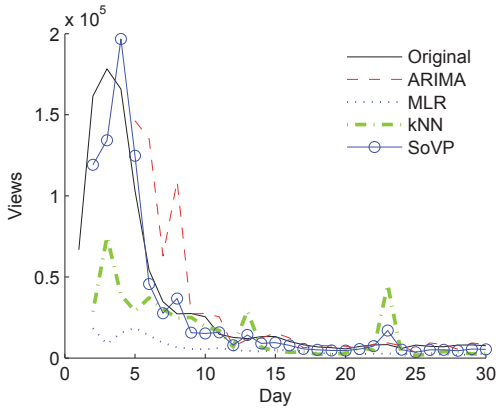


Figure 4.7: Type-1 video prediction

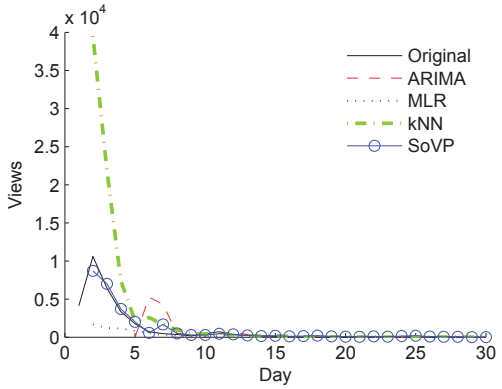


Figure 4.8: Type-2 video prediction

We also apply prediction models to the three typical videos that are depicted the early section. The original daily views as well as the prediction results are shown in Fig. 4.7, 4.8, and 4.9 respectively. Overall, we can see that the predictions of the three conventional models deviate a lot from the real values, while SoVP works much better than other three models, especially when predicting during the request bursting periods. Since views during the short-term bursts usually count for most proportion of the video's life-time views, we further give the RAEs of the four models during three videos' bursting days, in Table 4.2, 4.3, and 4.4 respectively. It confirms our observations in the figures. While some further optimizations can be made on those views-based models, they have inherent limits in predicting views with highly

Table 4.3: RAE of predictions for the type-2 video

	day 2	day 3	day 4	day 5	day 6
kNN	2.729	2.386	1.199	0.212	2.659
MLR	0.843	0.811	0.661	0.538	0.233
SoVP	0.179	0.087	0.108	0.129	0.183

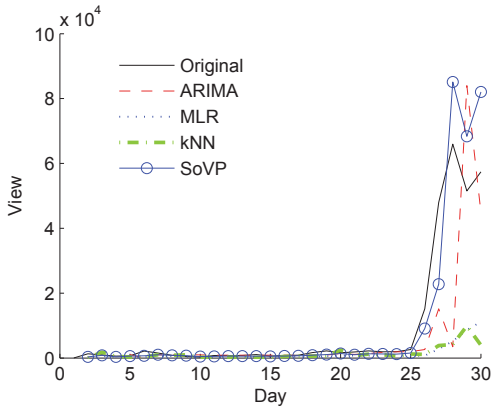


Figure 4.9: Type-3 video prediction

Table 4.4: RAE of predictions for the type-3 video

	day 26	day 27	day 28	day 29	day 30
kNN	0.926	0.920	0.937	0.808	0.932
MLR	0.951	0.942	0.921	0.832	0.805
ARIMA	0.826	0.684	0.947	0.631	0.219
SoVP	0.400	0.525	0.290	0.327	0.429

dynamic evolution. Solely based on early views, they have difficult to judge a video's sudden increase or decreases in views from its own early evolution pattern, or learning from other early published videos. By contrary, SoVP knows exactly the video's propagation process in the OSN and can extract useful statistics, so that can easily judge whether a video is on increasing stage or decreasing stage, and how fast of this trend.

4.5 Summary

In this chapter, we presented an initial study on popularity prediction of videos shared in OSNs. Our measurement results in chapter 2 suggested that the video views in early and later times exhibits much less correlation than that in VSSes, which poses significant challenges on conventional views-based prediction models. Our experiments with such conventional prediction models as ARIMA, MLR, and kNN confirmed their ineffectiveness in this new context, especially when predicting the requests bursts that are common for the evolutions of videos shared in OSNs. To overcome the limits, we developed a dynamic model to analyze the video propagation process, and accordingly presented a propagation-based prediction framework, SoVP. SoVP considers both video attractiveness and social context in predicting future video views, whose accuracy has been demonstrated by our trace-driven experiments.

Chapter 5

Cloud Assistance for Video Sharing in OSNs

5.1 Introduction

Traditionally, VSS videos are mainly discovered through search engines, front pages, and related videos [69]. The OSNs however offer quite different sharing mechanisms, where the video links are propagated through chains of friends. The coverage of such OSN-shared videos can be much broader with much faster propagation speed. It also leads to more micro- and macro-dynamics in the access pattern, as a super user with a great number of friends can easily trigger a surge of accesses [31], and in the long run, a video often has a series of peaks in terms of user access. Given that the video contents are still hosted by VSSes, such distinct access patterns from OSNs have created significant challenges to VSS service providers, particularly for resource provisioning.

There have been pioneering works on joint design and optimization for both VSSes and OSNs with shared information [64][57][58]. In the real market, however, VSS and OSN operators are not necessarily close collaborators, nor the VSSes are to be urgently and completely re-engineered for OSN shared videos given that the demands from traditional users remain strong. On the other hand, for OSN operators, building their own video storage and distribution services is not necessarily the best business model, either, not to mention the complexity and cost involved in joint design. Instead, we believe that, since an OSN knows best about the video sharing patterns from its users, it should provide necessary assistance for its users to access the external VSSes, which in turn will also mitigate the impact to the VSSes.

To this end, we develop SNACS (social network-aware cloud assistance for video sharing), which provides a cost-effective enhancement for video accesses from an OSN. The SNACS module sits between VSSes and an OSN, and is managed by the OSN to improve its users' experience in retrieving the videos from the VSSes. It utilizes both centralized cloud resources (e.g., Amazon S3 [4]) and edge servers (e.g., Amazon CloudFront [3]) to collectively serve

video accesses from within the OSN, which otherwise cannot be well served by the external VSSes. Given the strong dynamics of the access patterns, we are particularly interested in the content management and update strategies in the SNACS' implementation. Motivated by data traces from real world measurement, we show that the conventional cache replacement for video object can be quite inefficient in SNACS. We then develop an optimal offline replacement algorithm that generates minimum misses in this new context. We further offer guidelines to minimize replacements among the solutions of the lowest misses. The optimal offline solutions not only provide a benchmark for comparison but also motivate the design of an online replacement algorithm, which makes effective use of the video sharing patterns in the OSN. Our design has been extensively evaluated and its superiority has been validated under diverse network and user configurations.

The rest of the chapter is organized as follows. We present background and motivation in Section 5.2. Section 5.3 proposes our framework and discusses major design issues. We develop optimal offline replacement algorithms in Section 5.4, and an online algorithm in Section 5.5, respectively. The results and discussions for performance evaluations are presented in Section 5.6. Finally, we conclude in Section 5.7.

5.2 Background and Motivation

There have been significant data-driven measurement and modelling studies on the video content shared through OSNs, e.g., tracking social cascades of YouTube links over Twitter [49] and video popularity distribution and propagation in TencentWeibo, a Twitter-like OSN in China [58]. Our earlier works have also examined videos propagated over RenRen, a Facebook-like social network [30], which leads to the design of a synthetic traffic generator for video requests from OSNs [34].

Our work is motivated by these studies. To further understand the distinct characteristics of video request patterns from OSNs as compared with traditional video accesses and their impact to resource provisioning, we closely collaborate with *56.com*, one of the most popular VSSes in China to analyze its server access logs. The logs record video requests within 56.com website as well as requests from external OSNs. Our analysis shows that among all the video requests, over 36% are from the OSNs, most notably RenRen. For individual videos, however, the ratio of requests from RenRen to the total requests varies significantly, as shown Fig. 5.1. We have calculated the Pearson correlation coefficient[1][46] between video views in RenRen and the total views, with a result of 0.59, which is statistically insignificant. In other words, while statistical histories have often been used to predict the video popularity [7], it can

[1]It has been widely used for measuring the strength of linear dependence between two variables. The range is from -1 to 1, where a value greater than 0 indicates positive correlation, and less than 0 indicates negative correlation.

hardly predict the percentage of the requests from OSNs for a newly uploaded video.

