Link Prediction and Hub Detection: Analysis of the YouTube Graph Network

Ethan Brown (esbrown), Brendan Edelson (bedelson), and Elijah Taylor (elijaht)

Sunday 9th December, 2018

Abstract

This report outlines our investigation of the following problem: Can we accurately predict correlated YouTube videos and video importance by formulating the YouTube video network as a directed graph?

We have reviewed current literature regarding link prediction, hubs and authorities, and webpage reputation in directed graphs. We have formulated opinions on this literature and speculated how we will be able to contribute to it. We have applied experimental algorithms to these problems and compared them to widely acknowledged algorithmic standards. In doing this, we have applied state-of-the-art literature and analyzed how generalizable these algorithms are on the YouTube video network.

1 Introduction

Over 1 billion hours of YouTube video content are consumed per day worldwide [2]. As such, the YouTube video network stands as one of the larger content networks on the web. Due to the growing amount of literature in the analysis of social graphs, we have decided to see if current findings in the field generalize to the YouTube video network. Specifically, we aim to analyze two related problems.

Firstly, we investigate whether we can accurately predict correlated YouTube videos. YouTube relies heavily on users continuing to browse on its platform through mechanisms such as the related videos section on the browser and auto-playing videos after a current video ends. As such, the problem of accurately finding correlated videos is of utmost importance for the platform, especially one of its size.

Secondly, we investigate the notion of authorities and hubs within the YouTube video network. Just as it is important to refer users to related videos on the YouTube platform, it is also critical to refer them to hubs within their various communities. Whether the notion of community is through local graph structure or video category, we want to be able to refer users to other similar videos that are reputable within the network. We also examine what metrics can be used to evaluate a video as "reputable".

2 Related Work

2.1 Link Prediction

In the 2016 paper "A Survey of Link Prediction in Complex Networks", Martínez, Berzal, and Cubero analyze the different link prediction algorithms that previously existed in terms of effectiveness and computational complexity. Ultimately, the authors conclude that the effectiveness of a particular technique depends highly on the context of the problem and the structural properties of the network, which highlights the importance of analyzing the network before choosing a technique. However, while choosing a link prediction algorithm depends highly on context, in general, local approaches performed much better

than global approaches, and quasi-local techniques, which balance local and global techniques, also perform well on average. In addition to similarity-based methods, the paper also analyzes and suggests that statistical and algorithmic methods can be effective as well.

The analysis of these techniques addressed focused on undirected and unweighted networks, while not all networks fall under this category. Noticing this, we decided to try and apply the most successful algorithms from this paper to a directed graph from YouTube data, and to see whether we got similarly successful results.

2.2 Page Reputation: Authorities and Hubs

Both "Finding Authorities and Hubs From Link Structures on the World Wide Web", by Borodin, Roberts, Rosenthal, and Tsaparas as well as "What is this Page Known for? Computing Web Page Reputations" by Rafiei and Mendelzon deal with the idea of a web page's "authority" or "reputation" for specific topics.

The Borodin, Roberts, Rosenthal, and Tsaparas paper compares the existing algorithms for evaluating link analysis and formalizes criteria for comparing respective approaches. It starts by analyzing the PageRank algorithm, and moves on to analyze Kleinberg's algorithm, which it states as a "more refined notion" for the importance of web pages. The key finding in this paper is that the use of potential exploratory algorithms in "authority" analysis opens the door to finding successful statistical and machine learning techniques for ranking of linked documents. One warning presented by Borodin, Roberts, Rosenthal, and Tsaparas is that nearly all of the algorithms tested suffer from "topic drift", which limits the ability of the algorithms to correctly identify "authorities" of a specific field or topic. In order to minimize this effect, we decided to run "hub" detection algorithms only on networks of videos belonging to a single category, such as "Comedy".

The paper by Rafiei and Mendelzon presents much of the same information, referring to authorities on a certain topic "t" as pages with "high reputation on t". A page can acquire a good reputation by being linked to from many pages on the topic, or being linked to by pages with high reputations on the topic.

