Social Interaction Based Video Recommendation: Recommending YouTube Videos to Facebook Users

Bin Nie*, Honggang Zhang*, Yong Liu†
* Fordham University, Bronx, NY. Email: {bnie, hzhang44}@fordham.edu
‡ NYU Poly, Brooklyn, NY. Email: yongliu@poly.edu

***Abstract*—Online videos, e.g., YouTube videos, are important topics for social interactions among users of online social networking sites (OSN), e.g., Facebook. This opens up the possibility of exploiting video-related user social interaction information for better video recommendation. Towards this goal, we conduct a case study of recommending YouTube videos to Facebook users based on their social interactions. We first measure social interactions related to YouTube videos among Facebook users. We observe that the attention a video attracts on Facebook is not always well-aligned with its popularity on YouTube. Unpopular videos on YouTube can become popular on Facebook, while popular videos on YouTube often do not attract proportionally high attentions on Facebook. This finding motivates us to develop a simple top-k video recommendation algorithm that exploits user social interaction information to improve the recommendation accuracy for niche videos, that are globally unpopular, but highly relevant to a specific user or user group. Through experiments on the collected Facebook traces, we demonstrate that our recommendation algorithm significantly outperforms the YouTube-popularity based video recommendation algorithm as well as a collaborative filtering algorithm based on user similarities.**

I. Introduction

Recommender systems (RSs) address the information overload problem by suggesting to users items of their potential interests. Recent advances in recommender systems have shown that information derived from online social networks can be leveraged to improve recommendation accuracy [1]–[7]. However, there is still a lack of detailed empirical analysis of how much an RS can benefit from mining user interactions in real-world, general-purpose social networking sites. Meanwhile, videos are increasingly streamed online to users through various platforms, e.g., YouTube [8]. And online videos are hot topics of social interactions among users of OSNs, e.g., Facebook [9]. Therefore, online video is a perfect subject to study how social interaction information can be exploited to improve recommendation accuracy.

Towards this goal, we first conduct a measurement study to understand whether a video's popularity on OSNs (Facebook in our study) is well aligned with its popularity on video sharing websites (YouTube in our study). We first sample a subset of Facebook users, and then collect all videos that are shared/discussed among the sampled users. We further collect those videos' viewing/like statistics reported on YouTube for popularity comparison. Our empirical data have shown that video popularity distribution on Facebook has significant discrepancies from video popularity distribution on YouTube. Unpopular videos on YouTube can become popular on Facebook, and are shared and discussed heavily among friends. On the other hand, popular videos on YouTube often do not attract proportionally high attentions on Facebook, and as a result, their popularities get decreased.

This finding is rather encouraging for social network enhanced RSs. It is a well-known challenge for the traditional RSs to recommend *niche items*, that are globally unpopular, but highly relevant to a specific user or user group [10]. Our empirical finding suggests that those unpopular niche items for a user can potentially be identified by mining the social interactions, in the forms of rating, sharing, and posting, among the target user and his/her online social friends.

To verify this conjecture and see the potential recommendation accuracy improvement, we develop a simple top-k video recommendation algorithm, referred to as *SocialRecommend*, that exploits user social interaction information. *SocialRecommend* generates a list of k recommended videos for a target user by aggregating the videos liked by the target user's friends who either have similar video tastes to the target user (through mining the history data of 'liked videos'), or have high video-related interactions with the user. Through offline experiments over the collected Facebook trace, we are able to demonstrate that *SocialRecommend* can achieve better recommendation accuracy (in terms of top-k hit ratios and recall) than a pure similarity-based collaborative filtering [11] voting algorithm (without using any social interaction information). And *SocialRecommend* also significantly outperforms a naive YouTube-popularity based video recommendation algorithm. Different from the existing literature on applying social trust information to recommender system design [1]–[5], our algorithm exploits a much richer set of social interaction information from a real-world, general-purpose online social network. Our preliminary study demonstrates that, with the rich social interaction data, even simple recommendation algorithm can achieve significant recommendation accuracy improvement.

The rest of the paper is organized as follows. Section II describes our data collection process. Section III presents our empirical data to compare the popularity of a video on YouTube and on Facebook. Section IV presents our case study of applying users' social interaction to top-k video recommendation design. The paper concludes in Section V.