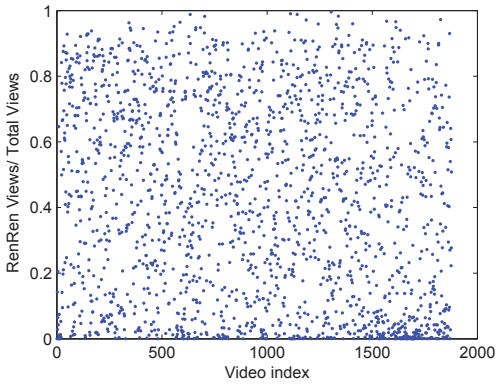


Figure 5.1: Distribution of fraction of RenRen views over the total views

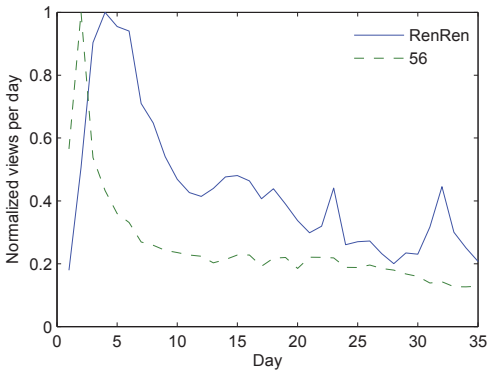


Figure 5.2: Video popularity evolution (normalized by maximum values of daily views)

We also examine how the popularity evolves for videos shared by both 56.com and RenRen. Fig. 5.2 compares the overall video popularity evolutions over 5 weeks. We can see that the views from RenRen users exhibit much stronger dynamics, with more peaks when compared to the overall views in 56.com. We then take a closer look at how the popularity evolves for a single video in a smaller time scale. Fig. 5.3 shows the results of a typical video investigated during our analysis. We find that, in OSNs, there are super users with a large number of friends or followers, and such users, once propagate a video, can trigger a significant number of follow-up accesses. This can lead to a peak of accesses even long after the release of the videos, in which stage the accesses from traditional VSSes users have long decayed. As such, today's VSSes, even equipped with the state-of-art prediction and resource provisioning modules, can still experience frequent under-provisioning.

A series of pioneer works have offered service enhancement of social video sharing by joint design and optimization for both VSSes and OSNs with mutually-shared information [64][58][57]. Since video services are critical to social network users, OSN operators do have

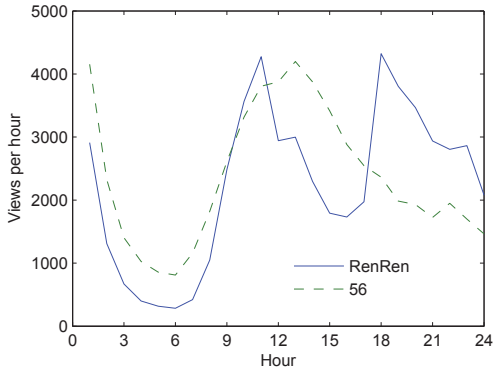


Figure 5.3: Video popularity evolution of a single video in one-day period

strong motivation to offer better service quality to its users. Yet whether they need to fully disclose the social information to external VSS providers remain questionable in the current market, and building their own video content services is not necessarily the best practice, either. On the other hand, the accesses from traditional VSS users remain strong (over 50%), and there is no immediate need for a VSS to re-engineer its services. Therefore, we need a new framework which still works effectively without the assumption that OSN and VSS must be fused together into a unified system. Considering that OSN has the best knowledge of its video requests and the propagation patterns, we suggest that the OSN should take the initiative to offer assistance to its users accessing the videos. In turn, it will also benefit external VSSes given a large portion of accesses from the OSN are absorbed by OSN servers.

5.3 SNACS: Social Network-Aware Cloud Assistance for Video Sharing

Motivated by above ideas, we propose a new framework called SNACS: social network-aware cloud assistance for video sharing. The SNACS module sits between VSSes and an OSN, and is managed by the OSN to improve its users' experience in retrieving the videos from the VSSes. It utilizes cloud resource to serve video accesses within the OSN that otherwise cannot be well served by the external VSSes. Fig. 5.4 offers a more detailed view of the work flow in SNACS. An OSN user will initially request video data from the VSS servers. According to feedback of the downloading speed, the OSN user may redirect the video request to the OSN-operated content cloud if it cannot be well served by VSSes.

As illustrated in Fig. 5.5, the cloud service for content (e.g., video) delivery usually consists of an origin server, and a distributed delivery network which includes multiple edge servers distributed in different geographical locations [1]. Initially, a cloud customer should apply an origin server to store its video files, and choose several edge locations to serve its user requests.

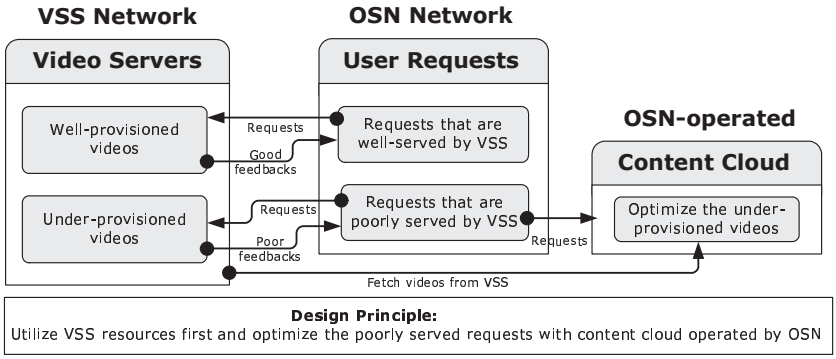


Figure 5.4: SNACS: Social Network-aware Cloud Assistance for Video Sharing

When the videos are ready to be delivered, they will be first uploaded to the origin server, and then copied to the edge locations. Take Amazon's Cloud as an example, to delivery video content globally, it suggests Amazon S3 [4] as the origin server, and Amazon CloudFront [3] as the distributed delivery network.

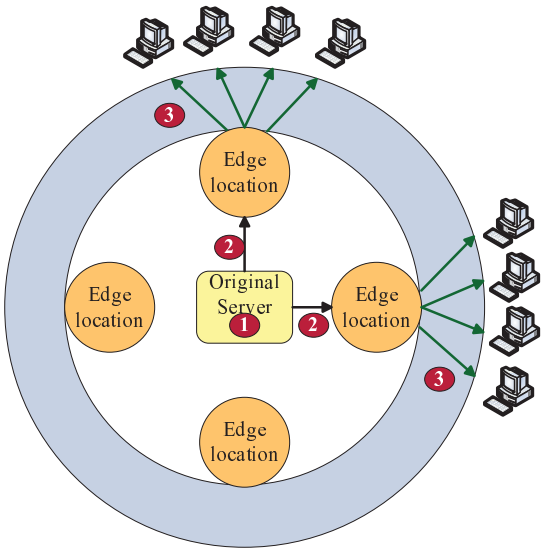


Figure 5.5: System model of content cloud

Considering the strong dynamics of the access patterns of OSN users, we are particularly interested in the content management and update strategies in the SNACS' implementation, including reducing the number of misses, selecting the right edge locations for each request, and running at a low cost. The cost to use content cloud service consists of three parts (1) charges for storing objects with original servers (e.g., Amazon S3), (2) charges for data transfer between original servers and edge locations (e.g., CloudFront), and (3) charges for serving data from edge locations. The storage is charged by usage time on a per unit time rate; and we let P_s denote the unit storage price of storing objects in cloud origin servers. The last

two are by traffic volume on a per byte rate; and we let P_e be the unit data transfer price of serving objects from edge locations, and P_c be the unit data transfer price of copying objects to edge locations. Given a sequence that has a length of time T, the cost during time T can be formulated as

$$Cost = S_z \cdot P_s \cdot T + B_c \cdot P_c + B_e \cdot P_e \tag{5.1}$$

where S_z is the storage size of the original server; B_c is the data transfer of copying objects from the original servers to edge locations; B_e is the data transfer of serving videos from edge locations.

