Social Connections in User-Generated Content Video Systems: Analysis and Recommendation

Zhenyu Li, Jiali Lin, Gaogang Xie

Social Connections in User Generated Content Video Systems: Analysis and Recommendation

Zhenyu Li, *Member, IEEE,* Jiali Lin, and Gaogang Xie, *Member, IEEE,*

***Abstract*—User Generated Content video systems are by definition heavily depending on the input of their community of users and their social interactions for video diffusion and opinion sharing. Nevertheless, we show in this paper, through a measurement and analysis of YouKu, the most popular UGC video system in China, that the social connectivity of its users is very low. These results are consistent with what reported about YouTube in previous works. As a UGC system can benefit from audience increase through improved connectivity, our findings motivate us to propose a mean to enhance the connectivity by taking benefit of friend recommendation. To this end, we assess two similarity metrics, where users' interests are derived from their uploads and favorites tagging of videos, to evaluate the interest similarity between friends. The results consistently show that friends share to a great extent common interests. Two friend recommendation algorithms are then proposed that propose potential friends with similar interests as measured by the similarity metrics that can be derived by publicly information provided by users. Experiments on the dataset of Youku desmonstrate that the social connectivity can be greatly enhanced by our friend proposition and that users can access to a larger set of interesting videos throight their recommendations.**

***Index Terms*—UGC systems, social connection, user interest, friend recommendation, tag augmentation.**

I. Introduction

THe past few years have witnessed the remarkable growth of the user generated content (UGC) video systems. For example, YouTube [4], the world's largest UGC video system, has attracted over millions of users with 3 billion video views a day and the equivalent of 48 hours of video uploaded per minute [2]. In China, Youku, the most popular UGC video system, attracts also over 40 million users per day, viewing 240 million videos [3].

UGC video systems differ from traditional Internet video services in that videos are generated and uploaded by the community of users. Users share their videos using the UGC system, that is also use to make social connections that we will abusively name friendship relations. Because of this ability such systems are called *social video* systems. Social video systems by definition heavily depend on the input of the community of their users and their social interactions for video diffusion and opinion sharing. For example, we observed that YouKu users with more friends tend to attract a larger audience for their videos. In [14] Crane *et al.* have shown the existence of a social cascade propagation of videos over the YouTube social video platform,changing the way video content are classically diffused. Nonetheless, as proposed in [7], social relationships can also be used as to build video spam detection schemes based on trust.

However, recent study [13] have shown weak social connectivity in YouTube. Up to 58% of Youtube users have no friends at all, and the average number of friends is only 4.3. Our analysis of Youku shows even weaker connectivity: about 75% of the users have no friends and the largest weakly connected component (WCC) [1] contains only 15% of the users of Youku. However, one can expect to improve the audience of the UGC by enhancing the social connectivity between members. Unlike classical social networks, like Facebook and LinkedIn, where social relationships of users in the real world form the basis of relations in the social network, the rlationships in UGC systems are formed by common users interests relative to contents [18]. This motivate us to propose a friend recommendation scheme that will improve social connectivity by connecting users with similar interests.

To this end, we have crawled the YouKu site and collected 627,000 user profiles, 3 million social connections and 13.6 million videos' information. We have found that the out-degree distribution of users does not follow a power-law distribution, that social connectivity is low and identified the impacts of friendship on user popularity and activity. We have derived two type of users interests from the users' activities. The videos uploaded by a user gives its *producer* interests, while the favorites of a user represents its *consumer* interests.

Each video content is mapped by the user that upload it to some *tags*. These tags as well as the of users favorite videos are publicly available in Youku and not protected. A user's interests are represented by an *interest vector*, containing couples (t_i, w_i), where t_i is a tag among the tags used the the user for its uploaded videos, or a tag among the tags of its favorite videos, w_i is a weight derived for t_i. With the interest vectors, one can compute the interest similarity, *i.e.* proximity, between any pair of users. However, as video tags are defined by the users themselves and are not controlled by the UGC system, they are *ambiguous* and *heterogenous*. This last point brings a major challenge for measuring the interest similarity [5].

We therefore need to introduce some semantics in form of an ontology in order to disambiguate the video tags. For this purpose, we crawled Baidu Encyclopedia [1], a Chinese language web-based encyclopedia. We obtained all 4 million entries and their associated ontologies. We disambiguate the

Zhenyu Li, Jiali Lin and Gaogang Xie are with Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China. E-mail: {zyli, linjiali, xie}@ict.ac.cn.

Jiali Lin is also with Graduate School of Chinese Academy of Sciences.

[1] A weakly connected component of a unidirectional graph is a set of nodes where each node will have a path to every other node in the set if all links are taken as bidirectional.

video tags by adding semantics coming from this huge source of knowledge. This eases greatly the accurate computation of interest similarity.

We propose two similarity metrics and compare them. We present in this paper results that show that to a great extent friends consistently share common interests. We therefore proposeto recommend those with similar interests as potential friends for individual users. Two recommendation algorithms are then proposed. Both of them locate potential friends with similar interests. It should be noted that such link information is always publicly available and not protected by users. As the interest similarity metric is derived on publicly available informatio, our proposed algorithms is attractive to use.

We thereafter show the results of applying friend recommendation algorithms on our YouKu dataset. Experiment results show that if the proposed recommendation is followed, the fraction of users with no friend drops to 8% and 95% of the users become present in the largest WCC. Moreover, users take advantage of their recommended friend to find more videos of interests. These results demonstrate that the social connectivity is effectively enhanced. To the best of our knowledge, this work is the first to measure interest similarity between friends in UGC video systems and to study friend recommendation for such systems.

The rest of the paper is organized as follows. Section II provides a survey of related works. Section III measures YouKu and describes the dataset. In Section IV, we statistically analyze the social connectivity of YouKu, followed by interest similarity analysis in Section V. We then present the friend recommendation algorithm in Section VI. The algorithm is applied on our data set of YouKu in Section VII. Finally, we conclude our work in Section VIII.

II. Related Work

YouTube has been extensively studied. Most of the works focus on video pattern, *e.g.* video popularity and video interaction. Cha et al. [11] analyze the video popularity pattern in YouTube. The results provide insights on the potential for using caching and Peer-to-Peer in such systems. Video interaction pattern in YouTube, which allows users to respond a video with another one, is analyzed in [7]. While video interaction offers much richer way for reviews provision, it facilitates the content pollution.

Online video systems always provide a list of related videos for each video. This prominent feature is deemed to be the key point of the success for such systems [12]. By analyzing the sources of video views in YouTube, Zhou *et al.* [31] show that related video recommendation is the most important source. The related videos form a video social network, where videos are vertexes and related video links are edges. The analysis results in [12] demonstrate that this social network exhibits small-world characteristics.

Besides the video pattern, the user pattern has also been analyzed. Users' long term behavior is analyzed in [8] according to user categories. Paolillo [24] empirically investigates the social structure of YouTube. The results indicate that users are like to make friends with others who upload videos with similar contents. Users are linked with friendship and they form user social networks. Along with other three online social networks based on social relationship, the social network in YouTube is measured by Mislove *et al.* [22]. Different from other three networks, the social network of YouTube shows a smaller scale-free metric, a negative assortativity coefficient and a lower clustering coefficient. These results demonstrate the differences between social relation-based networks and content-oriented networks. A recent work by Ding *et al.* [15] characterizes the uploaders in YouTube and has found that uploaders highly concentrated in very few categories.

Low social connectivity of YouTube have been found in [13] and [29]. The results in [13] show that 58% of the users have no friends, while in [29] the results show that about 60% of the users in YouKu have less than 1 friend. However, to the best of our knowledge, the interest similarity between friends and friend recommendation have not been considered so far.

