YouTube as an Indirect Social Network: The Video-Video Network and Related Video Prediction

Fabian Chan (fabianc), Karan Rai (karanrai), Stephanie Tang (svtang)

1 Introduction

The YouTube social network is often seen as a typical publisher-subscriber network. Research has shown that while YouTube deviates from typical online social networks due to its publisher-subscriber nature, social interactions occur in the form of indirect relationships through comments on videos or video reaction to other videos. However, not much in-depth work has been done examining the indirect social aspects of the video-video network of YouTube. By analyzing the network of related videos, we can learn about its structural properties, especially around popular videos, and in turn, hypothesize about indirect user relationships through videos and also features that potentially lead to popularity of videos.

Related videos serve as a very important aspect of YouTube allowing users to easily navigate between videos that they would enjoy watching hence, being able to predict related videos for new videos is a very important problem. This is an application of Link Prediction but due to the nature of this graph, being across videos rather than people, standard techniques may not be enough since we should be looking at both structural graph properties as well as node/video properties. In this report, we compare different link prediction algorithms and adapt them for the application of predicting related YouTube videos.

2 Related Work

Previous research focused more on the publisher-subscriber model of YouTube and the comment network of YouTube. Wattenhofer, Wattenhofer and Zhu [2] found that YouTube differed from traditional online social networks with its lack of homophily, reciprocative linking, and assortativity and social interactions occur in the form of indirect relationships through comments on videos or video reaction to other videos. However, this paper looks at popularity based on subscriptions only, and does not consider video popularity based on views and with respect to the comment network.

Other research has also looked at some characteristics of the video to video network in the form of related video networks and video response networks. Benevenuto et al [3] examines a characterization of a social network created by video response interactions among users in YouTube, finding that the network's degree distributions follow power laws and fall in line with many other real-world networks. The user interaction network shows a structure similar to the Web graph, where pages with high in-degree tend to be authorities and pages with high out-degree act as hubs, whereas other social networks exhibit a significant degree of symmetry. Chen et al. [1] looks at the social networking aspect of YouTube and analyzes different kinds of YouTube networks (users, videos, subscriptions etc.). Using counts of views/ratings/comments for the YouTube network from 2007 and 2008 along with the fact that 40.3% of users have no friends, the researchers conclude that user-user social networks have less impact than video-video networks in YouTube. Hence, we are more interested in doing more in-depth work to examine the indirect social aspects of the video-video network of YouTube and using these insights to better predict related videos.

Proximity based link prediction is a quite common method and are generally presented in most link prediction surveys [5][7]. These techniques generally calculate some scores between all pairs of nodes and then predict the top scores as new edges. A lot of work has been done to develop new and better such scores as in [7][8][10]. More recent work focuses more on using supervised learning where they employ these proximity scores as features leading to higher accuracy ([8]).

One challenge in link prediction is in building a model that cleverly combines both information about the network structure and information about the nodes (videos) themselves in a principled manner. Backstrom and Leskovec [9] propose an algorithm based on supervised random walks that performs link prediction by combining these two kinds of information. Predictions are made based on the scores of a random walk, but the edge transition probabilities between pairs of nodes are learned based on a similarity score between their associated metadata. This similarity score is learned using a gradient-based optimization technique that minimizes a loss function defined over the set of training examples. The supervised random walk algorithm mentioned in the paper outperformed many other approaches, both supervised and unsupervised.

According to Lichtenwalter et al. [8], while unsupervised techniques are popular in link prediction, those techniques struggle to take into account the unique properties found in networks. Supervised learning is not new to link prediction, but typical problems arise around high class skew (sparse networks). However, training a model can capture important interdependency relationships between topological properties. [8] also presents a list of features used in supervised learning for link prediction. Furthermore, the authors discuss many local proximity based scoring techniques to predict new edges and then propose a new scoring method based on a depth-limited BFS. A high score is given to a node based on amount of flow possible between the two nodes. They go on to use this along with other features in a supervised training model as well. We make use of a modified version of this algorithm for our application.