Given the storage size and the user request sequence, our SNACS needs to maximize the video sharing performance while still maintaining low costs. Since Amazon's CloudFront has already offered edge locations selection algorithms that are known to be effective [44], we mainly focus on minimizing the misses and replacements so as to reduce the cost of SNACS.

5.4 Optimal Off-line Scheduling Algorithm

In this section, we propose off-line solutions that can yield optimal results if the user requests are known *a priori*, which then motivates our online algorithm design in the next section.

5.4.1 Scheduling with Minimum Miss Rate

We start by proposing a scheduling algorithm that minimizes the miss rate and proving its optimality. We then extend the algorithm to also minimize the replacement rate in the next subsection. It worths noting that our problem is different from the classic miss and replacement problem [26], since in our problem, even a miss happen, we may not always do the replacement as in the classic problem. For example, we assume the request sequence is "$A\ B\ C\ B\ A$". The storage is of size 2 and initially empty. When a miss happens, the optimal solution (known as S_{FF} for Farthest-in-Further scheduling [39]) for the classic problem will always take in the missed video to replace the video in the storage whose next request occurs furthest in the future request sequence, which leads to 4 misses and 1 replacement for the aforementioned example. Yet we can easily find the optimal solution for our problem is 3 misses and 0 replacement, if we do not update C into the storage when its request misses. This shows that the solution for the classic problem actually does not work well in our problem. To this end, we propose a new algorithm S_{OPT_M} as shown in Algorithm 2 to address this new problem optimally.

For ease of exposition, we use dist[x] to denote the distance from the current position to the position where the first request to video x after the current position. We also define *reduced scheduling algorithm* if in the algorithm, a replacement can only happen when a request misses

Algorithm 2 S_{OPT_M}

```
if current request to video d misses then
  search the rest request sequence until each video v_i currently in the storage and d occur
  at least once;
  if ∃ v_i such that dist[d]< dist[v_i] then
    find the video v in the storage that maximizes dist[v];
    replace v for video d;
  else
    do no replacement;
  end if
else
  do no replacement;
end if
```

(although when a request misses, a replacement may not happen.). We thus have the following two lemmas:

Lemma 1. *For any giving scheduling algorithm S, there exists a reduced version $\bar{S}$ of S, where $\bar{S}$ is a reduced scheduling algorithm that brings in at most as many videos as the scheduling algorithm S does.*

Proof. We prove the lemma by constructing $\bar{S}$ as follows: each time when S replaces a video d that has not been requested into the storage, we can defer the replacement of video d until d is actually requested. Hence, the number of replacements by $\bar{S}$ is at most as many as S. □

Lemma 2. *Let S be a reduced scheduling algorithm that makes the same decision as S_{OPT_M} through the first j requests in the request sequence, for a number of j. Then there is a reduced scheduling algorithm S' that makes the same decisions as S_{OPT_M} through the first $j+1$ requests, and incurs no more misses than S does.*

Proof. To prove that there must exist such a reduced scheduling algorithm S', we should construct S' by trying to get the storage content back to the same state as S as quickly as possible, while not incurring more misses. If the storages of S and S' are the same, we can finish the construction of S' by just making it behave exactly same as S afterwards.

The detailed proof can be found in the Appendix.

□

The optimality of the scheduling algorithm S_{OPT_M} can then be shown by the following theorem:

Theorem 3. *S_{OPT_M} incurs no more misses than any other schedule S and hence is optimal in terms of achieving the minimum miss rate in our problem.*

Proof. To prove that the scheduling algorithm S_{OPT_M} is optimal, we begin with an optimal schedule S^*, and use Lemma 2 to construct a schedule S_1 that agrees with S_{OPT_M} through

the first step. We then continue applying Lemma 2 inductively for $j = 1, 2, 3, \cdots, m$, producing schedules S_j that agree with S_{OPT_M} through the first j requests. Each scheduling algorithm incurs no more misses than the previous one. We then have $S_m = S_{OPT\_M}$, since it agrees with it through the whole sequence. □

5.4.2 Extension to Minimize Miss Rate and Replacement Rate

Although S_{OPT_M} minimizes the miss rate, we find it may not always minimize the replacement rate. For example, assume the video request sequence is "*A B C B C A*", and the storage size is two. S_{OPT_M} will put video C into the storage at the third request by replacing A, and it will lead to 3 misses and 1 replacement. While a better way could be 3 misses and 0 replacement if there is no replacement when the third request to C misses in the storage. To further reduce the cost according to Eq. 5.1, it is necessary to find a solution which incurs both minimum miss rate and replacement rate.

To this end, one naïve approach is to use the exhaustive search on the scheduling decision tree, where for each request in the request sequence, we need to explore decision options such as whether to do replacement and if so, which video in the storage should be replaced out and which video should be taken in. However, even the exhaustive search can be further improved by using our S_{OPT_M} as a bound on the miss rate, the solution space can still be very large. We thus introduce two rules that can actually help analyze and improve the optimality of S_{OPT_M} for the replacement rate, which, together with S_{OPT_M}, will be incorporated into a guided search algorithm that can greatly shrink the solution space and efficiently find the optimal solution. We start from the first rule:

Rule 1: If a miss for requesting video d happens, and for each video v currently cached in the storage, we have $dist[d] \geq dist[v]$, then there is no replacement for d.

And we have the following lemma:

Lemma 4. *Given a reduced scheduling algorithm S, if Rule 1 is broken at least once, then there always exists a scheduling algorithm S' that never breaks Rule 1 and incurs no more misses and replaces than S does.*

To prove Lemma 4, we can assume that when a miss for requesting video d happens, S breaks Rule 1 and replaces video f for d. To construct S', we still try to have S' agrees with S in the storage content as quickly as possible. We can then finish the construction of S' by setting $S' = S$ thereafter. Note that, after requesting d misses, S and S' are slightly different in that S has video d and S' has video f. We can then use a similar approach to the proof for Cases 2-4 in Lemma 2 to prove this lemma. Due to the space limitation, we omit this proof as well as the proofs for Lemma 5 and Theorem 6, where the latter two can be proved by approaches similar

to proving Lemma 2 and Theorem 3, respectively. The detailed proofs for Lemma 4, Lemma 5 and Theorem 6 can be found in the Appendices.

From Lemma 4, we can directly get that by enforcing Rule 1 in Algorithm S_{OPT_M}, we can not only achieve minimized miss rate, but also incur no unnecessary replacements if $dist[d] \geq dist[v]$ for a missed request to video d and any video v in current storage. We now go on with the second rule:

Rule 2: If there is a miss and a replacement is required, then only replace the video v in current storage such that $dist[v] \geq dist[v']$ for any other video v' in current storage.

And we then have the following lemma:

Lemma 5. *Let S be a minimum-miss scheduling algorithm and agrees with Rule 2 through the first j requests. There exists a scheduling algorithm S', which agrees with Rule 2 through the first $j+1$ requests and incurs the same number of misses and no more replacements than S.*

Algorithm 3 S_{OPT_MR}

```
if reach the end of the request sequence then
  compare the current schedule with the best schedule found till now and keep the better
  one;
else
  if current request to video d misses then
    search the rest request sequence until each video v_i currently in the storage and d
    occur at least once;
    if ∃ v_i such that dist[d]< dist[v_i] then
      recursively handle next request with S_OPT_MR;

      // enforce Rule 2
      find the video v in storage that maximizes dist[v];
      replace v for video d;

      recursively handle next request with S_OPT_MR;
    else
      // enforce Rule 1
      do no replacement;

      recursively handle next request with S_OPT_MR;
    end if
  else
    do no replacement;
    recursively handle next request with S_OPT_MR;
  end if
end if
```

Lemma 5 tells that when Rule 2 is enforced in Algorithm S_{OPT_M}, if a replacement is necessary, replacing by Rule 2 can still keep the solution optimal in terms of both miss rate

and replacement rate. Therefore, instead of using the naïve exhaustive search, we can apply Algorithm S_{OPT_M} with Rule 1 and Rule 2 to do an efficient guided search that only explores the branches potentially leading to the optimal solution on both miss rate and replacement rate, while intelligently cutting off all the others. We call this new algorithm S_{OPT_MR} and summarize it in Algorithm 3[2]. We then have the following theorem:

Theorem 6. *S_{OPT_MR} incurs no more misses and replacements than any other schedule S and hence is optimal in terms of achieving the minimum miss rate and replacement rate in our problem.*

5.5 Online Scheduling Implementation

Different from the offline scheduling algorithm for which the optimality on the miss rate and replacement rate is of the first importance, the online scheduling implementation requires that the solution is simple and highly efficient yet achieving reasonably good performance and only based on the information that the system currently has, i.e., not relying on the future information as the offline algorithm does.