II. Data Collection from Facebook and YouTube

We first collect a subset of Facebook users via a random-walk based sampling. Then we further crawl the users who had video related activities with the sampled users. This process gives us a collection of users, and for those users, we collect all their friend relationships (if publicly available), and

978-1-4799-3088-3/14/$31.00

their mutual online interaction activities (if publicly available) including posts, tags, etc. Then we collect the information of all those videos that were shared, posted, or commented by users in this collection during a sampling period. Among those videos, we identify a subset of videos that are shared from YouTube [8]. Our study is based on these sets of users and videos.

A. Initial Random User Sampling

We first randomly select six Facebook users as root users. From each root user, we initiate a random walk in the Facebook OSN, following a neighbor-limited breadth-first search process. Specifically, from a root user u,

1) We first collect all friends of u and randomly select three of them, denoted by u_{f1}, u_{f2}, and u_{f3}. We say these three users are level-1 friends of user u, found at depth-level 1.
2) At depth-level 2, we then collect all friends of u_{f1} and randomly select three of them as u_{f1}'s sampled friends. Similarly we collect u_{f2}'s and u_{f3}'s sampled friends.
3) Continue the above random sampling steps till depth-level 6.

Note that for each user at each depth-level, we aim to find three newly sampled users who have never been found before. Ideally, this sampling process enables us to collect about 1093 sampled users starting from each root user ($\sum_{d=0}^{6} 3^d = 1093$). However, when we check a user's friend list, we might not always be able to find 3 friends that have not been selected before, thus, the actual number of sampled users from a root user might be smaller than 1093. Eventually we have identified $6,466$ users. Let U_{orig} denote the set of those users.

B. Glean Final User Sets and Video Sets

For all users in U_{orig}, we first identify all those videos published by them (including video posts and some swf posts) or shared by them (via URLs) between April 18, 2013 and May 30, 2013 (42 days in total). Let V_{orig} denote the set of those videos. This 42-days time period is referred to as the *sampling time interval*.

Then, we identify all users that have interactions (that are related to those videos) with the users in U_{orig} during the sampling time interval. Those newly added users might have posted their comments to, or liked or be tagged in the videos in V_{orig}. Adding those new users gives us an expanded user set, denoted by U_{all}. Note that $|U_{all}| = 32,209$ users. We have collected all the friend links (there are $111,486$ links) among them. Let E_{all} denote the set of those links. The social network formed by users in U_{all} and the friend links among them is referred to as $G_{all} = (U_{all}, E_{all})$. For all those users in U_{all}, we have collected their public profiles and their posts, together with the comments, likes, tags that were updated online in the sampling time period. There are mainly 7 different types of Facebook posts: *status, video, swf, photo, link, checkin, offer*. Throughout the rest of the paper, this network is referred to as *the sampled Facebook network* or *Facebook network* in short.

Then we collect all those videos that were shared or uploaded by users in U_{all} during the sampling time interval, and let V_{all} denote the set of those videos, and let V_y denote the set of all YouTube videos out of the videos in V_{all} (note that $V_y \subseteq V_{all}$). Note that $|V_{all}| = 294,826$, and the total number of YouTube videos (with unique links) is $134,151$.

We further derive a subset U_y of users in U_{all} who had activities (on Facebook) related to the videos in V_y during our sample interval. We use E_y to denote the set of all friend links among users in U_y. Then we have a sub-network of G_{all}, denoted by $G_y = (U_y, E_y)$, where $|U_y| = 19,609$ and $|E_y| = 101,176$. Note that $E_y \subseteq E_{all}$ and $U_y \subseteq U_{all}$. Each user in G_y has at least one activity related to a YouTube video.

The total numbers of collected posts by users in sets U_{all} and U_y are presented in Table I. The numbers of comments and likes that are related to those posts by users in U_{all} are $1,083,853$ and $9,558,767$ respectively. And the numbers of comments and likes related to those posts in U_y are $278,149$ and $543,633$ respectively.

If two users do not identify themselves as friends to each other on Facebook, but they have at least one common friend identified on Facebook, then we say that these two users have a two-hop friend relationship. We have further identified that there are $4,512,762$ two-hop friend relationship for users in U_{all}, and $238,552$ two-hop friend relationship in U_y.

TABLE I
COLLECTED POSTS BY USERS IN SETS U_{all} AND U_y.

