CS 224W Project Milestone
Analysis of the YouTube Channel Recommendation Network

Ian Torres [itorres]
Jacob Conrad Trinidad [j3nidad]

December 8th, 2015

I. Introduction

With over a billion users, YouTube is one of the largest online communities on the world wide web. For a user to upload a video on YouTube, they can create a channel. These channels serve as the home page for that account, displaying the account's name, description, and public videos that have been uploaded to YouTube. In addition to this content, channels can recommend other channels. This can be done in two ways: the user can choose to feature a channel or YouTube can recommend a channel whose content is similar to the current channel. YouTube features both of these types of recommendations in separate sidebars on the user's channel. We are interested analyzing in the structure of this potential network.

We have crawled the YouTube site and obtained a dataset totaling 228575 distinct user channels, 400249 user recommendations, and 400249 YouTube recommendations. In this paper, we present a systematic and in-depth analysis on the structure of this network. With this data, we have created detailed visualizations, analyzed different centrality measures on the network, compared their community structures, and performed motif analysis.

II. Literature Review

As YouTube has been rising in popularity since its creation in 2005, there has been research on the topic of YouTube and discovering the structure behind its network. Thus, there exists much research analyzing YouTube as a social network. Cheng looks at videos as nodes and recommendations to other videos as links [1]. Paolillo looks at channels as nodes and friends as links [2]. In addition, there are several other papers that examine finding structure among networks that will prove useful in the results of this paper. This section is a short summary and discussion on these works.

i) Statistics and Social Network of YouTube (Cheng, Dale, & Liu, 2008)

Cheng et al. [1] present a study of various statistics of YouTube videos. Additionally, the paper investigates the social network structure of YouTube videos, finding evidence of small-world characteristics. Data from over 3 million videos was collected and analyzed. The network analyzed in the paper is a directed graph where each YouTube video is a node, and an edge from a to b exists if b is in the first 20 "related videos" listed for a. Analysis was done on 10 cumulative datasets, each consisting of all previously collected data up to that point, with the last containing all data. Measurements were made on the Largest Strongly Connected Component of each graph. The clustering coefficient of the YouTube network was found to be much higher than a random graph's, while the characteristic path length was

only slightly larger. Thus the network has small-world characteristics, as it contains cliques and nodes are linked to all others by relatively short paths.

ii) Structure and Network in the YouTube Core (Paolillo, 2008)

Paolillo [2] gives an empirical investigation into the social structure of YouTube in terms of friend relations and comments. He investigates social interactions on the network and their relation to the content they upload. Data from 9948 videos was collected and resulted in the identification of 82 thousand users and 273 thousand edges directed friend edges. This paper illustrates that YouTube appears to have a social core among its users. This can be seen in consistent interactions of groups of users in similar types of video content. Such interactions include the friending between users and commenting on the same video threads. It reveals that these social interactions are similar to that of social media networks, but with more coherence on content.

iii) Finding and Evaluation Community Structure in Networks (Newman & Girvan, 2003)

Newman and Girvan [3] give three algorithms for finding community structure in networks using different edge betweenness measures. The defining features of the algorithms are that they are divisive, i.e. they break up a network into communities by iteratively removing edges, and that betweenness scores are recalculated after the removal of each edge. They also give a method for measuring the strength of these algorithms. The general community-structure algorithm using any of these measures consists of calculating betweenness scores for all edges, removing the highest-score edge, recalculating betweenness scores, and repeating. In practice, the paper recommends using the shortest-path betweenness measure, as its corresponding algorithm has the fastest runtime of $O(m^2n)$. The other measures tend to be about as good or worse at finding community structure, and have poorer performance. Reading this paper inspired us to look further into community structures, which lead us to the following paper.

iv) Fast Unfolding of Communities in Large Networks (Blondel, 2008)

Blondel et al. [4] devised a simple algorithm to extract community structure in large networks. Their algorithm works as follows. In the first phase, we begin by assigning a different community to each node of the network. Then, for each node i, we consider its neighbor j and evaluate the gain in the modularity measure by removing i from its community and placing it with j. We repeat this process until there is no further improvement. In the second phase, we build a new network whose nodes are the communities found in the first phase. We can then reapply the first phase to this network, and repeat these phases until there is no improvement. The accuracy of this algorithm has been shown to be very good in comparison with other community detection algorithms. Since we would like to understand the community structure of our large network, we plan to use this algorithm later in this paper. This is also known as the Louvain method for community detection.