Friend suggestion is studied in [25] for email communication networks. The proposed algorithm leverages implicit graph formed by users' historical interactions. The affinity between two users is then estimated with an interaction-based metric. Since in UGC video systems users interact less than in email networks, this algorithm is not suitable. A recent work by Yao *et al.* [30] presents a friend recommendation algorithm for Flickr photo sharing system. For each query user, the algorithm should compute his similarities with all others in terms of visual and geo similarities. This would be very cost-consuming for large-scale online social networks. Besides, users' interests are only derived from their uploads, which ignores users' interests as consumers. Link predication approaches [26] predict the exists of links based on users' current friendship, *i.e.* the current social network among users. However, in practice, users would like to protect their friendship from outside for privacy concerns [5].

Our work derives users' interests from both their uploads and favorites. These information is publicly available since users believe it harmless [5]. The friend recommendation algorithms leverage the links of related videos and favorite videos, yielding cost-effective and accurate identification of potential friends.

III. Data Collection from YouKu

This section briefly introduces YouKu, describes the crawling process and the resuling dataset.

A. Brief Introduction of YouKu

YouKu can be described as the YouTube's Chinese twin. It is a UGC that enables registered users to upload videos and comment/rate videos uploaded by other members. Each registered user has a profile page and have an assigned unique ID. The user's profile page contains the list of his uploaded videos, its favorites (the list of videos he has tagged as favorite), his friends (the list of Youku users with social links through Youku) and his aggregate popularity. When we collected our dataset, all the above information were publicly available. Social relationships in Youku are made by the following process: A user X sends a friendship request to another user *Y*. Once *Y* accepts the request a friendship link is build in the Youku system. However, the friendship is not necessarily reciprocal.

Users upload video in Youku system and provide for them a title and a up to 10 terms coined tags and choose for it one category among 20. Youku assigns a unique ID to each videos uploaded by users. Each video has a profile page, which contains its title, the tags, the category, information relative to the uploader, the popularity of the video in terms of number of views, and a list of up to 9 related videos assigned by YouKu. The link between two related videos is not necessarily reciprocal. The comments made by Youku users on each video along with the ID of the commenters is also provided in the profile page. YouKu offers APIs to its business partners, but not to the research community. We have therefore to develop a multi-thread crawler which mimics multiple web browsers to request video/user profile pages.

B. Crawling Process and Data Set

Our goal with crawling is to collect a large number of Youku's user profiles. However, as the social connectivity among users is extremely weak, simple application of the Breath First Search (BFS) to user profiles is not applicable. In place, we crawled Youku by applying the BFS to the video profiles that contain the URLs of the uploader and the 9 related videos. We therefore started by gathering a set of videos by following the related video links, and thereafter we crawled the users who uploaded these videos and followed all the videos uploaded by these users.

The developed multi-thread crawler run on a Linux server with 16GB memory and 2 4-core CPUs. Each thread initiates a web client that follows the above described BFS-like process. A common video ID queue is shared among the crawling threads for coordination. We begun the crawling begun with 326 seed videos in the queue, which were selected from "Most Recent", "Most Viewed", "Most Hot", "Most controversial", "Most Favorites", "Most Recommended", video lists. The variety of seed videos ensures fast crawling and enough diversity of user profiles. To eliminate duplicated crawls of the same videos, we constructed a Bloom filter [9] to quickly determine whether a video ID has been crawled or not.

We started our crawler on Nov. 1, 2010. The crawler digs more than 8 depths, crawling about 3 million unique videos,uploaded by 626,990 thousand users. From Nov. 4 to Nov. 7, we crawled the profile pages of these users. For each user we obtained his friends list, its uploaded videos, its favorites and its popularity, measured by the sum of the number of views of all his videos. At the time we crawled Youku, all these users have uploaded 13,594,037 videos and faved 12,102,651 videos . Among the favorite videos of these users, only 591,652 (4.9%) videos were not uploaded by users in the set. Finally, we crawled all the 13,594,03+591,652=14,185,689 videos and got their video tags. This last steps lasted 20 days. Since our crawler was aggressively requesting information from the servers, it was sometimes temporally (about 1 hour) banned by YouKu. When detecting an access ban, the crawler paused automatically and was awaken after 1 hour. Once the crawler was able to access YouKu again, it resumed.

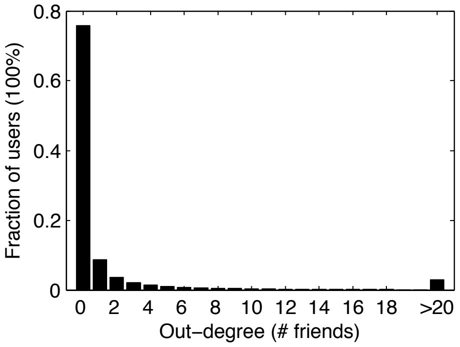

Fig. 1: Distribution of out-degree for individual users

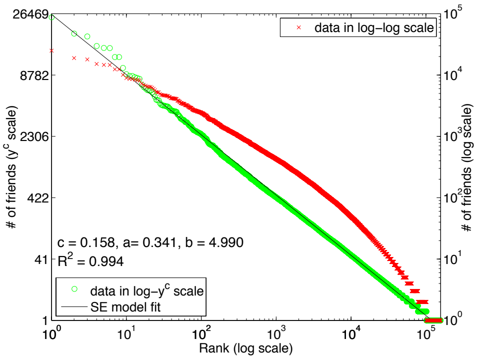

Fig. 2: Rank-ordering distribution of out-degree

There are two source of bias and limitation with our dataset. First, it does not contain users who have never uploaded nor faved any videos. However these users are not very active and cannot be a target for friendship recommendation. Second, The BFS crawling is biased toward videos that are related to many other videos [17]. However, the videos crawled by BFS process were used only for finding users and our work focuses on the social connections among users. Therefore, we believe that the bias does not affect the results.

IV. Analysis of YouKu Social Graph

A. Statistics of the Social Graph

The social relationship among users defines a directed graph. We define a user's *out-degree* as its friends number. Throughout this paper, we use "out-degree" and "number of friends" interchangeably. Figure 1 plots the distribution of out-degree for individual users. We stratify the users into groups according to their out-degrees. The users with out-degrees larger than 20 are gathered into 1 group. The figure shows that about 75% of users have no friends at all. We also observed that only 15% of the users are present in the largest Weakly Connected Component (WCC). Hence, the social connectivity of YouKu is even weaker than that of YouTube [13], in which the last percentage is 58%. can also be found that the fraction of users in a group decreases with the growth of the out-degree the group represents. We plot in figure 2 the out-degree CCDF in log-log scale. As can be seen, the distribution curve is not a straight line, meaning that the out-degree does not fit well to a power-law model. Instead, we find it can be well-fitted with a *Stretched Exponential* (SE) distribution. The complementary cumulative distribution function (CCDF) of the stretched exponential distribution [16] is given as

$$P(X \geq x) = e^{-(\frac{x}{x_0})^c} \tag{1}$$

where c is the stretching factor and x_0 a constant parameter. Now let's suppose we order N observed values x that follows an SE distribution in decreasing order. It is therefore expected that $P(X \geq x_i) = i/N$, where i $(1 \leq i \leq N)$ is the rank of the value x_i in the decreasing order ranking. Fitting SE distribution values we will have $\log(i/N) = -(\frac{x_i}{x_0})^c$. It will therefore be expected that the rank-ordered distribution follow the below curve:

$${x_i}^c = -a \log i + b \tag{2}$$

where $a = {x_0}^c$ and $b = {x_1}^c$, where x_1 is the largest observed value. Hence, if we plot the x_i^c *vs. its rank i in a loglog way we can expect to observe a straight line. In this paper, we use the method proposed by Guo* et al. *in [18] for fitting SE distribution information. The quality of the fitting is assessed as usual by the R^2 goodneed of fit value. We have also plotted in Figure 2 the ranked distribution plot in log-x^c scale. As can be see the fit is very good with a value of exponent $c = 0.158$ and $R^2 = 0.994$.*

We also computed the average shortest path length, the clustering coefficient and the reciprocity rate of the YouKu social network. The clustering coefficient for a node is defined as the ratio of the number of links that exist between its one-hop neighbors and the maximum number of links that could exist; a network clustering coefficient is the mean of nodes clustering coefficients. The reciprocity rate is the ratio of mutual friend pairs to all friend pairs. The measured average path length in the largest WCC is 4.25 and its clustering coefficient is 0.117. This shows that the WCC of the YouKu social network exhibits small-world properties, consistent with the results for YouTube [22]. The reciprocity rate is 38.8%, bigger than that of Twitter [19], but smaller than that of YouTube. The possible reason is that users in YouKu are more likely to take "celebrities" as friends than those in YouTube.

