Exploring the Emerging Type of Comment for Online Videos: *DanMu*

MING HE, Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China
YONG GE, The University of Arizona
ENHONG CHEN and QI LIU, Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China
XUESONG WANG, China University of Mining and Technology

DanMu, an emerging type of user-generated comment, has become increasingly popular in recent years. Many online video platforms such as Tudou.com have provided the *DanMu* function. Unlike traditional online reviews such as reviews at Youtube.com that are outside the videos, *DanMu* is a scrolling marquee comment, which is overlaid directly on top of the video and synchronized to a specific playback time. Such comments are displayed as streams of moving subtitles overlaid on the video screen. Viewers could easily write *DanMu*s while watching videos, and the written *DanMu*s will be immediately overlaid onto the video and displayed to writers themselves and other viewers as well. Such *DanMu* systems have greatly enabled users to communicate with each other in a much more direct way, creating a real-time sharing experience. Although there are several unique features of *DanMu* and has had a great impact on online video systems, to the best of our knowledge, there is no work that has provided a comprehensive study on *DanMu*. In this article, as a pilot study, we analyze the unique characteristics of *DanMu* from various perspectives. Specifically, we first illustrate some unique distributions of *DanMu*s by comparing with traditional reviews (TReviews) that we collected from a real *DanMu*-enabled online video system. Second, we discover two interesting patterns in *DanMu* data: a herding effect and multiple-burst phenomena that are significantly different from those in TRviews and reveal important insights about the growth of *DanMu*s on a video. Towards exploring antecedents of both th herding effect and multiple-burst phenomena, we propose to further detect leading *DanMu*s within bursts, because those leading *DanMu*s make the most contribution to both patterns. A framework is proposed to detect leading *DanMu*s that effectively combines multiple factors contributing to leading *DanMu*s. Based on the identified characteristics of *DanMu*, finally we propose to predict the distribution of future *DanMu*s (i.e., the growth of *DanMu*s), which is important for many *DanMu*-enabled online video systems, for example, the predicted *DanMu* distribution could be an indicator of video popularity. This prediction task includes two aspects: One is to predict which videos future *DanMu*s will be posted for, and the other one is to predict which segments of a video future *DanMu*s will be posted on. We develop two sophisticated models

This research was partially supported by grants from the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61325010) and the National Natural Science Foundation of China (Grants No. U1605251 and 61672483). Qi Liu gratefully acknowledges the support of the Youth Innovation Promotion Association of CAS (No. 2014299). Yong Ge acknowledges the support of the Natural Science Foundation of China (NSFC, Grants No. 61602234 and No. 61572032).
Authors' addresses: M. He, Q. Liu, and E. Chen (corresponding author), Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China, Hefei, Anhui 230027, China; emails: mheustc@gmail.com, {qiliuql, cheneh}@ustc.edu.cn; Y. Ge (corresponding author), Eller College of Management, The University of Arizona, Tucson, Arizona 85721, America; email: yongge@email.arizona.edu; X. Wang, School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, Jiangsu, China; email: wangxuesongcumt@163.com.

https://doi.org/10.1145/3098885

to solve both problems. Finally, intensive experiments are conducted with a real-world dataset to validate all methods developed in this article.

CCS Concepts: • **Information systems** → **Incentive schemes**; *Data stream mining*; Web crawling; • **Theory of computation** → *Probabilistic computation*; • **Computing methodologies** → *Maximum likelihood modeling*;

Additional Key Words and Phrases: *DanMu*, herding effect, burst detection, leading *DanMus*

ACM Reference format:
Ming He, Yong Ge, Enhong Chen, Qi Liu, and Xuesong Wang. 2017. Exploring the Emerging Type of Comment for Online Videos: *DanMu. ACM Trans. Web* 12, 1, Article 1 (August 2017), 33 pages.
https://doi.org/10.1145/3098885

1 INTRODUCTION

Recent years have witnessed the rapid development of the online video platform. Examples include *Youtube.com*[1] and *Tudou.com*.[2] On these online video platforms, users have generated a large number of reviews for videos. These traditional reviews (TReviews) usually include textual comments and numerical ratings and are often displayed under videos. Recent literature has demonstrated that these reviews have a great impact on the popularity of videos, as users often check reviews before watching videos (Borghol et al. 2012; Cha et al. 2009a; Figueiredo et al. 2011).

Other than TReviews, one emerging type of user-generated comment named *DanMu* becomes available at many online video platforms. *DanMu* originated from the online video platform Niconico[3] in Japan, which was set up in 2006 and now has more than 48 million users. At present, the *DanMu* function has been adopted by most online video platforms in China, such as *Iqiyi*,[4] *Acfun*,[5] and *Bilibili*.[6] Differing from TReviews that are displayed in a separate space outside the video, *DanMu*s are scrolling marquee comments, which are overlaid directly on top of video and synchronized to specific playback times as shown in Figure 1. *DanMu* comments are displayed as streams of moving subtitles overlaid on the video screen. In *DanMu*-enabled online video platforms, viewers could write their own comments while watching the video. Each *DanMu* comment is associated with two time stamps, which are video time and natural time separately. The video time of a *DanMu* is the playback time around which the writer sends out the *DanMu*. Once a user finishes his or her *DanMu* comment for a video and sends it out, it will be synchronized to the associated video time and immediately displayed onto the video. All viewers (including the writer him- or herself) of the video will be able to see the *DanMu* when they watch around the associated video time. As a video is played, *DanMu*s associated with the current playback time usually fly into the screen from the right, and *DanMu*s associated with past playback times fly out of the screen from the left. This is why *DanMu* is also referred to as a bullet curtain.

A similar application as *DanMu* is the live media streaming of big events such as political debates and urban protests. Many online users may respond to such live media streaming via writing real-time messages (e.g., tweets on Twitter). Similarly to *DanMu*, users who are paying attention to the

[1] http://www.youtube.com/.
[2] http://www.tudou.com/.
[3] http://www.nicovideo.jp/.
[4] http://www.iqiyi.com/.
[5] http://www.acfun.tv/.
[6] http://www.bilibili.com/.

live media streaming could see both live streams and user's messages simultaneously. Such an application has drawn much attention in the literature in recent years. For instance, Jungherr (2015) and Trilling (2015) investigate the connection between live media coverage and Twitter activity; Ortiz et al. (2015) explores the characteristics of the conversation and political discussion on Twitter during the television presidential debate; Hawthorne et al. (2013) investigates whether there are significant differences between the conversation posted on Twitter by elite and non-elite users during a presidential debate. However, an important difference between such live media streaming and *DanMu* is that most user responses occur only within a short time period (i.e., during the live media streaming) and that few users will pay attention to the historical live media streaming after the events. On the contrary, more and more users may watch the same video and write their *DanMu*s over a long time period. This difference could lead to very different distributions and patterns between online *DanMu* and user's response to live media stream. In fact, most distributions and patterns shown in this article may not exist in user's responses to the live media streaming of events.

The simultaneously displayed *DanMu* comments could greatly strengthen viewers' watching experience of videos, as many *DanMu* comments convey interesting and useful information about the content of videos. In fact, the unique nature of *DanMu* has greatly promoted users' activities in online video platforms. For example, there are news[7, 8, 9] and reports[10] revealing that the *DanMu* increases the videos' popularity and that *DanMu* has increased user activity by 100 times in many online video platforms. Consequently, traffic volume has significantly increased in these *DanMu*-enabled online video platforms, which will eventually benefit these platforms in different ways such as by increasing the number of clicks on displayed ads.

Although we see the unique nature of *DanMu* and its great impact on online video systems, to the best of our knowledge, *DanMu* data are still under-explored in the literature. For instance, Lv et al. (2016) proposes a video understanding framework to assign temporal labels onto highlighted video shots; Wu et al. (2014) leverages the textual content of *DanMu* to extract time-sync video tags; Wu and Ito (2014) investigates the co-relation between the volume of one particular *DanMu* (i.e., 233, which is Chinese network jargon denoting "a lot of laughs") and video popularity measures such as the number of replays and bookmarks. However, none of them has explored the unique characteristics of *DanMu* and provided a comprehensive analysis of *DanMu*.

In this article, as a pilot study, we aim at discovering the unique characteristics and dynamics of *DanMu* from various perspectives. First, we analyze how *DanMu* comments distribute over natural time and video time and how the distributions differ from those of TReviews. Second, we identify and measure the herding effect of *DanMu* versus both natural time and video time, which reveals important insights about the growth of *DanMu*s on a video. Third, we discover the burst patterns from the temporal distributions of *DanMu*s that are different from those of TReviews. Fourth, to explore the antecedents of *DanMu* bursts and the herding effect, we propose to detect leading *DanMu*s within a burst that play as leading opinions and that make the most contribution to the occurrence of a *DanMu* burst. Finally, based on the identified characteristics of *DanMu*, we propose to predict the distribution of future *DanMu*s over both videos and video segments, which could provide important indications for understanding the trend of video popularity and network traffic. The prediction of the distribution over videos is to predict which videos future *DanMu*s will be posted for. The prediction of the distribution over video segments is to predict

[7] https://newmedia.cityu.edu.hk/com5101/updates/13a/techdetails.cfm?EID=yslin5-c.
[8] http://socialbrandwatch.com/the-explosion-of-danmu-in-chinese-social-media/.
[9] http://technode.com/2014/08/07/others-theater-can-see-comments-screen-real-time/.
[10] http://digi.163.com/14/0915/17/A66VE805001618JV.html.

which segments of a video future *DanMu*s will be posted on. We develop two sophisticated models to solve these two problems, respectively, that both take into account the unique characteristics of *DanMu*.

Towards our aims, we crawl a large amount of data from acfun.tv, which includes 6,506 videos, 1,704,930 *DanMu*s, 320,000 writers, and 155,455 traditional reviews. Detailed information about the videos, *DanMu*s, and TReviews is contained in this dataset: (1) the uploaded time of each video that ranges from January 1, 2014, to January 1, 2015; (2) the natural time, video time, writer ID, and text content of each *DanMu*; and (3) the natural time and writer ID of each traditional review. As videos are uploaded at different times, there will be different numbers of *DanMu*s and traditional reviews for different videos in this dataset. More details about the data will be introduced in Section 2.

By conducting the aforementioned comprehensive analysis of *DanMu* data, we believe this work could provide valuable insights for online video system operators and content providers, who can improve the effectiveness of many services, including caching, network traffic management, content recommendation, and ads display, by leveraging the growth patterns of *DanMu* in a video, the identified leading *DanMu*s and their writers, and the predicted distributions of future *DanMu*s over different videos and over different segments of one video. For instance, system operators may encourage leading *DanMu* writers to post more *DanMu*s and display relevant ads around videos or video segments that will receive more *DanMu*s.

The rest of this article is organized as follows. Section 2 describes the details of our dataset and some comparisons between *DanMu*s and traditional reviews. In Section 3, we introduce the herding effect and bursts of *DanMu*s and investigate the correlation between bursts. In Section 5, we propose one model to identify leading *DanMu*s. In Section 6, we propose two models for predicting the growth of *DanMu*s: one predicting the growth of *DanMu*s over videos and the other one predicting that over segmentations of a video. In Section 7, we draw the conclusions.

2 DATA AND STATISTICS

In this section, we first illustrate the *DanMu* comment and introduce the collected dataset and then present the distributions of *DanMu* versus video/reviewer and compare them with those of TReview. Finally, we demonstrate the temporal distributions of *DanMu* and TReview over both natural time and video time.

2.1 *DanMu* Illustration and Data Description

In Figure 1, we show two snapshots of a sample video[11] that include several Chinese *DanMu*s on top of the video. We pick up four *DanMu*s by different viewers, translate them to English, and show them at the bottom. The green axis at the bottom indicates the video time, where the selected four *DanMu*s are aligned based on the associated video time. The *DanMu*s “God, Norton” and “Edward Norton” by user A and user B are about the actor of the officer in the snapshot. We can observe that previous *DanMu*s have direct influence on future *DanMu*s. For instance, after users A and B mention the name of the actor in their *DanMu*s, user C mentions the movie “Fight Club” featuring the same actor, and then user D mentions another movie (i.e., “Red Dragon”) featuring the same actor. From this example, we can see that *DanMu* enables much more intensive communication among users than traditional review does.

As we introduced in Section 1, we have collected the data of 6,506 videos uploaded between January 1, 2014, and January 1, 2015, from *acfun.tv* that includes both *DanMu*s and traditional reviews. The *acfun.tv* actually includes multiple categories of videos, such as “Television,” “Movie,” “Animation,” “Variety Shows,” and so on. As the “Movie” category are more popular than other

[11] Available at http://www.acfun.tv/v/ac1731008.

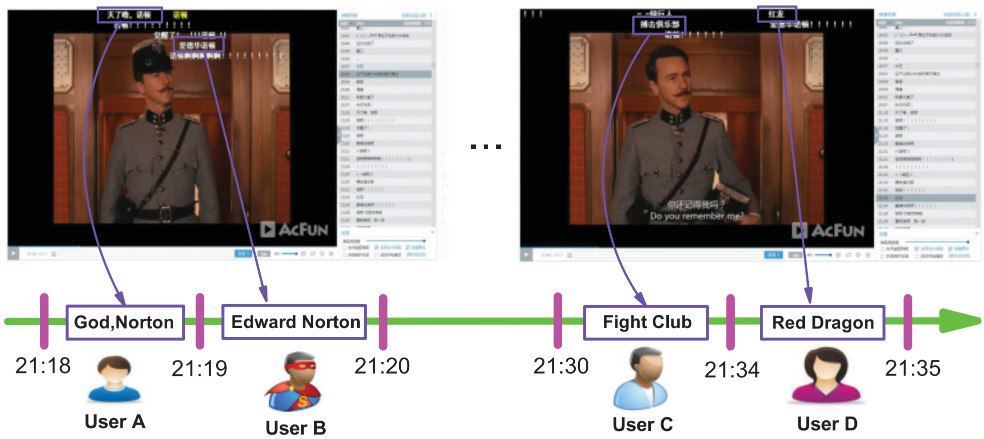


Fig. 1. Two snapshots of a *DanMu*-enabled video.

