Link Prediction between YouTube Videos using Node Features and Role Attributes

Han Lin Aung, James Li, Justin Xu
Stanford University
hanlaung, dawwctor, justinx

ABSTRACT

YouTube is unequivocally one of the most prominent content creation and sharing sites on the Internet. This prominence has transformed YouTube from a video site to a different kind of social network, connecting subscribers, commenters, and content creators together to form a huge network of intermingling social circles. We are interested in learning more about how videos within these different social circles influence each other, and what roles they play within their communities. In this paper, we utilize this information to develop and compare several link prediction algorithms suggesting new related videos to users. To tackle this, we looked at a YouTube dataset containing graphs of related videos and analyzed the emergent roles and communities within this network to see how their interplay affected the rest of the graph's connections. We found that using K-nearest neighbors on a combination of RolX roles and genres performed the best at predicting accuracy, although Random Forests performed better when we also incorporated node aggregate features (additional node features).

INTRODUCTION

Our goal is to extract large scale relationships between communities and roles on YouTube to help us understand how views flow and progress throughout the YouTube network through the links offered by the related videos section. An influx of popularity and links in one genre may signal an equivalent rise in other communities, signalling to content creators that collaboration between different subsectors of YouTube communities may be beneficial. Revealing individual metrics of related content creators or videos can help stretch the bounds of intersectional content creation, drawing together communities which may seem disparate but actually have great influence on each other.

We extracted this information by implementing a variety of supervised machine learning approaches for link prediction using the features extracted within communities, including role counts obtained from RolX. A link prediction model will point to the relative relationships between communities. We utilized various algorithms, described later in our paper, to evaluate the efficacy of each method by using our temporal data to maximize view increase correlations between different communities.

Moreover, our link prediction model could also be helpful for everyday users by helping them find additional related videos not originally anticipated by the original YouTube algorithm. A problem often stated among YouTube users is the lack of genre diversity of videos suggested by YouTube. By also including suggestions to videos with similar roles and attributes, but unrelated genres, a YouTube user can take greater advantage of the YouTube platform, enhancing their viewing experience and world view.

Related Work

Characterization of the YouTube Video Community

This paper studies YouTube's topology to analyze its structural properties and the nature of the social relationships among users and between users and videos. It also analyzes various network properties, including user profiles and video popularities, in order to highlight the impact of social relationships on a content-sharing network like that of YouTube's [5].

This is relevant to our work as it deals with the relationships between videos in terms of relatedness, the biggest attribute of which is genre. This paper notes that these videos are heavily influenced by social relations, although the tags are determined algorithmically and not directly through human decision [5]. Other previous work on YouTube mostly considered the YouTube network from the perspective of users and subscribers rather than the perspective of videos and how they relate to each other, as we do.

Characterizing Links, Roles, and Communities

While not explicitly pertaining to YouTube, this paper nonetheless covers the relationship between links, roles, and communities in social networks, which we found to prove useful in conducting our own analysis of roles and communities within YouTube for link prediction. This paper analyzes the network structure and focuses on the roles of key actors in communities in order to glean insights into the link structure and behavioral characteristics of nodes within the graph across time [2].

It also identifies nodes acting as key roles in the graph and displays the relationship between the combination of influence and roles with recurrent links between sections of the graph [2]. Moreover, the paper found that discovered communities in social networks could be applied for link recommendations for bridging new communities together. This analysis shows that roles, communities, and links are all interconnected in social network graphs, justifying our focus on analyzing these three particular attributes in our YouTube dataset.

RolX

In order to determine which roles are present within our data, we used RolX, which has previously proven to be a reliable, scalable way of determining node roles using unsupervised learning. This paper developed an algorithm for extracting roles from a graph which could then be interpreted and used to classify and search for similar nodes [3]. We were interested in this because it provided a way to extract additional node features that could be used to identify relationships between graph sections [3]. This can easily be repurposed to using these node properties on YouTube videos to help predict whether two nodes were likely to form a link, which is equivalent to determining whether two videos were likely to be related.