B. Correlation Analysis

We plot in Figure 3 the out-degree vs. the user popularity, measured as the sum of the number of views of all videos uploaded by this user. We bin the out-degree in log-scale and plot the median per bin in the solid line. It can be clearly found that user popularity and out-degree are almost linearly correlation. Besides the average user popularity against out-degree is always above the median, meaning that there are some outliers whose videos get views far more than expected.

Figure 4 plots the number of uploaded videos against the out-degree. As in Figure 3, we bin the out-degree in log-scale and plot the median per bin in the solid line. The number of uploaded videos grows with the out-degree before the out-degree reaches 100. Beyond this value, the number of uploaded videos fluctuates around 80. The above correlation results have shown the impacts of social connections on user popularity and activity.

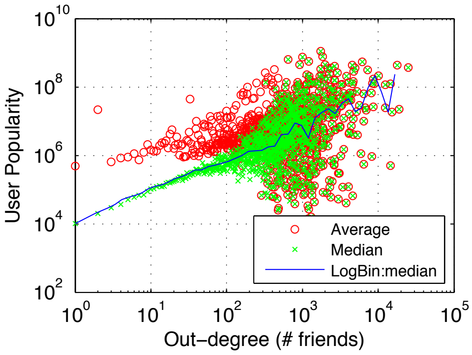

Fig. 3: Out-degree versus user popularity

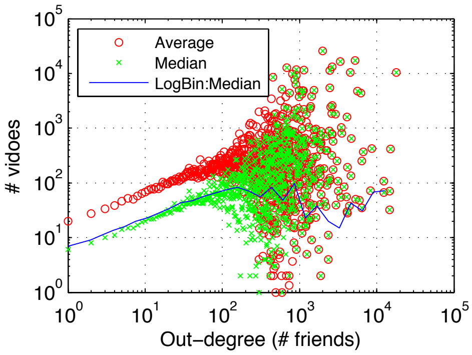

Fig. 4: Out-degree versus the number of uploaded videos

In summary, we have found the weak social connectivity and the importance of social connections. We thus aim to recommend friends to individual users to enhance it. To this end, we should first identify and measure the proximity or relevance between friends.

V. Interest Similarity Between Friends

Users act as both video producers and video consumers in UGC video systems. We derive a user's interests from both his uploads and favorites, which capture the user's interests with respect to both producers and consumers. Each video is associated with a tag provided by the uploader. The tag consists of up to 10 terms and represents the content of the video. A user's interests are formally represented by an *interest vector*. A typical element is (t_i, w_i), where t_i is a term from the tags of the user's uploaded videos and favorite videos, w_i is the weight for this term. There are various methods to compute the weight for a term. A good example is term frequency, *i.e.* the number of times that the term appears in the user's video tags. With the interest vectors, one can compute the interest similarity between any pair of users using similarity measures, *e.g.* cosine similarity.

In what follows, we first study the users' upload and favorite patterns, then analyze the video tags, followed by tag augmentation with a source of encyclopedia knowledge.

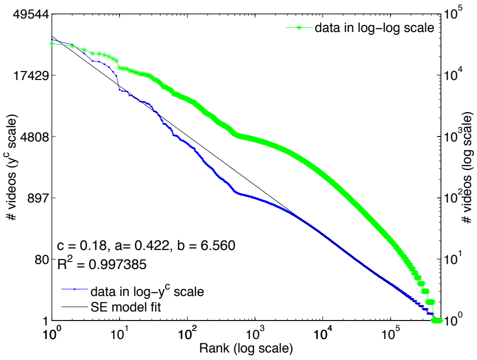

Fig. 5: Rank-ordering distribution of uploaded videos by individual users

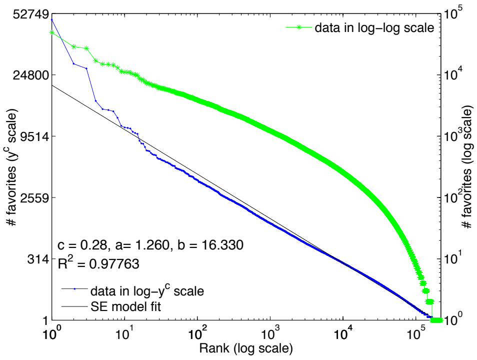

Fig. 6: Rank-ordering distribution of favorite videos by individual users

Finally, we analyze the interest similarity between friends using two different similarity measures.

A. Analysis of Upload and Favorite Patterns

We derive users' interests from their uploads and favorites. It should be noted that not all users equally act as both producers and consumers. For example, TV stations only act as video producers, while some are only interested in viewing videos. To gauge users' upload and favorite patterns, we show their rank-ordering distributions in Figure 5 and Figure 6, respectively. Both the upload and favorite behaviors follow stretched exponential distributions instead of power-law distributions, meaning that a small number of core users cannot dominate the system. The first several points in Figure 6 are higher than the stretched exponential model predicts due to the "King effect" [18].

We have found that the out-degree, user upload and favorite behaviors follow stretched exponential distributions. However, the three distributions have different stretched factors c. Guo *et al.* in [18] conjectured the stretched factor reflects the effort required to do something: the more effort that are required, the smaller c is. This conjecture can also be applied in our context. The distribution for favorite behavior has the largest c. In UGC video systems, if a user likes a video, it is easy to add a video to his favorite list. Users also prefer to add videos to favorite lists, since it enables users to easily find the videos that they like. It however requires a great effort to make a video clip and upload it to the system, resulting in a smaller stretched factor than that in favorite behavior. For out-degree, although the operation of making a friend is simple, users in current systems infrequently make friends with others as analyzed in Section IV. Users are unaware of the impacts of social connections and take the systems simply as a video pool. Besides, a friendship from user X to Y between two users is established only when X has sent a request to Y and the request is approved by Y. Thus, making a friend is more complicated, resulting the smallest stretched factor.

In order to quantify the correlation between upload and favorite behavior, we use the Spearman's rank correlation coefficient (Spearman's ρ) [6]. Spearman's ρ is defined as

$$\rho = 1 - \frac{6\sum(x_i - y_i)^2}{n(n^2-1)} \tag{3}$$

where x_i and y_i are the ranks of users according to the number of uploads and the number of favorites for a n-user system. It is a non-parametric measure of correlation, which shows how well an arbitrary monotonic function could describe the relationship between two variables. The coefficient lies in between [-1,1], where "1" indicates perfect positive correlation and "-1" means perfect negative correlation.

TABLE I: Spearman's Rank Correlation coefficients

Correlation	All	Top 10%	top1%
upload vs favorite	0.37	0.14	0.045

Table I lists the Spearman's correlation coefficients between upload behavior and favorite behavior. To avoid the tied ranks among the users who have uploaded the least number of videos [10], we also considered top users based on the number of uploads. The coefficients are small, especially for the top users, indicating the low correlation between two metrics. This is because two metrics reflect two different patterns of users, *i.e.* producer pattern and consumer pattern. We thus need to capture user interests from both aspects.