Table 1. A Summary of Our Data

# Videos	6,506
# *DanMus*	1,704,930
Avg # of *DanMus* per video	262.06
# *DanMu* writers	280,806
Avg # of *DanMus* per writer	6.07
# Traditional reviews	155,455
Avg # of Traditional reviews per video	23.89
# Traditional reviews writers	43,776
Avg # of Traditional reviews per writer	3.55

categories on this website, we focus our videos crawling on the "Movie" category. Furthermore, to avoid the bias on crawling "Movie" videos, we have crawled all new uploaded videos (6,506) under the category "Movie" during the observed period (January 1, 2014, to January 1, 2015). Table 1 shows a summary of statistics, where we observe much more user activities on *DanMu* than on TReview. For instance, the total number of *DanMus* is 10 times that of traditional reviews; the total number of *DanMu* writers is 6 times that of traditional reviews; and the average number of *DanMus* per writer is nearly 2 times that of traditional reviews. All these comparisons reflect that the *DanMu* function has attracted much more contributions from users and that online viewers prefer to write *DanMus* rather than traditional reviews for *DanMu*-enabled videos.

2.2 Volume Distributions

Let us first show the distributions of the number (i.e., histograms) of *DanMus* and traditional reviews per video. In Figure 2, we show the percentage of videos versus different numbers of *DanMus* and traditional reviews. As can be seen, while there are around 98.6% of the videos, of which the traditional review number is less than 200, there are only around 79% of the videos with less than 200 *DanMus*. This indicates that the volume of *DanMus* is much larger than that of traditional reviews. Furthermore, we zoom in and show more detailed volume distributions within [0,200) on the x-axis at the right-top of Figure 2. We find that over 59% of the videos have fewer than 10 traditional reviews and that over 77% have fewer than 20 ones. But for *DanMu*, there are only 25% videos with fewer than 10 *DanMus* and 37% videos with fewer than 20 *DanMus*. According

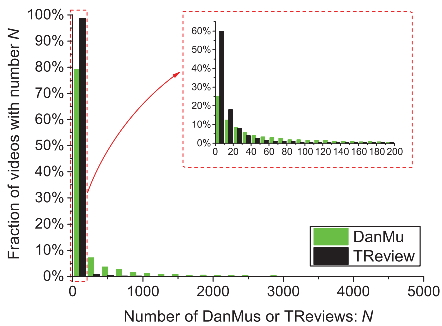


Fig. 2. Fraction of videos versus the number of *DanMus* and TReviews.

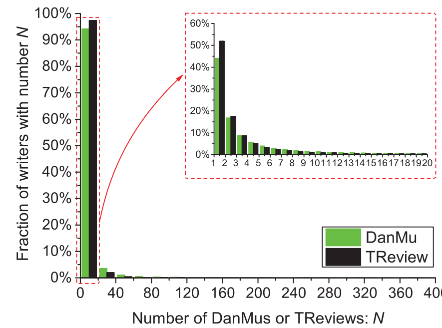


Fig. 3. Fraction of writers versus the number of *DanMus* and TReviews.

to the definition of power law in Clauset et al. (2009) and Li et al. (2016), the video proportion distributions on *DanMu* and TReview both follow a power law.

Also we show the histograms of *DanMu*s and traditional reviews per writer in Figure 3. These distributions more directly reflect the frequency of users' activities. As we can see, most writers of *DanMu*s and traditional reviews have written fewer than 20 *DanMu*s or traditional reviews. In particular, 97% writers have fewer than 20 traditional reviews. From the detailed distributions within [0,20) on the x-axis, we can see that 78% of users have fewer than 4 traditional reviews. But 69% of the writers have fewer than 4 *DanMu*s. It indicates that *DanMu* writers are relatively more active than traditional review writers. Besides, the total number of *DanMu* writers is 280,806, but the total number of traditional review writers is only 43,776, which also indicates that the *DanMu* function attracts more users to contribute than the traditional review does.

2.3 Temporal Distributions

Here we would like to demonstrate and compare the growth of *DanMu*s and traditional reviews over natural time. Specifically, in Figure 4, we show the temporal distributions of *DanMu*s and TReviews over natural time for four different videos. The x-axis denotes the number of days since the time when a video is uploaded, and the y-axis denotes the counts of *DanMu*s or TReviews. As we can see from all four distributions, there are usually a large number of both *DanMu*s and TReviews on the first day. This happens probably due to the fact that online video systems usually have different mechanisms to promote the newly uploaded videos such as being shown on the main page and being recommended to various users. Also, we can observe that the number of *DanMu*s is higher than that of TReviews almost every day after a video is uploaded. Furthermore, there are different growth patterns between *DanMu*s and TReviews. Specifically, most TReviews are generated by users during the first several days, and very few TReviews are added after this period. In contrast, the number of *DanMu*s may still dramatically increase after the early period, and the maximum daily growth of *DanMu*s may even happen many days later after a video is uploaded. For instance, the video 984507 gets the most daily growth of *DanMu*s on the 25th day after it is uploaded, and for the video 1005627 it is on the 250th day. In addition, we observe many more bursts from the distributions of *DanMu*s than those of TReviews. There is usually only one burst for TReviews, and the growth of TReviews will usually keep decreasing after the burst. On the contrary, we can see multiple bursts in all four distributions of *DanMu*s. Therefore, we conjecture that other than the mechanisms by online systems that drive the initial burst of *DanMu* (or TReview), there are some other factors that drive the significant growth of *DanMu*. In fact, as we show in Figure 1, some interesting *DanMu*s themselves may motivate online viewers

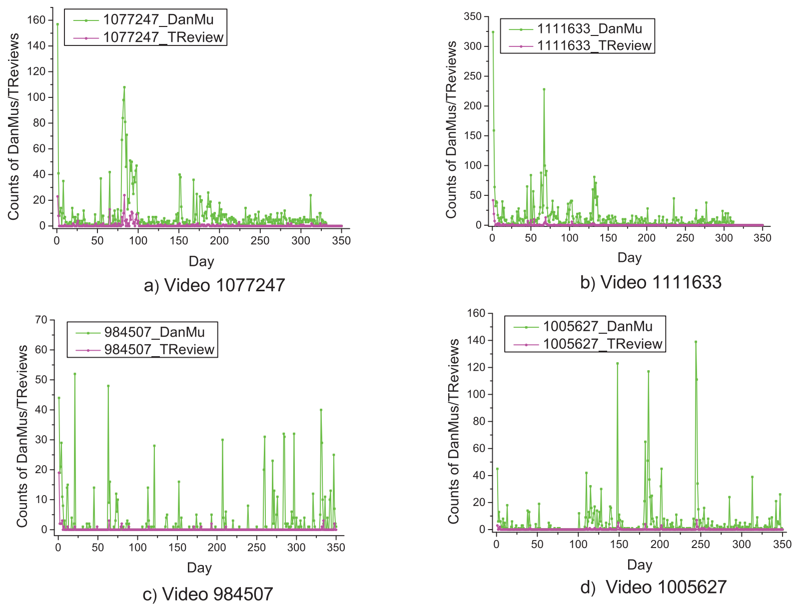


Fig. 4. Examples of temporal distribution.

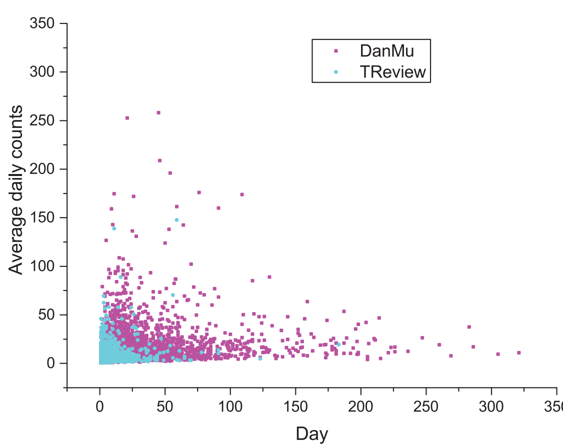


Fig. 5. A scatter distribution.

to generate more *DanMu*s that are related to previous ones. From this point of view, *DanMu* could be considered as a kind of "second-creation" that is also consumed by viewers together with video content.

To further examine the longer and higher growth of *DanMu*s, we define and calculate the number of growth days for *DanMu* and TReview, respectively. A growth day is a day where the video receives more than one *DanMu* or TReview. We count the number of growth days with respect to *DanMu*s or TReviews for each video and the average daily number of *DanMu*s or TReviews on all growth days for each video. We show the scatter plot in Figure 5, where each dot corresponds to one video; the x-axis denotes the number of growth days and the y-axis is the average daily number of *DanMu*s or TReviews. As can be seen, most points (purple) with respect to *DanMu*s are

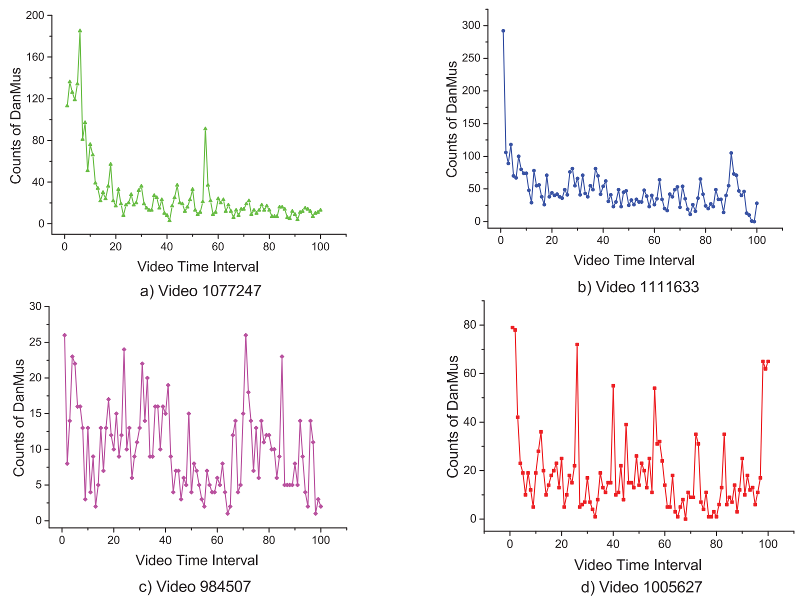


Fig. 6. Examples of temporal distribution.

spread on the top-right of those ones (blue) with respect to TReviews, which indicates that users generate more *DanMu*s daily over a longer period. In fact, the average number of growth days and the daily increased number of *DanMu*s are 15.56 and 9.88, respectively. But the average number of growth days and the daily increased number of TReviews are 3.93 and 4.44.

On the other hand, we also look into the distribution of *DanMu*s over video time, which is a unique feature of *DanMu*. In Figure 6, we show the distributions of *DanMu*s on video time for the same videos used in Figure 4. The x-axis is the video time that is equally discretized into 100 bins. The y-axis denotes the counts of *DanMu*s in each interval. For all videos, we observe a large number of *DanMu*s during the early video time, but the number may decrease later for some videos. One possible reason is that users watch videos from the beginning and abandon the watch later after realizing that the videos are not that attractive. We also find the distribution of *DanMu*s on video time may include more than one burst. The occurrence of a burst over video time is probably due to the video content around particular video times and potential herding effect.

As the user's comments may indicate the view of videos, we also examine the correlation between video's view count and *DanMu*s and TReviews, respectively. We adopt the Pearson correlation measure (Cohen et al. 2013) to calculate the correlation between video's daily view count and the daily volume of *DanMu* and TReview. Consequently, we get the Pearson correlation of (*DanMu* count, view count) and (TReviews count, view count) as 0.70 and 0.72, respectively, which demonstrates that both types of comments have a strong positive correlation with the video view.

3 THE HERDING EFFECT OF *DANMU*

Unlike TReviews, viewers could directly see *DanMu*s by others and easily write their *DanMu*s while watching, which makes it much more convenient for users to virtually interact with each other via *DanMu*s. Based on this unique feature of *DanMus*, we expect that there is a stronger herding effect with *DanMu*s than TReviews, as the viewers who want to write *DanMu*s can be easily affected by observed *DanMu*s. To this end, we adopt the definition of the herding effect in Banerjee (1992), where the herding effect is defined as *everyone doing what everyone else doing, even when their private information suggests they are doing something quite different* and examine this phenomena with quantitative methods and our dataset.

3.1 Herding Effect Model

In the literature, different successful methods have been proposed to the herding effect in different domains (Andersson et al. 2006; Hey and Morone 2004; Hsieh et al. 2008), such as social media and finance. For instance, researchers developed methods to quantify the herding effect in the social media domain (Wang et al. 2014; Wu et al. 2017). Christie and Huang (1995) test whether the equity returns indicate the presence of herd behavior on the part of investors during periods of market stress. In this article, we adopt the herding effect model in Christie and Huang (1995) to quantify the herding effect of *DanMu*, because we could naturally analogize the video set as a stock market and each video as a stock and #*DanMu*s as equity returns.

Specifically, we first calculate the herding effect of each video m at time interval t (i.e., video time interval or natural time interval) as

$$H_{mt} = abs(d_{mt} - d_t), \tag{1}$$

where d_{mt} is the number of *DanMu*s or TReviews of the video m during time interval t and d_t is the average number of *DanMu*s of all videos during time interval t. If H_{mt} is larger, then the herding effect is stronger. As H_{mt} reflects the variation between the video m and all videos, if H_{mt} is larger, then it indicates that more users write or do not write *DanMu* for video m by following other users who have done the same (i.e., herding effect as defined in Banerjee (1992)). Thus large H_{mt} indicates a strong herding effect. If $H_{mt} = 0$, then it indicates that there is not variation, that is, there is no herding effect.

Then we obtain the herding effect for each video m by averaging H_{mt}:

$$H_m = \sum_{t=1}^{T} \frac{H_{mt}}{T} = \sum_{t=1}^{T} \frac{abs(d_{mt} - d_t)}{T}, \tag{2}$$

where T is the counts of time intervals.