Link Prediction via Supervised Learning

This paper relies on three sets of features: proximity, aggregated, and topological features [1]. These features respectively describe the similarity between nodes, individual properties of nodes, and relational structures between nodes, which are all then combined to help predict link formation [1]. We took a similar approach, especially with the topological and aggregated features, as the roles and communities a node belongs to can have a great effect on whether a specific YouTube video will become related to another one.

Furthermore, for our own link predictions models, we considered many of the machine learning models tested in this paper, as even though our features were different, the underlying models proved robust enough to continue providing good performance.

Data

The dataset we used is the "Statistics and Social Network of YouTube Videos" dataset located at http://netsg.cs.sfu.ca/youtubedata/.

Each file in the dataset consists of a directed graph of crawled YouTube videos, with each video corresponding to a node. Each node contains information on the uploader, category, length, view count, and other information we might consider useful when analyzing the different video roles. A directed edge in the graph exists from node a to b whenever a video b is in the first twenty videos of the related video list of a video a. We did not have to perform any data collection for our dataset, although we still aggregated all the graph files to work with the complete YouTube network.

Mathematical Background and Algorithms

RolX

To determine what roles each video node contained, we used the RolX algorithm to automatically discover each video's structural roles in the YouTube network. To do this, RolX recursively extracts features based off the network connectivity features, which are a combination of local features, like degree, and egonetwork features, features of the node's neighbors and edges located within [3]. It then uses these features to generate additional ones by aggregating surrounding features, as intuitively, nodes with similar roles must also have similar neighbors [3].

After reaching the recursive depth limit, RolX takes the resulting feature vectors and partitions the network by assigning each node a vector of roles it most closely fits. RolX provides us with a soft clustering of each node to each role and provides a role-feature matrix, or alternatively the sense-making matrix, that maps raw features to each role [3].

Feature Selection

The paper [1] constructs a supervised machine learning model for link prediction by first selecting which features are appropriate to use. These features include proximity features, aggregate features, and topological features.

Proximity features include keywords, which can be extracted from the underlying description of the videos, like keywords in a name that point to the similarity between nodes. Aggregate features are those which sum the values for particular metrics within a set of nodes, like view counts and comments for our graph. We also included role and genre information into our feature set as categorical variables. Topological features include those featuring the underlying connections in the graph. The paper [4] provides various distance metrics, which we list here, with the most relevant being the Shortest Distance.

Let $G = (V, E)$ be a directed, unweighted graph. We denote the set of node v's direct neighbors as $N(v)$.

1) Shortest Distance: $S(A, B)$, the shortest path connecting node A to B
2) Common Neighbors: $C(A, B) = |N(A) \cap N(B)|$
3) Jaccard Distance: $J(A, B) = \frac{|N(A) \cap N(B)|}{|N(A) \cup N(B)|}$

Binary Classification

We model link prediction as a binary classification problem, as we determine whether a connection between two nodes will exist, giving a label of 1 or 0 based on the features between those nodes. Our dataset uses $n(n-1)/2$ points, where n is the number of nodes in our graph G. The paper [1] explains multiple approaches we used, including SVMs with a linear/RBF kernel, Random Forest Decision Trees, K-Nearest Neighbors, and Naive Bayes, with SVM RBF kernel performing the best overall.

To evaluate each model, we tested with links between popular YouTube videos from our dataset, both from 2007 and 2008. We then measured the recall, precision, and F1-score of our models to see how accurately we predicted link formation. We additionally considered popularity metrics to see not only what was currently popular, but what would become popular through the links we predicted in the graph.