B. Analysis of Video Tag

Video tags are generated by uploaders and under no control of the systems. In fact, the tags are short sentences (or even one word). We analyze the video tag length, *i.e.* the number of terms in a tag, in Figure 7. Videos are clustered into four groups based on the number of views. Two observations are notable. First, in total, 90% of the videos are with tags less than 5 terms, and as many as 50% video tags only contains one word. Second, popular videos tend to have longer tags.

Video tags are also *ambiguous* and *heterogenous* [5]. For a MV with the tag as "Beat it", without a knowledge that this is a song of Michael Jackson, one cannot relate this video to Michael Jackson's other MVs. In addition, users may tag videos in different granularity. For example, for a MV of Michael Jackson's song "Beat it", one user may tag it as the song name "Beat it", while another may tag it as the

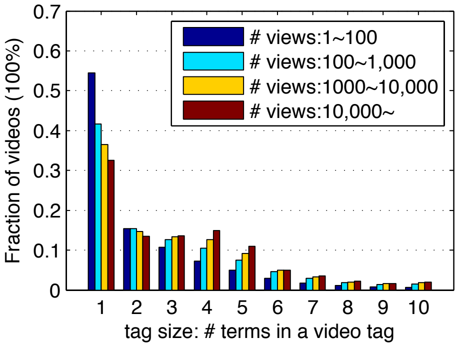

Fig. 7: Distribution of tag length

singer's name "Michael Jackson" or even the type of the video "Music".

The ambiguousness and heterogeneity of short tags bring a great challenge to measure the interest similarity. We should find a source of knowledge solves the challenge by augmenting video tags with semantic knowledge.

C. Tag Augmentation with Baidu Encyclopedia

We crawled Baidu Encyclopedia [1] to get a source of semantic knowledge. Baidu Encyclopedia, also called Baidu Baike, is a Chinese language web-based encyclopedia. It is a self-evolving encyclopedia that covers the most up-to-date human knowledge of concepts in Chinese. At that time, it contains about 4 million entries. Each entry consists of a lexical item, an article describing the lexical item, and the open categories (similar to folksonomy) the item belongs to. The articles are written and edited by registered users collaboratively, but reviewed by administrators before release. The categories of a lexical item do not necessarily have hierarchical ontology structure, although some do have. For example, the lexical item "Heal the world", the name of a Michael Jackson's song, is tagged with "Dangerous" (the album name), "Michael Jackson", "music" and "song" as the term's open categories.

Each entry has a web page, which contains the lexical item, article, the open categories and reference links. Entries are numerically assigned global unique ID, starting from 1. With an ID of an entry, we can generate the URL of the web page for that entry. This enables us to easily crawl all available entries. The crawling process lasted about one, starting at the middle of Nov., 2011. All 4 million entries were crawled and their categories were extracted.

Figure 8 shows the distribution of open categories for all the lexical item. Although the number of categories a lexical item belongs is limited to 5 by Baidu Encyclopedia, we still find some exceptions (less than 1%). The percentage of the lexical items with one category is a little higher than others, which are close to uniform distribution. This is totally different from the video tag length distribution in Figure 7, although both of them are provided by users. There are two possible reasons. First, the open categories in Baidu Encyclopedia can be further edited by others after its first publish. Second, it seems that the administrators of Baidu Encyclopedia review the article more carefully since it is a resource of knowledge.

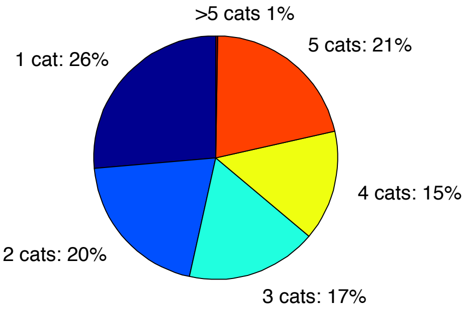

Fig. 8: # open categories for terms in Baidu Encyclopedia

In total, there are 585,066 unique open categories. We count the number of lexical items belonging to each category and plot the rank-ordering distribution in Figure 9 in log-log scale. The Zip'f distribution with coefficient $\alpha = 1.1$ well fits the empirical data. This indicates that most of the lexical items belong to a very small number of open categories.

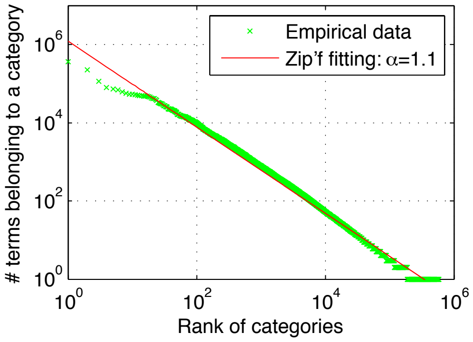

Fig. 9: Rank-ordering distribution of the number of terms of open categories

With such a great source of knowledge, one can add semantics to video tags to augment them. We in this paper use a simple method. For each term in a video tag, we lookup the knowledge base that we obtained from Baidu Encyclopedia. If there is a lexical item which is the same as the term, we take the open categories of that lexical item to augment the term. In particular, a term is augmented to a term collection which contains the term itself and the open categories. The augmented video tag is then the concatenation of the term collections. For example, suppose that a video v's tag consists of term A and B, A's open categories include C and D, B only belongs to one category D, then after augmentation, the video tag is $\{A, C, D, B, D\}$.

However, there are two exceptions. The first exception is about *polysemous* entries. For example, a lexical item (or term) "love" refers to 16 different meanings in Baidu Encyclopedia, such as an English word describing emotion, the name of a NBA player, a song of Beyond rock band and so on. Different meanings correspond to different open categories. If a video's tag contains the term “love”, without investigating the video content, we have no idea about its exact meaning and thus cannot augment it. Aiming at accurate augmentation of video tags, we filtered out all the polysemous entries, which amount to 1 million. The second exception is *synonym*, *i.e.* different lexical items refer to the same meaning, or even the same thing. For example, in Baidu Encyclopedia, the lexical item “MJ” and “Michael Jackson” both refer to the pop music star “Michael Jackson”. Synonymous lexical items have different entry ID, but are referred to the same web page and have the same open categories. For a lexical item that has synomymous items, besides adding open categories for augmentation, we add all the synonymous items. For example, if a video tag contains “MJ”, after augmentation, the tag would contain “MJ”, “Michael Jackson”, and the open categories of “Michael Jackson”. Out of 4 million entries, there are about 600K synonymous entries.

The augmentation adds semantics to video tags. For example, before augmentation, it is impossible to relate the MV with tag “Heal the World” to the MV with tag “Black or White”, although both of them are in Michael Jackson’s album “Dangerous”. Following the above procedure, both video tags after augmentation would include “Dangerous” and “Michael Jackson”, and thus they are related now. One can also use more sophisticated augmentation methods such as LDA (Latent Dirichlet Allocation)-based augmentation as in [5]. However, since our simple method already adds useful semantics, we believe it is sufficient in our context.

Not every term in video tags has a match in Baidu Encyclopedia. We define *tag augmentation rate (tar)* as follows to gauge the effect of augmentation.

$$tar = 1 - \frac{|tag_{ori}|}{|tag_{aug}|} \tag{4}$$

where tag_{aug} and tag_{ori} are the video tag after augmentation and before augmentation. This metric captures the fraction of new terms added in augmented tags. Figure 10 plots the results. About 30% video tags are not augmented. Most of these video tags only contain 1 single user-generated word, which could not be found in Baidu Encyclopedia. There are as many as 60% of video tags containing more than 60% new terms after augmentation. Thus, following our augmentation method, a large portion of video tags are augmented with semantic knowledge.