v) FANMOD: A Tool for Fast Network Motif Detection (Wernicke, 2006)

Wernicke and Rasche developed FANMOD [5], a tool for fast motif detection in large networks. FANMOD implements an algorithm also devised by Wernicke [6] called RAND-ESU, a fast, unbiased subgraph-sampling algorithm. When calculating the significance of motifs, FANMOD compares the frequency of a motif in the original network to the frequency in a random network generated by performing a series of random edge swaps in the original network. FANMOD runs orders of magnitude faster than similar tools on large networks. We chose to use FANMOD for motif analysis because of its fast runtime given the size of our networks and support of directed graphs.

III. Data Collection

We crawled YouTube's website and scraped users' channel pages by following a breadth first search algorithm. The algorithm began by initializing a queue with the top 1000 most subscribed YouTube channels and each channel was given a depth of 0. Then, for each channel in the queue, we scrape its YouTube page to find its recommended channels. For each recommended channel, if it's not a generated YouTube channel and its not seen before, we add it to the queue with an increased depth. We add a directed edge from the current channel to each valid recommended channel. Once we reach a depth of 5, we stop adding to the queue. The data crawl is done once the queue is empty and outputted a key file, which notes id/url pairs, and an edges tsv file, which notes source and destination ids and the type of edge (user or YouTube recommended).

On November 1st, we collected from the website *www.vidstatsx.com* the top 1000 most subscribed YouTube channels. We collected the data by running a Python script utilizing Beautiful Soup 4 on an Amazon Web Service EC2 virtual server. We found a total of 228575 channels. 1000 channels were at a depth of 0 and used to initialize the algorithm. 4497 were at a depth of 1, 11722 were at a depth of 2, 29166 were at a depth of 3, 63071 were at a depth of 4, and 119119 were at a depth of 5. With these channels, we found 905853 recommendations, where 400249 where user recommended and 505604 were YouTube recommended. In addition, we found that 86 channels were failed to have their recommended channels scraped during the data collection. This occurred because these channels were deleted and no longer existed. Thus, they did not have any outgoing edges.

Note that we ignore those that are generated automatically by YouTube's video discovery system. Such examples include "#Music", "#Gaming", and "#Sports", which automatically aggregate videos that fall under their categories into their channel. We are ignoring these channels because we want to focus on channels that have humans curating and controlling user recommendations rather than automated channels.

This approach to constructing the YouTube Channel Recommendation Network also has a particular disadvantage. We can see that the resulting network is only a small subset of the entire YouTube channel recommendation graph. However, by targeting the most subscribed YouTube channels and doing a BFS outward, we hope to capture the most active region of YouTube. YouTube users have a tendency to collaborate with each other and have similar types of content, thereby allowing the recommendation system to create these communities that we will examine. We will further examine the implications of this in our results.

IV. Models

We are using three models to describe the YouTube channel network. Let all three of these models be directed graphs where the nodes are channels, and there is an edge from channel c_i to channel c_j if c_i recommends c_j. In $model_{user}$, c_i recommends c_j if the user of channel c_i recommends c_j. In $model_{YouTube}$, c_i recommends c_j if YouTube's system recommends channel c_j on channel c_i. In $model_{all}$, c_i recommends c_j if c_j is recommended by c_i either by the user or by YouTube's system. In each of these networks, let the set of nodes be all of the channels found in the data collection process. Let the set of edges in $model_{user}$ be the set of user recommendations found in the data collection. Let the set of edges in $model_{YouTube}$ be the set of YouTube recommendations found in the data collection. Let the set of edges in $model_{all}$ be all of the recommendations found in the data collection, i.e. the union of the edges in the other two models. In our networks, we did not assign any weights to the edges, so they all have a default weight of 1.

We have two primary goals for this project. The first goal is to discover the underlying structure behind each of these three networks. We will do so by analyzing different properties of each model, in particular centrality measures, community structure, degree distribution, and motifs. The second goal is to compare and contrast the three models using these structural properties that we have discovered. Using this, we hope to shed light on their network structures and why they are structured in such a manner.