The Rafiei and Mendelzon report includes a section about the drawbacks of the concept of reputation. For example, the authors point out that if a company has a high reputation on a topic "t", it still may not receive a lot of recognition if topic "t" is not well represented on the internet. On the other hand, it is also emphasized that websites with many incoming links (like Twitter) will end up with high reputations for many topics, but will likely not truly be a good resource for said topics. Despite the fact that all of our data comes from the same website (YouTube), we took these potential downfalls into account as we attempted to compute and examine the reputations the pages on various topics.

Inspired by the Borodin et. al. paper, we decided to apply exploratory algorithms in the "hub" detection field to the YouTube video network, with the goal of comparing their results to Kleinberg's Algorithm. Kleinberg's Algorithm is widely considered the industry standard for "hub" detection, and we attempted to see if it outperformed other potential metrics on the YouTube network as well. We mixed in both established metrics and other newer, more experimental ones.

Included in these metrics are ClusterRank, introduced by Chen et. al., which outperformed PageRank and LeaderRank in their 2013 study on large scale directed networks, Lin Centrality, which was used successfully to identify nodes with "high centrality" by the Vigna and Boldi paper "Axioms for Centrality", and Leverage Centrality, which most effectively identified neighborhood hubs in the Joyce et. al. study on functional brain networks from 2013 [6][7][8].

3 Data Collection

We are using a dataset called "Statistics and Social Network of YouTube Videos", in which the data from a normal crawl of 100,382 YouTube videos is used to represent the videos on the website as a directed graph. A directed edge from video A to video B signifies that video B is in the "related videos" section for video A. The data is readily available at http://netsg.cs.sfu.ca/YouTubedata. We are using the crawl data from March 13, 2007.

4 Models, Algorithms, and Methods

4.1 Link Prediction

4.1.1 Jaccard Similarity

We can assign a Jaccard Coefficient to each pair of nodes (a, b) to find how similar the neighbors of a and b are by using the following equation:

$$J(a, b) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

4.1.2 Node2Vec & Data Formatting

To format the data so that it would be appropriate to use in logistic regression and machine learning models, we took advantage of node2vec. We used node2vec embeddings to represent each node in the YouTube network. In order to apply these embeddings to the task of link prediction, the edges were represented four different ways: as the concatenation, sum, average, and Hadamard product of two node embeddings. The output variable corresponding to each input (edge embedding), was a binary variable, where 1 indicates that an edge exists in the graph, and 0 indicates the absence of said edge. We then created a dataset of size corresponding to the true number of edges in the network, consisting of 50% edges that exist in the network (positive examples) and 50% edges that don't exist (negative examples). This dataset was then split into test and train sets using the test_train_split method from sklearn, with an 70/30 split between the train and test set.

4.1.3 Logistic Regression

We used scikit-learn's version of logistic regression as a binary classifier to predict whether or not an edge exists in the YouTube network. The goal of this model is to find a θ such that we maximize the following function:

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \tag{2}$$

The value of $\theta^T x$ is mapped to a value in the range $[0, 1]$. This allows the algorithm to serve as a binary classifier, where $h_\theta(x)$ gives us the probability that the the binary output variable is 1, in our case indicating that en edge exists. The model then uses stochastic gradient ascent to adjust the value of θ, as follows:

$$\theta_j = \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)})x_j^{(i)} \tag{3}$$

This gradient ascent step occurs several times in order to maximize the probability that the hypothesis is correct.

4.1.4 MLP Classifier

We used scikit-learn's implementation of a Multi-layer Perceptron classifier, which optimizes the log-loss function using the Adam optimization algorithm in order to predict an edge exists between two given nodes in the network. The log-loss function with c being the class label, and p being the probability

predicting c, is as follows:

$$L(p) = -(c * log(p) + (1 - c)log(1 - p)) \tag{4}$$

Additionally, We used a 3-layer fully connected neural network, with layer sizes 15, 15, and 5, learning-rate of 0.001, batch size of 200, L2 Regularization penalty of 0.01, and a Relu activation function.