Finally, the herding effect over the whole video set H is the average of the herding effect for each video H_m,

$$H = \sum_{m=1}^{M} \frac{H_m}{M} = \sum_{m=1}^{M} \frac{\sum_{t=1}^{T} \frac{abs(d_{mt} - d_t)}{T}}{M}, \tag{3}$$

where M is the number of videos. Also, we use the following equation to quantify the herding effect over all videos within the time interval t, which helps us to learn the dynamic trend of the overall herding effect,

$$HT_t = \sum_{m=1}^{M_t} \frac{H_{mt}}{M_t} = \sum_{m=1}^{M_t} \frac{abs(d_{mt} - d_t)}{M_t}, \tag{4}$$

where M_t is the number of videos during the time period t.

3.2 Results of the *DanMu* Herding Effect

*DanMu*s have two time dimensions: One is natural time and the other one is video time. Thus we show the results of the herding effect along both time dimensions with our dataset.

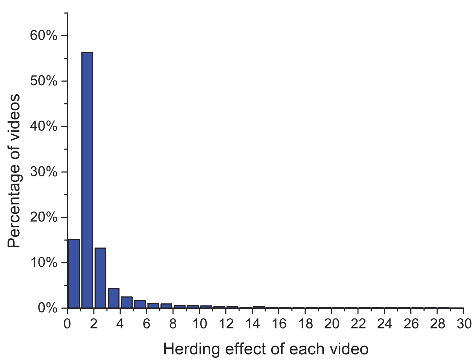


Fig. 7. Histogram of the herding effect of H_m.

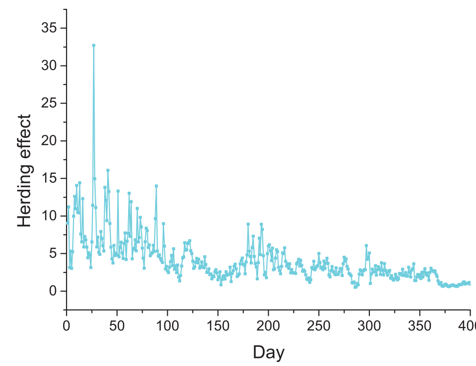


Fig. 8. Distribution of the herding effect of HT_t.

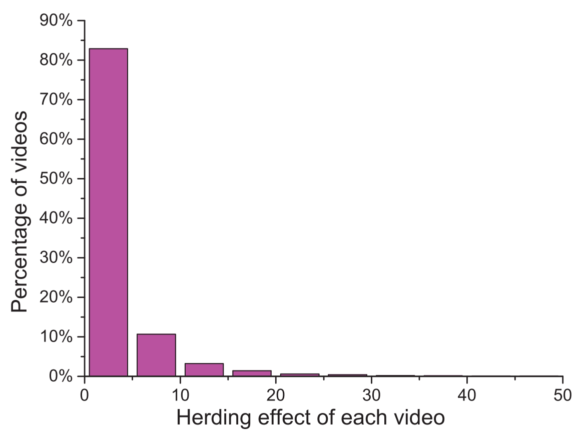


Fig. 9. Histogram of the herding effect H_m.

3.2.1 Results Along Natural Time. On average, the herding effect of TReviews over all videos is only 0.67, but the herding effect of *DanMu*s over all videos is 2.93. We also calculate the herding effect of each video and the herding effect over all videos each day and show both distributions in Figure 7 and Figure 8. As can be seen from Figure 7, more than 85% of the videos have a herding effect larger than 1. The herding effect is larger in the first 90 days and is more fluctuant than over the succeeding days, as shown in Figure 8. One possible reason is that, as time goes on, the percentage of old videos becomes larger and these old videos decrease the herding effect.

3.2.2 Results Along Video Time. We first equally split the video time by 1 minute for each video. As different videos have different durations, the number of time intervals T is different among videos. So we only calculate the herding effect on video H_m and the herding effect over all videos H. The herding effect overall all videos is 2.98, which is slightly larger than the herding effect (i.e., 2.93) along natural time. Figure 9 shows the histogram of H_m, where we can see that the herding effect of almost 64% of the videos is within [0,2). If we compare with Figure 7, then we can see a higher herding effect along video time than along natural time.

In addition, we examine whether the herding effect along video time and that along natural time have some correlation. Specifically, we show the scatter plot of H_m with all videos in Figure 10, where the x-axis is the herding effect H_m along video time and the y-axis is H_m along natural time.

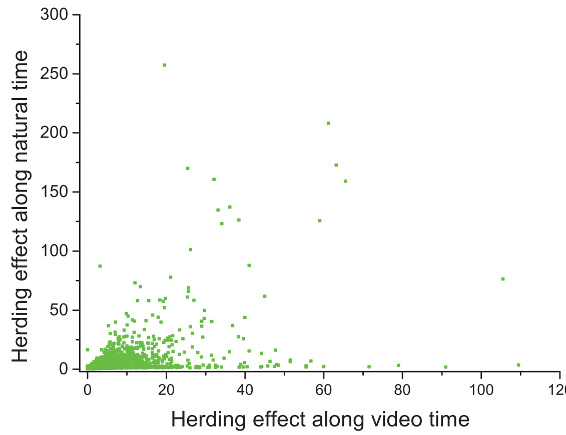


Fig. 10. A scatter plot of herding effect.

The Pearson correlation is 0.48. These results indicate that the herding effect on video time and that on natural time have a positive correlation.

4 THE BURST OF *DANMU*

As shown in Section 2.3, there are often multiple bursts of *DanMu* along both video time and natural time. For the same video, there should be intrinsic relationship between the distributions of *DanMu* over video time and that over natural time as both distributions are formed with the same set of *DanMu*s. In this section, we first introduce an algorithm to detect bursts and then examine the relationship between both distributions.

4.1 Detection Algorithm

In the literature, many existing works address the burst detection of time series (Palshikar et al. 2009; Sboner et al. 2007; Vlachos et al. 2004; Zhu and Shasha 2003), and many of these techniques could be applied here. Considering efficiency and effectiveness, we simply adopt the burst detection method proposed in Vlachos et al. (2004) to discovery the bursts of *DanMu*s. This approach (denoted as VTBM) is based on the computation of moving average (MA) and consists of the following three steps:

(1) Calculate Moving Average MA_{wi} of length w for element i within sequence $t = (t_1, \ldots, t_i, \ldots, t_n)$; if $i < w$, then we set $MA_{wi} = t_i$, else the $MA_{wi} = \sum_{i-w+1}^{i} t_i/w$;
(2) Set $cutoff = mean(MA_{wi}) + x * std(MA_{wi})$;
(3) $Bursts = \{t_i | MA_{wi} > cutoff\}$.

As suggested in Vlachos et al. (2004), we set $x = 1.5$ and $w = 3$ for our data. Since the *DanMu*s has two time dimensions, we detect the bursts of *DanMu* distributions over both natural time and video time, respectively.

4.2 Results of Burst Detection

4.2.1 Results over Natural Time. First, let us show detected bursts of two videos in Figure 11, where the red parts are detected bursts. To validate the effectiveness of VTBM, we also adopt another method in Kleinberg (2003) to detect bursts, which is denoted as KSBM. We compare the results of both methods and show them in Figure 11. As can be seen, while these two methods

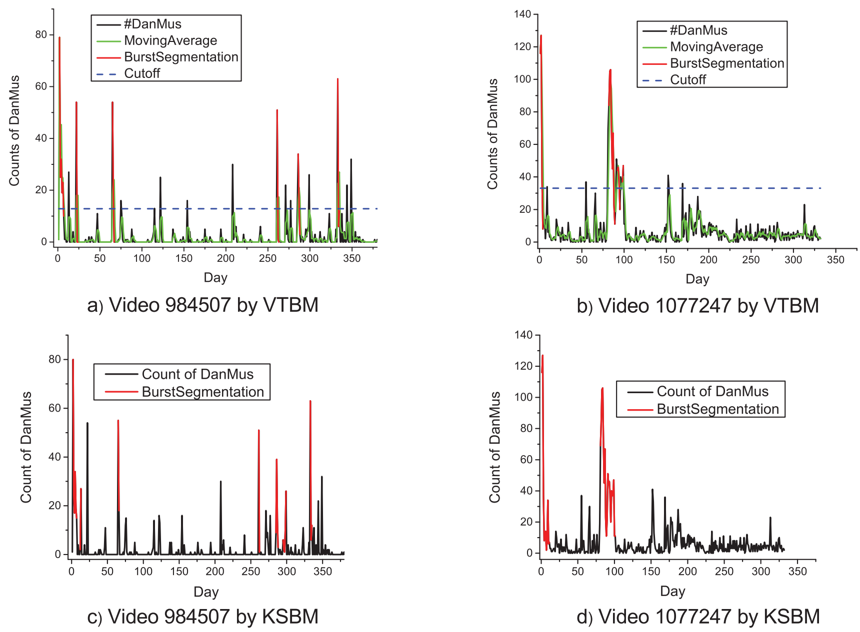


Fig. 11. Detected bursts of two videos.

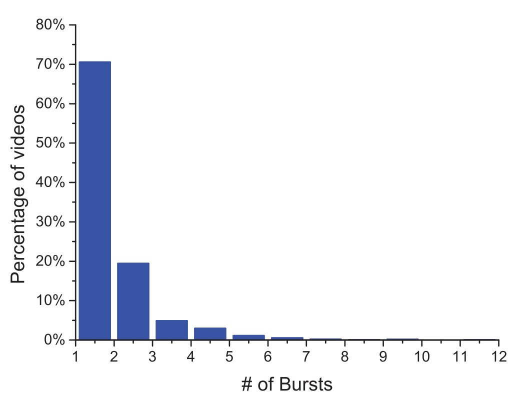


Fig. 12. Histogram of burst number per video.

have similar identified bursts, VTBM produces more robust results. With the VTBM method, we can get reasonable bursts based on a good cutoff line.

After applying the detection method to all data, we get the average number of bursts as 1.49. Also, we show the histogram of the number of bursts with all data in Figure 12. As we can see, 70% of the videos have only one burst. This is reasonable, because only a small portion of online videos draw most attention from users. In fact, the number of bursts is not greater than two for more than 80% of the videos. But there are still 30% of the videos with two or more bursts that lead us to explore more about *DanMu* bursts. Based on the definition of power law in Clauset et al. (2009) and Li et al. (2015), we found that the percentage of videos on bursts follows a power-law distribution.

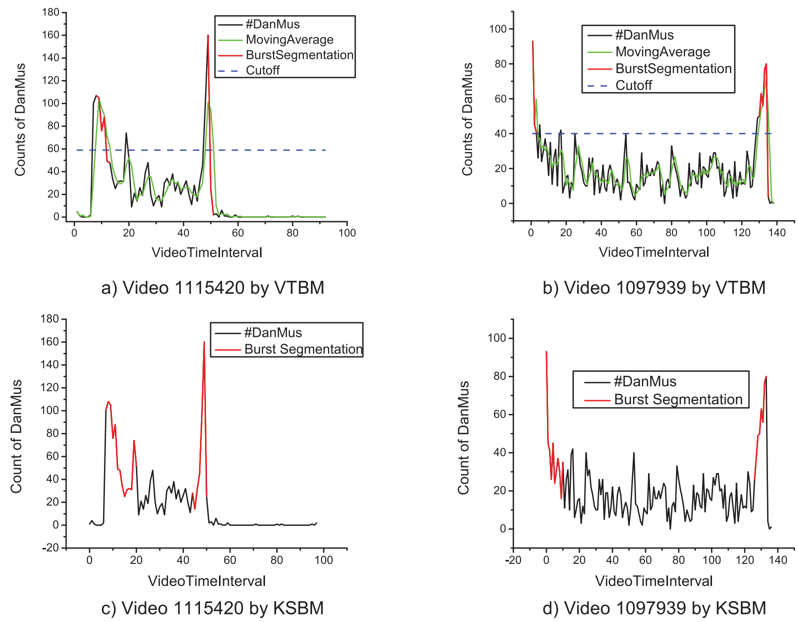


Fig. 13. Detected bursts of two videos.

As the distribution of *DanMu*s over natural time exhibits a multi-burst characteristic, it will be interesting to explore if the old or new *DanMu*s contribute to the multiple bursts. To achieve this goal, we first obtain the time difference d_{ij}(in days) between the date when the first *DanMu* of an ith burst is posted and the date when the corresponding video j is uploaded. For instance, if we obtain $d_{ij} = 10$ for an ith burst of video j, it means that the first *DanMu* is posted 10 days later after the video is uploaded and that all *DanMu*s included in the ith burst are posted 10 days later as well. After we obtain d_{ij} for all bursts of all videos, we find that the mean of d_{ij} for the second burst (i.e., d_{2j}) over all videos is close to 50, and the mean of d_{ij} for the third burst (i.e., d_{3j}) is greater than 100. These results demonstrate that *DanMu*s included in the bursts that occur later tend to be new *DanMu*s. Consequently, we can draw a rough conclusion that multiple bursts are contributed by new *DanMu*s.

4.2.2 Results over Video Time. Next we turn to investigate the distribution of the number of bursts over video time. Similarly, we first show the detection results of two videos in Figure 13, which demonstrates that the burst detection method also works well for detecting bursts over video time. The average number of bursts over video time per video is 2.56, which is larger than over natural time. This actually is consistent with the strong herding behaviors by viewers; that is, users tend to follow existing *DanMu*s while watching. Again, we show the histogram of the number of bursts with all data over video time in Figure 14. As can be seen from Figure 14, about 73% of the videos have more than one burst. Compared with results over natural time, there are much more bursts over video time than over natural time.

4.3 The Correlation of Bursts

As it is the same set of *DanMu*s that form distributions along two time dimensions, we would like to examine the intrinsic correlation between both distributions. First, let us show an example with the *DanMu*s of one video. For this video (with an ID of 1114190), we detect two bursts along

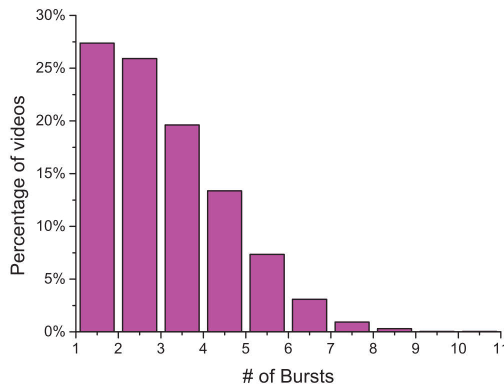


Fig. 14. Histogram of burst number per video.