We must recognize that there are some caveats that come with the nature of channel recommendations on YouTube that may affect these differences. There are a limited number of outgoing edges for each YouTube channel. Based on our observations, there appears to be at most 10 user recommended channels and 6 YouTube recommended channels. In addition, YouTube recommendations do not include channels that the user recommended. This means that YouTube recommendations will by nature connect channels that are more similar in content, but not directly connected. In contrast, the nature of user recommendations is that users tend to know and collaborate with each other, thus creating a community collaborations, such that there would tend be more reciprocated edges and connected triads. Lastly, note that this community structure is based on the time we collected the data, as these recommended channels could very well change over time. We will explore in further detail the implications of these aspects.

V. Results

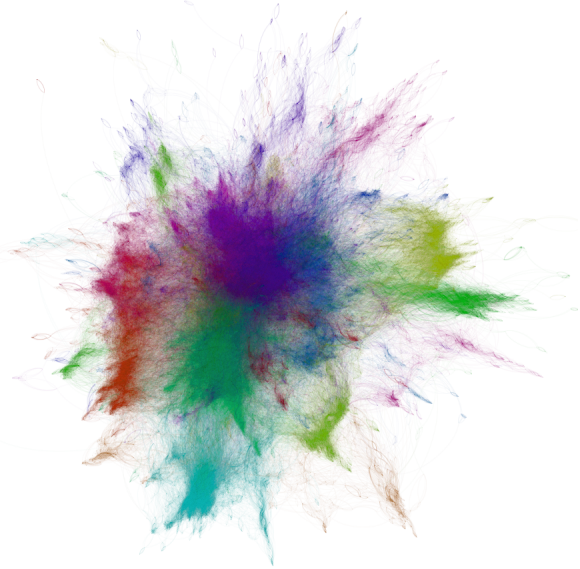
This section entails the results of our analysis on the YouTube channel recommendation network. Here we discuss our visualizations, the community structure, centrality measures, degree distributions, and motif analysis.

i) Visualizations

We used Gephi to produce the visualizations seen in Figure 1. We focused on the giant connected component and eliminated any nodes with no in-degree or out-degree (and their corresponding edges). This resulted in 75791 nodes and 544020 edges in the graph in Figure 1a, 43685 nodes and 198310 edges in Figure 1b, and 45497 nodes and 200660 edges in Figure 1c.

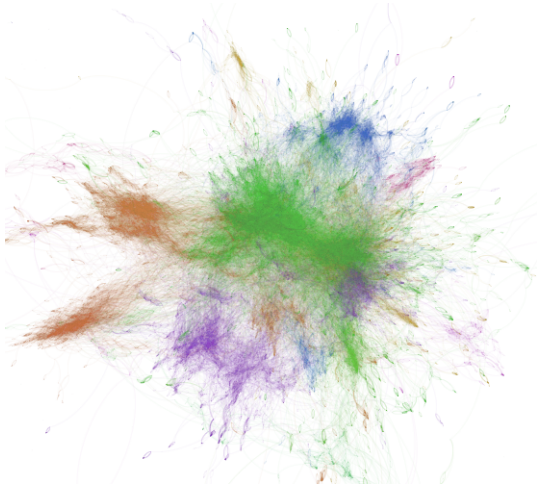
To produce the colored communities, we used the modularity functionality of Gephi. The modularity function of Gephi allows us to detect community structure. It does so by using the algorithm based on Blondel's paper[4] as mentioned in the related works. The resolution limit of modularity imposes a limit on the size of the smallest community one can obtain by modularity optimisation. Note that we used higher resolution limits to allow for more pronounced communities in our visualizations, since higher resolution limits corresponds to communities of larger sizes and thus a smaller number of communities. Using a resolution limit of 3.0, we find 26 communities with a modularity of .812 in figure 1a. Using a resolution limit of 6.0, we find 206 communities with a modularity of 0.797 in figure 1b. Using a resolution limit of 6.0, we find 207 communities with a modularity of 0.789 in figure 1c. Each of these communities is colored differently and illustrated in the figures above. To produce the layout, we began with a random layout and then used the ForceAtlas2 algorithm as it appeared to be the best as separating the communities.