For the classic miss and replacement problem, LRU (Least Recently Used) is a well accepted implementation that approximates the optimal offline scheduling algorithm S_{FF}. Thus one straightforward solution is to directly apply LRU to our problem. However, like S_{FF}, LRU also has the same limitation for solving our problem, i.e., it always does replacement when a miss happens. In addition, LRU simply replaces the least recently used item when a miss occurs, and hence fails to consider the specific features of online social video sharing. Yet one major principle that we can still learn from LRU is that it actually uses historical statistics to approximate the future request sequence used in S_{FF}. Therefore, in this section, we will first discuss how we can approximate future user requests in our problem. We will then propose our online scheduling implementation that can successfully incorporate what we have learned from the optimal offline scheduling algorithm discussed in the previous section.

5.5.1 Approximate Future User Requests

To incorporate the lessons learned from the offline optimal algorithm, for each video v currently cached in the storage and the missed video d, we need to know $dist[v]$ and $dist[d]$, respectively. In other words, we need to know how soon each of these videos will be requested in the future. To achieve this, we need to predict the popularity of these videos based on the information in the OSN. There are a number of studies [31][64] to address this problem. Yet most of them are based on a relatively large time scale, say, one hour or even one day. To afford a finer

[2]For ease of exposition, we use the recursive version here, while the non-recursive version can be more efficient for implementation.

time granularity which is essential to our online solution, we develop an efficient approximation algorithm based on the approach proposed in [31].[3]

The approximation solution works as follows: we first search backwards within previous K video requests and identify those users who recently issued the requests for the videos currently cached in the storage as well as for the missed video. If a user decides to share the video after watching the requested video, we then look at its neighbors in the OSN and count those who have not requested this video. We also maintain the popularity of this user. In particular, for each video previously shared by this user, we count the number of neighbors who actually viewed the video and divide it by the total number of neighbors. The popularity of this user is thus the average value on all the videos previously shared by this user. For each video that we are interested in, we calculate the sum for each user who requests this video by adding the number of the potential viewers of this user weighted by the popularity of this user. We then use the reciprocal value of this sum to approximate how soon this video will be requested in the future. For brevity, we call this reciprocal value as the approximation value.

Note that the rationale of this approximation is that, if a video is very popular in the OSN, i.e., there are a large number of OSN users who tend to request it (which is different from a large number of users who have viewed and shared it), then it is likely for this video to be requested soon in the future. In the next subsection, we will use this approximation algorithm to incorporate what we have learned from the optimal offline solution, which leads to our online scheduling implementation.

5.5.2 Incorporate Lessons Learned from Offline Optimal Solution

Algorithm 4 S_{OSN_MR}

if current request to video d is not in the storage **then**
  search the request sequence backwards for K requests;
  calculate the approximation value for each video v_i currently in the storage and for the missed video d;
  if $\exists\ v_i$ such that approx[d]$<$ approx[v_i] **then**
    find video v in the storage that maximizes approx[v];
    replace v for video d;
  else
    do no replacement;
  end if
else
  do no replacement;
end if

Remember that, from the design of optimal offline scheduling algorithm, we have learned

[3]Note that our focus here is on how to use a highly efficient prediction solution to effectively approximate the future user requests for our online implementation. While proposing an algorithm with better predication results for OSNs is very important and can be useful to our solution, it is generally out of the scope of this chapter.

the following lessons:

1. When a video request misses, a replacement may not always happen. Especially when the situation in Rule 1 happens, we should never do the replacement.

2. When a replacement must be done, we should always do the replacement according to Rule 2.

With the approximation algorithm proposed in the last subsection, we can now interpret Rule 1 as follows: if the approximation value of the missed video is larger than that of any video currently in the storage, then there is no replacement for the missed video. Similarly, Rule 2 can be interpreted as follows: if there is a miss and a replacement is required, then we should replace the video with the largest approximation value in current storage.

Based on these interpretations, we can then propose our online scheduling implementation for OSN video sharing, called S_{OSN_MR}, as shown in Algorithm 4, where approx[x] denotes the approximation value of the video x. Compared to the aforementioned straightforward solution using LRU, our solution is still reasonably simple and highly efficient. More importantly, it fully exploits the specific features of online social video sharing and successfully incorporates what we have learned from the optimal offline solution. In the next section, we will further demonstrate its superiority by our extensive performance evaluation with the real traces from one of the largest OSN in China.

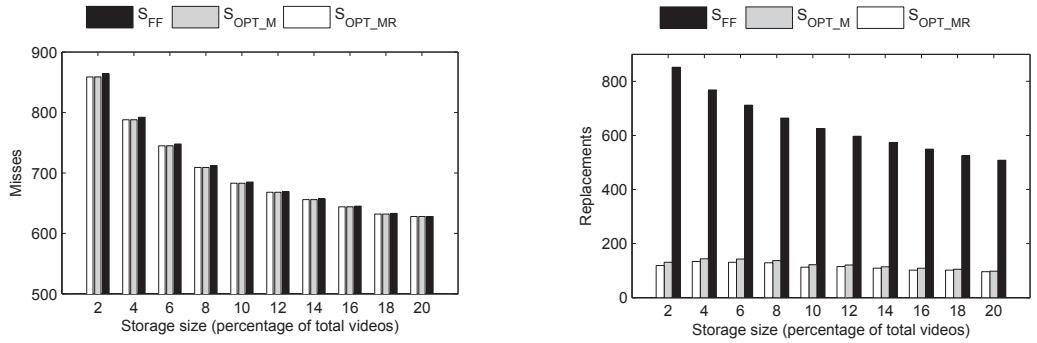
5.6 Performance Evaluation

We have conducted extensive trace-based simulations to evaluate our solutions for SNACS. To this end, we collected the traces of video requests in the RenRen OSN [33] over three months. We find the video requests show a strong daily pattern [34] and thus choose a trace in a typical one-day period (April 19^{th}, 2011) for our simulations, which contains $19,473,512$ requests and involves $278,922$ unique videos.

5.6.1 Comparison of Offline Algorithms

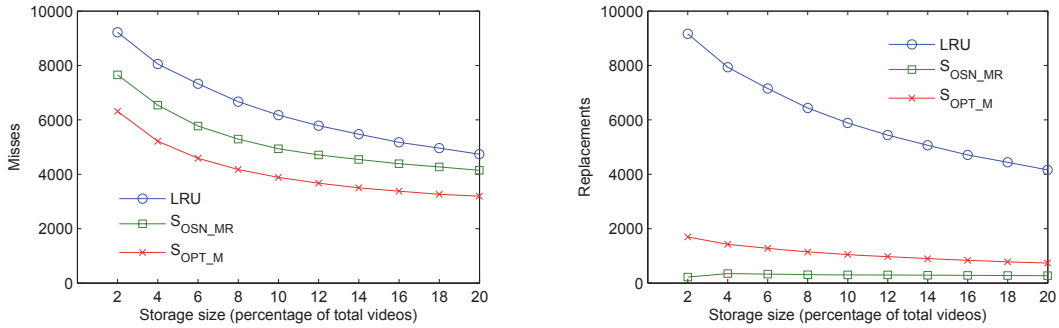
To evaluate our offline algorithms S_{OPT_M} and S_{OPT_MR}, we also implement the optimal offline solution S_{FF} for the classic miss and replacement problem. Fig. 5.6 shows the results on the miss number and the replacement number[4]. It is clear to see that both S_{OPT_M} and S_{OPT_MR} outperform S_{FF} for solving our problem. In particular, although S_{FF} is the optimal solution for the classic miss and replacement problem, we can see that by allowing no replacement,

[4]In worst case, each offline algorithm may traverse the whole trace to find if a video is requested in the future and thus causes enormous time to finish. To this end, we shrink the trace to a subset by randomly sampling 10% of the requests in the April 19^{th} trace.