3 Problem Statement

We are given a YouTube graph where nodes are videos and edges represent links between a video to its related videos. We analyze this video-related video network in two main phases. In the first half we characterize our network in terms of structural properties to understand how they relate to other Online Social Networks (OSNs) and Random Graphs. More specifically, we look at degree distributions, centrality/reciprocity measures etc. We also compute PageRank/HITS scores and explore any trends that they follow with respect to video properties like age, views, comments etc. Finally, we calculate clustering coefficients and check for the presence of small world phenomenon and compare our results with those of random graphs and other OSNs.

The second phase entails link prediction. Given a set of already existing videos and their related videos, the aim is to predict new edges i.e. more related videos based on the graph structure as well as node properties. The main application of this is to provide better related video coverage so that both publishers and viewers can benefit. Also, this can be used to predict links for new videos that get added to the network. We survey four methods and evaluate them in context of the YouTube related video network: proximity scoring, supervised random walks, PropFlow, and supervised learning algorithms, along with variations in scoring/weighting methods and different combination of video metadata and graph structure features.

4 Dataset

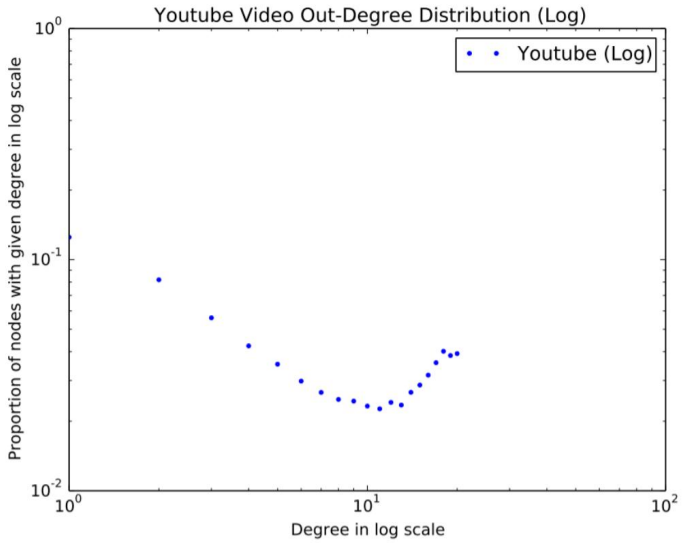
We will be using the related video YouTube data set from Simon Fraser University [4]. This data set's graph is defined as follows: each node represents a video and each directed edge from A to B denotes that A's top 20 related video list includes B. Each video has some metadata, including the uploader, video age, category, length, number of views, rating, number of ratings, and number of comments. For our analyses, we will be looking specifically at the related video data acquired in March 1, 2007, which has 155513 videos (crawled over 3 depths of a BFS starting from an initial set of videos).

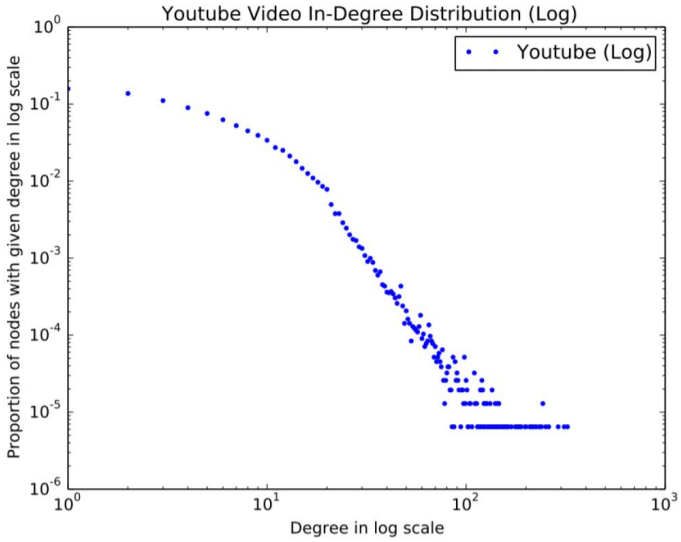
5 Related Video Graph Characteristics

5.1 Degree Distribution and Power Law

The out-degree distribution of the related video network has a somewhat even distribution. There are more videos with just a few related videos, which makes sense considering there are many newer videos that have not yet established connections with other nodes. We see a slight increase of nodes with 20 out-degree, but the data set only looked at the top 20 related videos for each video, where the initial video set is the crawl day's popular videos. The distribution does not look like other graph models encountered in class, but this is most likely due to the nature of the data set.