In figure 1a, the purple color on the top of graph consists of 19% of the graph and consists of popular entertainment and comedy channels such as JennaMarbles, TheEllenShow, and PrankvsPrank. The green color to the left of the graph occupies 17% of the network and consists of popular gaming channels such as PewDiePie, VanossGaming, and CaptainSparklez. The central blue community occupies 10% of the graph and consists of musical channels such as RihannaVEVO, JustinBieberVEVO, and ArianGrandeVevo.



(a) Includes both user and YouTube recommendations

(b) Only includes YouTube recommendations



(c) Only includes user recommendations

Figure 1: Network Visualizations for the $model_{all}$ (a), $model_{YouTube}$ (b) and $model_{user}$ (c)

In figure 1b, we can see similar categorization of communities. For instance, the purple on the right side consists of 15% of the graph and consists of the entertainment channels while the green color consists of the 11% of the graph and consists of the music channels. This similarity is likely due to YouTube's recommendation algorithm keeping these similar types of content together. In addition, it possibly reveals that the YouTube recommendations heavily influenced the community detection in the previous model.

Interestingly, figure 1c is has different types of communities. The large green section consists of 32% of the nodes and contains all of the previously mentioned types of channels (gaming, entertainment, and music). Meanwhile, the concentrated red section on the left side of the graph takes up 11% of the graph and features Spanish channels. This is likely a result of the user recommendations forming small, tight knit communities, that when combined due to the high resolution limit, created one giant community. We will examine further such phenomena in the section (iii).

ii) Community Detection

We have touched the topic of community detection slightly in section (i). We can note the clear presence of different communities within the network. In this section, we will further explore effects of the different types of recommendations on communities as well as determining the different types of communities in the network. Once again, we used Gephi's functionality to compute the communities present within our three models, this time, using a resolution limit of 1.0. We found 319 communities in $model_{all}$, 61020 communities in $model_{user}$, and 84242 communities in $model_{YouTube}$. Further detailed results can be seen in Table 1.Referencing our modularity scores for all three of our models, we notice that they are all positive and close to 1. This means that there is a strong sense of various communities within each model.

	Modularity Score	Number of communities within the size range					
		(0,1]	(1,10]	(10,100]	(100,1000]	(1000,10000]	(10000, ∞)
$model_{all}$	.852	248	13	2	9	42	5
$model_{user}$	.903	56804	3840	288	43	44	1
$model_{YouTube}$	.896	83517	612	49	19	44	1

Table 1: Data in regards to each models community structure

The second thing to note is the large number of communities within each model that have a size less than or equal to 10. There are communities of size 1 because they have no incoming edges and no outgoing edges. The 248 singleton communities in $model_{all}$ are isolated because they were a part of the initial 1000 YouTube channels we seeded our BFS with. The large number of communities whose size is 1 in $model_{user}$ and $model_{YouTube}$. This is due to the fact that there are many more nodes within these networks that have no incoming or outgoing edges of that type. There are a large number of communities within $model_{user}$ of size between (1,10] because these are a lot of small communities who just recommend each other and small communities where there is 1 channel recommending other channels with no user recommendations. The latter is likely due to the fact of our data collection, and all of our nodes at a depth of 5 have no outgoing edges.

Focusing on the larger communities for each of these models, we notice that there are not that many. In fact, there are only 5 communities in $model_{all}$ larger than 10000 (with the largest of size 20946), 1 community in $model_{user}$ larger than 10000 that is of size 12449, and 1 community in $model_{YouTube}$ that is of size 12360. Looking at the 5 large communities in $model_{all}$, we notice that the channels within each community share similar types of content, which are communities of entertainment/gaming, foreign entertainment, foreign gaming, movies and TV shows, and then music channels, with the entertainment/gaming community being the largest. We notice similar trends in the largest communities in the $model_{user}$ and $model_{YouTube}$ networks, where in both of these communities focus on entertainment and gaming. This illustrates the strong presence of such communities on YouTube. These also reflect the network visualizations seen in the prior section.

iii) Centrality

We used SNAP to calculate the in-degree, out-degree, betweenness, and closeness centralities of the nodes in the giant connected component of each graph. To gain some initial insight on this data, we then found the top 20 nodes for each centrality measure in each graph. Here we list the top 5. We omit out-degree as many channels had the maximum number of recommended channels imposed by YouTube, making this measure ineffective.