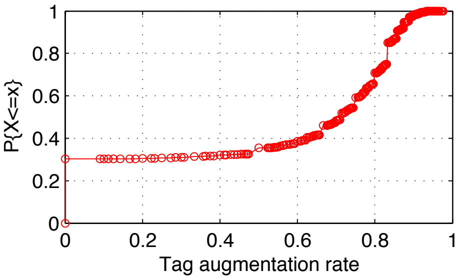

Fig. 10: Video tag augmentation rate

D. Analysis of Interest Similarity

A user’s interest vector contains the $(term, weight)$ pairs, where terms are from the augmented tags of the user’s uploaded and favorite videos. By considering both uploads and favorites, we capture the user’s interests as both producers and consumers in UGC video systems. In a user’s interest vector, the term frequency, *i.e.* the number of times a term appears in the user’s uploaded and favorite videos, may or may not be evenly distribution. If the distribution is skewed and a subset of terms appear more than others, the user’s interests are then concentrated. Otherwise, the user falls in a large range of diverse interests. To gauge how even the distribution of term frequency is, we compute disparity [19] of term frequency.

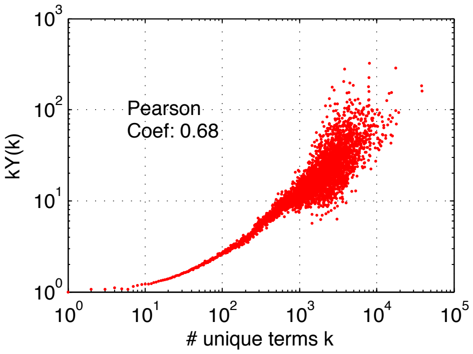

Fig. 11: Disparity of term frequency

We define $|r_{ij}|$ as the frequency of term j in a user i’s videos (include both uploaded videos and favorite videos). Then, we define $Y(k, i)$ as follows:

$$Y(k,i) = \sum_{j=l}^{k} \left[\frac{|r_{ij}|}{\sum_{l=1}^{k} |r_{ij}|} \right]^2 \tag{5}$$

where k is the number of (unique) terms. Let $Y(k)$ be the average of $Y(k, i)$ for all users having k unique terms. If the term frequency distritbutes evenly, then $kY(k) \sim 1$. If a subset of terms appear more frequently than others, then $kY(k) \sim k$. Figure 11 depicts $kY(k)$ against k in log-log scale. The linear correlation is notable. We quantify the correlation with Pearson correlation coefficient, a linear correlation measurement and defined as follows for two variables

$$\rho_{XY} = \frac{E((X - E(X))(Y - E(Y)))}{\sqrt{D(X)}\sqrt{D(Y)}} \tag{6}$$

where $E(X)$ is the expectation of X and $D(X)$ is the deviation of X. The Pearson correlation coefficient is 0.68, meaning the high linear correlation between $kY(k)$ and k. The results indicate the concentration or locality of users’ interests.

With the interest vector, one can estimate the interest similarity between any pair of users who have non-empty vectors. A typical estimation method in this context is *cosine similarity*, where the similarity between two users u_1 and u_2 is the cosine of the their vector representations $\overrightarrow{V}(u_1)$ and $\overrightarrow{V}(u_2)$. Here, the weight of term t in user u’s interest vector is assigned in the form of $w_{t,u} = 1 + \log f_{t,u}$, where $f_{t,u}$ is the number of times that t appears in user u's videos (include both uploaded and favorite videos). Formally, the cosine similarity between two users u_1 and u_2 is as follows.

$$sim(u_1, u_2) = \frac{\overrightarrow{V}(u_1) \cdot \overrightarrow{V}(u_2)}{|\overrightarrow{V}(u_1)||\overrightarrow{V}(u_2)|} = \sum_{t \in u_1, u_2} \bar{w}_{t,u_1} \times \bar{w}_{t,u_2} \tag{7}$$

where t is a term appearing in both users' video tags, $\bar{w}_{t,u}$ is the normalized weight of t. The cosine similarity lies in between $[0, 1]$. The higher the similarity is, the more common interests two users share.

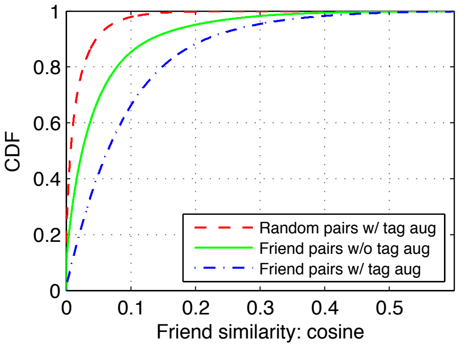

Fig. 12: Cosine similarity between friends

We use cosine similarity to quantify the interest similarity between friends. The distribution of friend similarity is shown in the dash-dotted line in Figure 12. It can be found that friends share considerably common interests: the similarity values of more than 35% of the friend pairs are bigger than 0.1, and a few (5%) pairs even have similarity values bigger than 0.3.

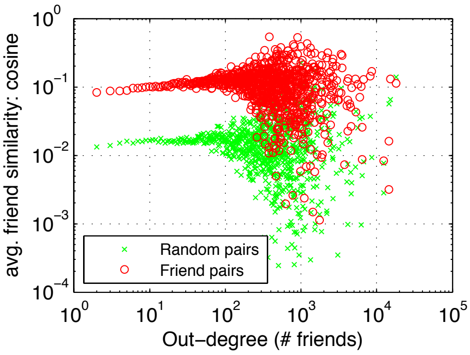

Fig. 13: Scatter distribution of cosine similarity of YouKu friend pairs and random friend pairs

To better assess the similarity between friends, we leverages hypothesis testing. The null hypothesis would be a scenario where users select friends randomly. The number of random friend pairs is the same as that of the dataset. We compute the cosine similarity between random friend pairs. If the observed similarity values substantially exceed the random expectation, we can safely say that friends indeed share common interests. The results are shown in Figure 12 and in Figure 13. The difference between the random model and the observed similarity pattern is notable. The observed similarity values between friends are one order of magnitude larger than that in the random model. The percentage of friend pairs with similarity values bigger than 0.1 is only 2%. In summary, we can safely conclude that users are likely to make friends with those sharing similar interests.

In order to demonstrate the effects of the augmentation, we also compute the cosine similarity values between friends *without* video tag augmentation, The similarity distribution is also plotted in Figure 12 in dashed line. Although we can still identify the high similarity values, it is not as obviously as that with tag augmentation due to the ambiguous and heterogeneity of video tags. The tag augmentation add semantic knowledge to tags and thus facilitates exploiting of user interests.

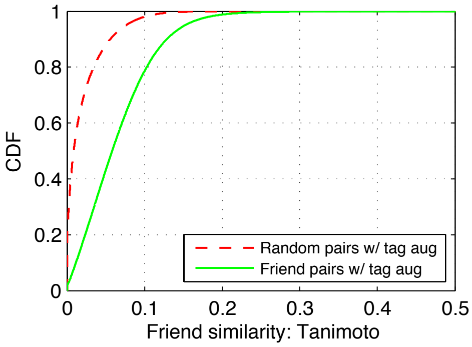

Fig. 14: Tanimoto similarity between friends

Besides cosine similarity, we also use Tanimoto similarity [27] to quantify the interest similarity between friends. In this context, if a term t appears at least once in the tags of a user u's videos, the term weight in u' interest vector is 1, *i.e.* $w_{t,u} = 1$. The Tanimoto similarity between user u_1 and u_2 is defined as follows

$$t\_sim(u_1, u_2) = \frac{R}{|\overrightarrow{V}(u_1)| + |\overrightarrow{V}(u_2)| - R} \tag{8}$$

where R is the number of terms appearing in both users' video tags at least once. The similarity value lies in between [0,1]. The higher the similarity value is, the more interests two users share. Tanimoto similarity differs from cosine similarity in that it ignores term frequency.