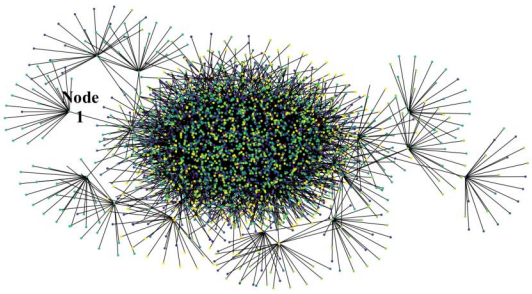

Fig. 1. Colors represent different genres

APPROACH

We started by analyzing our data by visualizing various graph attributes. The dataset includes several depths of the crawl through the related videos over different time frames. Some statistics we extracted are shown below (on the lowest two depth crawl):

1) Number of nodes: 3356
2) Number of edges: 21115
3) Clustering coefficient: 0.69329563953

To cluster nodes, we relied on the given genres to identify graph communities. For time and memory performance reasons, we assumed that videos of the same genres would naturally form related clusters, and to confirm this, we calculated the proportions of related videos with the same genres in Figure 1. We found that for a given video, over half of its related videos, specifically 53.35%, had the same genres. While difficult to see the connected nodes in the graph's center, the peripherals show us that the related videos of each nodes are likely to be of the same genre, such as Node 1 in Figure 1. Hence, videos with the same genres will be likely to form clusters, so we incorporate this information as a feature representing communities into our link prediction algorithm instead of extracting them through other methods, like Stochastic Block Modeling.

We also ran RolX on our graph to categorize our nodes into specific roles to be used as features for our link prediction algorithms. The recursive features of RolX incorporated both local and global network structures, helpful in encapsulating the structure of the network. Furthermore, we constructed a role-features matrix for each role we extracted, which included raw features, such as degrees and PageRank, to see which roles specific roles were actually relevant and which could be consolidated as one metric for the number of roles. We also tested with different number of roles to determine the optimal number of roles to use for RolX. The results are based on the number of roles (three roles) that gave us the best output though there is not a significant difference as we tweak the number of roles. The number 3 was chosen due to the fact that it was less computational energy to use fewer number of roles in our feature set.

The feature vector for each of the node pairs was:

$$v(n_1, n_2) = \begin{bmatrix} role(n_1) + role(n_2) \\ agg(n_1) + agg(n_2) \\ abs(agg(n_1), agg(n_2)) \\ genre(n_1) == genre(n_2)) \end{bmatrix}$$

Where $role(n_1)$ represents the 3 sized vector representing the role vector of node n_1. $agg(n_1)$ represents the 3 aggregate statistics (comments, views, ratings) for n_1 and $genre(n_1)$ is the genre of node n_1.

We call the RolX features the one generated from our Roles. RolX + genre including the genre feature, and RolX + agg + genre including all of the above features.

RESULTS

We partitioned our results by features and algorithms used, as well as by the total number of RolX roles incorporated into our features, as you can see in Table I.

To get the test set, we constructed features on the video list for a separate timestamp to see how well our data generalizes to a different graph.

To get our validation set, we first extracted 1% of the node pairs from our original graph, using them as our validation set to test performance on unseen node pairs. We then took

RolX	Random Forest	Logistic Regression	KNN	Naive Bayes
Train acc	0.603253301	0.582569028	**0.958943577**	0.527346939
Train precision	0.715769404	0.615970864	**0.924815539**	0.520364742
Train recall	0.342521008	0.438559424	**0.999111645**	0.69877551
Train F1	0.463324727	0.512341524	**0.960529049**	0.596515679
Val acc	**0.860691835**	0.727669851	0.856172533	0.358747381
Val precision	0.010529695	0.007109082	**0.025421687**	0.004404982
Val recall	0.37020316	0.492099323	**0.952595937**	0.720090293
Val F1	0.020476963	0.014015687	**0.049521798**	0.008756399
Test acc	0.748806597	**0.880894346**	0.862333257	0.246217111
Test precision	0.002316192	**0.002971768**	0.001171097	0.001765886
Test recall	0.429133858	0.25984252	0.118110236	**0.984251969**
Test F1	0.004607516	**0.00587633**	0.002319199	0.003525447