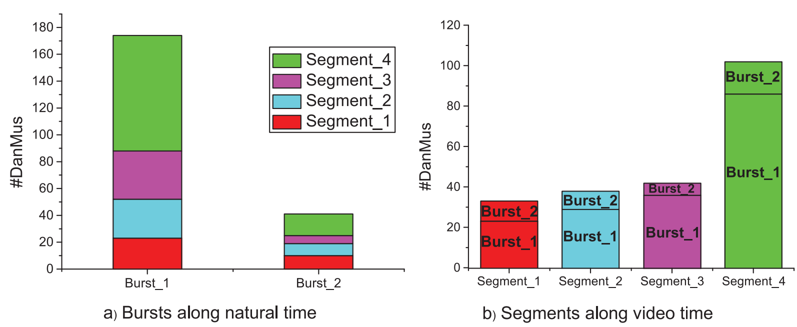


Fig. 15. An illustration example.

natural time. We partition the video time into four successive segments for each video in our data. But it is possible to choose different numbers of segments for other datasets. In Figures 15(a) and (b), we use a bar chart to represent each burst along natural time and each segment along video time ((a) for natural time and (b) for video time). As can be seen, the *DanMus* within each burst along natural time consist of *DanMus* from all four segments along video time. On the other hand, the *DanMus* within each segment along video time contain *DanMus* from two bursts along natural time. Furthermore, we can see that both bursts along natural time have a similar percentage of *DanMus* from different segments along video time. In other words, *DanMus* in both bursts spread over four segments in a similar way. In fact, it is very likely that *DanMus* in the second burst that comment on a segment (e.g., the red bar in right panel) of the video are motivated by *DanMus* in the first burst that comment on the same segment (e.g., the red bar in right panel) of the video. Therefore, we infer that there should be some correlation between both bursts along natural time.

To further generally examine the possible correlation among all bursts in our data, we propose to represent each burst along natural time with a vector, each attribution of the vector is the number of *DanMus* within a segment along video time, and then compute Pearson correlation between each pair of bursts based on their vectors. Specifically, we conduct the following steps:

(1) Partition *DanMus* over video time based on the burst detection method in Section 4.1 for video m as $S_m = \{s_{m1}, \ldots, s_{mj}, \ldots, s_{m|S_m|}\}$, where each segment is either the segment of detected burst or the one before/after a burst and $|S_m|$ is the number of of segmentations for video m.

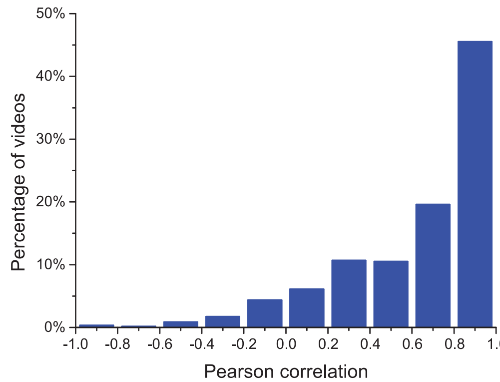


Fig. 16. Histogram of the Pearson correlation.

(2) Obtain bursts $B_m = \{b_{m1}, \ldots, b_{mi}, \ldots, b_{mm_b}\}$ over natural time for video m based on the same burst detection model in Section 4.1. m_b represents the number of bursts for video m.
(3) Generate a vector for each burst b_{mi} $(1 \leq i \leq m_b)$ as $\boldsymbol{v}_{mi} = (v_{mi1}, v_{mi2}, \ldots, v_{mis_m})$, where v_{mij} is the number of *DanMu*s that are within burst b_{mi} along natural time and belong to the segmentation j along video time.
(4) Compute Pearson correlation between each pair of two bursts b_{mi} and $b_{mi'}$ based on their vectors v_{mi} and $v_{mi'}$ for video m.

The average Pearson correlation of bursts with all data is 0.63. Also, Figure 16 shows the histogram of the correlation. As can be seen, only 7.5% videos have negative correlation, and more than 75% videos have Pearson correlation over 0.5. These results strongly indicate that bursts of a single video share very similar composition of *DanMu*s within different segments along video time. In other words, there are significant linear correlation among bursts per video, and it is very likely that the *DanMu*s in one burst are motivated by those in previous bursts along natural time.

5 LEADING *DANMU* IDENTIFICATION

What causes the strong herding effect and multiple-burst phenomenon, particularly with video time? By carefully examining our real-world data, we find there exist some *DanMu*s that often act as leaders and stimulate subsequent *DanMu*s. We call this kind of *DanMu* a *leading DanMu*. For instance, the *DanMu* "God, Norton" shown in Figure 1 is likely a leading *DanMu*, as it stimulates the following three *DanMu*s by users B, C, and D. We conjecture that such leading *DanMu*s make great contributions to both the herding effect and the occurrence of bursts. Detecting those leading *DanMu*s is very important, as it will not only help us understand the phenomenon of herding effect and burst but also help system operators to improve the effectiveness of services. For example, with the identified leading *DanMu*s, the website operators can know the leading writers who provide leading *DanMu*s and encourage them to write more *DanMu*s, which may further attract other users to write more *DanMu*s and watch videos. In addition, we can understand users' preference via these leading *DanMu*s to help people make and upload more attractive videos in the future, which is similar to Zhao et al. (2016) learning user preference for video recommendation. To this end, in this section, after introducing three types of factors including *DanMu* content, user information, and time information of *DanMu*, we will propose a framework that takes into account those three types of factors for detecting leading *DanMu*s and then present the experimental results.

5.1 Three Types of Factors

5.1.1 DanMu *Content.* As the content of *DanMu* can be viewed directly by viewers, if one *DanMu* is interesting and informative, then it may attract viewers to follow it by adding subsequent *DanMu*s that are related to the *DanMu*. First, if the content of one *DanMu i* is similar to the subsequent *DanMu*s in terms of semantics, then the *DanMu i* is more likely to be a leading *DanMu* as its subsequent ones are possibly following the *DanMu i*. On the other hand, if the content of one *DanMu i* is similar to the antecedent *DanMu*s (with both natural time and video time), then it is less likely to be a leading one. In addition, for one *DanMu i*, if there are more subsequent *DanMu*s and fewer antecedent ones, it is very likely to be a leading one, and vice versa. Based on these intuitions, we define the following three features based on content for identifying leading *DanMu*s.

(1) Content similarity (f_1). Within each burst segmentation s of video m, we first compute the content similarity between *DanMu i* and each of subsequent *DanMu*s as

$$cs_{ij} = \frac{|w_i \cap w_j|}{|w_j|}, \qquad s.t.\{vt_j \geq vt_i, nt_j \geq nt_i\}, \tag{5}$$

where j denotes one of subsequent *DanMu*s. w_i and w_j denote the set of words in *DanMu i* and *DanMu j*. $|w|$ denotes the count of words, vt and nt represent the associated video time and natural time for one *DanMu*. Then we sum up all similarity between *DanMu i* and each of its subsequent ones as

$$cs_i = \sum_{j=1}^{n_i} cs_{ij}, \tag{6}$$

where n_i is the number of subsequent *DanMu*s of *DanMu i*. Finally, we consider the content similarity defined in Equation (6) as a feature for leading *DanMu* detection.

(2) Content novelty (f_2). For one *DanMu i* within a burst, we first compute the similarity between this one and each of its antecedent *DanMu*s with Equation (5), and then we measure the novelty of this *DanMu* as

$$cn_i = 1 - Max(s_{ik}), \tag{7}$$

where k is the index of antecedent *DanMu*s of *DanMu i*.

(3) *DanMu* position (f_3). For one *DanMu i* within a burst, we take the following ratio as a feature for leading *DanMu* detection:

$$dp_i = \frac{sdn_i}{adn_i + 1}, \tag{8}$$

where sdn_i is the number of subsequent *DanMu*s and adn_i is the number of antecedent *DanMu*s. The bigger dp_i is, the more likely *DanMu i* is a leading one.

5.1.2 *User Information.* Different users have different experiences with *DanMu*s. While some of them are good at writing interesting and informative *DanMu*s that will encourage others, some of them are not. Thus, we look into user information for leading *DanMu* identification. First, if one user has written a lot of *DanMu*s, then it is more likely that he or she will contribute leading *DanMu*s in the future. Also, if one user's *DanMu*s frequently appear in bursts with video time, it is more likely that he or she is a writer of a leading *DanMu*. Based on these intuitions, we define the following features based on user information to distinguish leading *DanMu*s from others:

(1) Total number of *DanMu*s by user u (f_4). We denote this feature as tdn_u. If the value of tdn_u is larger, then it indicates that the user is more active with writing *DanMu*s.

(2) Total number of *DanMu*s for video m by user u (f_5). We denote this feature as tdn_{mu}, which reflects the interest of the user u on video m. A larger tdn_{mu} indicates that the author is more interested in and knowledgeable about video m.
(3) The ratio of *DanMu*s within bursts (f_6). Basically for each user u, we count the number of his *DanMu*s and the number of his *DanMu*s within different bursts. Then we define the following ratio as a feature

$$br_u = \frac{tbdn_u}{tdn_u}, \tag{9}$$

where tdn_u is the total number of *DanMu*s by user u and $tbdn_u$ is the total number of *DanMu*s within bursts by user u.

5.1.3 Time Information. *DanMu* as a new type of comment has two time dimensions: One is natural time and the other is video time. Both of them are of great importance for *DanMu* leading identification. As one *DanMu* appears earlier with both natural time and video time within one burst, it should have a higher possibility to be a leading one than others that appear later with both natural time and video time within the same burst. Along this line, we define the following two features based on time information.

(1) Natural time (f_7). If the natural time nt_i of *DanMu* i is close to the upload time ut_m of video m, then the writer of *DanMu* i probably knows the video earlier than others and has more information about video m. Also, this *DanMu* will have a longer time to be shown to other viewers. Therefore, the difference $ntD = nt_i - ut_m$ is negatively correlated with the leading possibility. Since we want to get a positively correlated feature, we first normalize ntD based on Equation (10) and then define f_7 as $f_7 = 1 - ntD'$, where ntD' denotes the normalized value of ntD.
(2) Video Time (f_8). If the video time vt_i of *DanMu* i is close to the starting time st_{ms} of a burst s, then this *DanMu* will be seen earlier by users than other *DanMu*s and thus has a higher possibility to be a leading one. Similarly, we first take the different $vtD = vt_i - st_{ms}$ and then transfer it to $f_8 = 1 - vtD'$, where vtD' is the normalized value of vtD based on Equation (10).

5.2 A Framework

In the literature, many works have been proposed to identify leading opinions or opinion leaders. For instance, Chan and Misra (1990) empirically examined the role of a personality trait and public individuation for understanding opinion leadership. Matsumura et al. (2002) aimed at mining and characterizing opinion leaders from threaded online discussions on the Internet. Song et al. (2007) proposed a novel algorithm called InfluenceRank to identify opinion leaders in the blogosphere. Yu et al. (2010) used sentiment analysis methods and the weight of links between users for discovering the authority value of user. Moon et al. (2014) proposed an advanced method for measuring the leader in online course discussion based on people's choice of words given a semantic topic of interest. However, most of them could not be directly applied for leading *DanMu* identification, because we are lack of relevant information such as link information among users and the explicit thread between messages.

To this end, we adopt the hot topic identification model proposed in Li and Du (2011), which introduced the weight for individual feature into the conventional Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) method proposed in Zanakis et al. (1998). The basic idea of the TOPSIS method is to measure the Euclidean distance between measurements and the ideal solution (positive or negative) and consider the one blog that is close to the positively ideal solution, which is made up of the best value for each feature, and far from the negatively ideal solution, which is made up of the worst value for each feature as optimal. To adopt this method, we first use sigmoid curve transformation to normalize the features f_1, f_3, f_4, and f_5 into the range $[0, 1]$. The sigmoid curve transformation is defined as

$$c_i' = 1 \Big/ \left(1 + e^{-\frac{c_i - \overline{c}}{\overline{c}}}\right), \tag{10}$$

where c_i is the counts of the feature i, $\overline{c}$ is the average counts of c_i.

Based on Section 5.1, all designed features are positively correlated with the leading probability. Thus we could have a positively ideal solution for each feature i as $Z_i^+ = max(F_j^i)$, $(i = 1, \ldots, n)$, where j is the index of *DanMu* and n is 8 as we have 8 features. F_j^i represents the value of feature i of *DanMu* j. Also, we could have a negatively ideal solution for each feature i as $Z_i^- = min(F_j^i)$, $(i = 1, \ldots, n)$. For each *DanMu* j, we compute the distance between this one and the negatively ideal solution as $s_j^- = \sqrt{\sum_{i=1}^{n} (Z_i^- - F_j^i)^2}$. Similarly, we get the distance between this one and positive solution as $s_j^+ = \sqrt{\sum_{i=1}^{n} (Z_i^+ - F_j^i)^2}$. Finally, we define a score to measure the leading possibility of *DanMu* j as

$$c_j = s_j^- / (s_j^+ + s_j^-). \tag{11}$$

Furthermore, as suggested in Li and Du (2011), we could also apply a weight for c_j to overcome the scale difference between different features. Specifically, we define a new score as follows:

$$c_j' = \left(1 - \frac{SF_j}{\overline{F}_j}\right) * [s_j^- / (s_j^+ + s_j^-)], \tag{12}$$

where $\overline{F}_j$ is the average of n features $\overline{F}_j = (\sum_{i=1}^{n} F_j^i)/n$ and SF_j is the standard deviations of all features $SF_j = (\sqrt{\sum_{i=1}^{n} (\overline{F}_j - F_j^i)^2})/(n-1)$. As c_j is defined based on Content, User and Time information of *DanMu* and TOPSIS, we simply denote this method as CUTT. Similarly we denote the method based on c_j' as CUTW. For both scores, a higher value indicates a larger leading possibility.

5.3 Experimental Results

In this section, we first present the results of leading *DanMu* identification with different methods and then evaluate the power of different combinations of features for leading *DanMu* identification.