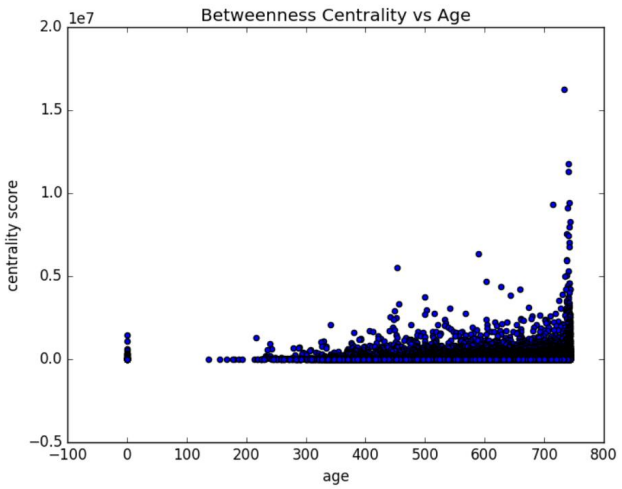
We do see something that resembles a power-law distribution in the in-degree distribution. Since new videos arrive one at a time and form edges with existing nodes, older videos are expected to have higher in-degree values. Compared to other random graph models and real-world networks encountered in class, the in-degree distribution most closely resembles the collaboration network (from PSET1).

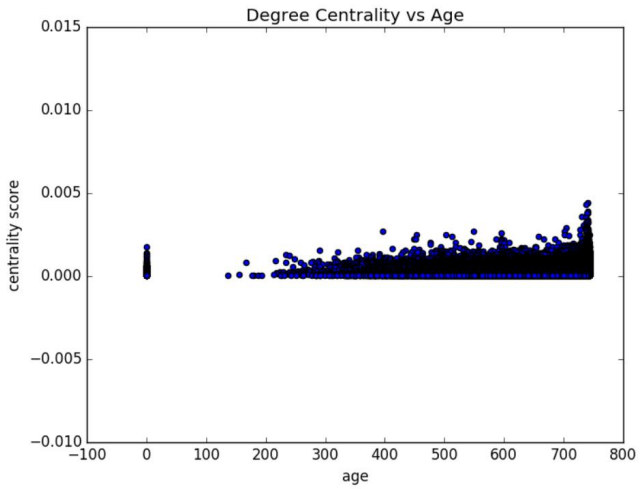




We found that the average path length between any pair of nodes on a directed path is approximately 14.174991 and on an undirected path is approximately 7.574534. Even in the undirected case, small world (approx. 6 degrees of separation) does not seem to quite hold for the whole related video network. This may have to do with the fact related videos have less of a social aspect than person-to-person relations and that related videos are likely found via keywords and other topic tags computed by YouTube's algorithms.

5.2 Measures of Centrality Compared to Video Age





We hypothesized that the network may exhibit some characteristics of the Preferential Attachment

model. For the network of depth 2, we computed the betweenness and degree centrality of each video and plotted them against their age. The two graphs are presented above and both show a pattern where videos of higher centrality are more likely to be of higher age compared to those of lower centrality, suggesting that the longer a video has existed, the higher centrality score it tends to hold, which is a trend that agrees with the preferential attachment model.