These results yield some interesting insights. One distinction we noticed in Table 2 is that between

$model_{user}$	$model_{Youtube}$	$model_{all}$
Vevo	VanossGaming	VanossGaming
Machinima	LittleBabyBum	LittleBabyBum
Warner Music Group	BuzzFeedVideo	Bratayley
PewDiePie	ChuChu TV Kids Songs	ChuChu TV Kids Songs
TGN	Bratayley	BuzzFeedVideo

Table 2: In-Degree Centrality: Top 5 Channels for each model

$model_{user}$	$model_{Youtube}$	$model_{all}$
Fabinho da Hornet	Sony MAX	PewDiePie
StreetfrassMusic	Sony AATH	BuzzFeedVideo
MafiaTheViper Ent	SET India Classics	jacksepticeye
Krish Genius	SET India	VanossGaming
Jahvis Crushroad	TV Shows	Markiplier

Table 3: Closeness Centrality: Top 5 Channels for each model

$model_{user}$	$model_{Youtube}$	$model_{all}$
PewDiePie	PewDiePie	PewDiePie
BuzzFeedVideo	Machinima	BuzzFeedVideo
VanossGaming	Miranda Sings	VanossGaming
jacksepticeye	LittleBabyBum Â®	jacksepticeye
elrubiusOMG	TGN	Vevo

Table 4: Betweenness Centrality: Top 5 Channels for each model

$model_{user}$	$model_{Youtube}$	$model_{all}$
Vevo	VanossGaming	AviciiOfficialVEVO
Vevo UK	jacksepticeye	jacksepticeye
Vevo dscvr	Markiplier	VanossGaming
IISuperwomanII	PewDiePie	DrakeVEVO
Vevo Polska	SkyVSGaming	PewDiePie

Table 5: PageRank Centrality: Top 5 Channels for each model

the highest in-degree nodes is that the top nodes in the user-only network contain music/gaming networks (Vevo, Machinima, TGN), while the top nodes in the Youtube-only network and the combined network have the same channels, just in a different order. The latter highlights the influence of the YouTube recommendations on the overall network in terms of in-degree. Notably in Table 3, PewDiePie is the top node for betweenness centrality for all three graphs. As PewDiePie is the most subscribed channel on YouTube, perhaps we might be able to use betweenness centrality in our networks to help predict the popularity of a channel.

The top nodes for closeness centrality for the user- and YouTube-only networks do not appear for any other centrality measure, which suggests that closeness centrality may not be a very useful measure for these networks. This makes sense, as the maximum out-degree imposed by YouTube greatly limits the closeness centrality of the nodes. Another interesting aspect is the high closeness of Fabinho da Hornet in the user model. This could potentially explain the strong presence of a Spanish centric community within figure 1c. More generally, the top 5 channels in the Table 4 for the user and YouTube models were not

seen before, highlighting the interesting structure of these channels maintaining high centrality despite the BFS search from the top 1000 subscribed channels. We can also see in Table 5 an interesting division between each of the models. We notice that the VEVO channels have high PageRank in $model_{user}$, the gaming channels have high PageRank in $model_{YouTube}$, and $model_{all}$ is a mix of the two. This appears to be because in terms of user recommendations, VEVO controls all of its channels thus tends to recommend itself, whereas all the top gaming channels in PageRank also appear to be top in other centrality measures, which illustrates their prominence in the YouTube community.

iv) Degree Distributions

We have briefly discussed degree distributions in section (iii) when examining degree centrality. In this section, we will analyze the degree distributions of the network and note their impact on the overall structure of the graphs. To see these degree distributions, we decided to plot them as seen in Figure 2.