4.2 Hub Detection

4.2.1 Hyperlink-Induced Topic Search (Kleinberg's algorithm)

The Hyperlink-Induced Topic Search (HITS) algorithm calculates authority and hub scores for web pages for a certain topic. The set of pages to be evaluated is created by the taking the top pages returned by a text-based web search, and then adding to the set all of the pages that are linked to from these pages as well as some of the pages that link to them. The authority and hub scores are calculated and updated over a series of iterations with three steps per iteration. Before the first iteration, we initialize the hub and authority scores of each page to 1. In each iteration we do the following:

Update the authority score of each page p with the following equation (n is the number of pages that link to p and each i is one of those pages):

$$auth(p) = \sum_{i=1}^{n} hub(i) \tag{5}$$

Next, we update the hub scores of each page p with the following equation (n is the number of pages p links to and each i is one of those pages):

$$hub(p) = \sum_{i=1}^{n} auth(i) \tag{6}$$

The last step of each iteration is to normalize the values. We can do so by dividing each hub score by square root of the sum of the squares of all hub scores, and the same thing with the authority scores.The hub and authority values will eventually converge after some number of iterations.

4.2.2 ClusterRank

ClusterRank is a local ranking algorithm which takes into account not only the number of neighbors and the neighbors' influences, but also the clustering coefficient of each node. This helps to quantify the influence of a node by taking into account not only its direct influence and influences of its neighbors, but also its local network structure.

Mathematically, it is calculated using the following equation:

$$s_i = f(c_i) * \sum_{j \in \Gamma_i} (k_j^{out} + 1)$$

where c_i is the clustering coefficient of c, $f(c_i) = 10^{-c_i}$, and k_j^{out} is the out degree of node j.

4.2.3 Lin Centrality

Lin Centrality represents the influence domain of an organization and is defined as the extent to which the opinion of an organization is sought both directly and indirectly. It considers closeness to be the

inverse of the average distance, instead of the inverse of a sum of distances. It also values nodes with a larger coreachable set. Mathematically, it is defined as:

$$\frac{|y|d(y,x) < \infty|^2}{\sum_{d(y,x)<\infty} d(y,x)}$$

Lin Centrality has largely been ignored in popular literature, but it provides a reasonable solution to the problem of node centrality [7].

4.2.4 Leverage Centrality

Leverage Centrality considers the degree of a node relative to its neighbors and operates under the principle that a node in a network is central if its immediate neighbors rely on that node for information [5]. It is a measure of the relationship between the degree of a node and the degree of each of its neighbors. Mathematically, it is defined as:

$$l_i = \frac{1}{k_i} \sum_{N_i} \frac{k_i - k_j}{k_i + k_j}$$

where k_i is the degree of node i.

5 Results & Findings

5.1 Link Prediction

5.1.1 Jaccard Similarity

For our baseline method, we calculated the Jaccard similarity score for each pair of nodes, and predicted an edge if the score for a pair of nodes is > 0.5, and no edge otherwise. We then built a confusion matrix using our predicted edges and actual edges.