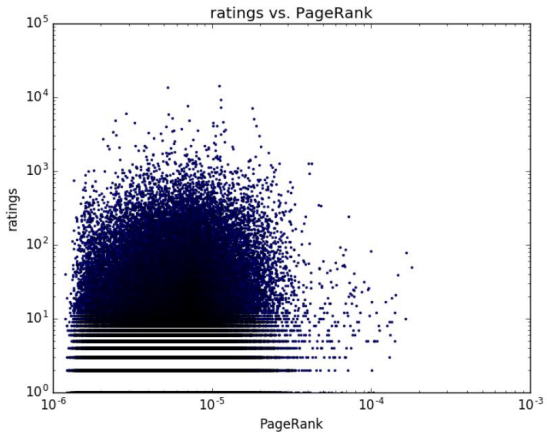
5.3 Clustering Coefficient

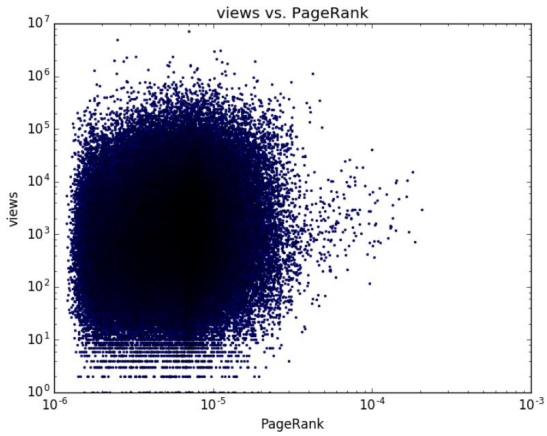
We find that average clustering coefficient for the entire graph is about 0.52. Average clustering coefficient for each category subgraph is also very similar, lying in the range of 0.46 for the "Comedy" subgraph up to a value of 0.57 for the "Music" subgraph, but the difference is not significant enough to make any conclusions. Also, the high clustering coefficient for our graph is likely an artifact of the crawling process since we are starting from random videos and performing a BFS for up to a depth of 3 for 20 neighbors each.

As expected for random graph models, the Erdos-Renyi model has a very low clustering coefficient in the order of 10^{-5} and the Preferential Attachment model has about 0.0022. This can be explained by the small-world phenomenon that is present in real-world networks. Comparing with other OSNs, we find that it is even higher; for example, [6], user-network such as Flickr has a clustering coefficient of 0.313 while the YouTube user network has a clustering coefficient of 0.136. This suggests that the video network of YouTube is much more connected, which makes sense as related videos are vital to YouTube. This also further strengthens our thought that video-video networks may be more important than user networks in YouTube.

5.4 PageRank and HITS

We computed the PageRank and Hub/Authority scores for all videos in the graph. We were not able to detect any special characteristics of the videos with high scores but we did observe that high PageRank scores did not necessarily correlate with videos that have high rating/comments/views. We found videos with either high ratings/comments/views or high PageRank scores but not both which was a bit surprising. Intuitively, we would expect videos with high views to also have high PageRank. Also, we observed that on a log scale, views vs. PageRank seems to very closely resemble a bivariate normal distribution. Such a distribution along with video properties could be potentially used to predict views for new videos that come up.





6 Link Prediction

6.1 Edge Prediction with Proximity

The aim is to predict more edges, i.e. predict more related videos for a given video. We will first do this via proximity score calculations. We compare certain scores between each pair of videos representing how

close they are based on the graph structure such as common neighbors etc. ([5]) Then, we sort the pairs based on this score and predict new edges for the highest scoring pairs. We will be using and comparing some of the most popular proximity score functions:

- Graph distance: (negated) Shortest path length between 2 nodes
- Common number of neighbors - For node x and node y, if we use Γ to denote set of neighbours, this would be $|\Gamma(x) \cap \Gamma(y)|$
- Jaccard's coefficient for the sets of neighbors - $|\Gamma(x) \cap \Gamma(y)|/|\Gamma(x) \cup \Gamma(y)|$
- Adamic/Adar score - $\sum_{z \in \Gamma(x) \cap \Gamma(y)} 1/\log(|\Gamma(z)|)$ [10]
- Preferential Attachment - $|\Gamma(x)|.|\Gamma(y)|$

This procedure has run-time quadratic in the number of nodes since we need to calculate the score for each pair of nodes, and therefore too time taking for our entire graph. Another critique of this method is that it does not use the video properties such as length, category, age etc. in any way. It does have an advantage in its simplicity yet being versatile since the score could be potentially defined in any way.