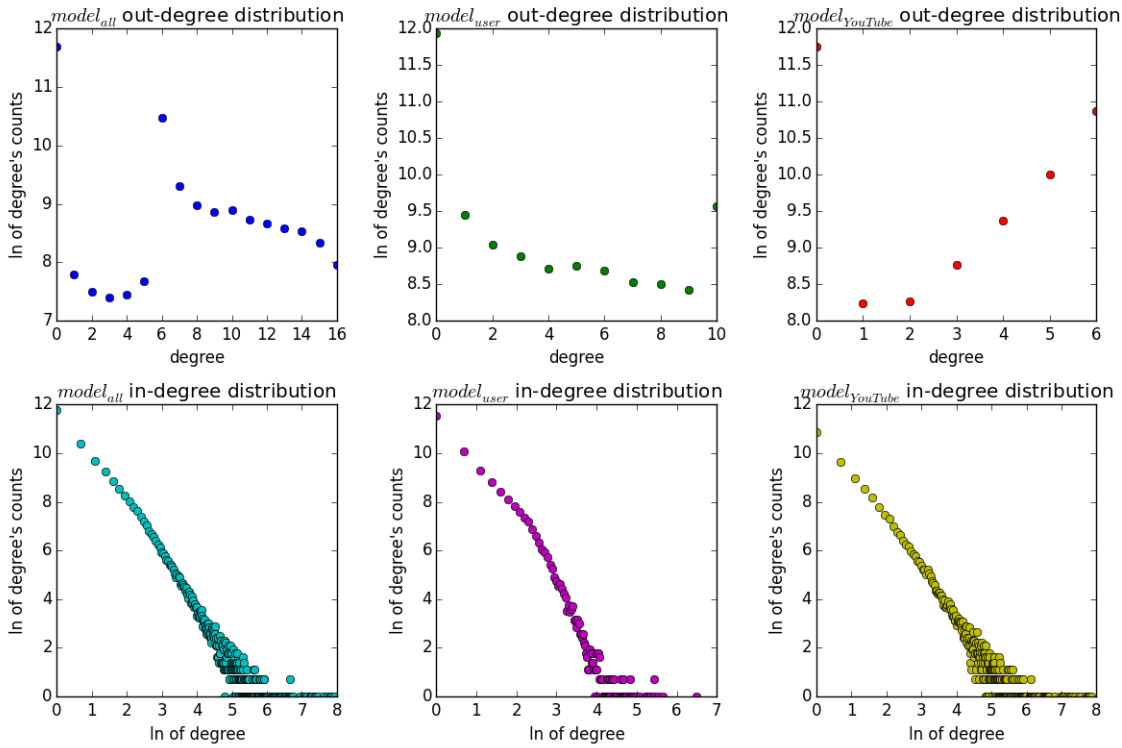


Figure 2: Degree Distributions: the top row are out-degree distributions and the bottom row are log-log plots of their in-degree distributions. Left column is $model_{all}$, middle column is $model_{user}$, and right column is $model_{YouTube}$

These plots reveal some very interesting information about the in-degree and out-degree distributions of the models. Looking at the out-degree distribution, we notice that there are a large number of channels that do not have recommendations in the first place. In addition, We can see a trend where users tend to recommend a smaller number of channels or use all 10 of their recommendations. In contrast, YouTube tends to recommend closer to their max of 6 channels. This shows that YouTube attempts to recommend more channels when they can in contrast to users, who only either recommend their close community or they max out their recommendations.

Examining the log-log plots of the in-degree distributions, we see an interesting feature. It appears that the degree distributions follow a power-law distribution. When fitting a power law distribution to

it, all of the models have $xmin = 1$ and the an α exponent of 1.757 for $model_{all}$, 1.845 for $model_{user}$, and 1.708 for $model_{YouTube}$ as calculated by the plfit library for Python. This highlights the fact that there this social network has aspects of preferential attachment, as people tend to recommend more popular channels since they are in a community with them (such as BuzzFeedVideo) and YouTube tends to recommend the popular channels because they are simply great within their genre, so YouTube promotes them. We must note however that these figures may come from the nature of our data collection, such that for all of the channels that we have examined, we have all of their outward edges but only a portion of their inward edges. So there is a chance that there we see this preferential attachment phenomena because the nodes closer to the center are those that we initialized the BFS algorithm with. Despite this, we believe that it is natural for the channels with higher subscription counts to have higher in-degree and that these results would still hold true if he had access to the network in its entirety.

v) Motif Analysis

We used FANMOD [5] to do motif analysis on the networks. FANMOD is a tool for detecting network motifs, small subgraphs that appear more frequently than in a random network. For each model, we used FANMOD to detect size-3 motifs in the original network and 100 random networks. In Table 6, we list the top 5 most frequent motifs with p-Value = 0, meaning that the motif appeared more frequently in the original network than any of the random networks. Note that for all motifs in each model, p-Values were either 0 or 1. For each motif, we list its ID (obtained from taking the adjacency matrix of the motif as a binary integer), frequency in the original network, and mean frequency in the random networks.