Post Type	Number of Posts by U_{all}	Number of Posts by U_y
status	$366,273$	$246,076$
video	$294,826$	$269,309$
swf	$52,847$	$51,932$
photo	$1,150,131$	$1,038,893$
link	$630,183$	$515,365$
checkin	2	1
offer	34	28
Total	$2,494,296$	$2,121,604$

The degree distribution of the sampled Facebook network $G_{all} = (U_{all}, E_{all})$ is shown in Figure 1. We observe that the complementary cumulative distribution function (CCDF) of the degree distribution roughly follows a power-law curve with a drop-off at the tail part, which is consistent with what had been observed in the past empirical studies of OSNs.

III. VIDEO POPULARITY AND SOCIAL INTERACTION

A. General Statistics

Facebook users can upload their own videos or share videos from other websites (e.g., YouTube, Twitter) on Facebook. Among all users in our sampled Facebook network $G_{all} = (U_{all}, E_{all})$, about 4% of them had uploaded or shared videos, and among those user, 81% of them had shared videos from YouTube. The average number of videos uploaded or shared (including videos from Facebook and other sources) per user is 17.033. And the average number of videos shared from YouTube per user is 16.430. Figure 2 shows the empirical complementary cumulative distribution function (CCDF) of the count of all videos and the YouTube videos per user. As a comparison, we use the averages of the set of all videos and the set of YouTube videos to generate two exponential distributions respectively, and we plot them in Figure 2 as well. Comparing the empirical CCDFs against their corresponding exponential CCDFs shows the long tail nature of the two empirical distributions. We observe that the majority of users only upload or share very few videos, whereas a small number of users upload or share a large amount of videos.

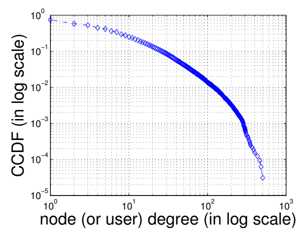

Fig. 1. Empirical CCDF of the degree distribution of the sampled Facebook network.

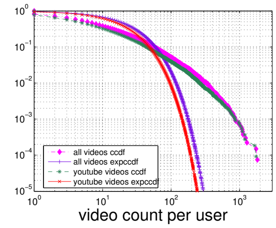

Fig. 2. Empirical CCDF of the numbers of all videos and YouTube videos shared/uploaded per user (in set U_{all}).

B. Videos and User Interactions

We next investigate the relationship between the total number of video-related interactions between two users and the numbers of videos shared or uploaded by those two users, on the sampled Facebook network G_{all}. Note that Facebook users are able to comment on or like a video, or to be tagged for a video in Facebook posts. We call these activities *video-related interaction* between users. The following example illustrates how to calculate video-related interactions. Suppose there are two users u_a and u_b. User u_a uploaded 2 videos, v_{a1} and v_{a2}. User u_b commented on, liked, or was tagged to v_{a1} 4 times, and v_{a2} 2 times. User u_b uploaded 1 video, v_{b1}. User u_a commented on, liked, or was tagged to v_{b1} 5 times. In this case, the number of total video-related interactions between u_a and u_b is $11 = 4 + 2 + 5$.

The scatter plot shown in Figure 3 illustrates that there is a correlation between video-related interactions between two users and their total number of uploaded/shared videos. In the figure, x-axis shows the video-related number of interactions between a pair of users, and y-axis shows the total number of videos shared or uploaded by the two users. The Pearson coefficient of this correlation is 0.125, and statistically significant (P-value $\ll 0.01$). We see that the more videos the two users upload or share, the more video-related interactions between them. We have observed similar correlation between YouTube-video-related interactions and the total number of YouTube videos shared or uploaded by two users.

Similarly, we also calculate other types of interactions between two users. There are seven different types of posts in Facebook, thus, we calculate seven types of interactions for each user pair. More specifically, if two users both commented, liked, or were tagged in a common post, we say that these two users have *one interaction*. We divide interactions according to the type of post. Then, we can sum up all those seven interaction counts to get the number of all types of interactions between two users.

We next compare all types of interactions and video-related interactions between any two users in set U_y, shown in Figure 4. We observe that the more interactions between two users, the more video-related interactions between them. The correlation between these two kinds of interactions is statistically significant (P-value $\ll 0.01$) and strong (Pearson coefficient is 0.674). Among all user pairs who have different types of interactions, the percentage of those pairs who have at least one video-related interactions is 56.2%.