Figure 14 plots the distribution of Tanimoto similarity of YouKu friend pairs and random friend pairs. The random model is used for hypothesis testing as previous. Again, the difference between observed pattern and the similarity pattern in random model is obvious.

In summary, using two different similarity measures, we have found consistently that friends share common interests to a great extent. Thus, we should find users with similar interests as potential friends for individual users.

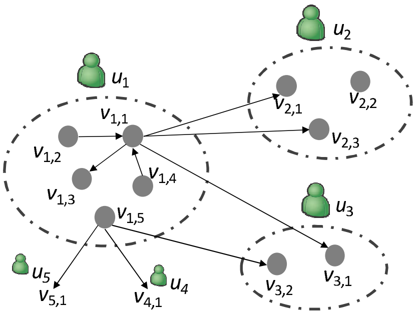

Fig. 15: An example of related videos

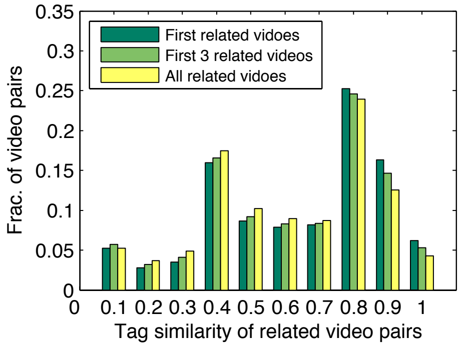

Fig. 16: Tanimoto similarity between video tags of related video pairs

VI. Friend Recommendation for UGC Video Systems

An intuitive way to find the users with similar interests for individual users is through users' friend connections. For example, the friends' friends of a user are taken as the new friends to recommend. Since friends share similar interests, this method is reasonable. However, in practice it is not applicable to UGC systems for two reasons. First, as we analyzed in Section III, a large portion of users have no friends at all. Second, users may protect their friendship from outside for privacy concerns [5].

The uploads and favorites are always publicly available in UGC video systems. Thus, we prefer to search possible friends through users' uploads and favorites.

A. Searching Possible Friends

If a user u_1 adds another user u_2's videos as favorites, as a consumer, u_1 is deemed to have similar interests with u_2. However, favorites do not capture users' interests as video producers. To find potential friends for individual users as video producers, we take advantages of related video links, the most unique feature toward the success of UGC video systems [12].

Each video in YouKu is assigned 9 related videos by the system. Figure 15 gives an example of video relationship. Two related videos are linked with a directed connection. A video may either take videos uploaded by the same user or by others as related videos. For example, the video $v_{1,1}$'s related videos include $v_{1,3}$, uploaded by the same user u_1, and $v_{3,1}$, uploaded by another user u_3.

It is expected that systems assign related videos based on content similarity. We quantify the similarity of related video pairs by applying the Tanimoto similarity (Eq. 8) on their augmented video tags. Figure 16 plots the distribution of similarity. Related videos are grouped into 3 categories according to their positions on the related video lists. More than 60% of the related videos have similarity values larger than 0.5, meaning a great similarity. Besides, although the related videos at the front positions in the related lists are more similar to the video, the differences are small.

The above results demonstrate that one can search friends through the related videos and favorites for recommendation to individual users. Before delving into the details, we define some notations. We refer to the user that will be recommended friends as *query user*. A *related user* for a query user is a user who shares at least one related video pairs with the query user. In Figure 15, both u_2 and u_3 are related users for u_1. A *favorite user* for a query user is a user who has at least one video in the query user's favorite list, *i.e.* liked by the query user. A *related user pair* consists of a query user and one of his related user, while a *favorite user pair* is formed by a query user and one of his favorite user.

We compute the cosine similarity of interests for all possible related user pairs and favorite user pairs in our dataset. The interest similarity distribution curves are plotted in Figure 17. Note that, as previous, users' interests include both their interests as producers and the interests as consumers.

The high similarity values are obvious for related user pairs and favorite user pairs. For example, the percentage of user pairs which have similarity values higher than 0.2 is more than 60%. A small fraction of user pairs even have similarity values as high as 0.8. Another observation is that the two distribution curves are very close to each other. The results collectively demonstrate that users found through related videos and favorites indeed have similar interests with query users. Thus, they can be recommended to query users.

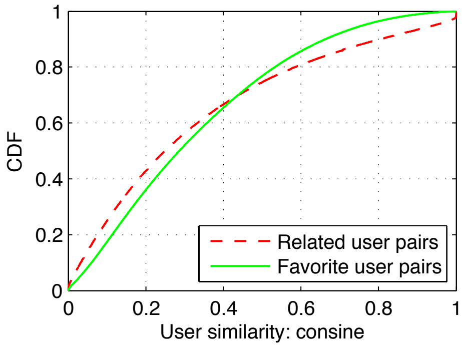

Fig. 17: Interest Similarity for related user pairs and and favorite user pairs

B. Friend Recommendation

One can directly recommend users found through related videos and favorites to query users as new friends. However, this is rather coarse. For more accurate recommendation, we leverage the cosine similarity and finely rank the users found according to their interest similarity values to the query user. Finally, the top users are selected to be recommended. This yields the `BasicFR` friend recommendation algorithm for the query user u. The pseudo-code is listed in Algorithm 1.

Algorithm 1 `BasicFR`: Basic Friend Recommendation

```
Input: User $u$'s uploaded video set $S(u)$ and favorite video set $F(u)$
Output: The friends recommended to $u$
1:  for each video $v \in S(u)$ do
2:      locate the related users of $u$ through the related video list of $v$ and insert them into the set $R(u)$
3:  end for
4:  for each video $v \in F(u)$ do
5:      insert the uploader of $v$ into the set $R(u)$
6:  end for
7:  for each user $u_i \in R(u)$ do
8:      get the uploaded video set $S(u_i)$ and favorite video set $F(u_i)$
9:      derive interest vector of $u_i$ from $S(u_i)$ and $F(u_i)$
10:     compute the cosine similarity between $u$ and $u_i$ using
11: end for
12: Rank the users in $R(u)$ according to their similarity values
13: recommend top $m$ users to the query user $u$ as friends
```

The `BasicFR` considers every user who may share similar interests with the query user. The user set $R(u)$ consists of all the related users and favorite users of u. We call this set of users as *potential friends* of u. The algorithm returns m users with highest similarity values as new friends to recommend, where m is a design parameter and can be tuned by system designers. While it is very effective to locate the most similar users, it may require to using a relative longer time. This is because for every potential friend, we need to get all their uploads and favorites, compute the cosine similarity values to the query user (line 7-11 in Algorithm 1).

Related user pairs share different numbers of related video pairs. For example, in Figure 15, the user u_4's videos appear in the related video lists of u_1's videos only once, while u_3's videos appear 3 times. In other words, the related user pair of u_1 and u_4 share 1 related video pair and the pair of u_1 and u_3 have 3 related video pairs. It is reasonable to assume that related user pairs which share a larger number of related videos are more likely to have similar interests. In Figure 15, u_3 are likely to be more similar to u_1 than u_4 in terms of interests on contents. Likewise, the users having more videos that the query user liked are likely to share common interests to a greater extent with the query user.