(a) misses (b) replacements

Figure 5.6: Comparison between offline algorithms



(a) misses (b) replacements

Figure 5.7: Comparison between online algorithms and the optimal algorithm

our S_{OPT_M} and S_{OPT_MR} can achieve even fewer misses (Fig. 5.6(a)), which becomes even observable as the storage size decreases. This result further confirms with our theoretical analysis on optimality in Section 5.4.

In terms of replacement number (Fig. 5.6(b)), our S_{OPT_M} and S_{OPT_MR} perform even much better than S_{FF}, by successfully reducing the replacement number for 75% to 85%. One interesting observation is that for the replacement number, S_{OPT_M} performs very close to S_{OPT_MR}. This further demonstrates the effectiveness of our Rule 1 and Rule 2 derived in Section 5.4.2, since these two rules are also reflected in S_{OPT_M} implicitly.

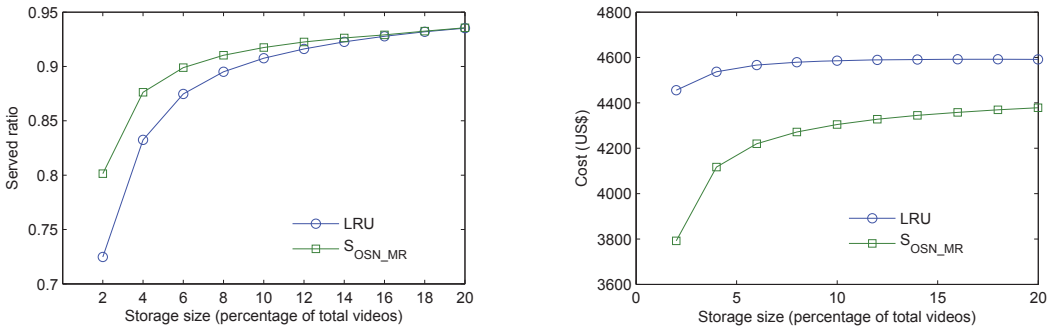
5.6.2 Online Implementation vs. Offline Algorithm

Now we go on to compare the performance of our online implementation S_{OSN_MR} with the offline solution. Since S_{OPT_M} performs very close to the optimal offline algorithm S_{OPT_MR} and is more efficient, we thus use S_{OPT_M} for this comparison. In addition, we also implement LRU. The results on miss number and replacement number are shown in Fig. 5.7. It is easy to

see that our S_{OSN_MR} performs much better than LRU and stays close to S_{OPT_M}, especially in terms of the replacement number. This is because our solution can well approximate the video popularity and thus the user requests in the future with the information from OSNs, while LRU always replaces the least recently requested video when a miss occurs.

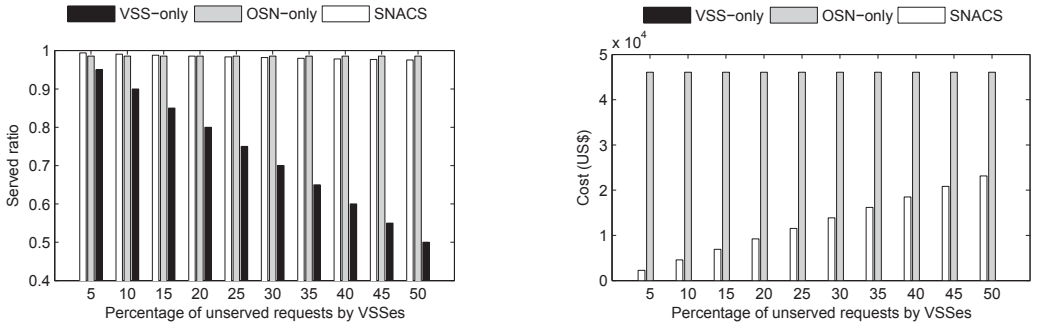
Another observation comes from comparing the miss number and replacement number together. Unlike LRU, which performs bad on both the miss number and replacement number, S_{OSN_MR} incurs fewer replacements than S_{OPT_M} at a cost of a slight increase in the miss number. Therefore, in the future work, it is interesting to investigate such tradeoffs between miss rate and replacement rate.

5.6.3 Impacts to Served Ratio and Cost of OSN



(a) performance (b) cost

Figure 5.8: Comparison between online algorithms



(a) performance (b) cost

Figure 5.9: Comparison between three architectures

Besides the miss number and replacement number, we also investigate the impacts to the cost by using the cloud assistance. To this end, we adopt a typical setting as used in [64]. We assume each video has the same file size denoted as F_z. Then the storage size can be

represented as the number (denoted as N_s) of stored videos in the cloud. The data transfer of copying objects to edge locations can be represented as the product of the number of edge locations (N_e) and the number (N_r) of video replacements in the cloud. The data transfer of serving videos from edge locations can be represented as the number (N_h) of hit requests by the cloud. The cost in Eq. 5.1 can be rewritten as Eq. 5.2.

$$Cost = N_s \cdot F_z \cdot P_s \cdot T + N_r \cdot F_z \cdot N_e \cdot P_c + N_h \cdot F_z \cdot P_e \tag{5.2}$$

According to Amazon pricing model [2], we set P_e=\$0.12 per GB, P_s=\$0.08 per GB per month, P_c=\$0.02 per GB. For other parameters, we set N_e=5, F_z=20MB in this chapter. We also conduct simulations under other parameter settings and find the results generally follow a similar trend.

Fig. 5.8 shows the results of the served ratio (the fraction of the video requests from the OSN that are well served by VSS or OSN servers over the total requests from the OSN) and the money spent by the OSN. Again our S_{OSN_MR} still outperforms LRU especially when the storage size is small. Moreover, compared to LRU, our S_{OSN_MR} can also greatly reduce the total costs by 5% to 15%. Besides comparing with the SNACS framework, we also conduct simulations to compare our SNACS solution with the *VSS-only* and *OSN-only* solutions. The *VSS-only* solution, which is the current development architecture in real life, assumes that all the video requests from OSNs are served by VSSes. The *OSN-only* solution, which idea is reflected in early work [64], assumes all the video requests from OSNs are served by the video servers operated by the OSN. We vary the percentage (5% to 50%) of daily requests that are unserved by VSSes due to under-provision, and examine how the served ratio and cost would change. Fig. 5.9 shows the results. We can see that although both *OSN-only* and SNACS can fundamentally improve the user experience, SNACS can achieve similar performance with significantly less cost.

5.7 Summary

In this chapter, we proposed a framework, called SNACS, for the OSN to cost-effectively enhance its video viewing experience by leveraging content cloud service. Given the strong dynamics of the video access patterns in the OSN, we were particularly interested in the content management and update strategies in the SNACS'implementation. We showed that conventional cache replacement strategies can be quite inefficient in SNACS. We then developed an optimal offline replacement algorithm that generates minimum misses in this new context. We further offered guidelines to minimize replacements among the solutions of the lowest misses. The optimal offline solutions not only provide a benchmark for comparison but also motivate the design of an online replacement algorithm, which makes effective use of the video sharing patterns in the OSN. The superiority of our design was confirmed by trace-driven simulations.

Chapter 6

Conclusion

In this thesis, we investigated a broad spectrum of issues about video propagation in OSNs, from perspectives of measurement, modeling analysis, and system enhancement. Even though, this is a relatively new research topic and many further works are needed.