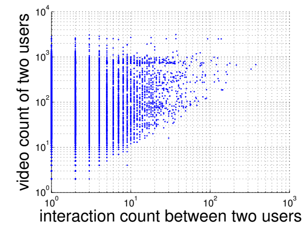

Fig. 3. The total number of uploaded/shared videos of two users vs. the number of video-related interactions between them.

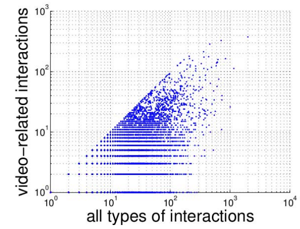

Fig. 4. The number of video-related interactions between two users vs. all types of interactions between them.

C. Video popularity on Facebook and YouTube

One might expect that a video that is more popular on YouTube is also more popular on Facebook, which however is not true in our data. We find that less popular videos on YouTube can become popular on Facebook, and are shared and discussed heavily among friends. On the other hand, popular videos on YouTube often do not attract proportionally high attentions on Facebook, and as a result, their popularities get decreased. One explanation can be that discussing a very popular YouTube video on Facebook is probably not too *cool* from a user's point of view. This finding is rather encouraging for social network enhanced Recommender Systems (RSs). It is a well-known challenge for the traditional RSs to recommend *niche items*, that are globally unpopular, but highly relevant to a specific user or user group [10]. Our empirical finding suggests that those unpopular niche items for a user can potentially be identified by mining the social interactions, in the forms of rating, sharing, and posting, among the target user and other online users.

We study video set V_y and network $G_y = (U_y, E_y)$ in this section. For a video v_i in set V_y, let c_i denote the total number of sharing, comments, likes, tags of video v_i. Then, the popularity of video v_i on Facebook is calculated as $F_i = c_i / \sum_{j=1}^{n} c_j$, where $n = |V_y|$. We can also measure the popularity of video v_i according to its watch count and like count on YouTube. Let w_i denote its watch count on YouTube, and l_i denote its like count on YouTube. We define the popularity of v_i in terms of watch count as $W_i = w_i / \sum_{j=1}^{n} w_j$, and its popularity in terms of like count as $L_i = l_i / \sum_{j=1}^{n} l_j$.

We next present scatter plots in Figures 5 and 6 to show the popularity differences of all those collected videos on Facebook and YouTube. These two figures show that there is no strong correlation between F_i and L_i (Pearson coefficient 0.03), nor between F_i and W_i (Pearson coefficient 0.02).

Furthermore, we divide videos in V_y into 60 groups $G_f^1, G_f^2, ..., G_f^{60}$ according to their popularity values on Facebook. We make sure each group has at least 100 videos. If $i < j$, then all videos in group G_f^i are *no more popular* than all videos in group G_f^j. We then plot in Figure 7

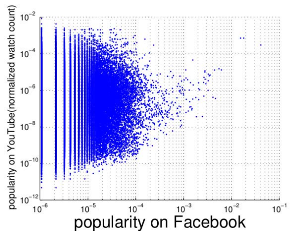

Fig. 5. F_i (x-axis) on Facebook vs. W_i (y-axis) on YouTube.

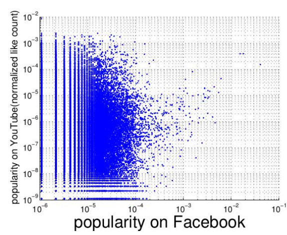

Fig. 6. F_i (x-axis) on Facebook vs. L_i (y-axis) on YouTube.

the average YouTube popularity values of each group (in terms of like, watch, and dislike count). Figure 7 shows that the average YouTube popularity value does *not* have any functional relationship with Facebook popularity groups.

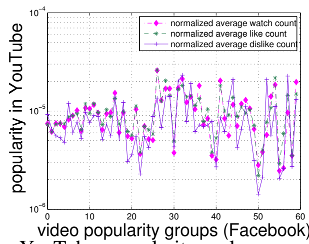

Fig. 7. Average YouTube popularity value across different Facebook popularity groups (those groups sorted in increasing order of F_i).

We next present a very interesting finding that a very popular video on YouTube is not necessarily popular on Facebook, and on the other hand, a video that is not very popular on YouTube can become quite popular on Facebook, due to social interaction.