RolX + genre	Random Forest	Logistic Regression	KNN	Naive Bayes
Train acc	0.599747899	0.585498199	**0.959039616**	0.529651861
Train precision	0.710749252	0.619041252	**0.924942212**	0.522157236
Train recall	0.336398559	0.444609844	**0.999159664**	0.69877551
Train F1	0.456659551	0.517522777	**0.960619561**	0.597691707
Val acc	**0.858711866**	0.725183791	0.858205775	0.358747381
Val precision	0.010319068	0.007108469	**0.025777289**	0.004404982
Val recall	0.367945824	0.496613995	**0.952595937**	0.720090293
Val F1	0.020075128	0.01401631	**0.050196265**	0.008756399
Test acc	**0.955421385**	0.878563542	0.864354709	0.246217111
Test precision	**0.004766561**	0.002914422	0.001386248	0.001765886
Test recall	0.153543307	0.25984252	0.137795276	**0.984251969**
Test F1	**0.009246088**	0.005764192	0.002744883	0.003525447

RolX + agg + genre	Random Forest	Logistic Regression	KNN	Naive Bayes
Train acc	0.602641056	0.583613445	**0.958955582**	0.527130852
Train precision	0.71389973	0.616968394	**0.924873864**	0.520197326
Train recall	0.342569028	0.441032413	**0.999063625**	0.69877551
Train F1	0.462976183	0.514372121	**0.960538313**	0.596405664
Val acc	**0.859590866**	0.725663245	0.856962745	0.358747381
Val precision	0.010446525	0.007057072	**0.025443751**	0.004404982
Val recall	0.37020316	0.492099323	**0.948081264**	0.720090293
Val F1	0.020319663	0.013914598	**0.049557522**	0.008756399
Test acc	0.748646587	**0.880472988**	0.863351983	0.246217111
Test precision	0.002314717	**0.002961235**	0.001258356	0.001765886
Test recall	0.429133858	0.25984252	0.125984252	**0.984251969**
Test F1	0.004604596	**0.005855736**	0.002491824	0.003525447

TABLE I
TRAINING, VAL, AND TEST RESULTS WITH 3 ROLX ROLES

the remaining 99% of the node pairs, down-sampling the 0 labeled edges to the number of 1 labeled edges so that we had a balanced dataset to train on.

Due to the fact that our validation set had such a few number of 1 labels, the precision was ultimately very low across the board. However, we can see that when evaluating the recall values, the models performed better, but the K-Nearest Neighbors model actually gave us very high recall values (around .95 across the board). To interpret this, our model was able to capture the majority of the edges in the validation set, but also tended to label some other node pairs as edges. Comparing the f1 scores (an aggregate of the recall and precision scores), we see that K-Nearest Neighbors model performed the best.

However, when also looking at our performance on the test set, we see that K-Nearest Neighbors did a lot worse when generalizing to a new graph. The precision was just as low and the recall scores dropped down very low, (to .13 or .12) and the f1 scores of either the logistic regression or random forest outperforming KNN. This makes sense as the nearest neighbors of our original graph do not really correspond to anything in a new graph, so we can't really use that information. However, some of the information does translate when we use random forest or logistic regression, and it seems like our model was able to capture some of that. All in all, it doesn't seem like the algorithm was able to generalize very well though.

However, when we used all of the RolX, genre, and aggregate features, we found that performance across the board decreased significantly for every algorithm except for Naive Bayes, which already performed relatively poorly, and Random Forest.

We can see that there are miniscule differences between the different features we are using, however, in general we see that the RolX + genre performed greater had greater statistics across the board when compared to just the RolX features. However, when also adding in the agg features to the mix, we see that performance across the board tends to decrease, suggesting that the aggregate features don't add much to the model and the RolX features capturing the majority of the predictive power.