6.1 Summary of this Thesis

First, we provided the first major stab at characterizing video requests from OSNs, by analyzing the logs of video viewing and sharing behaviors in a large-scale OSN over several months. Our measurement unveiled both static and temporal characteristics of video requests from OSNs, highlighting several distinctive features from the requests directly from VSSes. To better understand the characteristics observed in our empirical data, we built an emulator to model video viewing and sharing behaviors in OSNs. Although simple, our emulator well captures the observed characteristics in the empirical data, including the video popularity distribution and dynamics. This emulator can work as a video request generator.

Second, we further explored video propagations in OSNs from both measurement and modeling. Specifically, we measured video propagation structure, factors influencing video popularity, and user sharing and viewing behaviors. We further proposed an S^2I^3R model which extends the conventional epidemic models to accommodate diverse types of users and their probabilistic viewing and sharing behavior. This model can serve as a valuable foundation for such applications as traffic prediction, and resource provision of video servers.

Third, we conducted an initial study on popularity prediction of videos shared in OSNs. Our measurement results show the video views in early and later times exhibit a much less correlation than that in VSSes, which poses significant challenges on conventional views-based prediction models. Our experiments with such conventional prediction models as ARIMA, MLR, and kNN confirmed their ineffectiveness in this new context, especially when predicting the requests bursts that are common for the evolutions of videos shared in OSNs. To overcome the

limits, we developed a dynamic model to analyze the video propagation process, and accordingly presented a propagation-based prediction framework, SoVP. SoVP considers both video attractiveness and social context in predicting future video views, whose accuracy has been demonstrated by trace-driven experiments.

Fourth, we proposed a framework, called SNACS, for the OSN to cost-effectively enhance its video viewing experience by leveraging content cloud service. Given the strong dynamics of the video access patterns in the OSN, we were particularly interested in the content management and update strategies in the SNACS'implementation. We showed that conventional cache replacement strategies can be quite inefficient in SNACS. We then developed an optimal offline replacement algorithm that generates minimum misses in this new context. We further offered guidelines to minimize replacements among the solutions of the lowest misses. The optimal offline solutions not only provide a benchmark for comparison but also motivate the design of an online replacement algorithm, which makes effective use of the video sharing patterns in the OSN. The superiority of our design was confirmed by trace-driven simulations.

6.2 Future Directions

Extending measurements to other OSNs. Our current measurements are mainly based on RenRen OSN. First, it might be interesting to compare video propagations in Facebook and RenRen. They have similar layouts and functions but different user bases (RenRen users are mainly from China, while rarely Chinese users use Facebook). By comparing this, we can see how the culture difference affects user behaviors and thereby video propagations. Second, Twitter and lots of its copycats around the world (e.g., Weibo [1] in China) are another kind OSN, which provide microblogging services. Extending measurements to Twitter-like OSNs can provide more comprehensive understanding on video propagation in social networks.

Enhancing current system with closer collaboration between OSNs/VSSes. Enabling users sharing videos from VSSes to OSNs has brought additional traffic for both of them. Yet, their collaborations are still limited. A closer collaboration between them could benefit them more in enhancing their current systems. First, they can collaborate by sharing statistics such as the total views, likes, shares in their system. With such information, we can design more efficient recommendation and prediction strategies. Second, they can make joint effect to provide storage and networking service for the shared videos.

Further works on video popularity prediction. We have made a first attempt to make video popularity predictions using its propagation information in OSNs, but the current work still has limitations. First of all, although the proposed SoVP model can generally get better prediction than the conventional views-based prediction models, its complexity and scalability are not as good as them. Therefore, a compromised solution between SoVP and the conventional

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

models may be a better choice. For example, one possible solution could simplify SoVP by only leveraging recent video propagation information; another choice is to incorporate propagation information into conventional prediction models. Furthermore, our current work only considered predictions in the granularity of one day. It is also important to predict finer granularity such as one hour or longer granularity such as one week or one month.

Bibliography

[1] http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Paying.html. [Online; accessed 25-July-2013]. 62

[2] http://aws.amazon.com/pricing/. [Online; accessed 25-July-2013]. 73

[3] Amazon-CloudFront. http://aws.amazon.com/cloudfront/. 59, 63

[4] Amazon-S3. http://aws.amazon.com/s3/. 59, 63

[5] E. Bakshy, I. Rosenn, C. Marlow, and L. Adamic. The Role of Social Networks in Information Diffusion. In *Proc. of WWW*, 2012. 4

[6] Y. Borghol, S. Mitra, S. Ardon, N. Carlsson, D. Eager, and A. Mahanti. Characterizing and modeling popularity of user-generated videos. In *Proc. of IFIP Perfromance*, 2011. 5

[7] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. B. Moon. I Tube, You Tube, Everybody Tubes: Analyzing the Worlds Largest User Generated Content Video System. In *Proc. of IMC*, 2007. 4, 8, 12, 14, 15, 16, 18, 60

[8] M. Cha, A. Mislove, B. Adams, and K. P. Gummadi. Characterizing Social Cascades in Flickr. In *Proc. of WOSN*, 2008. 4

[9] M. Cha, A. Mislove, and K. P. Gummadi. A Measurement-driven Analysis of Information Propagation in the Flickr Social Network. In *Proc. of WWW*, 2009. 4

[10] V. Chaoji, S. Ranu, R. Rastogi, and R. Bhatt. Recommendations to Boost Content Spread in Social Networks. In *Proc. of WWW*, 2012. 4

[11] X. Cheng, C. Dale, and J. Liu. Statistics and Social Network of YouTube Videos. In *Proc. of IWQoS*, 2008. 4, 38

[12] X. Cheng, H. Li, and J. Liu. Video Sharing Propagation in Social Networks: Measurement, Modeling, and Analysis. In *Proc. of INFOCOM mini-conference*, 2013. 4, 43, 50, 53

[13] comScore. http://www.comscore.com/insights/press_releases/2013/10 /comscore_releases_september_2013_us_online_video_rankings. 1

[14] R. Crane and D. Sornette. Viral, Quality, and Junk Videos on YouTube: Separating Content From Noise in an Information-Rich Environment. In *AAAI Spring Symposium*, 2008. 4

[15] D. J. Daley, J. Gani, and J. M. Gani. *Epidemic Modelling: An Introduction*. Cambridge Studies in Mathematical Biology. Cambridge University Press, 2001. 38

[16] F. Figueiredo, F. Benevenuto, and J. Almeida. The Tube over Time: Characterizing Popularity Growth of YouTube Videos. In *Proc. of WSDM*, 2011. 4

[17] W. Galuba, D. Chakraborty, K. Aberer, Z. Despotovic, and W. Kellerer. Outtweeting the twitterers - predicting information cascades in microblogs. In *Proc. of WOSN*, 2010. 4, 5

[18] A. Ganesh, L. Massoulie, and D. Towsley. The Effect of Network Topology on the Spread of Epidemics. In *Proc of IEEE INFOCOM*, 2005. 5

[19] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. YouTube Traffic Characterization: a View from the Edge. In *Proc. of IMC*, 2007. 4, 5

[20] B. Golub and M. O. Jackson. Using Selection Bias to Explain the Observed Structure of Internet Diffusions. In *Proc. of National Academy of Sciences of the United States of America*, 2010. 5, 28

[21] G. Grsun, M. Crovella, and I. Matta. Describing and Forecasting Video Access Patterns. In *Proc. of INFOCOM*, 2011. 46, 47

[22] T. Hogg and K. Lerman. Stochastic Models of User-Contributory Web Sites. In *Proc. of ICWSM*, 2009. 49

[23] L. Hong, O. Dan, and B. D. Davison. Predicting Popular Messages in Twitter. In *Proc. of WWW*, 2011. 4, 5

[24] J. Jiang, C. Wilson, X. Wang, P. Huang, W. Sha, Y. Dai, and B. Y. Zhao. Understanding Latent Interactions in Online Social Networks. In *Proc. of IMC, 2010*. 9, 32, 42

[25] S. Jin and A. Bestavros. GISMO: A Generator of Internet Streaming Media Objects and Workloads. In *Proc. of SIGMETRICS Performance Evaluation Review*, 2001. 5, 21