5.3.1 Identification Performance. As we have no ground truth of leading *DanMu*s, we conduct a user study to manually label a set of leading *DanMu*s which are then considered as ground truth. Specifically, we select 10 experienced users who like to view *DanMu*-enabled videos and write *DanMu*s to label leading *DanMu*s from 46 randomly selected bursts. These 46 selected bursts belong to 42 different videos and each participant independently labels all 46 bursts. For the sake of accurately labeling the leading *DanMu*s, we first ask the 10 participants to watch each burst's corresponding video segment including the *DanMu*s. Statistics show that the total time of 46 corresponding video segments is 107.8 minutes. Note that for each burst, we ask these participants to watch the corresponding video segment instead of the whole corresponding video, as *DanMu*s within a burst are strongly correlated to the corresponding video segment, not the video's other parts. Then we show them more detailed information of the *DanMu*s in a burst as shown in Table 2. Based on the provided information, they label those *DanMus* that they think affect the occurrence of other *DanMus* with high likelihood as leading ones. Finally, we combine the labeled leading *DanMus* by all participants and pick those where at least 50% (5/10) of users label them as leading. These chosed leading *DanMus* are finally considered as our ground-truth data.

In our experiments, we first randomly select 46 bursts that have the average number of *DanMu*s per burst as 34.85 and then ask users to label leading *DanMu*s from each one. In total, we get

Table 2. Detailed Information of *DanMu*s within a Burst

Movie name	Burst ID	Movie time (Seconds)	Natural time	Content
The Grand Budapest Hotel	1	1273	2016-01-12 22:08	Norton is coming!
The Grand Budapest Hotel	1	1276	2016-05-27 21:24	My Norton!
The Grand Budapest Hotel	1	1278	2016-02-27 23:20	So cool
The Grand Budapest Hotel	1	1279	2016-01-25 16:51	I love Norton!!!

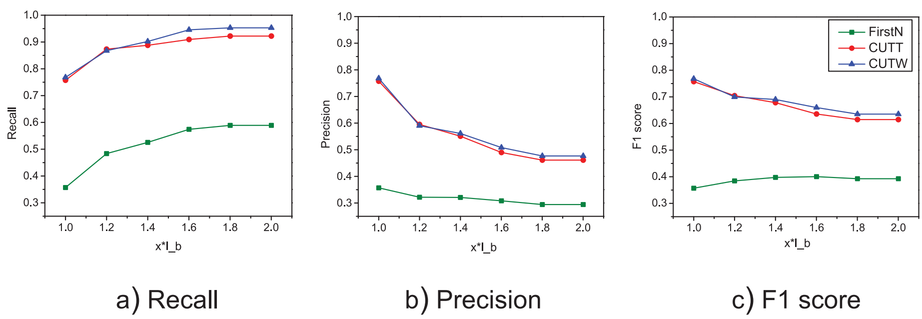


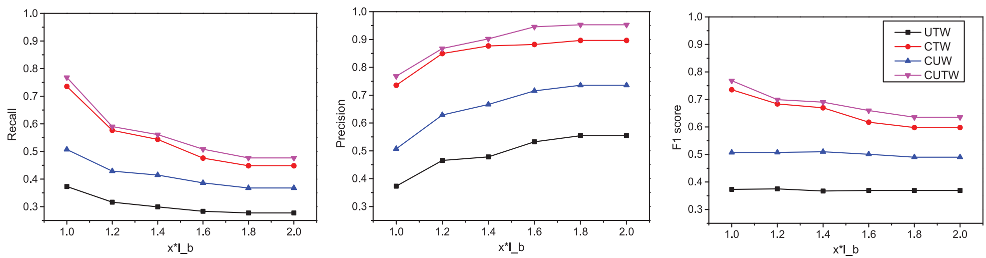
Fig. 17. Experimental results with different sizes $x * l_b$.

110 leading *DanMu*s from these bursts. Note that different bursts may have a different number of leading *DanMu*s. For each burst, we will rank all *DanMu*s based on the score c_j or c'_j. Suppose the number of manually labelled leading *DanMu*s is l_b, we will return top $x * l_b$ (like the notion *topK* (Liu et al. 2011)) *DanMu*s from the ranked list of *DanMu*s, where $x \in [1.0, 1.2, 1.4, 1.6, 1.8, 2.0]$ is the magnification of l_b. Then we compute three validation metrics, including recall (Liu et al. 2012), precision, and F_1, to evaluate the performance of ranking. For the purpose of comparison, we also include a baseline method that is simply selecting top $x * l_b$ *DanMu*s based on the associated video time of *DanMu*. For simplicity, we denote this method as FirstN.

As shown in Figure 17, the CUTW model is the best to identify the leading *DanMu*s in terms of all metrics. Overall, the results show that both CUTT and CUTW can effectively capture the *DanMu*s that lead bursts and validate the effectiveness of *DanMu* content and user and time information for leading *DanMu* detection. In particular, when we choose the top $1.6 * l_b$ *DanMu*s, the CUTW model almost returns all leading *DanMu*s, as the average recall is 95%. Compared to CUTT, we find incorporating the weight of features could lead to a slight better result. On the contrary, the FirstN method only leads to the average recall as 0.52, which suggests that *DanMu*s appearing earlier with video time of a burst are not always the leading one.

5.3.2 Impact of Different Features. With the results in the "Identification Performance," we learn that the CUTW model can capture the leading *DanMu*s. Now we want to examine the power of different combinations of features for leading *DanMu* detection. Specifically, we use two types of features and compute the score c'_j based on them and then get the ranked list of *DanMu*. As we have three types of features, we get three different combinations that are denoted as UTW, CTW, and CUW, respectively. The UTW indicates that all content features (i.e., f_1, f_2, and f_3) are removed. CTW and CUW mean that user information features and time information features are removed, respectively.

Finally, we get a group of comparison shown in Figure 18. As can be seen, UTW leads to the worst performance. This indicates that the content-based features are important for identifying leading *DanMu*s. As the result of CTW is close to that of CUTW, the features based on user information are



a) Recall b) Precision c) F1 score

Fig. 18. Experimental results with different sizes $x * l_b$ and features.

not that important. One possible explanation for this is that all features based on user information only indirectly reflect user's experience in writing *DanMu*, which may be missing in some cases. Besides, the features based on time information are also important as the performance of CUW is much worse than CUTW. Overall, the results in Figure 18 generally suggest that we should explore more in content and time of *DanMu* for identifying leading *DanMu*s for real practice.

6 PREDICTING THE GROWTH OF *DANMUS*

In this section, we introduce how to utilize the identified characteristics and patterns of *DanMu* for predicting the growth of *DanMu*. We first introduce our dynamic herding effect model, denoted as the DHEModel, and the experimental results. Then we present the TVHAD model and results on the growth over video segments. Both predictions for *DanMu* growth will provide important insights for online video systems, for example, the growth of *DanMu* can be an indicator of a video's popularity, because the view count and the *DanMu* count have a strong positive correlation, as demonstrated in Section 2.

6.1 Dynamic Growth over Videos

6.1.1 Method for Predicting the Growth of DanMus *over Videos.* There are some existing works for predicting the popularity growth of videos. For instance, Borghol et al. (2012) develop a model that could integrate the impacts of various content-agnostic factors for video popularity. In Cha et al. (2009b) and Szabo and Huberman (2010), the herding effect (Barabási and Albert 1999) is explored for predicting video popularity. Szabo and Huberman (2010) utilize the observation that the total views received after a video is uploaded provide a strong sign of its future views for building the prediction model. While these methods have been used for video popularity prediction, we cannot apply them for the prediction of *DanMu* growth due to the unique characteristics of *DanMu*, such as the herding effect over both natural and video time. In contrast, in Welch (2000), a herding effect model is proposed to study the influence of the prevailing consensus on current analysts' recommendations (strong buy, buy, hold, sell, and strong sell) for a stock. By carefully examining the herding effect in stock data and *DanMu* data, we find a very close analogy between them: A user chooses a video to write *DanMu*s, which is also affected by the prevailing consensus (i.e., popular videos). Therefore, we decide to explore the model in Welch (2000) to predict the dynamic growth of *DanMu*s over videos. However, there are still three major differences between the setting in Welch (2000) and ours: The first one is that the model of Welch (2000) computes the transition probability between different items (i.e., stock recommendations, such as hold, strong buy, buy and so on), while our problem is to compute the proportion of *DanMu*s for different videos. the second difference is that the items are the same over time in Welch (2000), while new videos may be added in our problem setting. The last one is that the herding effect parameters in our setting change over time, which could be updated by daily data. Once we learn the herding effect based on daily data, we can capture the herding effect of the prevailing videos on other videos every day. Along this line, we adjust the model in Welch (2000) to Equation (13) to accommodate these differences. As we want to investigate the herding effect caused by *DanMu*, the popular videos are measured by the number of *DanMus*, not the number of views. Specifically, we select the top_k videos as popular videos ranked by the number of each video's *DanMu*s. And the value of k may vary among different applications,

$$\widetilde{p}_{mt} = \frac{p_{m(t-1)}[1 + dis(m, o_{t-1})]^{-\theta_{t-1}}}{D_t}, \tag{13}$$

$$D_t = \sum_{m'=1}^{M_{t-1}} p_{m'(t-1)}[1 + dis(m', o_{t-1})]^{-\theta_{t-1}},$$

where $\widetilde{p}_{mt}$ is the predicted proportion of *DanMu*s for video m in day t and $dis(m, o_{t-1})$ is the distance between video m and popular videos based on the number of *DanMu*s in day $t-1$. D_t ensures that the sum of all videos' proportion is equal to 1. o is the centroid of popular videos representing the prevailing consensus on day $t-1$, and M_{t-1} represents the number of videos on day $t-1$. The proportion of *DanMu*s for the video m on day $t-1$ is equal to $d_{m(t-1)}/d_{t-1}$ and p_{mt} is d_{mt}/d_t, where d_{mt} is the number of *DanMu*s for the video m until day t and d_t is the number of *DanMu*s of all videos until day t.

Comparing to the original model of Welch (2000), we change the transition probability between items to the proportion of each video's *DanMu*s to represent the dynamic growth over videos. As we have mentioned that new videos may be added over time, we redefine p_{mt} as Equation (14) to eliminate the affect of new videos:

$$p_{mt} = \frac{d_{mt}}{d_t - d_{new,t}}, \tag{14}$$

where $d_{new,t}$ is the number of *DanMu*s for the new videos on day t. If there is no herding effect, then the proportion of video m does not change over time.

As we want to check whether the popular videos have an effect on other videos based on the number of *DanMu*s of each video, we compute the distance between the video m and popular videos based on the number of *DanMu*s as follows:

$$dis(m, o_{t-1}) = \frac{\overline{d}_{o_{t-1}} - d_{m(t-1)}}{\overline{d}_{o_{t-1}}}, \tag{15}$$

where $\overline{d}_{o_{t-1}}$ is the average number of *DanMu*s of popular videos on day $t-1$.

6.1.2 Parameter Estimation. The value of θ stands for the herding effect of the popular videos on other videos: If $\theta = 0$, then the popular videos have no herding effect on other videos. If $\theta > 0$, then the popular videos have a positive herding effect on other videos. And if $\theta < 0$, then the popular videos have a negative herding effect on other videos. As we could directly compute $p_{mt}, p_{m(t-1)}$ and the distance $dis_{m,o_{t-1}}$, the only parameter to learn is θ. Specifically, we adopt Equation (16) to estimate the herding effect parameter θ as follows:

$$\min_{\theta_{t-1}} \frac{1}{2} \sum_{t=2}^{T} \sum_{m=1}^{M_{t-1}} (\widetilde{p}_{mt} - p_{mt})^2, \tag{16}$$

where T is the longevity of the data by the day. For simplicity, we set $J = \frac{1}{2}\sum_{t=2}^{T}\sum_{m=1}^{M_{t-1}} (\widetilde{p}_{mt} - p_{mt})^2$. We use the gradient descent algorithm to learn θ. First, we calculate the partial derivative of θ as follows:

$$J_{\theta_{t-1}} = \sum_{m=1}^{M_{t-1}} (\widetilde{p}_{mt} - p_{mt}) * \frac{\partial \widetilde{p}_{mt}}{\partial \theta_{t-1}}, \tag{17}$$

$$\frac{\partial \widetilde{p}_{mt}}{\partial \theta_{t-1}} = \frac{p_{m(t-1)}[1 + dis(m, o_{t-1})]^{-\theta_{t-1}}}{D_t^2} * \left\{ \sum_{m'=1}^{M_{t-1}} p_{m'(t-1)}[1 + dis(m', o_{t-1})]^{-\theta_{t-1}} \ln [1 + dis(m', o_{t-1})] - \ln [1 + dis(m, o_{t-1})] D_t \right\}. \tag{18}$$

Then we set the update rule of θ_{t-1} as follows:

$$\theta_{t-1}^{(n+1)} = \theta_{t-1}^{(n)} - \eta * J_{\theta_{t-1}}^{(n)}, \tag{19}$$

where η is the learning rate. To set a proper starting point for learning, we set $\theta_{t-1} = 1$ for $t = 2, 3, \ldots, N$ at the beginning. Algorithm 1 sketches the procedure of the model.

ALGORITHM 1: Framework of Learning Parameters for Our Model.