We first rank all videos in V_y in terms of their like counts on YouTube. Then we select the videos that are ranked among the top 100 on YouTube and plot their popularity values on Facebook in Figure 8. Note that there might be multiple videos with the same rank on YouTube, so there might be multiple Facebook popularity values for each YouTube rank (i.e., x-axis value). In Figure 9, we plot the Facebook popularity values of the videos that are ranked among the lowest 100 on YouTube in terms of their like counts.

It is interesting to see from Figure 8 that those top most popular videos on YouTube are significantly less popular on Facebook, whereas Figure 9 shows that those least popular videos on YouTube become significantly more popular on Facebook. Similar patterns exist when the videos are ranked by their YouTube watch counts.

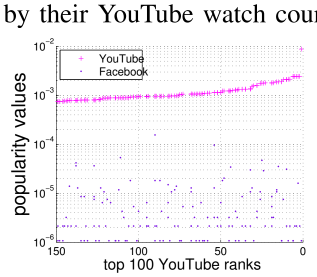

Fig. 8. The blue dots shows the Facebook popularities of those videos ranked as the top 100 in terms of their *like* counts on YouTube.

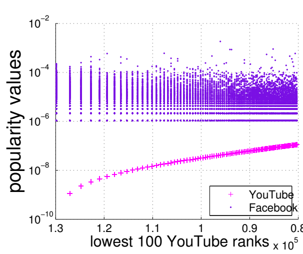

Fig. 9. The blue dots shows the Facebook popularities of those videos ranked as the lowest 100 in terms of their *like* counts on YouTube.

Furthermore, for those videos with the same YouTube popularity rank and the rank being one of the lowest 100 ranks on YouTube (in terms of their like counts), we plot in Figure 10 their average percentile rank in terms of their popularity values on Facebook. A video that has percentile rank x is more popular than or equally as popular as $x\%$ of videos in our video set V_y. We see from this figure that on average, the videos that have the lowest 100 ranks on YouTube can achieve about 50 percentile ranks on Facebook, contrasting sharply with their ranks on YouTube.

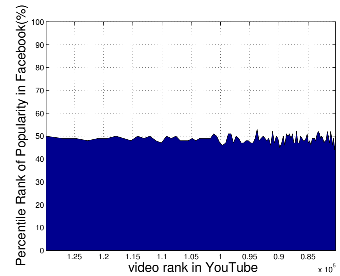

Fig. 10. Facebook popularity *percentile ranks* of the videos with ranks among the lowest 100 ranks on YouTube.

Finally, to get an overall picture of popularity change, we divide videos into 50 groups G_y^k ($k = 1, 2, ..., 50$) in an *increasing* order of their like counts on YouTube (i.e., if $i > k$, all videos in group G_y^i are *more or equally popular* than all videos in group G_y^k on YouTube). For each video $j \in V_y$, we calculate its popularity increases on Facebook from YouTube as $inc_j^L = F_j - L_j$ (we use like count L_j of video j). We calculate the average popularity increase of the videos in each group $G_y^k, \forall k$, and plot in Figures 11, 12, and 13 those average increases. We can see that on average, a less popular video on YouTube (in $G_y^1, G_y^2, ..., G_y^{45}$) enjoys popularity increase on Facebook (compared against its popularity on YouTube). However, note that the amount of popularity increase of a video does not appear proportional or inversely proportional to its popularity level on YouTube. On the other hand, a more popular video on YouTube (in groups $G_y^{46}, G_y^{47}, ..., G_y^{50}$) becomes less popular on Facebook on average. These results are consistent with what we observe from Figures 8 and 9. We observe similar patterns when a video's YouTube popularity is measured in terms of watch count.

The findings in this section imply that *mining the social interactions (on a social networking site like Facebook) might potentially help to identify and significantly improve the recommendation of those niche videos, that are globally unpopular (on a video sharing site like YouTube), but highly relevant to a specific user or user group.*

D. Similarity of Interest among Facebook Users

We are further interested in the relationship between users' similarities and their social distances (measured as the number of links or hops on the shortest path between two users on Facebook). We measure the similarity between two users via cosine similarity [11], [12].