6.2 Edge Prediction with Supervised Random Walks

Next, we predict new edges using Supervised Random Walks ([5]). We combine node/edge features with the network structure, and run personalized PageRank from a node v' in order to compute the strength of candidate edges. The goal is to recommend a list of possible edges by interpreting the network as a set of labeled training examples.

For every source node v, we take a portion of its edges $(v, e^+) \in$ E and take them as positive training examples while taking all $(v, e^-) \notin$ E as negative examples. Our aim is to obtain higher proximity values for positive training examples compared to negative examples when we run the personalized PageRank algorithm. The PageRank algorithm essentially performs random walk with restarts from v using a set of transition probabilities we derive and estimate from a weight function $f(u, v)$ that assigns strength a_{uv} to edge (u, v):

$$a_{uv} = f(u, v) = \frac{1}{1 + \exp(-\beta^T x_{uv})} \tag{1}$$

where x_{uv} is a feature vector that gives some measure of similarity between nodes u and v, and β is the weight vector which we aim to optimize. The personalized PageRank algorithm is performed by the following sequence of steps:

1. Initialize transition matrix A where $(A)_{uv} := a_{uv} * 1[(u, v) \in E]$
2. Initialize PageRank proximity vector $p^{(0)}$ where $(p)_j^{(0)} := 1[j = v]$
3. Run until convergence:
$$p^{(t+1)} := (1 - \alpha)Ap^{(t)} + \alpha * p^{(t)} \tag{2}$$
4. Finally, rank nodes j by decreasing p_j.

In order to optimize our weight function $f(u, v)$, we want $p_i < p_j$ for every positive example j and negative example i so we define a loss function $h(x)$ where

$$h(x) = x^2 * 1[x > 0] \tag{3}$$

We will also add regularization to β with a parameter λ to prevent over-fitting; then our total loss function $J(\beta)$ is the following:

$$J(\beta) = \sum_{i,j} h(p_i - p_j) + \lambda ||\beta||^2 \tag{4}$$

This loss function is minimized using stochastic gradient descent. Although the partial derivatives ∇p_i with respect to β cannot be computed in an exact manner, they can be approximated with an iterative process as described in details in the Backstrom and Leskov [8] paper. However, we took the simpler approach of numerically computing the gradient using the difference quotient with a sufficiently small delta. This technique is easier to implement while still running with acceptable speed and accuracy.

The model parameters α, λ and the learning rate used in gradient descent were tuned using standard machine learning techniques such as cross-validation and grid search.

6.3 Edge Prediction with PropFlow Algorithm

The PropFlow method computes a score from a node to all other nodes in the network from which the top nodes can be predicted as new edges. This is based on a depth-limited BFS performed from the source node and scores are assigned according to the amount of flow that reaches a particular node. Flow is distributed evenly among a node's outgoing edges proportional to the edge weights. This method was introduced in [8]. We believe their algorithm presented in their paper had a few mistakes so we have corrected those shown below in Algorithm 1. We further modify the algorithm to calculate edge weights based on node/video properties. This allows us to include similarity of nodes into our edge weights thus allowing our flow to be distributed to nodes which are similar. In doing so, we get better predicted edges since we are using more information (i.e. video properties) to score them.

Data: Graph $G = (V, E)$, Source node v_s, Max depth d
Result: Score S_u for all nodes u reachable from v_s by at most d edges
insert v_s into $Found$
push v_s into $NewSearch$
set $S[v_s] \leftarrow 1$
for $CurrentDepth = 0$ *to* d **do**
  $OldSearch \leftarrow NewSearch$
  $NewSearch \leftarrow []$
  for $v_i \in OldSearch$ **do**
    $NodeInput \leftarrow S[v_i]$
    $SumOutput \leftarrow 0$
    for $v_j \in$ *neighbours of* v_i **do**
      $w_{ij} \leftarrow GetWeight(v_i, v_j)$
      $SumOutput += w_{ij}$
    end
    for $v_j \in$ *neighbours of* v_i **do**
      $w_{ij} \leftarrow GetWeight(v_i, v_j)$
      $Flow \leftarrow NodeInput \times (\frac{w_{ij}}{SumOutput})$
      $S[v_j] \leftarrow Flow$
      if $v_j \notin Found$ **then**
        insert v_j into $Found$
        push v_j into $NewSearch$
      end
    end
  end
end