Input:
 The *DanMu* proportion of video m by the day t, p_{mt};
 The distance between the video m and the *DanMus*' herding target by day $t-1$, $dis_{m,o_{t-1}}$;
Output:
 Setting of each parameter θ_{t-1} for $t = 2, 3, \ldots, N$;
// Initialization
Initialize each parameter $\theta_{t-1} = 1$ for $t = 2, 3, \ldots, N$;
// Iterative optimization
repeat
 // Update parameters;
 for *each* $t \in [2, T]$ **do**
  Update θ_{t-1} according to Equation (19);
 end
until *not converged yet*;
Return Each parameter θ_{t-1} for $t = 2, 3, \ldots, N$;

6.1.3 Experimental Results. To validate the proposed prediction model, we adopt two linear regression models as baselines. The first one utilizes the *DanMu*'s proportion information, which is linearly correlated to previous I days' *DanMu*'s proportions as $\tilde{p}_{mt} = \sum_{i=1}^{I} a_i * p_{m(t-i)}$. This baseline is denoted as pmtREG. The other one is based on the video's features as $\tilde{p}_{mt} = \sum_{n=1}^{N} \lambda_n * f_{mn}$, where λ_1 stands for the coefficient of the number of views, λ_2 is the TReview's coefficient, λ_3 is the collect's coefficient, λ_4 denotes the banana's coefficient, and the λ_5 is the *DanMu*'s coefficient. This baseline is denoted as featureREG,

To avoid the potential sparseness issue, we remove the videos whose view number is less than 500 and where *DanMu*'s number is less than 50. Then we choose the initial 130 days' data to train the model and the remaining 53 days' data to test the model. When we rank all videos by the number of each video's *DanMu*s on the website acfun.tv, the website will list the top 20 videos on the first page. Because the videos on the first page will mitigate users' activity on videos on the following pages, it motivates us to set the threshold k for selecting popular videos as 20. As we predict the *DanMu*'s proportion of the video m on the next day, the most direct and effective

Table 3. The Results of *aadr*

Name	*aadr*	Relative improvement
pmtREG	9.30%	-
featureREG	5.26%	43.44%
DHEModel	4.08%	56.13%

Table 4. The Results of *a* for pmtREG

a	a_1	a_2	a_3
Value	0.332	0.318	0.268

Table 5. The Results of λ for FeatureREG

λ	λ_1	λ_2	λ_3	λ_4	λ_5
Value	0.002	−0.002	0.001	−0.001	0.147

evaluation metric is to compute the average absolute difference ratio between the observed values and the predicted values, which is denoted as *aadr*:

$$aadr_t = \frac{\sum_{m=1}^{M_t} \frac{Abs(\tilde{p}_{mt} - p_{mt})}{p_{mt}}}{M_t}. \tag{20}$$

The comparisons in terms of *aadr* are shown in Table 3. Besides the *aadr* index, we also show the relative improvement in terms of *aadr*. From the table, we find that the DHEModel is the best among all models with the greatest improvement 56.13%, which indicates that the dynamic herding effect is an extremely important factor for predicting the growth of *Danmu*s over videos. The pmtREG model is the worst among all models, which reflects that the linear regression model based on the previous proportions does not work well.

The training values of $\boldsymbol{a}$ for the pmtREG model are shown in Table 4, where we can see that the growth of *DanMu*s depends more on the volume of *DanMu*s in nearly previous days.

The training values of $\boldsymbol{\lambda}$ for the featureREG model are shown in Table 5. From this table, we can see that the volume of *DanMu*s in the previous day and the feature of view play important roles in predicting the growth of *DanMu*s. The other three features have a relatively small impact on predicting the growth of *DanMu*s. In particular, the traditional review has a negative impact on predicting the growth of *DanMu*s.

For the DHEModel, we learn the herding effect parameters for each day. We find that the learned herding effect parameters for each day do not have the same sign. In other words, the values on some days are positive, while the values on other days are negative. This is reasonable, because popular videos may have a positive effect on the other videos on some days and a negative effect on the other videos on other days. We show the dynamic change of the herding effect parameters on each day in Figure 19. From Figure 19, we can draw several implications: First, in the initial few days, the fluctuation of parameters is stronger than in subsequent days. The possible reason is that the number of videos is not large during the first few days, which leads to more fluctuation.

6.2 Dynamic Growth over Video Segments

As introduced in Section 2, *DanMu*s may be distributed around different segments of a video. Predicting such distributions of *DanMu*s over video segments may benefit online video system operation such as to optimize the online ads display. To this end, we propose a systematical framework to integrate a variety of factors for predicting the growth over video segments, namely the Time, Video quality, and Herding effect Aware *DanMu* dynamics model (TVHAD).

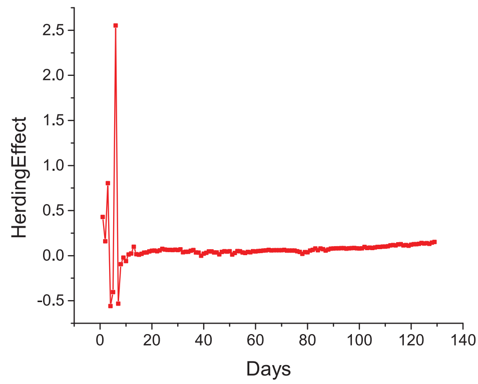


Fig. 19. The dynamics of herding effect parameters.

6.2.1 Method for Predicting the Growth of DanMus *over Segments.* As shown in Section 2, there is a clear herding effect on the growth of *DanMu*s over video segments. To address such a herding effect, we adopt and extend a model proposed in Wang et al. (2014). The research focus of Wang et al. (2014) is to model the social influence of prior ratings on subsequent ratings in common e-commerce systems. While the herding effect of *DanMu* is strongly related to the social influence in Wang et al. (2014), there are still some unique features (such as the mutual influence between segments) we need to address for the *DanMu* data.

Specifically, we first uniformly divide each video into S segments along the video time. The description of all parameters used in our TVHAD model is given below.

(1) *DanMu* history $\boldsymbol{x}_i$. The first $(i-1)$ *DanMu*s form the history for r_i: $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{iS}]^T$, where x_{is} represents the proportion of *DanMu*s on segment s among the first $(i-1)$ *DanMu*s. It is obvious that $\sum_{s=1}^{S} x_{is} = 1$ for $i > 1$ and $\boldsymbol{x}_1$ is a zero vector.
(2) $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_S]^T$ represents the coefficients of all segments, which is used to represent the true quality of the video.
(3) $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_S]^T$ represents the coefficient of an time distribution, which is assumed related to the upload time of the video.
(4) f is the magnitude function. f_i represents the relationship between the strength of the global herding effect and the number of all historical *DanMu*s. And f_{i_s} represents the relationship between the strength of segment herding effect and the number of historical *DanMu*s for the segment s, where i_s is the number of historical *DanMu*s for the segment s.
(5) $\boldsymbol{\theta}$ represents the weights of segment of $\mathbf{x}_i$. The $\boldsymbol{\theta}$ can capture both positive and negative influences between segments. If $\theta_{s,s'} > 0$, then the *DanMu*s on the previous segment s has a positive effect on the formation of *DanMu*s on segment s'. On the contrary, if $\theta_{s,s'} < 0$, then the preceding *DanMu*s on segment s has a negative effect on the formation of *DanMu*s on segmentation s'.

Then we present the TVHAD model based on the parameters described above to predict the probability of the ith *DanMu* r_i over different segments:

$$p(r_i = s|\mathbf{x}_i) = \frac{exp\left(\mu_s + \beta_s t(i) + f_i f_{i_s} \boldsymbol{\theta}_s^T \mathbf{x}_i\right)}{\sum_{s'=1}^{S} exp\left(\mu_{s'} + \beta_{s'} t(i) + f_i f_{i_{s'}} \boldsymbol{\theta}_{s'}^T \mathbf{x}_i\right)}. \tag{21}$$

The $p(r_i = s|\mathbf{x}_i)$ shows the probability of the *DanMu* i on the segmentation s given *DanMu* history $\mathbf{x}_i$, where t is the time span from the video's upload time to the *DanMu* ith time. Comparing to the model in Wang et al. (2014), we make two adjustments on the original model. First, we add the time distribution $\boldsymbol{\beta}$ on the model, which represents the contribution that the time span t influences the generation of *DanMu*s on video's segments. The model in Wang et al. (2014) has considered the magnitude function f_i, which reflects the herding effect based on all historical *DanMu*s, but the model has not captured the specified herding effect based on each segment's historical *DanMu*s for prediction. For capturing the specified herding effect, we add the magnitude function f_{i_s} in our proposed model, which reflects the herding effect based on the number of *DanMu*s on the segment s. These two adjustments in our model make the prediction of the proportion of *DanMu*s on video segments more accurate and efficient.

6.2.2 Parameter Estimation. We need to learn the parameters $\boldsymbol{\mu}$, $\boldsymbol{\beta}$, $\boldsymbol{\theta}$ and the magnitude functions f_i and f_{i_s}. For simplicity, we denote $\boldsymbol{\Psi} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_S, \boldsymbol{\mu}, \boldsymbol{\beta}]$.

Based on the dataset, we can get the ordered sequence of N *DanMu*s $\{r_i\}_{i=1}^N$ by time. Then we give the log-likelihood of parameters $\boldsymbol{\Psi}$ given $\{r_i\}_{i=1}^N$ as follows:

$$\begin{aligned} J(\boldsymbol{\Psi}) &= \frac{1}{N} \log \prod_{i=1}^{N} p(r_i|\mathbf{x}_i, \boldsymbol{\Psi}) \\ &= \frac{1}{N} \sum_{i=1}^{N} \sum_{s=1}^{S} y_{i,s} \log \frac{\exp\left[\mu_s + \beta_s t(i) + f_i f_{i_s} \boldsymbol{\theta}_s^T \mathbf{x}_i\right]}{\sum_{s'=1}^{S} \exp\left[\mu_{s'} + \beta_{s'} t(i) + f_i f_{i_{s'}} \boldsymbol{\theta}_{s'}^T \mathbf{x}_i\right]}, \end{aligned} \tag{22}$$

where $y_i \in \{0,1\}^S$ is an indicator with $y_{i,s} = 1$ if $r^i = s$ and $y_{i,s} = 0$ otherwise.

We optimize the penalized log-likelihood function (Moons et al. 2004; Wang et al. 2014) to estimate the $\boldsymbol{\Psi}$ as follows:

$$J_\lambda(\boldsymbol{\Psi}) = -J(\boldsymbol{\Psi}) + \frac{\lambda}{2}\left[\|\boldsymbol{\Psi}\|_F^2 + R(f_i) + R(f_{i_s})\right], \tag{23}$$

where λ is the balance parameter to prevent overfitting. $\|\cdot\|_F$ is the matrix Frobenius norm and $R(f_i) = \int_0^\infty (f_i')^2 di$.

$J_\lambda(\boldsymbol{\Psi})$ is similar to the softmax regression, which has the integral of an unknown function and all the parameters coupled. These characteristics makes $J_\lambda(\boldsymbol{\Psi})$ hard to adopt off-the-shelf optimization methods. While Wang et al. (2014) provides a surrogate function to handle these characteristics, we refer to the surrogate function $Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(n)})$, which is a tight upper bound of $J_\lambda(\boldsymbol{\Psi})$ to decouple the parameters, where $\boldsymbol{\Psi}^{(n)} = [\boldsymbol{\theta}_1^{(n)}, \boldsymbol{\theta}_2^{(n)}, \ldots, \boldsymbol{\theta}_S^{(n)}, \boldsymbol{\mu}^{(n)}, \boldsymbol{\beta}^{(n)}]$ denotes the current parameter setting:

$$\begin{aligned} Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(n)}) = &\frac{1}{N} \sum_i \sum_s \left[\phi_{i,s}^2 + \left(\zeta_{i,s}^{(n)} - 2\phi_{i,s}^{(n)} - y_{i,s}\right)\phi_{i,s}\right] \\ &- \frac{1}{NS} \sum_i \left(\sum_s \phi_{i,s} - 2\sum_s \phi_{i,s}^{(n)}\right)\left(\sum_s \phi_{i,s}\right) + \frac{\lambda}{2}\left[\|\boldsymbol{\Psi}\|_F^2 + R(f_i) + R(f_{i_s})\right] + \frac{1}{N}\sum_i C_i(n), \end{aligned} \tag{24}$$

where terms $\phi_{i,s}$, $\phi_{i,s}^{(n)}$, $\zeta_{i,s}^{(n)}$, and $C_i^{(n)}$ are defined below,

$$\phi_{i,s} = \mu_s + \beta_s t(i) + f_i f_{i_s} \boldsymbol{\theta}_s^T \mathbf{x}_i,$$

$$\phi_{i,s}^{(n)} = \mu_s^{(n)} + \beta_s^{(n)} t(i) + f_i^{(n)} f_{i_s}^{(n)} \boldsymbol{\theta}_s^{(n)T} \mathbf{x}_i,$$

$$\zeta_{i,s}^{(n)} = \frac{\exp\left(\phi_{i,s}^{(n)}\right)}{\sum_{s'} \exp\left(\phi_{i,s'}^{(n)}\right)},$$

$$C_i^{(n)} = \sum_s \left(\phi_{i,s}^{(n)2} - \zeta_{i,s}^{(n)}\phi_{i,s}^{(n)}\right) - \frac{1}{S}\left(\sum_s \phi_{i,s}^{(n)}\right)^2 + \log\sum_s \exp\left(\phi_{i,s}^{(n)}\right).$$

Updating Parameters and Magnitude Functions. To obtain $\mathbf{\Psi}$'s update rule, it is explicit to extract the derivatives of $Q(\mathbf{\Psi};\mathbf{\Psi}^{(n)})$ with respect to u_s, β_s, and $\theta_{s,s'}$, which are defined as

$$\mu_s^{(n+1)} = \frac{S\sum_i \left(y_{i,s} - \zeta_{i,s}^{(n)}\right) + 2N(S-1)\mu_s^{(n)}}{2N(S-1) + NS\lambda}, \tag{25}$$

$$\beta_s^{(n+1)} = \frac{S\sum_i t(i)\left(y_{i,s} - \zeta_{i,s}^{(n)}\right) + 2(S-1)\beta_s^{(n)}\sum_i t(i)^2}{2S\sum_i t(i)^2 - 2\sum_i t(i)^2 + NS\lambda}, \tag{26}$$

$$\theta_{s,s'}^{(n+1)} = \frac{S\sum_i \left[f_i f_{i_s} x_{i,s'}\left(y_{is} - \zeta_{i,s}^{(n)}\right)\right] + 2(S-1)\sum_i f_i^2 f_{i_s}^2 x_{i,s'}^2 \theta_{s,s'}^{(n)}}{2(S-1)\sum_i f_i^2 f_{i_s}^2 x_{i,s'}^2 + NS\lambda}. \tag{27}$$