Analysis

Based off of our results, we found that K-Nearest Neighbors performed the best when using RolX roles or RolX roles combined with genre communities and considering generalizing to unseen node pairs in our current graph. Because videos with similar structural roles and genres in the graph are likely to be related to each other, it is more likely that a directed edge will form between the two corresponding nodes. Thus, we believe K-Nearest Neighbors performed so well overall because it focused purely on finding videos that shared these similarities, so the features included were well-suited for the algorithm's results. Essentially, because KNN focuses so much on similarity and edges in our YouTube network are based upon presence in the related video list, it makes intuitive sense that an algorithm for determining what other nodes are similar to the current node would perform so highly on a link prediction task based on related videos.

For very much the same reasons, it is not surprising that the Random Forest algorithm also performed well in predicting related videos, and thus links, in the graph, as the algorithm also considers nodes based off of similarity, except between decision trees instead of the nearest neighbors.

One interesting note is that as soon as we incorporated aggregate features into our algorithms, the performance of almost all the machine learning algorithms we used showed little change, decreasing in some metrics, with the important exception of Random Forest. As the aggregate features includes sums and differences of comment and view counts, we suspect that this discrepancy in performance is due to

the fact that videos are not necessarily related to each other based off of comment and view counts, so incorporating this information is mostly irrelevant. This also explains why KNN performance decreased, as comment and view count similarities do not contribute meaningfully to video relatedness, and will thus detract from link prediction accuracy. Furthermore, it also explains why Random Forest performance did not decrease, as the multiple decision trees could simply ignore these additional meaningless features, thus accounting for only the ones that dealt directly with accurate link prediction, like roles and genres.

When running our algorithm with a different number of roles for RolX, we saw little change in the predictive scores. This suggests that the patterns in performance decreases we saw stem from the additional aggregate features used rather than from the number of total roles incorporated.

Conclusion

In conclusion, because our link prediction problem is in essence a problem about determining whether two videos are related, the algorithms that were more suited towards determining similarity performed better overall, like KNN and Random Forest. Moreover, only the features that directly dealt with similarity were useful for our predictions, as videos with similar genres and similar placements in the YouTube network were more likely to be related to each other, as not only the videos, but the videos they were related to, were similar in many aspects. Overall, we have seen that RolX is suitable for predicting related videos in YouTube and that genre can act as an appropriate substitute for communities when it comes to deciding whether two YouTube videos are related. Our algorithms also showed little promise in the ability to predict nodes for an unseen graph, which made sense because the features for the RolX in different graphs were in different spaces.

Future Work

In the future, with more time and computing resources, we could experiment with using other methods of community extraction, like Stochastic Block Modeling, to use as our community features for link prediction, instead of explicitly using genre. Unfortunately, due to time and memory constraints, we were unable to incorporate the entire dataset over all time frames, since the resources needed to load all of the graphs was rather massive, but we could attempt this in the future with additional computing resources. Finally, we could also experiment with using the raw features extracted from the role-to-features (sense-making) matrix generated from RolX.

Contributions

1) Han Lin Aung: Performing preliminary data analysis and coding initial infrastructure to begin processing of the YouTube related videos network.
2) James Li: Initial problem formulation, data gathering, researching background information on RolX, analyzing results, writing up the report, and making the poster.
3) Justin Xu: Also coding up the different algorithms, running tests comparing their accuracies, and creating graphs of the YouTube related videos based off of the resulting information.

Project Code

https://github.com/HanLinAung88/CS224W-Youtube-Project

REFERENCES

[1] Al Hasan, Mohammad, et al. "Link prediction using supervised learning." *SDM06: workshop on link analysis, counter-terrorism and security*. 2006.

[2] Atzmueller M. (2014) Social Behavior in Mobile Social Networks: Characterizing Links, Roles, and Communities. In: Chin A., Zhang D. (eds) Mobile Social Networking. Computational Social Sciences. Springer, New York, NY

[3] Henderson, Keith, et al. "Rolx: structural role extraction & mining in large graphs." *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012.

[4] Sanders, Lloyd, et al. "Introduction to Link Prediction." D-GESS: Computational Social Science.

[5] Santos, R. L., et al. "Characterizing the YouTube video-sharing community." *Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil, Tech. Rep* (2007).