Table 6: Top 5 Most Frequent Size-3 Motifs with p-Value = 0

$model_{user}$			$model_{YouTube}$			$model_{all}$		
ID	Freq.	Freq. (Random)	ID	Freq.	Freq. (Random)	ID	Freq.	Freq. (Random)
36	36.51%	34.51%	36	95.70%	94.78%	36	88.62%	87.07%
38	1.4994%	0.0022%	38	0.3601%	0.0035%	38	0.4526%	0.0055%
174	1.1985%	0.0004%	166	0.2134%	0.00003%	166	0.3234%	0.0001%
46	1.0674%	0.0006%	46	0.057%	0.0003%	46	0.1930%	0.0014%
166	1.0625%	0.0003%	174	0.0444%	0.000001%	174	0.1638%	0.000001%

The results from FANMOD show significant differences in the motif distributions between $model_{user}$ and $model_{YouTube}$. Although the top 5 motifs are the same for both models, the less frequent motifs appear much more frequently in $model_{user}$ than in $model_{YouTube}$. The most frequent motif in both models, 36, has adjacency matrix $\left(\begin{smallmatrix}0&0&0\\1&0&0\\1&0&0\end{smallmatrix}\right)$, consisting of only two edges directed toward a single node. Motif 36 makes up 95.70% of motifs in $model_{YouTube}$, but only 36.51% of motifs in $model_{user}$. In contrast, $model_{user}$ contains significantly higher frequencies of motifs with bidirectional edges. For example, motif 238 (not in the table), in which all vertices share bidirectional edges, has frequency 1.0468% in $model_{user}$ and only 0.0215% in $model_{YouTube}$.

We believe that this distinction is due to the tendency of users to reciprocate channel recommendations. Users that collaborate in some way are more likely to recommend each other, so a bidirectional edge in $model_{user}$ may be an indicator of a strong tie between two channels.

VI. Conclusion

YouTube has an interesting channel recommendation network structure. As we have seen, channel recommendations tend to be towards other channels that contain similar types of content. In particular, our community structure analysis revealed the strong presence and community of entertainment channels,

gaming channels, and music channels (in particular Vevo). The results of our examination of centrality measures show that the channels with high subscribers tend to have high centrality measures, such as PewDiePie's (the most subscribed channel) presence in each of the top 5 centrality measures. The degree distribution of our models are shown to be power law, illustrating the feature of preferential attachment within these networks. Our motif analysis shows that the user model had a richer size-3 motif distribution than the YouTube model, possibly indicating the tendency for users to mutually recommend each other.

We see several opportunities as to how one can extend this work. The first would be to simply to collect more data. This could be done by letting our algorithm run until all possible channels have been examined. This way, we could explore the entirety of the network rather than a subset of it. Another extension would be to examine the evolution of the network over time. This could be done by capturing different snapshots of the network at different times and analyzing the changes (if any) to the network. We could also look at the community structure in certain subnetworks. For example, motivated by our results in the motif analysis, we could see if there is a more interesting community structure in the user model if we only include bidirectional edges. One last possible extension is to examine if meta-information about channels, such as subscriber count, age, and number of videos, could be predicted by the structure of the network.

References

[1] Xu Cheng, C. Dale, and J, Liu., Statistics and Social Network of YouTube Videos. in *Quality of Service, 2008. IWQoS 2008. 16th International Workshop on.* pp.229-238, June 2008.

[2] John C. Paolillo, Structure and Network in the Youtube Core. in *HICSS 2008 - 41st Hawaii International International Conference on Systems Science.* pp. 156. January 2008

[3] M.E.J. Newman and M. Girvan., Finding and evaluating community structure in networks. *Phys. Rev. E* 69, 026113 (2003).

[4] Vincent D Blondel, J. Guillaume, R. Lamboitte, E. Lefebrve. Fast unfloding of communities in large networks in *Journal of Statisical Mechanics: Theory and Experiment.* 2008.

[5] S. Wernicke, F. Rasche. FANMOD: a tool for fast network motif detection. *Bioinformatics*, 22(9):1152-1153, 2006.

[6] S. Wernicke. A Fast Algorithm for Detecting Network Motifs. In *Proceedings of the 5th Workshop on Algorithms in Bioinformatics (WABI '05), Lecture Notes on Bioinformatics.* Vol. 3692, pp. 165-177.