Algorithm 1: PropFlow algorithm

6.4 Edge Prediction with Supervised Learning

Finally, we will train a model to predict edges. Using L1 regularized logistic regression, we can classify pairs of nodes as 0=noEdge or 1=edge. We will use video metadata as features as well as proximity scores. For a pair of nodes u and v, we will use the following metadata-based features:

1. whether u and v belong to the same category
2. whether u and v were uploaded by the same user
3. inverse difference in video length
4. inverse difference in the number of views
5. inverse difference in video ratings
6. inverse difference in the number of ratings
7. inverse difference in the number of comments.

In addition, to further improve recall while maintaining high precision, we will also use a pair of node's common neighbor score, Jaccard score, Adamic/Adar score, and their node degrees as features (as described

in section 6.1).

As with the previous methods, we will use 80% of edges present in the graph as positive training examples, and an equal amount of random node pairs that do not have an edge as negative training examples. To limit computation time and memory usage, we find the core nodes (defined as nodes that have at least k positive examples in training and testing sets). We train on all edges with at least one core node endpoint and on an equal random subset of no-edge node pairs with at least one core node. Our test set consists of any of removed edge with at least one core node end point and a equal random subset of no-edge node pairs with at least one core node.

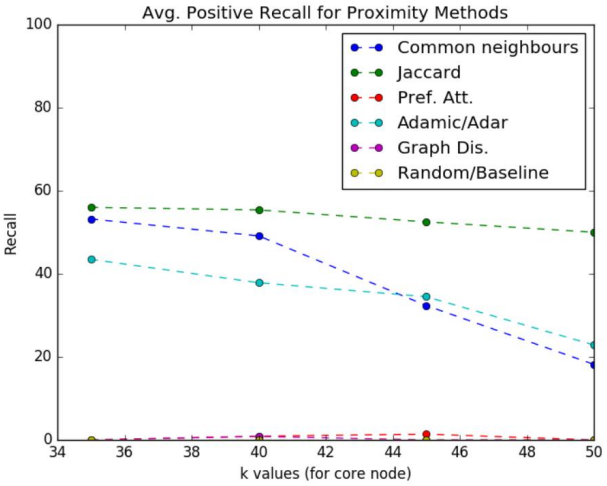
7 Results and Discussion

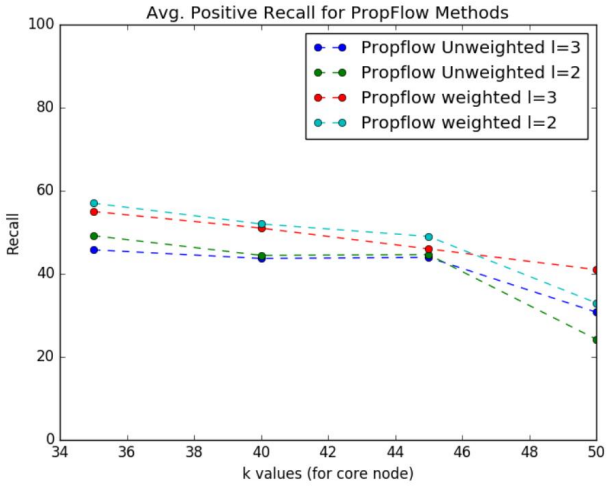
7.1 Evaluation Criteria

We follow evaluation criteria similar to those in [7]. First, we assign 80% of our edges as Training edges and the 20% as Testing edges uniformly at random. We then perform training using only our training edges. For testing, we only look at the "core" set of nodes. This only includes nodes that have at least k_{train} training edges and at least k_{test} testing edges. We count the number of testing edges occurring between these "core" nodes as k and then predict top k edges based on our training data. In the supervised learning algorithms, we classify potential edges into edges or non-edges. In all cases, we define our success as the Recall rate (i.e. fraction of correctly predicted edges over all true test edges). We also compare our results with a baseline which is just a random edge predictor between these "core" nodes.