KEY	
True +	False +
False −	True −

Jaccard	
6800	17270
17070	95094926

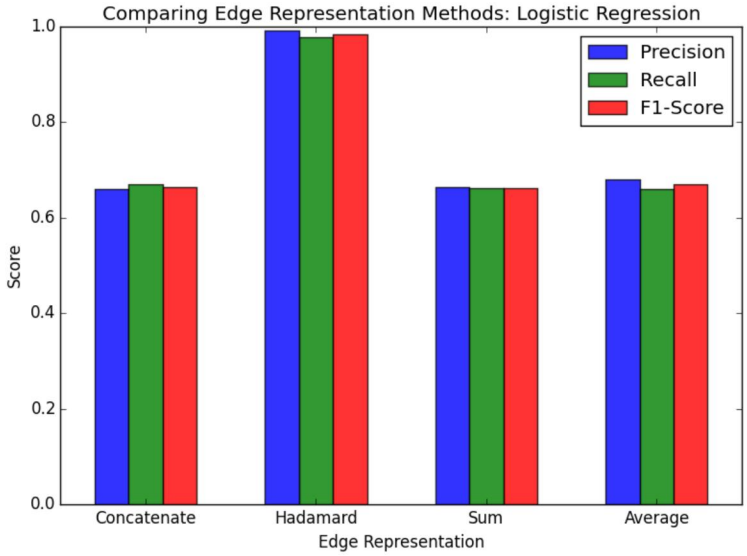
Precision = 0.282509348
Recall = 0.284876414
F1 = 0.283687943

As we can see, this is not a very effective measure for edge prediction, but it provided a useful baseline for us to improve upon with later methods.

5.1.2 Logistic Regression

We used our datasets of positive and negative examples from the 'Comedy' category, where n=13,796. We tried four different methods of representing an edge given the node representations of two nodes. We either concatenated, took the Hadamard product, summed, or averaged the components of the node2vec representations. We then computed and averaged the decision matrices for each of the four methods, computing the precision, recall, and f1 scores for each method.

In the chart below, the average precision, recall, and F1-scores are shown for each method of representing the edge.



Below we have included the average confusion matrices for each method of edge representation to show the number of true and false negatives and positives.

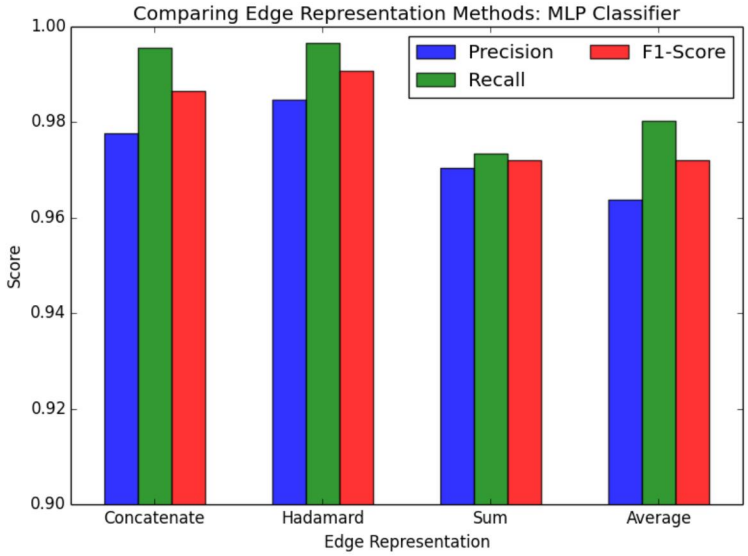
KEY		concatenation		Hadamard		sum		avg	
True +	False +	2427	1254	3619	32	2424	1230	2485	1169
False −	True −	1194	2447	81	3590	1243	2427	1284	2384

The Hadamard product performed by far the best with an F1 score of 0.985. Note that this product is an element-wise multiplication, so we are treating the directed edges as if they are undirected, as $a => b$ would have the same representation as $b => a$. In general, the concatenate, sum, and average methods performed similarly to each other, but significantly below the Hadamard product, having F1 scores of 0.665, 0.662, and 0.670, respectively.

5.1.3 MLP Classifier

We used our datasets of positive and negative examples from the 'Comedy' category, in which we either concatenated, took the Hadamard product, summed, or averaged the components of the node2vec representations. We then computed and averaged the decision matrices for each of the four methods, computing the precision, recall, and f1 scores for each method.

In the chart below, the average precision, recall, and F1-scores are shown for each method of representing the edge (note the y axis is from 0.9 to 1).