Then we optimize the magnitude function in an infinite space to extract its update rule. First, we devise the parts of $Q(\mathbf{\Psi};\mathbf{\Psi}^{(n)})$ relevant to magnitude functions. We reformulate the problem of minimizing $Q(\mathbf{\Psi};\mathbf{\Psi}^{(n)})$ with respect to f_i and f_{i_s} as

$$\min_{f_i} \sum_i A_i f_i^2 + \sum_i B_i f_i + \frac{\lambda}{2}\int_0^{+\infty} (f'(i))^2 di, \tag{28}$$

$$\min_{f_{i_s}} \sum_i A_{i_s} f_{i_s}^2 + \sum_{i_s} B_{i_s} f_{i_s} + \frac{\lambda}{2}\int_0^{+\infty} (f'(i))^2 di, \tag{29}$$

where A_i, B_i, A_{i_s}, and B_{i_s} are defined below:

$$A_i = \frac{1}{N}\sum_s \left(\boldsymbol{\theta}_s^{(n)T}\boldsymbol{x}_i f(i_s)^{(n)}\right)^2 - \frac{1}{NS}\left(\sum_s \boldsymbol{\theta}_s^{(n)T}\boldsymbol{x}_i f(i_s)^{(n)}\right)^2,$$

$$B_i = \frac{1}{N}\sum_s \left[2\left(\mu_s^{(n)} + \beta_s^{(n)}t(i) - \phi_{i,s}^{(n)}\right) + \zeta_{i,s}^{(n)} - y_{i,s}\right]\boldsymbol{\theta}_s^{(n)T}\boldsymbol{x}_i f_{i_s}^{(n)}$$
$$+\frac{2}{NS}\left(\sum_s \boldsymbol{\theta}_s^{(n)T}\boldsymbol{x}_i f_{i_s}^{(n)}\right)\left[\sum_s \phi_{i,s}^{(n)} - \sum_s\left(\mu_s^{(n)} + \beta_s^{(n)}t(i)\right)\right],$$

$$A_{i_s} = \frac{1}{N}\sum_s \left(\boldsymbol{\theta}_s^{(n)T}\boldsymbol{x}_i f_i^{(n)}\right)^2 - \frac{1}{NS}\left(\sum_s \boldsymbol{\theta}_s^{(n)T}\boldsymbol{x}_i f_i^{(n)}\right)^2,$$

$$B_{i_s} = \frac{1}{N}\sum_s \left(2\mu_s^{(n)} + 2\beta_s^{(n)}t(i) - 2\phi_{i,s}^{(n)} + \zeta_{i,s}^{(n)} - y_{i,s}\right)\boldsymbol{\theta}_s^{(n)T}\boldsymbol{x}_i f_i^{(n)}$$
$$+\frac{2}{NS}\left(\sum_s \boldsymbol{\theta}_s^{(n)T}\boldsymbol{x}_i f_i^{(n)}\right)\left[\sum_s \phi_{i,s}^{(n)} - \sum_s\left(\mu_s^{(n)} + \beta_s^{(n)}t(i)\right)\right].$$

After we reformulate the objective functions of the magnitude functions defined in Equation (28) and Equation (29), we need to devise the update rules of magnitude functions. As proved in Wang et al. (2014), the update rule of Equation (28) must satisfy the Euler-Lagrange equation (Zhou et al. 2013):

$$2A(i)f_i + B(i) - \lambda f_i'' = 0, \tag{30}$$

where $f''(\cdot)$ is the second-order derivative of $f(\cdot)$. $A(i) = A_i I\{i \leq N \wedge i \in \mathbb{N}\}$ and $B(i) = B_i I\{i \leq N \wedge i \in \mathbb{N}\}$ ($I\{\cdot\}$ is the indicator function that returns 1 if the predicate is true and 0 otherwise).

Because Equation (29) has the same type of function as Equation (28), Equation (29) also satisfies the Euler-Lagrange equation. To solve these differential equations, we use the Seidal type iteration, which is good at solving functions with discrete features:

$$\lambda(f_{i+1} - 2f_i + f_{i-1}) - 2A(i)f_i - B(i) = 0. \tag{31}$$

Based on the equations, we can employ curve fitting to solve the values of f_i and f_{i_s}.

Complete Algorithm. We set up the following initialization:

$$\begin{cases} \mu_s = \log\left(\frac{\sum_{i=1}^{N} y_{i,s}}{N}\right) & s = 1, 2, \ldots, S \\ \beta_s = \frac{1}{S} & s = 1, 2, \ldots, S \\ f_i = 0 & i = 1, 2, \ldots, N \\ f_{i_s} = 0 & i_s = 1, 2, \ldots, N_s \\ \boldsymbol{\theta}_s = Random\ value & s = 1, 2, \ldots, S \end{cases}. \tag{32}$$

Algorithm 2 shows how we initialize and learn parameters $\mathbf{\Psi}$ and magnitude functions f_i, f_{i_s}.

ALGORITHM 2: Framework of Learning Parameters for Prediction on Video's Segmentations.

Input:
DanMu history $\{r_i\}_{i=1}^{N}$;
t, the time span from the video's uploaded time to *DanMu* ith date;
Output:
Setting of parameters $\mathbf{\Psi}$ and magnitude functions f_i, f_{i_s};
// Initialization
Initialize parameters according to Equation (32);
Compute statistics $\{x_i\}_{i=1}^{N}$;
// Iterative optimization
repeat
    // Update parameters
    for *each* $s \in [1, S]$ **do**
        update μ_s following Equation (25);
        update β_s following Equation (26);
        for *each* $s' \in [1, S]$ **do**
            update $\theta_{s,s'}$ following Equation (27);
        end
    end
    // Update magnitude function;
    Compute $\{f_i\}_i$, $\{f_{i_s}\}_{i_s}$ by solving differential Equation (30);
until *not converged yet*;
Return Parameters $\mathbf{\Psi}$ and magnitude functions f_i, f_{i_s};

6.2.3 Experimental Results. In this section, we evaluate the effectiveness of the proposed model with the same data as in Section 2.1.

Baseline Methods. Other than the TVHAD model, we have four baseline models for predicting the *DanMu* growth.

(1) *DanMu* Dynamics model (DD). It assumes that new *DanMu* growth is only related to the quality of videos (μ).

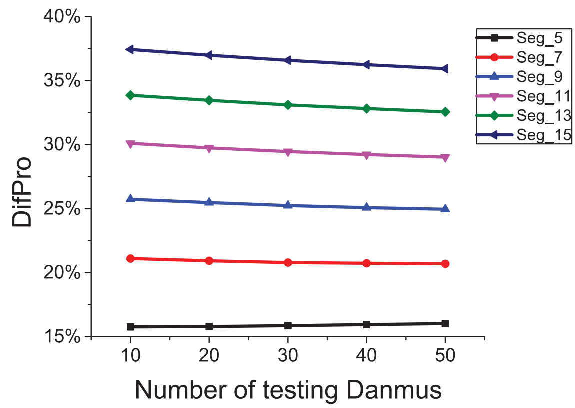


Fig. 20. The DifPro of DD with different numbers of segmentations.

(2) Time Aware *DanMu* Dynamics model (TADD). It assumes that new *DanMu* growth is related to the quality of videos (μ) and the time information (t), which follows the definition of Equation (21) and ignores the herding effect of f_i and f_{i_s}.
(3) Global Herding effect Aware *DanMu* Dynamics model (GHADD) (Wang et al. 2014). The GHADD model predicts the growth of *DanMu*s based on the combination of video's quality (μ) and global herding effect f_i.
(4) Segmentation Herding effect Aware *DanMu* Dynamics model (SHADD). SHADD assumes that new *DanMu* growth is related to the quality of videos (μ) and the segmentation herding effect (f_{i_s}), which also follows the definition of Equation (21) and ignores the time information and the global herding effect f_i.

The default value of parameter λ is 0.08. For evaluating the long-term prediction power of our proposed model, we dynamically adjust the length of testing *DanMu*s (i.e., 10, 20, 30, 40, and 50) and adopt six different segment numbers, including 5, 7, 9, 11, 13, and 15. Also 2/3 of the data is used to train models, and the remaining 1/3 of the data is used to test models.

Growth Prediction Results. To show the improvement clearly, we compute the percentage of improvement for other growth models compared to the DD model. If the percentage is greater than 0, then it means the growth model is better than the DD model. On the contrary, if the rate is less than 0, then it means the growth model is worse. To comprehensively evaluate the models, we propose two measures DifPro and DifRank, which can evaluate the accuracy and rank consistence of prediction, respectively. The results are summarized in the following:

— *Absolute Difference of Probability (DifPro).* DifPro is the absolute difference between each video's predicted probability and actual probability, which is used to test the accuracy of prediction. We first demonstrate DifPro of the DD model with different lengths of testing *DanMu*s and different segment numbers in Figure 20. As the number of testing *DanMu*s increases, the DifPro slowly decreases. Nevertheless, the DifPro increases faster with the increase of the segment number. For example, when the number of testing *DanMu*s is 20, the DifPro is 29.8% with the segment number as 11, which is 2 times that (15.7%) with the segment number as 5.

Then we give the rate of the improvement between the DD model and other models in Figure 21. We find the DifPro of the TADD model is greater than 0, which means the time information is very important and has a positive effect on predicting the growth of *DanMu*s

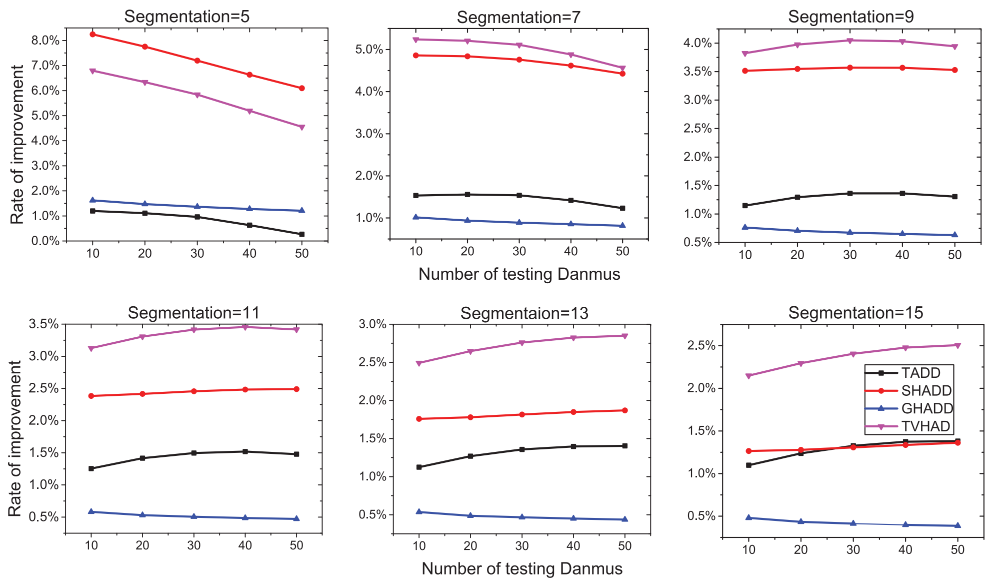


Fig. 21. The rate of DifPro's improvement with different numbers of segmentations.

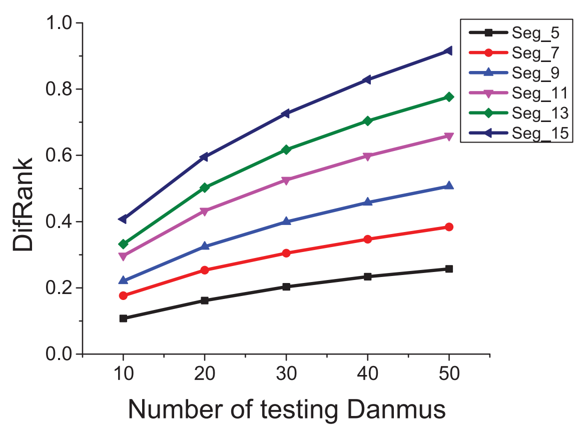


Fig. 22. The DifRank of DD with different numbers of segmentations.

over segments. The SHADD model is always better than the GHADD model, which proves our conjecture that the segment herding effect (f_{i_s}) is more effective than the global herding effect (f_i). It is noticed that with the increase of segments, our model TVHAD obviously outperforms other models in terms of DifPro. In particular, with the increase of testing *DanMu*s, the TVHAD appears more accurate compared to other models.

— *Average absolute difference of rank (DifRank).* DifRank is the average absolute difference between the rank of predicted values and actual ones, which is used to test the ranking consistence of segments based on prediction probability. We show the DifRank of the DD model in Figure 22 and the rate of the improvement of other models over the DD model in Figure 23. Based on Figure 22, we can conclude that with the increase of the number of testing *DanMu*s and segment number, the DifRank becomes worse, which is different

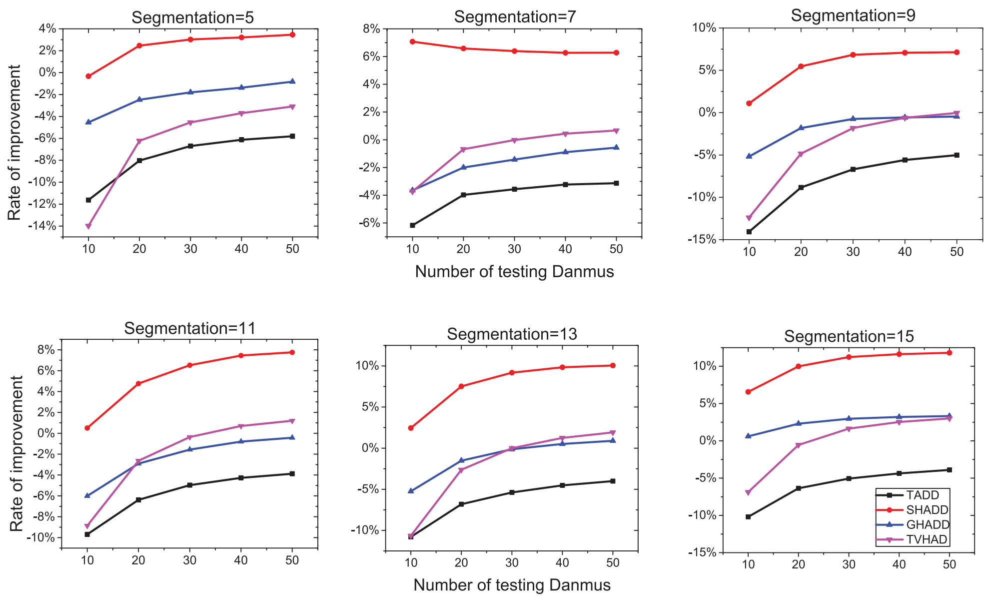


Fig. 23. The rate of DifRank's improvement with different numbers of segmentations.

from DifPro. A main reason may be that the index of DifRank is more sensitive than DifPro when the number of testing *DanMu*s and segments are changed. The improvement rates of other models over the DD model are shown in Figure 23. The TADD is the worst among all the other models, which means the time information is not as good as other factors for predicting the growth of *DanMu*s on segments.