To justify the reasonableness of the assumption, for each user u in the dataset, we rank the potential friends based on cosine similarity, the number of related video pairs shared and the number of videos liked by u, respectively. We then compute the Spearman's rank correlation coefficient (Eq. 3) to quantify the correlations among different measures. Figure 18 shows the distribution of the coefficients between cosine similarity and the number of videos liked by u. Users are grouped based on the number of potential friends and the average value over all users in each group is computed. The correlation is moderately high (around 0.55), especially for the users having a small number (< 200) of potential friends. The correlation between cosine similarity and the number of related video pairs shared is around 0.52. The distribution is closely similar to that in Figure 18, and thus is omitted here to avoid duplication. These results have shown the reasonableness of the assumption that we just have made. Based on this, we propose a more efficient friend recommendation algorithm `RelatFR` listed in Algorithm 2.

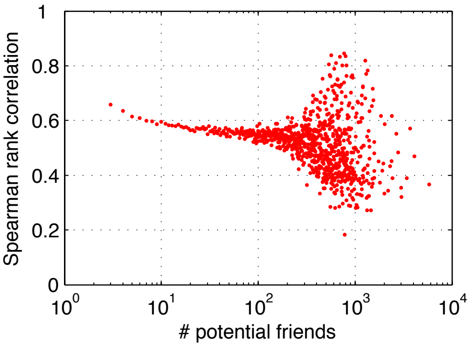

Fig. 18: Spearman's rank correlation between cosine similarity and the number of videos liked by u, for each user u's potential friends

Algorithm 2 `RelatFR`: Relationship-based Friend Recommendation

```
Input: User $u$'s uploaded video set $S(u)$ and favorite video set $F(u)$
Output: The friends recommended to $u$
for each video $v \in S(u)$ do
    for each related video of $v$ do
        get its uploader $u_i$ and insert $u_i$ into the set $R(u)$
        $RV[u_i] = RV[u_i] + 1$
    end for
end for
Rank the users in $R(u)$ according to $RV[\cdot]$
recommend top $m_1$ users to the query user $u$ as friends
for each video $v \in F(u)$ do
    get its uploader $u_i$ and insert $u_i$ into the set $T(u)$
    $FV[u_i] = FV[u_i] + 1$
end for
Rank the users in $T(u)$ according to $FV[\cdot]$
recommend top $m_2$ users to the query user $u$ as friends
```

The `RelatFR` algorithm recommend top m_1 related users and top m_2 favorite users to the query user, where m_1 and m_2 are design parameters. Compared with the `BasicFR` algorithm, the `RelatFR` algorithm no longer needs to get the uploads and favorites for every potential friend. It also no longer needs the computation of interest similarity. Thus, both the time and cost are greatly saved. Instead, the algorithm only requires the information of related video and favorite relationship, which is easy to obtain.

In both algorithms, the query user u's uploaded video set $S(u)$ and favorite video set $F(u)$ may be empty. If at least one of them is non-empty, the algorithms can still recommend friends to the query user. If both sets are empty, it is impossible to derive his interests on contents. For such users, we can learn the interests from his footprints in the systems, such as the videos he commented, the videos he voted. However, if a user leaves no footprint, one can only recommend the celebrities.

VII. Applying Friend Recommendation

We apply the friend recommendation algorithms on our YouKu dataset collected in Section III. Then, we analyze the updated social network graph and evaluate the effectiveness through one-hop video search.

A. Social Graph Analysis

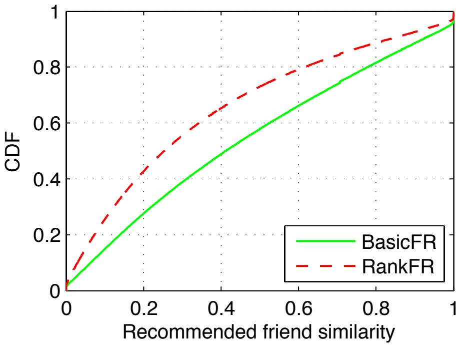

Fig. 19: Interest similarity between any pair of a query user and one of his recommended friend

We apply both algorithms proposed in the previous section on the dataset. Every user in the dataset is considered as a query user for friend recommendation. We recommend at most 10 friends to each query user. If the number of possible friends is even smaller than 10, then all the potential friends are recommended. In the BasicFR algorithm, m is set as 10, while in the RelatFR algorithm both m_1 and m_2 are set as 5.

To compare the two algorithms, we compute the cosine similarity between any pair of a query user and one of his recommended friend. The CDF (Cumulative Distribution Function) results are plotted in Figure 19. Although the friends recommended by the RelatFR share less similarity with query users than those recommended by the BasicFR, the differences are small. Another notable observation is that both algorithms are able to locate those users sharing great common interests with the query users. We thus here only use the RelatFR algorithm.

Users may or may not accept all the friends recommended by our algorithm as their new friends. In our experiments, we assume a user would accept a recommended friend as a new friend if their interest similarity value is higher than a threshold 0.2. This results in a updated social network of YouKu, which is denoted as "*YouKu w/ FR*" in the following figures. We analyze the graph properties of such a social network and compare it with other social networks, including the original YouKu social network without friend recommendation, YouTube and Flickr social networks measured by Mislove *et al.* in [22].

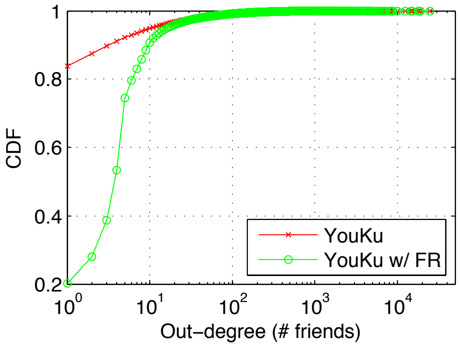

Fig. 20: Out-degree comparison

We first show the out-degree distribution for the YouKu social network and the updated social network in Figure 20. After friend recommendation, the fraction of users with no friends is decreased from 75% to 8%. Moreover, the percentage of the users presented in the largest WCC grows sharply from 15% to 95%. These results have demonstrated that the social connectivity is greatly enhanced.

Next, we analyze the small-world properties of the social networks. A network is considered as a small-world one if the clustering coefficient is as large as in regular graphs, but the average path length between nodes is as small as in random graphs [28]. The clustering coefficient, average path length, radius and diameter for different social networks are displayed in Table II. Radius and diameter are computed with the concept of node eccentricity, which is defined as the maximal shortest path length between a node and any other node. The radius of a graph is the minimum eccentricity over all nodes, and the diameter is the maximum eccentricity over all nodes. Average path length is the average of all shortest paths between any pair of nodes. The results are obtained by measuring the largest weakly connected component (WCC) of each network.

Three observations are notable. First, the YouKu social network exhibits similar properties as YouTube. Second, the friend recommendation increases the average out-degree and the clustering coefficient of the YouKu social network. Third, the average path length, radius and diameter of each network are low. In particular, the average path length is close or less than 6 in all networks, giving new evidences to the six-degrees of separation hypothesis for social networks [21]. The average path length in the YouKu social network slightly increases after friend recommendation due to the sharp growth of the WCC. In summary, the YouKu social network after friend recommendation exhibits more obvious small-world characteristics.

TABLE II: High-level statistics and Social graph measurements

Systems	Users crawled	Avg. Degree	Avg. Path Len.	Radius	Diameter	Clustering Coef.	Assort. Coef.
YouKu	0.63M	5.13	4.25	8	13	0.117	-0.074
YouKu w/ FR	0.63M	8.8	4.64	10	16	0.186	0.004
YouTube	1.16M	4.29	5.1	13	21	0.136	-0.033
Flickr	1.85M	12.24	5.67	13	27	0.313	0.179

We also study the degree correlations of different social networks by measuring the assortativity coefficient r, which is formally defined in [23]. The coefficient measures the probability that the nodes with similar degrees are connected. The coefficient r ranges from -1 to 1. A higher r indicates that nodes tend to connect with others of similar degree, while a negative coefficient means that nodes tend to connect with others of dissimilar degree. The last column of Table II shows the coefficient for different social networks. Again, YouKu is very similar to YouTube: both of the social network graphs have negative coefficients due to the celebrity-driven nature of such sites. After friend recommendation, the coefficient grows and becomes positive, meaning that high-degree users are highly connected to form the "core" of the small-world network. This enables the low radius and diameter of the updated social graph.