7.2 Proximity and PropFlow Algorithms

We plot the Recall rate obtained versus various values of $k_{train} = k_{test}$ in the Figure below. Among the traditional proximity measures, we find that Preferential Attachment and Graph Distance perform very badly giving us only 0-1 % recall, very similar to predicting just random edges. This is likely due to the way the data was crawled. For example, the graph has many pairs of nodes that are a certain graph distance apart, but that alone does not tell if the nodes themselves are likely to be related since many videos can cover different topics, resulting in a diverse related video list. For preferential attachment, because two highly popular videos in the same topic will result in a high score and two highly popular videos in different topics will also result in a high score, the preferential attachment score cannot tell the difference between the two cases; this is unlike a human social network, where if two people are popular, they are likely to know each other. Adamic/Adar measure gives intermediate results of about 30 % recall while Common number of neighbours and Jaccard score give us upto 50-55 % recall. Unweighted Propflow is also able to attain similar recall of about 45-50 %. Propflow with edges weighted according to the video metadata helps improve the recall rate by a bit to around 55 %.



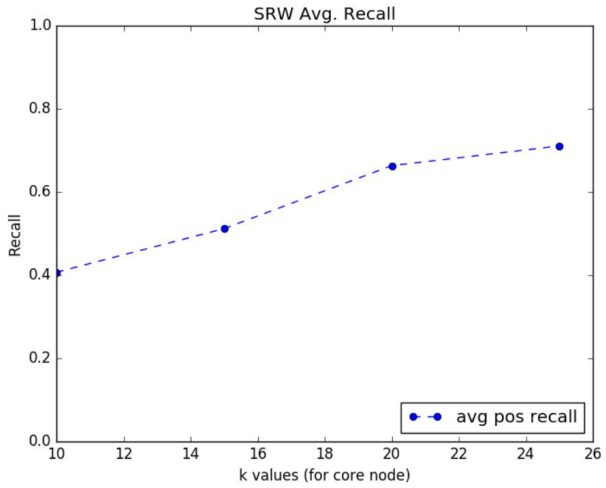


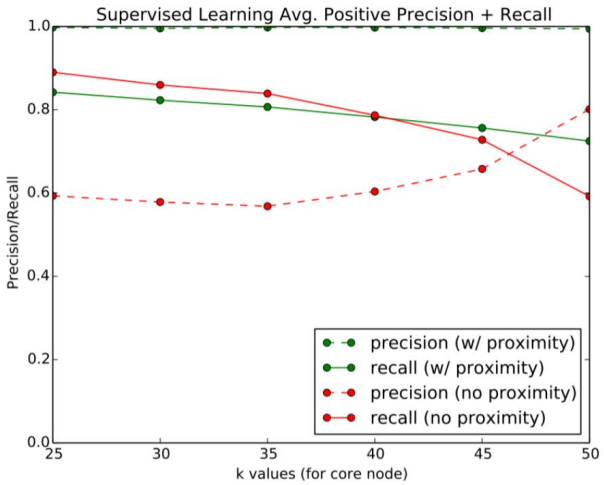
7.3 Supervised Random Walks (SRW)

We trained the algorithm using the set of metadata-based features defined in Section 6.4. The training process is quite slow and quadratic in the number of nodes in the graph. For every source node v, we ran supervised random walk algorithm on a subgraph defined by the set of nodes encountered by running BFS starting from v up to a depth of 4. The subgraphs obtained from this operation end up with roughly 500-800 nodes. Since the predictions for each node rely on every edge weight in the graph and because the number of features is very small, we believe there isn't much danger of overfitting despite doing this. Backstrom and Leskov similarly chose a reduced set of nodes to train on, for example only using 200 for the Facebook dataset. The algorithm generally converges within 50 iterations, where convergence occurs when the changes in total loss and in the norm of β became insignificantly small between iterations.

We plot the recall rate obtained verses various values of $k_{train} = k_{test}$ shown in the figure below. Note that unlike the other link prediction algorithms we have implemented, the supervised random walk algorithm only trains on out-going edges rather than all edges connected to the source node, so k_{train} is effectively halved during our training process. The features we trained are based purely on video metadata. As k increases from 10 to 25, recall increases. The model averaged over 70% recall for $k = 25$. This indicates that the model performs better for nodes with high degrees, and is a reasonable result because the random walk process is highly dependent on the out-degree of the source node. Furthermore, a high node degree provides more positive examples to feed into the training algorithm and conceivably results in a more generalizable vector of model parameters β.