KEY		concatenation		Hadamard		sum		avg	
True +	False +	5356	123	5400	84	5348	163	5284	199
False −	True −	24	5479	18	5480	146	5346	106	5394

Overall, all four methods performed well using the MLP Classifier, but surprisingly, the Hadamard product performed the best with an F1 score of 0.9906. Since the Hadamard product is an element-wise multiplication, we disregard the fact that we are looking at a directed graph, as $a => b$ would have the same representation as $b => a$. However, the concatenation method was close behind with an F1 score of 0.9865, so it is possible that the concatenation works better on some datasets. Meanwhile, the sum and average methods performed the worst (though still extremely well), with F1 scores of 0.9720 and 0.9719 respectively.

5.2 Hub Detection

For our authority and hub prediction, we ran Kleinberg's algorithm, which we used as a baseline algorithm due to it's well known reputation within the field of hub detection. To see how the algorithm performed while limiting potential topic drift, we decided to run our baseline test of Kleinberg's algorithm on videos tagged with the category "Comedy". To do this, we used the same videos as used for the Link Prediction.

As a first analysis using Kleinberg's algorithm, we graphed the authority and hub scores of videos against the number of views that the video had, using a video's view count as a rough proxy of its importance within the video network. We would expect that there would be a correlation between the authority ranking of a video and its view count.

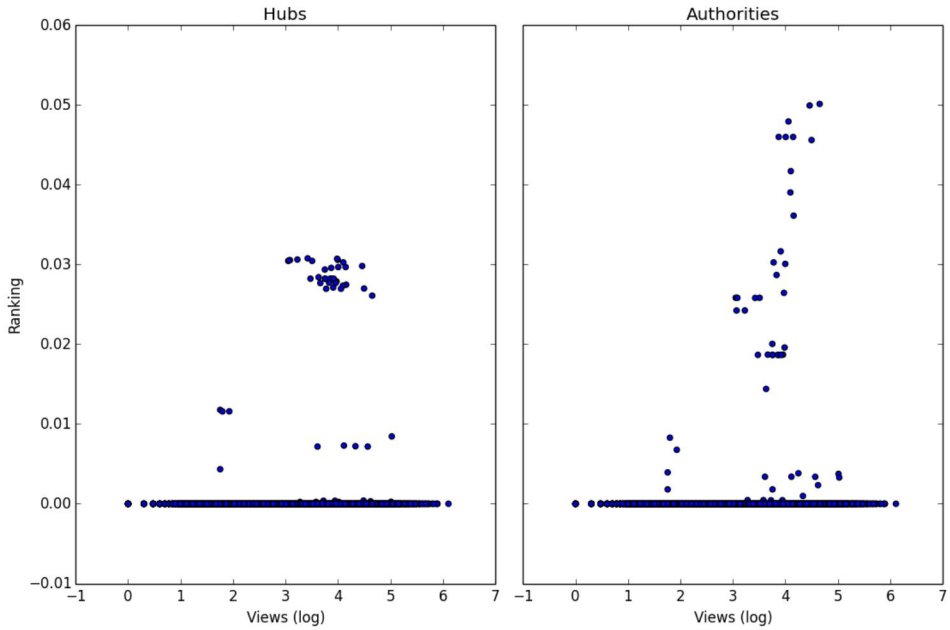


Figure 1: Authority and Hub Rankings vs. View Counts For Comedy Videos

As we can see in Figure 1, there is a correlation between authorities and view counts, but there are many videos with a authority score close to 0, regardless of the video's view count. This makes sense, as a video's view count is not directly tied to its role as an authority in its category. More importantly, we can see that videos with a high authority score almost universally have a high view count as well, which matches what we would expect. We can also notice that all of the nodes with high hub scores have similar view counts – this is not necessarily what we would expect but implies that nodes with high hub scores in the YouTube network tend to have relatively high view counts as well.