Considering all findings in terms of DifPro and DifRank, we can conclude that the segment herding effect (f_{i_s}) is more effective and powerful than the global herding effect (f_i), and the time information cannot be ignored in growth prediction models. We would emphasize that although TVHAD achieves good performance with respect to DifPro compared to other baselines, the TVHAD model is worse than SHADD with respect to DifRank. The possible reason is that the learning process in Algorithm 2 can ensure a good fitting of the probability of *DanMu*s as the optimization function is based on the probability, but it may not ensure the optimal parameters for a good fit with respect to DifRank.

Illustrating Herding Effect. Here we take an in-depth analysis of features of f_i and f_{i_s} as shown in Figure 24 and Figure 25. Besides showing learned values of the two herding effect functions, we also apply curve fitting to f_i and f_{i_s} with an exponential model $a * exp(i/b) + c$ (a, b and c are parameters) based on the shape of the learned values. Despite the slightly different parameter settings of a, b, and c, the exponential model can fit the learned values very well.

Based on Figure 24 and Figure 25, we can draw several implications: First, the strength of f_i and f_{i_s} can be explicitly quantified by the same formula. Second, it validates our hypothesis that the strength of the herding effect evolves with the cumulative number of *DanMu*s. Third, the strength of the herding effect becomes stronger quickly at the beginning, while the strength of the herding effect becomes steady when the number of *DanMu*s reaches a particular threshold. At last, because f_{i_s} is to capture the herding effect in growth models better than f_i, we observe that the strength of the herding effect from f_{i_s} is stronger than that from f_i. That is why the SHADD model is more effective and powerful than the GHADD model.

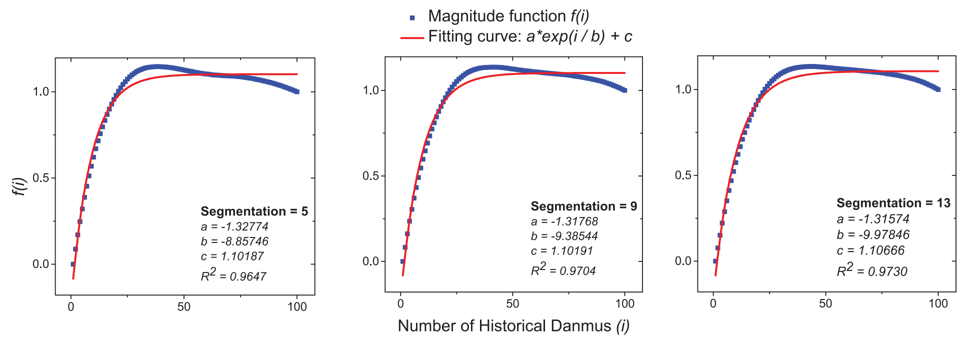


Fig. 24. Global Herding effect function f_i and fitting curve $a * exp(i/b) + c$.

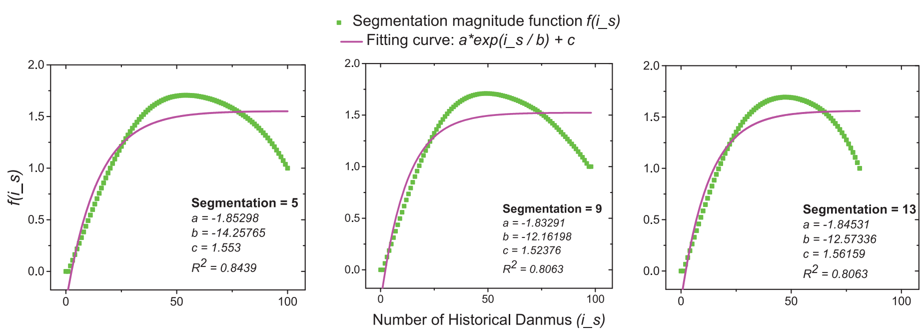


Fig. 25. Segmentation herding effect function f_{i_s} and fitting curve $a * exp(i_s/b) + c$.

7 CONCLUSIONS

In this article, for the first time, we provided a comprehensive study on *DanMu* data, which is an emerging type of user-generated comment. We collected a large set of data from a *DanMu*-enabled online video system (acfun.tv) that includes 6,506 videos, 1,704,930 *DanMu*s, 320,000 writers, and 155,455 TReviews. We first revealed interesting distributions of *DanMu*s over both time dimensions and make comparisons with TReviews. These distributions convey two types of interesting patterns of *DanMu* data: a herding effect and multiple-burst phenomena. We proposed practical methods to identify and measure both patterns that reveal insights about the growth of *DanMu* and proposed to detect leading *DanMu*s within bursts along video time, because these leading *DanMu*s very likely stimulate subsequent *DanMu*s and lead to both herding effect and multiple bursts. Towards detecting leading *DanMu*s, we explored three types of features and designed two methods to rank *DanMu*s within bursts according to the leading possibility. Experimental results with our data demonstrate the effectiveness of our detection methods and different powers of different combinations of features. Using the identified unique characteristics of *DanMu*, we further proposed two models to predict the dynamic growth of *DanMu* over videos and video segments. We conducted intensive experiments to demonstrate the effectiveness of both models with the collected data.

ACKNOWLEDGMENTS

The authors thank Le Wu and Yan Chen for their help in modification of the article and HongPing Guo for help with data analysis.

REFERENCES

Maria Andersson, Carmen Lee, Ted Martin Hedesström, and Tommy Gärling. 2006. Effects of reward system on herding in a simulated financial market. *Interaction on the Edge* (2006), 12.

Abhijit V. Banerjee. 1992. A simple model of herd behavior. *Quart. J. Econ.* 107, 3 (1992), 797–817.

Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *Science* 286, 5439 (1999), 509–512.

Youmna Borghol, Sebastien Ardon, Niklas Carlsson, Derek Eager, and Anirban Mahanti. 2012. The untold story of the clones: Content-agnostic factors that impact youtube video popularity. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1186–1194.

Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. 2009a. Analyzing the video popularity characteristics of large-scale user generated content systems. *IEEE/ACM Trans. Netw.* 17, 5 (2009), 1357–1370.

Meeyoung Cha, Alan Mislove, and Krishna P. Gummadi. 2009b. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of the 18th International Conference on World Wide Web*. ACM, 721–730.

Kenny K. Chan and Shekhar Misra. 1990. Characteristics of the opinion leader: A new dimension. *J. Advert.* 19 (1990), 53–60.

William G. Christie and Roger D. Huang. 1995. Following the pied piper: Do individual returns herd around the market? *Financ. Analyst. J.* 51, 4 (1995), 31–37.

Aaron Clauset, Cosma Rohilla Shalizi, and Mark E. J. Newman. 2009. Power-law distributions in empirical data. *SIAM Rev.* 51, 4 (2009), 661–703.

Jacob Cohen, Patricia Cohen, Stephen G. West, and Leona S Aiken. 2013. *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*. Routledge.

Flavio Figueiredo, Fabrício Benevenuto, and Jussara M. Almeida. 2011. The tube over time: Characterizing popularity growth of youtube videos. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*. ACM, 745–754.

Joshua Hawthorne, J. Brian Houston, and Mitchell S. McKinney. 2013. Live-tweeting a presidential primary debate: Exploring new political conversations. *Social Science Computer Review* 31, 5 (2013), 552–562.

John D. Hey and Andrea Morone. 2004. Do markets drive out lemmings or vice versa? *Economica* 71, 284 (2004), 637–659.

Sandra Hsieh, Yang Yu Tai, and Tam Bang Vu. 2008. Do herding behavior and positive feedback effects influence capital inflows? Evidence from Asia and Latin America. *Int. J. Bus. Financ. Res.* 2, 2 (2008), 19–34.

Andreas Jungherr. 2015. The media connection. In *Analyzing Political Communication with Digital Trace Data*. Springer, 155–188.

Jon Kleinberg. 2003. Bursty and hierarchical structure in streams. *Data Min. Knowl. Discov.* 7, 4 (2003), 373–397.

Feng Li and Timon C Du. 2011. Who is talking? An ontology-based opinion leader identification framework for word-of-mouth marketing in online social blogs. *Decision Support Syst.* 51, 1 (2011), 190–197.

Huayu Li, Yong Ge, Richang Hong, and Hengshu Zhu. 2016. Point-of-interest recommendations: Learning potential check-ins from friends. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 975–984.

Huayu Li, Richang Hong, Shiai Zhu, and Yong Ge. 2015. Point-of-interest recommender systems: A separate-space perspective. In *2015 IEEE International Conference on Data Mining (ICDM'15)*. 231–240.

Qi Liu, Enhong Chen, Hui Xiong, Chris H. Q. Ding, and Jian Chen. 2012. Enhancing collaborative filtering by user interest expansion via personalized ranking. *IEEE Trans. Syst. Man Cybernet. B* 42, 1 (2012), 218–233.

Qi Liu, Yong Ge, Zhongmou Li, Enhong Chen, and Hui Xiong. 2011. Personalized travel package recommendation. In *2011 IEEE 11th International Conference on Data Mining (ICDM'11)*. IEEE, 407–416.

Guangyi Lv, Tong Xu, Enhong Chen, Qi Liu, and Yi Zheng. 2016. Reading the videos: Temporal labeling for crowdsourced time-sync videos based on semantic embedding. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.

Naohiro Matsumura, Yukio Ohsawa, and Mitsuru Ishizuka. 2002. Mining and characterizing opinion leaders from threaded online discussions. In *Proceedings of the 6th International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies*. 1267–1270.

Seungwhan Moon, Saloni Potdar, and Lara Martin. 2014. Identifying student leaders from MOOC discussion forums through language influence. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 15–20.

K. G. M. Moons, A. Rogier T. Donders, E. W. Steyerberg, and F. E. Harrell. 2004. Penalized maximum likelihood estimation to directly adjust diagnostic and prognostic prediction models for overoptimism: A clinical example. *J. Clin. Epidemiol.* 57, 12 (2004), 1262–1270.

Júnia Ortiz, Solange Hamrin, Camilo Aggio, and Edson Dalmonte. 2015. Television experience and political discussion on Twitter: Exploring online conversations during the 2014 Brazilian presidential elections. (2015).

G. Palshikar and others. 2009. Simple algorithms for peak detection in time-series. In *Proceedings of the 1st International Conference on Advanced Data Analysis, Business Analytics and Intelligence*.

A. Sboner, A. Romanel, A. Malossini, F. Ciocchetta, F. Demichelis, I. Azzini, E. Blanzieri, and R. Dell Anna. 2007. Simple methods for peak and valley detection in time series microarray data. In *Methods of Microarray Data Analysis V*. Springer, 27–44.

Xiaodan Song, Yun Chi, Koji Hino, and Belle Tseng. 2007. Identifying opinion leaders in the blogosphere. In *Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management*. ACM, 971–974.

Gabor Szabo and Bernardo A. Huberman. 2010. Predicting the popularity of online content. *Commun. ACM* 53, 8 (2010), 80–88.

Damian Trilling. 2015. Two different debates? Investigating the relationship between a political debate on TV and simultaneous comments on Twitter. *Soc. Sci. Comput. Rev.* 33, 3 (2015), 259–276.

Michail Vlachos, Christopher Meek, Zografoula Vagena, and Dimitrios Gunopulos. 2004. Identifying similarities, periodicities and bursts for online search queries. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*. ACM, 131–142.

Ting Wang, Dashun Wang, and Fei Wang. 2014. Quantifying herding effects in crowd wisdom. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1087–1096.

Ivo Welch. 2000. Herding among security analysts. *J. Financ. Econom.* 58, 3 (2000), 369–396.

Bin Wu, Erheng Zhong, Ben Tan, Andrew Horner, and Qiang Yang. 2014. Crowdsourced time-sync video tagging using temporal and personalized topic modeling. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 721–730.

Le Wu, Yong Ge, Qi Liu, Enhong Chen, Richang Hong, Junping Du, and Meng Wang. 2017. Modeling the evolution of users preferences and social links in social networking services. *IEEE Trans. Knowl. Data Eng.* 29, 6 (2017), 1240–1253.

Zechen Wu and Eisuke Ito. 2014. Correlation analysis between user's emotional comments and popularity measures. In *Proceedings of the 2014 IIAI 3rd International Conference on Advanced Applied Informatics (IIAIAAI'14)*. IEEE, 280–283.

Xiao Yu, Xu Wei, and Xia Lin. 2010. Algorithms of BBS opinion leader mining based on sentiment analysis. In *Web Information Systems and Mining*. Springer, 360–369.

Stelios H. Zanakis, Anthony Solomon, Nicole Wishart, and Sandipa Dublish. 1998. Multi-attribute decision making: A simulation comparison of select methods. *Eur. J. Operat. Res.* 107, 3 (1998), 507–529.

Zhou Zhao, Hanqing Lu, Deng Cai, Xiaofei He, and Yueting Zhuang. 2016. User preference learning for online social recommendation. *IEEE Trans. Knowl. Data Eng.* 28, 9 (2016), 2522–2534.

Ke Zhou, Hongyuan Zha, and Le Song. 2013. Learning triggering kernels for multi-dimensional Hawkes processes. In *Proceedings of the 30th International Conference on Machine Learning (ICML'13)*. 1301–1309.

Yunyue Zhu and Dennis Shasha. 2003. Efficient elastic burst detection in data streams. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 336–345.

Received December 2015; revised February 2017; accepted May 2017