[26] J. Kleinberg and Éva Tardos. *Algorithm Design*. Addison-Wesley, 2005. 64

[27] F. Kooti, W. A. Mason, K. P. Gummadi, and M. Cha. Predicting Emerging Social Conventions in Online Social Networks. In *Proc. of CIKM*, 2012. 5

[28] K. Lerman and R. Ghosh. Information Contagion: an Empirical Study of the Spread of News on Digg and Twitter Social Networks. In *Proc. of ICWSM*, 2010. 4

[29] K. Lerman and T. Hogg. Using a Model of Social Dynamics to Predict Popularity of News. In *Proc. of WWW*, 2010. 4, 5

[30] H. Li, J. Liu, K. Xu, and S. Wen. Understanding Video Propagation in Online Social Networks. In *Proc. of IWQoS*, 2012. 4, 43, 49, 60

[31] H. Li, X. Ma, F. Wang, J. Liu, and K. Xu. On Popularity Prediction of Videos Shared in Online Social Netowrks. In *Proc. of ACM CIKM*, 2013. 59, 68, 69

[32] H. Li, F. Wang, Y. Le, J. Liu, and K. Xu. SNACS: Social Network-Aware Cloud Assistance for Online Propagated Video Sharing. Technical report, Simon Fraser University, 2013. [Online; https://s3-us-west-2.amazonaws.com/haitao-papers/SNACS.pdf]. 4

[33] H. Li, H. Wang, J. Liu, and K. Xu. Video Sharing in Online Social Network: Measurement and Analysis. In *Proc. of NOSSDAV*, 2012. 1, 4, 46, 55, 70

[34] H. Li, H. Wang, J. Liu, and K. Xu. Video Requests from Online Social Networks: Measurement, Analysis and Generation. In *Proc. of INFOCOM*, 2013. 4, 60, 70

[35] H. H. Liu, Y. Wang, Y. R. Yang, H. Wang, and C. Tian. Optimizing Cost and Performance for Content Multihoming. In *Proc. of SIGCOMM*, 2012. 46

[36] Z. Liu, Y.-C. Lai, and N. Ye. Propagation and Immunization of Infection on General Networks with Both Homogeneous and Heterogeneous Components. *Physical Review E*, 67(1):031911, 2003. 5

[37] X. Ma, H. Wang, H. Li, J. Liu, and H. Jiang. Enhancing Recommended Video Lists for Youtube-like Social Media. In *Proc. of IEEE MMSP*, 2012. 6

[38] J. S. Maritz. *Distribution-free Statistical Methods*. 1995. 15, 16

[39] R. L. Mattson, J. Gecsei, D. R. Slutz, and I. L. Traiger. Evaluation Techniques for Storage Hierarchies. *IBM Systems Journal*, 1970. 64

[40] S. A. Myers, C. Zhu, and J. Leskovec. Information Diffusion and External Influence in Network. In *Proc. of KDD*, 2012. 4

[41] A. Navot, L. Shpigelman, N. Tishby, and E. Vaadia. Nearest Neighbor Based Feature Selection for Regression and Its Application to Neural Activity. In *Proc. of NIPS*, 2006. 47, 48

[42] M. Newman. Spread of Epidemic Disease on Networks. *Physical Review E*, 66(1):016128, July 2002. 5

[43] D. Niu, Z. Liu, and B. Li. Demand Forecast and Performance Prediction in Peer-Assisted On-Demand Streaming Systems. In *Proc. of INFOCOM*, 2011. 46, 47

[44] D. Rayburn. Comparing CDN Performance: Amazon CloudFront's Last Mile Testing Results. Technical report, Frost & Sullivan, 2012. 64

[45] RenRen. http//www.renren.com. 1

[46] J. L. Rodgers and W. A. Nicewander. *Thirteen Ways to Look at the Correlation Coefficient*. The American Statistician, 1988. 15, 16, 60

[47] T. Rodrigues, F. Benvenuto, M. Cha, K. P. Gummadi, and V. Almeida. On Word-of-Mouth Based Discovery of the Web. In *Proc. of IMC*, 2011. 4

[48] M. Rowe. Forecasting Audience Increase on YouTube. In *Proc. of the International Workshop on User Profile Data on the Social Semantic Web*, 2011. 47, 48

[49] S. Scellato, C. Mascolo, M. Musolesi, and J. Crowcroft. Track Globally, Deliver Locally: Improving Content Delivery Networks by Tracking Geographic Social Cascades. In *Proc. of WWW*, 2011. 4, 60

[50] D. A. Shamma, J. Yew, L. Kennedy, and E. F. Churchill. Viral Action: Predicting Video View Counts Using Synchronous Sharing Behaviors. In *Proc. of ICWSM*, 2011. 46

[51] G. V. Steeg, R. Ghosh, and K. Lerman. What Stops Social Epidemics? In *Proc. of ICWSM*, 2011. 4, 43

[52] E. Sun, I. Rosenn, C. Marlow, and T. Lento. Gesundheit! Modeling Contagion through Facebook News Feed. In *Proc. of ICWSM*, 2009. 4

[53] G. Szabo and B. A. Huberman. Predicting the Popularity of Online Content. *Commun. ACM*, 2010. 46

[54] W. Tang, H. Labs, and Y. Fu. Medisyn: A Synthetic Streaming Media Service Workload Generator. In *Proc. of NOSSDAV*, 2003. 5

[55] P. van Kemenade. Predicting the Propagation of Video Content on Twitter. Technical report, University of Amsterdam, 2011. 4

[56] D. Wang, Z. Wen, H. Tong, C.-Y. Lin, C. Song, and A.-L. Barabasi. Information Spreading in Context. In *Proc. of WWW*, 2011. 5

[57] Z. Wang, B. Li, L. Sun, and S. Yang. Cloud-based Social Application Deployment using Local Processing and Global Distribution. In *Proc. of ACM CoNext*, 2012. 4, 59, 61

[58] Z. Wang, L. Sun, X. Chen, W. Zhu, J. Liu, M. Chen, and S. Yang. Propagation-Based Social-Aware Replication for Social Video Contents. In *Proc. of ACM Multimedia*, 2012. 4, 6, 9, 46, 59, 60, 61

[59] Z. Wang, L. Sun, C. Wu, and S. Yang. Guiding Internet-Scale Video Service Deployment Using Microblog-based Prediction. In *Proc. of ICWSM*, 2012. 9

[60] Z. Wang, L. Sun, W. Zhu, S. Yang, H. Li, and D. O. Wu. Joint Social and Content Recommendation for User Generated Videos in Online Social Network. *IEEE Transactions on Multimedia*, 2013. 6

[61] Z. Wang, C. Wu, L. Sun, and S. Yang. Peer-Assisted Social Media Streaming With Social Reciprocity. *IEEE Transactions on Network and Service Management*, 2013. 4, 6

[62] X. Wei, J. Yang, and L. A. Adamic. Diffusion Dynamics of Games on Online Social Networks. In *Proc. of WOSN*, 2010. 4

[63] T. Wu, M. Timmers, D. D. Vleeschauwer, and W. V. Leekwijck. On the Use of Reservoir Computing in Popularity Prediction. In *Proc. of ICCGI*, 2010. 46

[64] Y. Wu, C. Wu, B. Li, L. Zhang, Z. Li, and F. C. Lau. Scaling Social Media Applications into Geo-Distributed Clouds. In *Proc. of IEEE INFOCOM*, 2012. 4, 6, 59, 61, 68, 72, 73

[65] YouTube. http://www.youtube.com/t/press_statistics. 1

[66] G. Yule. A Mathematical Theory of Evolution Based on the Conclusions of Dr. J. C. Willis. *Philosophical Transactions of the Royal Society of London*, 1925. 21

[67] L. Zhang, T. Peng, Y. Zhang, and X. Wang. Content or Context: Which Carries More Weight in Predicting Popularity of Tweets in China. In *Proc. of WAPOR*, 2012. 4

[68] B. Zhao, Y. Li, , and J. C. Lui. Mathematical Modeling of Advertisement and Influence Spread in Social Networks. In *Proc of IEEE NetEcon*, 2009. 5

[69] R. Zhou, S. Khemmarat, and L. Gao. The Impact of YouTube Recommendation System on Video Views. In *Proc. of IMC*, 2010. 8, 59

[70] M. Zink, K. Suhb, Y. Gu, and J. Kurosea. Characteristics of YouTube Network Traffic at a Campus Network - Measurements, Models, and Implications. *Computer Networks*, 2009. 4, 5

Appendix A

Proof of Lemma in Chapter 5

A.1 Proof of Lemma 2

Consider the $(j+1)^{th}$ request to video d. If the request hits, then we can just let $S' = S$, since S and S_{OPT_M} make the same decision on the $(j+1)^{th}$ request and have the same storage before the $(j+1)^{th}$ request happens. Therefore, S' incurs no more misses than S does.