To compare Kleinberg's algorithm to the more exploratory algorithms, we first generated lists of the 50 nodes with the highest centrality/hub scores for each algorithm. To judge an algorithm's effectiveness, we first took each node in those lists of 50 and calculated the average view count of the videos that it points to in the network. We then used histograms, seen below in Figure 2, to help us view this distribution.

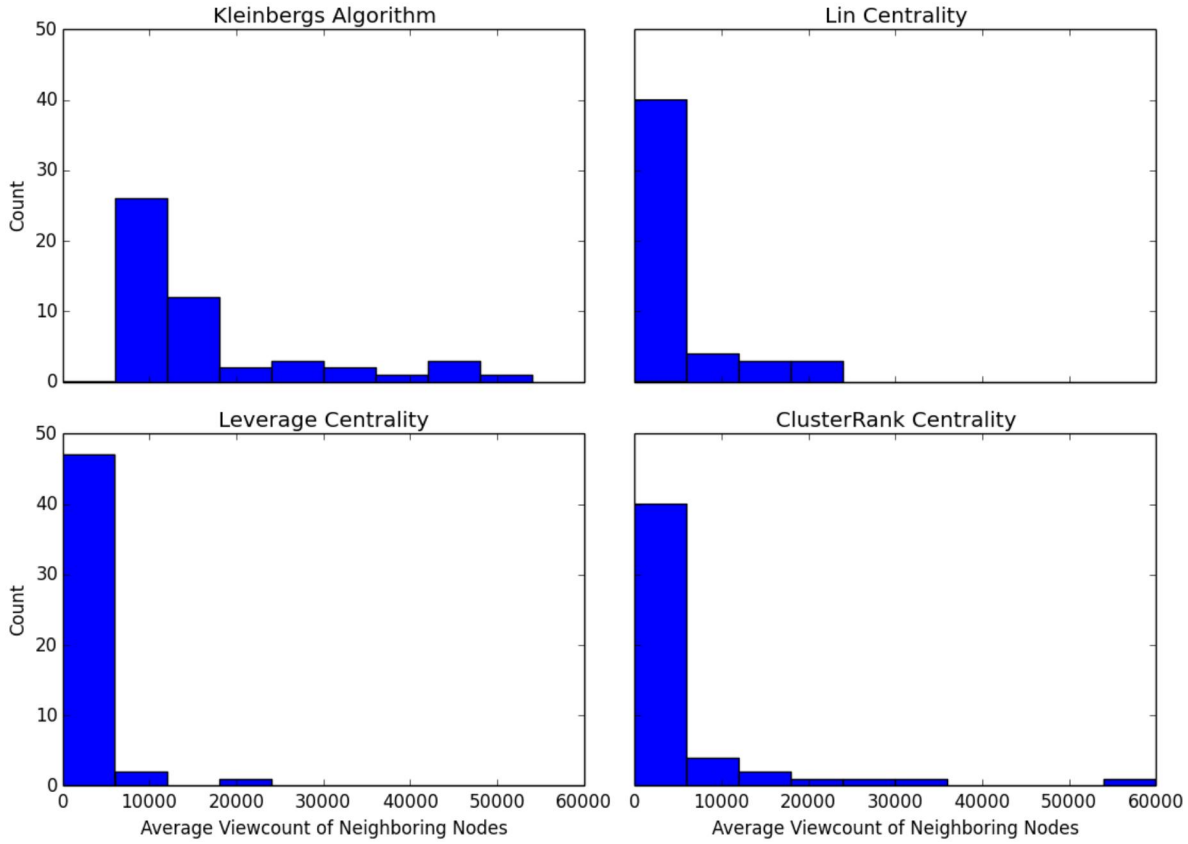


Figure 2: Avg. Neighbor View Counts For Top 50 "Hubs"

As we can see from Figure 2, Kleinberg's Algorithm is by far the most successful "hub" detection algorithm in finding nodes that point to videos with high view counts. ClusterRank performs slightly better than the other experimental algorithms, but still does not come close to matching the performance of Kleinberg's Algorithm.