B. One-hop Video Search

Users would like to view the videos of their interests. We thus evaluate our friend recommendation algorithms by counting, how many similar videos individual users can find on their one-hop friends. We uniformly selected 100,000 users at random from the dataset. For each user, we selected at most 10 videos randomly from his uploaded videos to generate query terms. For each selected video v which is uploaded by user u, a query is generated with up to h terms from its augmented tag. For each query, we counted the number of matched videos on friends of u. A video is matched a query if its tag contains all query terms.

We measured the performance of YouKu, YouKu with friend recommendation and YouKu with random friend recommendation (denoted as "*YouKu w/ Random FR*"). All three networks contain the same number of users and videos. The difference lies in the social graph. In the "YouKu w/ Random FR", each user has the same number of friends as in the "YouKu w/ FR", but the new friends are selected at random. This random model is used as the scenario of null hypothesis.

Figure 21 shows the distribution of the number of matched videos. The query length h is set as 2. We filtered out the queries which have no matches in our dataset except the query originator. The "YouKu w/ FR" greatly outperforms others. The mean for "YouKu w/ FR" is 110, while for YouKu and "YouKu w/ Random FR", the mean values are only 21 and 24, respectively. Moreover, for 90% of the users in YouKu, we could only find less than 1 matched video on their friends. After friend recommendation, the percentage of such users is decreased to 31%. These results indicate that users can quickly find videos of their interests on their friends, which in turn greatly improves the QoE (Quality of Experience).

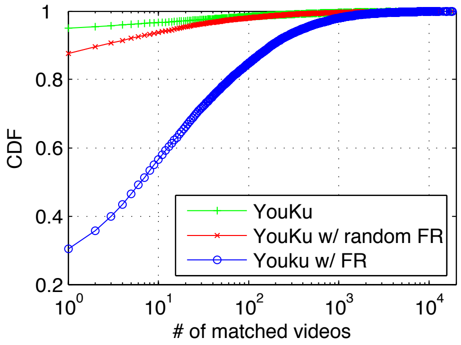

Fig. 21: Comparison of three graphs in terms of the number of matched videos on one-hop friends

VIII. Conclusion

We have made two technical contributories in this paper. First, through a measurement of YouKu, we have found that the social connectivity in UGC video systems is extremely weak and friends share common interests to a great extent. The interests of users capture their patterns as both video producers and consumers. Second, we propose two friend recommendation algorithms which find potential friends for query users through the related videos and favorites. And both algorithms only require to using publicly available information. Finally, we applied the algorithms on our dataset. The updated social network along with other three have been analyzed. The effectiveness is also demonstrated by one-hop video search experiments.

Acknowledgment

The authors would like to thank Rong Gu for the data collection. An early version of this work [20] was presented in the *proceedings of ACM/IEEE IWQoS 2010.*

References

[1] (2012) Baidu baike. [Online]. Available: http://baike.baidu.com
[2] (2012) New youtube statistics. [Online]. Available: http://searchenginewatch.com/article/2073962/New-YouTube-Statistics-48-Hours-of-Video-Uploaded-Per-Minute-3-Billion-Views-Per-Day
[3] (2012) Youku. [Online]. Available: http://www.youku.com
[4] (2012) Youtube. [Online]. Available: http://www.youtube.com
[5] G. A. A. Chaabane and M. A. Kaafar, "You are what you like! information leakage through users' interests," in *Proceedings of NDSS*, ser. NDSS '12, 2012.
[6] J. L. M. Arnold D Well, "Research design and statistical analysis, second edition," *Commun. ACM*, vol. 13, pp. 422–426, July 1970.
[7] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, and K. Ross, "Video interactions in online video social networks," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 5, pp. 30:1–30:25, November 2009.

[8] J.-I. Biel and D. Gatica-Perez, "Wearing a youtube hat: directors, comedians, gurus, and user aggregated behavior," in *Proceedings of the 17th ACM international conference on Multimedia*, 2009, pp. 833–836.

[9] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, pp. 422–426, July 1970.

[10] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi, "Measuring user influence in twitter: The million follower fallacy," in *Proceedings of international AAAI Conference on Weblogs and Social*, 2010.

[11] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. B. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system," in *ACM IMC*, 2007.

[12] X. Cheng, C. Dale, and J. Liu, "Statistics and social network of youtube videos," in *IEEE IWQoS*, 2008, pp. 229–238.

[13] X. Cheng, J. Liu, and C. Dale, "Understanding the characteristics of internet short video sharing: A youtube-based measurement study," *IEEE Transactions on Multimedia*, 2012.

[14] R. Crane and D. Sornette, "Robust dynamic classes revealed by measuring the response function of a social system," *PNAS*, vol. 105, pp. 15 649–15 653, Oct 2008.

[15] Y. Ding, Y. Du, Y. Hu, Z. Liu, L. Wang, K. Ross, and A. Ghose, "Broadcast yourself: understanding youtube uploaders," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, 2011, pp. 361–370.

[16] U. Frisch and D. Sornette, "Extreme deviations and applications," *Journal of Physics I France*, vol. 7, pp. 1155–1171, September 1997.

[17] M. Gjoka, M. Kurant, C. T. Butts, and A. P. Markopoulou, "Walking in facebook: A case study of unbiased sampling of osns," in *IEEE INFOCOM*, 2010, pp. 2498–2506.

[18] L. Guo, E. Tan, S. Chen, X. Zhang, and Y. E. Zhao, "Analyzing patterns of user content generation in online social networks," in *ACM KDD*, 2009.

[19] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?" in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 591–600.

[20] Z. Li, R. Gu, and G. Xie, "Measuring and enhancing the social connectivity of ugc video systems: a case study of youku," in *Proceedings of the Nineteenth International Workshop on Quality of Service*, 2011, pp. 36:1–36:9.

[21] S. Milgram, "The small world problem," *Psychology Today*, vol. 2, pp. 60–67, 1967.

[22] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 2007, pp. 29–42.

[23] M. E. J. Newman, "Assortative mixing in networks," *Phys. Rev. Lett.*, vol. 89, Oct 2002.

[24] J. C. Paolillo, "Structure and network in the youtube core," in *Proceedings of the Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, 2008, pp. 156–.

[25] M. Roth, A. Ben-David, D. Deutscher, G. Flysher, I. Horn, A. Leichtberg, N. Leiser, Y. Matias, and R. Merom, "Suggesting friends using the implicit social graph," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 233–242.

[26] H. H. Song, T. W. Cho, V. Dave, Y. Zhang, and L. Qiu, "Scalable proximity estimation and link prediction in online social networks," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, 2009, pp. 322–335.

[27] T. T. Tanimoto, "Ibm internal report," Tech. Rep., 1957.

[28] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, June 4 1998.

[29] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao, "User interactions in social networks and their implications," in *Proceedings of the 4th ACM European conference on Computer systems*, 2009, pp. 205–218.

[30] T. Yao, C.-W. Ngo, and T. Mei, "Context-based friend suggestion in online photo-sharing community," in *Proceedings of the 19th ACM international conference on Multimedia*, 2011, pp. 945–948.

[31] R. Zhou, S. Khemmarat, and L. Gao, "The impact of youtube recommendation system on video views," in *ACM IMC*, 2010, pp. 404–410.