Consider user set U_y and video set V_y, and let m and n denote the sizes of these two sets respectively. Then we use a $m \times n$ matrix A to model users' video related activities, in which $A_{ij} = 1$ means that user i has at least one interaction with video j and otherwise $A_{ij} = 0$ ($i = 1, 2, \cdots, m$ and $j =$

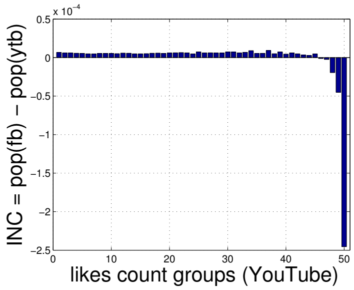

Fig. 11. Popularity increase on Facebook from YouTube measured as inc_i^L.

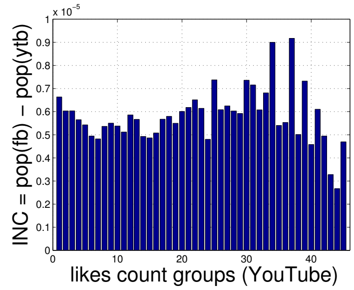

Fig. 12. Video groups that have increased average popularity on Facebook.

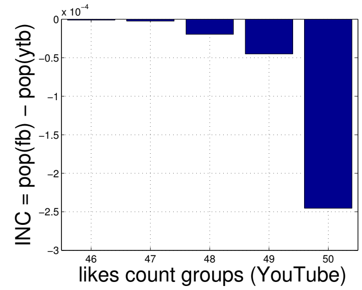

Fig. 13. Video groups that have decreased average popularity on Facebook.

$1, 2, \cdots, n$). Note that $n = |V_y|(= 134, 151)$. The row vector A_i of matrix A represents a binary vector of user i's interaction with all videos in V_y. In this section, we only consider those users who have interactions with at least 10 videos during our sampling period. The cosine similarity between users i and ℓ is calculated by:

$$cosineSim(i, \ell) = \frac{\sum_{j=1}^{n} A_{ij} \times A_{\ell j}}{\sqrt{\sum_{j=1}^{n} A_{ij}^2} \times \sqrt{\sum_{j=1}^{n} A_{\ell j}^2}} \quad (1)$$

The range of *cosineSim* is $[0, 1]$. If the *cosineSim* value between two users is close to 1, it means that the two users are very similar to each other. In Figure 14 we plot the boxplots of *cosineSim* between two users that are one hop away (i.e. direct friends), two hops away, and more than two hops away on Facebook. Note that it is computationally too expensive to search the whole Facebook network in order to find if two users are exactly x-hop away when $x \geq 3$. This figure shows that the similarity between two users in general does not depend on their social distances. This is not surprising as it is not unlikely for a person to find many strangers with similar interests. Thus it might not be effective to use social distance to infer user similarity and further help with video recommendation between users.

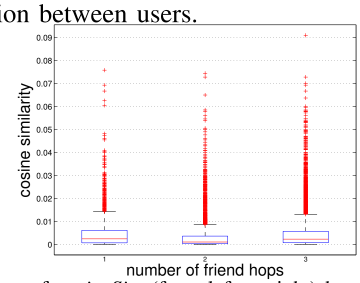

Fig. 14. Boxplots of *cosineSim* (from left to right) between two users (in U_y) that are direct friends, two hops away, and more than two hops away.

IV. Social Interaction based Recommendation

We next demonstrate that information about social interaction among users on OSNs can significantly improve video recommendation accuracy. We consider top-k recommendation [7], [13], as it is widely used in practice.

Recall that for user set U_y and video set V_y (m and n denote the sizes of these two sets respectively), we can get a $m \times n$ matrix A to model users' video related activities. The row vector A_i of matrix A represents a binary vector of user i's interaction with all videos in V_y. In this section, we only consider those users who have interactions with at least 50[1] videos during our sampling period. That is, the number of ones in each row vector must be at least 50. We say that user i *has interest* in video j if $A_{ij} = 1$; user i *has no interest* in video j if $A_{ij} = 0$.

For each user i, we randomly select 20% of videos that have value of 1 in A_i as our test dataset. Let $V_{i,test}$ denote this set of test videos. We use the set of the remaining videos as user i's training dataset, denoted by $V_{i,train}$. Note that $\forall i$, $V_y = V_{i,test} \bigcup V_{i,train}$, and $V_{i,test} \bigcap V_{i,train} = \emptyset$. In addition, let $V_{i,0}$ represents the set of all videos with which user i has no interaction. Note that $V_y = V_{i,test} \bigcup (V_{i,train} \setminus V_{i,0}) \bigcup V_{i,0}, \forall i$.