6 Conclusion

In our experimentation with applying machine learning models to the task of link prediction, we discovered that both the model and way we choose to represent the inputs to the model are highly consequential. In general, we saw that the MLP Classifier outperformed the Logistic Regression model, but due to our experiments with different ways to represent an edge using node2vec, we were able to learn a bit more about how to most effectively predict edges in the YouTube network. In both machine learning models, we saw the best results when representing the edge as the Hadamard product of the node2vec representations of both the source and destination nodes. As alluded to in the results section, this method removes directionality from our representation of an edge. The fact that removing the directionality improved how well the model performed was a bit surprising at first, but makes some sense after further consideration, as the problem becomes slightly less constrained when it doesn't consider which direction the edge is directed. Note that directionality is also lost when taking the sum or the average of two node2vec representations, but we found that the Hadamard product outperformed the other methods in both models.

With regards to Hub detection in a large scale directed network, there is no quantifiable metric we can use to state an algorithm's absolute performance on a given graph. However, we believe that we have

identified metrics that we can successfully use as a proxy for 'hub' success on the YouTube video network. The most impactful of these, seen in Figure 2, clearly identified Kleinberg's algorithm as the most successful 'hub' detection algorithm for the YouTube video network out of the ones that we implemented. This is significant because it reaffirms existing literature in the field. Kleinberg's algorithm is widely considered the gold standard for hub detection [1]. We took experimental algorithms that had performed well on other datasets and compared them to Kleinberg's algorithm on the YouTube video network to see if it confirmed existing literature or if these experimental algorithms outperformed expectations.

One factor that we controlled for in our analysis was Kleinberg's algorithm's susceptibility to topic drift. We did this by only analyzing the videos in the YouTube network with the 'Comedy' category tag. We are very interested in scaling this analysis to the full graph. However, it is also worth noting that in many ways, finding hubs within a category can be more useful and applicable in regards to video recommendation than finding hubs within the overall graph. To further this study, we would also like to incorporate more experimental algorithms in the hub detection field to see if we can find any that match the performance of Kleinberg's Algorithm on the YouTube video network.

7 References

[1] Borodin, A., Roberts, G. O., Rosenthal, J. S., and Tsaparas, P. (2001). Finding authorities and hubs from link structures on the World Wide Web. Proceedings of the Tenth International Conference on World Wide Web - WWW 01. doi:10.1145/371920.372096.

[2] https://techcrunch.com/2017/02/28/people-now-watch-1-billion-hours-of-youtube-per-day/.

[3] Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. 2016. A Survey of Link Prediction in Complex Networks. ACM Comput. Surv. 49, 4, Article 69 (December 2016), 33 pages. DOI: https://doi.org/10.1145/3012704.

[4] Davood Rafiei and Alberto O. Mendelzon. 2000. What is this page known for? Computing Web page reputations. Comput. Netw. 33, 1-6 (June 2000), 823-835. DOI=http://dx.doi.org/10.1016/S1389-1286(00)00078-5.

[5] CentiServer - Leverage Centrality. http://www.centiserver.org/?q1=centralityq2=Leverage-Centrality.

[6] Chen, D., Gao, H., Lü, L., Zhou, T. (2013). Identifying Influential Nodes in Large-Scale Directed Networks: The Role of Clustering. PLoS ONE, 8(10), e77455. doi:10.1371/journal.pone.0077455.

[7] Boldi, P., Vigna, S. (2014). Axioms for Centrality. Internet Mathematics, 10(3-4), 222-262. doi:10.1080/15427951.2013.865686.

[8] Joyce, K. E., Laurienti, P. J., Burdette, J. H., Hayasaka, S. (2010). A New Measure of Centrality for Brain Networks. PLoS ONE, 5(8), e12200. doi:10.1371/journal.pone.0012200.

8 Github

The link to the github repo for this project is https://github.com/esbrown/project224w

9 Teamwork

We all did equal work.