If the $(j+1)^{th}$ request to video d misses, then there are 4 more cases as following:

1. **S does not replace and S' does not replace**
   Similar to the above request hit case, we can set $S' = S$.

2. **S replaces and S' replaces**
   If S and S_{OPT_M} both replace the same video in the storage for d, again we can set $S' = S$.

   The interesting case arises when S replaces video f for d while S_{OPT_M} replaces video $e \neq f$ for d. Then S and S_{OPT_M} do not agree through the $(j+1)^{th}$ request. In this case, to make the S' has the same content of S as quickly as possible, we can let S' behave exactly the same as S until one of the following subcases happens for the first time from the $(j+2)^{th}$ request.

   (a) There is a request to video $g \notin \{e, f\}$ and S replaces e for g. In this subcase, we can let S' replace f for g. S and S' then have the same storage content. And we set $S' = S$ afterwards.

   (b) There is a request to video f and S does replacement. If S replaces video e, then after this step, S' agrees with S in the storage content. If the video replaced by S is not e, say e', then S' can replace e' as well, but for video e instead of f, which can make S' and S are the same. However, after this behavior, S' is no longer a reduced scheduler, because it replaces e' for e into the storage when it is not immediately needed. So we can further transform S' to $\bar{S}'$ using Lemma 1, which does not

increase the number of videos brought in by S'. $\bar{S}'$ also incurs no more misses than S does and still keeps the consistent with S_{OPT_M} during the $(j+1)^{th}$ request.

(c) There is a request to video e. Note that due to the property of S_{OPT_M}, before there is a request for e, the farthest video in the future request sequence (line 5 in Algorithm 2), one of the above 2 subcases must have happened and S' has already been constructed. We thus do not need to discuss this subcase.

Therefore, for all subcases, there must exist a new reduced schedule S' (or $\bar{S}'$) that makes the same decisions as S_{OPT_M} through the first $j+1$ requests, and incurs no more misses than S does.

3. S **does not replace and** S' **replaces**
Assume S_{OPT_M} replaces the video e and according to the property of S_{OPT_M}, e is farther than d. Here S and S_{OPT_M} do not agree at the $(j+1)^{th}$ request since S has video e in the storage while S_{OPT_M} has video d in the storage. So like the Case 2 where S has e and S' has f, we can use the same approach to prove the lemma holds for this case.

4. S **replaces and** S' **does not replace**
Assume S replaces the video e for d while S' did no replacement. Thus, S and S' have slight difference after the $(j+1)^{th}$ request since S has item d in the storage and S' has video e in the storage. Again, we can use the same approach to prove the lemma holds for this case as for Case 2 and Case 3.

This finishes the proof for the situation that video d misses and concludes the proof for the whole lemma.

A.2 Proof of Lemma 4

Proof. S' have agreed up to this point, they have the same cache contents.If d is in the cache for the both,then no decision is necessary.However, if d is missed, the interesting case happens when d needs to be bought into the cache,and to do this, S evicts item f while S' does not evict item .Here S and S' do not already agree through step $j+1$ since S has d in cache while S' has f in cache. Hence, we must actually do something nontrivial to construct S'.As the previous proof, we'll have S' try to get this cache back to the same state as S as quickly as possible, while not incurring unnecessary misses and replaces. Once the caches are the same, we can finish the construction of S' by just having it behave like S.
Without loss generality, we assume that when a miss for requesting video d happens, S breaks Rule 1 and replaces video f. To construct S', we still try to have S' agrees with S in the storage

content as quickly as possible. Once the storages of S and S' are the same, we can finish the construction of S' by making it behave exactly the same as S. Note that, after requesting d misses, S and S' have slight different that S has video d and S' has video f. Then there are 2 cases we need to consider.

1. **dist[d]=dist[f]**
   This case happens when d and f never occur again in the request sequence thereafter. Thus, after current request for d, S' behaves exactly like S until the following case happens for the first time.
   There is a request for video $g \notin \{d, f\}$ that is not in the storage of both S and S' and S replaces d for g. In this case, we can let S' replace f for g, and now the storages of S and S' are the same. We can then make S' behave exactly like S for the rest of the sequence. In this case, S' is better than S by having one less replace than S.

2. **dist[d] $>$ dist[f]**
   This case happens when all of the cached videos will appear in the rest of the request sequence. Thus, after current request for d, we still can let S' behave exactly like S until one of the following two cases happens for the first time. First, there is a request for video $g \notin \{d, f\}$ that is not in the storage of S and S replaces d for g. This proof is the same as case 1. Second, there is a request to f. There are three kinds of behaviors in S.

   (a) S does no replacement. To make the storage content of S' back to the same state as S as quickly as possible, S' replaces f for d. However, we must notice that S' is no longer a reduced scheduling algorithm, since it bought in d although it is not immediately needed. Thus, we use Lemma 1 to get the reduction $\bar{S}'$ of S' and this does not increase the number of videos brought in by $\bar{S}'$. Hence, we are done as $\bar{S}'$ has one less miss than S. longer a reduced schedule: it brought in d when it wasn't immediately needed. So to finish this part of the construction,we further transform S' to its reduction $\bar{S}'$ using Lemma 1; this doesn't increase the number of items brought in by S', and it still agrees with S_{OPT} through step $j+1$.Hence, we are done and S' is better than S by having one less miss than S.

   (b) S replaces d for f. In this case, we are all set: S' can simply access f from the storage, and after this step the storages of S and S' will be the same. Now, S' is better than S by having one less miss and one less replacement.

   (c) S replaces $e' \neq d$ for f. we have S' replaces e' as well, and bring in d to the storage. This results in S and S' having the same storages. We still should notice that S' is no longer a reduced schedule and again, we transform S' to its reduction $\bar{S}'$, thus we are done. brought in d when it wasn't immediately needed. So to finish this part of the construction,we further transform S' to its reduction $\bar{S}'$ using **Lemma 1**; this

doesn't increase the number of items brought in by S', and it still agrees with S_{OPT} through step $j+1$.

Hence, in all these cases, we have a new reduced schedule S' that incurs no more misses and replacements than S that breaks Rule 1 at least once.

□

A.3 Proof of Lemma 5

Proof. Without loss generality, we assume it requires a replace in step $j+1$. If S evicts an item, which is farthest in the future which agrees with Rule 2 through step $j+1$, then we can just set $S' = S$.

The interesting case happens when S evicts item f, which is not the farthest item in the future, while S' evicts item $e \neq f$ by Rule 2 and we assume e is the farthest item in the future. Hence, S and S' do not agree through step $i+1$ since S has e in cache while S' has f in cache. Our plan to construct S' is still to try to get this cache back to the same state as S as quickly as possible, while incurs same misses and no more replaces than S. Once the caches are the same, we can finish the construction of S' by just having it behave like S.
Specially, from request $i+2$ onward, S' behaves exactly like S until one of the following this happens for the first time.

1. There is a request to an item $g \neq e, f$ that is not in the cache both of S and S' and S evicts e. We can have S' evict f, and now the caches of S and S' are the same. We can then have S' behave exactly like S for the rest of the sequence.

2. There is a request to an item f. Then in this case, S' hits while S misses, which is impossible to happen, since it is against with the assumption that S is a miss-optimal schedule.

Hence, in both these cases, we have a new reduced schedule S' that agrees with Rule 2 through the first $j+1$ items and incurs the same misses and no more replaces than S. □