We then compare three top-k recommendation algorithms.

The first algorithm is a simple voting algorithm based on collaborative-filtering [11]. For each user i, we calculate his/her cosine similarity to the other users in U_y, using the $V_{i,train}$ video set, as in (1). We choose the set of those top 50 most similar users, denoted as S_i. Then we get a candidate video set $V_{i,rec} = V_{i,0} \bigcup V_{i,test}$ for user i. For each video $j \in V_{i,rec}$, we calculate its score as the number of users in S_i who *have interests* in video j (i.e., have at least one interaction with it). We then sort all videos in $V_{i,rec}$ in the non-increasing order of their scores and use the top k videos, denoted as $V_{i,rec}^k$, as the top-k video recommendation list for user i. This algorithm is referred to as *CollabRecommend*.

The second algorithm is an enhanced version of the first algorithm. It additionally considers social interactions between users. For a user i, this algorithm sorts all other users based on their numbers of interactions with user i. Then the top 50 users with the most interactions with user i are added into set S_i (in addition to the top 50 most similar users). Note that the set S_i may not contain 100 users as there might be overlap between the set of the top 50 most similar users and the set of the top 50 users with the most interactions. This algorithm is referred to as *SocialRecommend*.

The third algorithm is referred to as *NaiveYouTube*. It simply recommends the top k most popular videos to user i from its candidate set $V_{i,rec}$. We use either *like* counts or *watch* counts to measure video popularity on YouTube.

We use top-k Hit-Ratio and Recall [7] to evaluate the accuracy of the above three algorithms. Top-k Hit-Ratio is

[1]Measuring recommendation accuracy is problematic for a user that interacts with < 10 videos. On the other hand, only looking at users that interact with lots of videos can result in a small number of eligible users and hence unrepresentative results. We find that 50 appears to be a reasonable number.

defined as the fraction of *interested* items in the test set that are covered in the top-k recommendation list, i.e.,

$$H(i,k) = \left(|V_{i,rec}^k \bigcap V_{i,test}|\right) \Big/ \left(|V_{i,test}|\right) \qquad (2)$$

We further use *recall* to measure the recommendation accuracy for all users,

$$recall = \left(\sum_{i=1}^{|U_y|} |V_{i,rec}^k \bigcap V_{i,test}|\right) \Big/ \left(\sum_{i=1}^{|U_y|} |V_{i,test}|\right), \qquad (3)$$

which is essentially the weighted sum of the top-k hit ratios of all users, with the weight for each user being proportional to the number of interested videos in the test set. We vary k from 10 ,20, 30, 40, to 50.

Figures 15 to 17 show the boxplots of hit ratios of all users in U_y at different k values for *NaiveYouTube*, *CollabRecommend*, and *SocialRecommend* respectively. These figures show that both *CollabRecommend* and *SocialRecommend* are significantly better than *NaiveYouTube*, and the social interaction based algorithm *SocialRecommend* is the best among all the three. Figure 18 shows the *recall* values of each recommendation algorithm. We can see that the recalls of *NaiveYouTube* based on *like* counts and *watch* counts are very small, so their bars are nearly invisible. In addition, *SocialRecommend* algorithm performs better than *CollabRecommend*, with about 20% to 25% relative accuracy improvement across different k values.

Then for a target user (i.e., user i mentioned in the previous description of the three algorithms), besides selecting the top 50 (denoted by N_{sim}) most similar other users and the top 50 (denoted by N_{inter}) users that have the largest number of interactions with her/him, we also set both N_{sim} and N_{inter} to be $10, 30, 80$, or 100. When $N_{inter} = 10$ and $N_{sim} = 10$, *SocialRecommend* achieves highest *recall*, but it decreases and converges to a stable value when N_{inter} and N_{sim} both increase to 100 from 10. This is true for any k (of a top-k list). Part of our ongoing work is to explore the design space of *SocialRecommend* and improve its performance.

In summary, we observe that recommending YouTube videos to Facebook users only based on YouTube's global popularity ranking can have very poor accuracy. The recommendation accuracy improves significantly when we consider the similarity among users' video related activities on Facebook. Furthermore information about social interactions between Facebook friends can further improve YouTube video recommendation accuracy.

V. Conclusion

In this paper, we presented a case study on recommending YouTube videos to Facebook users based on their social interactions. We first showed through a measurement study that video popularity on Facebook is not always well-aligned with video popularity on YouTube, and unpopular videos on YouTube can get significant popularity boost on Facebook. Motivated by this finding, we developed a simple top-k video recommendation algorithm that exploits user social interactions on Facebook to improve the recommendation accuracy for YouTube videos. We demonstrated through experiments that social interaction information can be added on top of the existing similarity-based collaborative filtering algorithm to achieve significantly higher top-k recommendation accuracy. For future work, we will get more diverse datasets to improve the representativeness of our results.

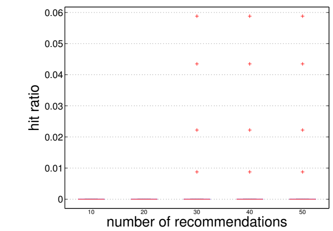

Fig. 15. Watch-count based *NaiveYouTube*'s top-k hit ratios at different k values.

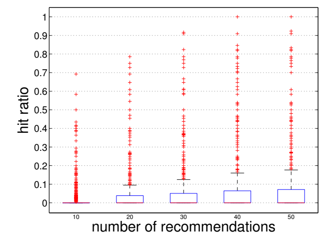

Fig. 16. *CollabRecommend*'s top-k hit ratios at different k values.

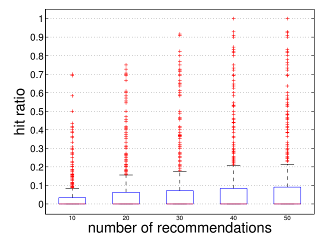

Fig. 17. *SocialRecommend*'s top-k hit ratios at different k values.

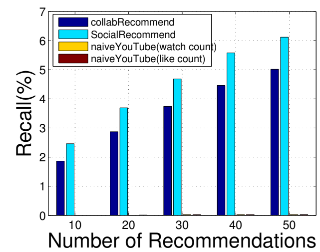

Fig. 18. *Recalls* of the recommendation algorithms. The recalls of *NaiveYouTube* based on *like* or *watch* counts are very small, so their bars nearly invisible.

VI. Acknowledgment

This research is supported in part by NSF under grants CNS-0845500, CNS-1018032, and an Faculty Research Grant of Fordham University (2013-2014).

References

[1] J. A. Golbeck, "Computing and applying trust in web-based social networks," *Doctoral Dissertation*, 2005.
[2] M. Jamali and M. Ester, "Trustwalker: a random walk model for combining trust-based and item-based recommendation." in *ACM Conference on Know ledge Discovery and Data Mining (KDD)*, 2009.
[3] M. Jamali and M. Ester., "A matrix factorization technique with trust propagation for recommendation in social networks." in *ACM Conference on Recommender Systems (RecSys)*, 2010.
[4] H. Ma, I. King, and M. R. Lyu., "Learning to recommend with social trust ensemble." in *ACM conference on Research and development in information retrieval (SIGIR)*, 2009.
[5] P. Massa and P. Avesani., "Trust-aware recommender systems." in *ACM Conference on Recommender Systems (RecSys)*, 2007.
[6] X. Yang, H. Steck, and Y. Liu, "Circle-based recommendation in online social networks," in *Proceedings of ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2012.
[7] X. Yang, H. Steck, Y. Guo, and Y. Liu, "On top-k recommendation using social networks," in *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 2012, pp. 67–74.
[8] [Online]. Available: www.youtube.com
[9] [Online]. Available: www.facebook.com
[10] H. Steck, "Item popularity and recommendation accuracy," in *Proceeding of RecSys 2011 Proceedings of the fifth ACM conference on Recommender systems*, 2011.
[11] A. Gunawardana and G. Shani, "A survey of accuracy evaluation metrics of recommendation tasks," *The Journal of Machine Learning Research*, vol. 10, pp. 2935–2962, 2009.
[12] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, "Effects of user similarity in social media," in *The Fifth ACM international conference on Web search and data mining*, 2012.
[13] M. Jamali and M. Ester, "Using a trust network to improve top-n recommendation," in *Proceedings of the third ACM conference on Recommender systems*. ACM, 2009, pp. 181–188.