One note that deserves some attention is that the recall scores depend heavily on the network structure (corresponding to the random walk part of the prediction algorithm) and not significantly on the similarity of videos based on metadata information (corresponding to the edge strengths of the algorithm). In fact, we discovered that if β is set to a zero vector, the prediction recall scores only drop by 10%. This suggests that the network structure of the YouTube related video network is more telling of the relationship between related videos rather than the video metadata, at least for Supervised Random Walks.





7.4 Supervised Learning Algorithms

We trained a logistic regression model on a variety of features. A baseline untrained binary classifier would uniformly and randomly classify node pairs as edge or no-edge, resulting in an average recall of approximately 50% and precision of 50%. With looking purely at video metadata and not considering the graph properties at all, the model averaged around $85 - 90\%$ recall rate (at approximately $59 - 60\%$ precision for $k = 25$ and is significantly different from the baseline model $p = 1.3734177721008256e - 12$). This high recall with lower precision indicates that the model is too liberal in classifying examples as having an edge (top right, red). As the number of training examples decreases (increasing k), recall drops while precision increases. To improve on this model, we combined previously implemented proximity scores as features as well, including common neighbors, Jaccard, and Adamic/Amar scores. This combination

implementation has better results, with a recall at around 85% and precision close to 100% at $k = 25$ (statistically different from baseline $p = 5.7030447076743596e-15$). Averaged over 10 trials for each k, we see that precision remains high, while recall fluctuates between $75 - 85\%$ (top right, green). The decrease in recall as k increases is likely explained by the fact that as k increases, the number of training examples decreases.

7.5 Comparision and Discussion

Overall, by comparing recall rates between the different algorithms, we find that the supervised learning method using a combination of video metadata and proximity scores as features does the best. Proximity scores using preferential attachment and graph distance did the worst (0 to 1 % recall), as bad as our baseline which is just a random edge prediction. The other proximity scoring methods do a bit better as they take into account who the neighbors are in some way, not just the number of neighbors. These proximity scoring methods do reasonably well in that they are simple and do not require training time, while the downside is that they do not take into account important video metadata such as category or uploader.

We see that PropFlow is comparable to proximity scores and that using video metadata-based features when generating weights in the modified PropFlow algorithm increases recall. The benefits of PropFlow include the depth-limited BFS which speeds up the process and that it also does not require training time. The recall is comparable to the best of the proximity scoring methods, but with some more features and hand-tuning of parameters for each feature, the modified PropFlow algorithm has the potential to predict edges better and faster.

Supervised random walks on average seems to work better on dense graphs (nodes with more edges). The best results from supervised random walks does even better than the best results from PropFlow or proximity scoring, but with not enough edges, supervised random walk does poorly in predicting edges on the YouTube related video network. Using video features does seem to help in recall for supervised random walks as compared to the previous algorithms.

The supervised learning algorithm using just video metadata as features is comparable to supervised random walks. Supervised learning actually does well with nodes with less edges, but with it's low precision, this is likely due to overfitting. The video features alone do not take into account the graph structure as proximity scoring does, so adding these proximity scores as features boosts recall while maintaining high precision. This combination of video meta-data and graph structure does best. This helps us confirm that traditional link prediction applications are not directly applicable to YouTube related videos networks due to the different graph structure, including higher clustering coefficient, degree distribution etc. and also the presence of video properties. We need to use video meta-data in some way or fashion (as we showed in Supervised Random Walks and Learning) in addition to graph structure in order to predict edges with high success rate.

8 Conclusion

We do not see small world phenomena in the YouTube related video network, but we found a power law distribution in degrees. The network has a high clustering coefficient. This is different from other OSNs which can be explained by the fact that the nodes are videos rather than people and hence, edges are not quite like friend relationships. In a survey of edge prediction algorithms, we find that a combination of video metadata and graph structure does the best in predicting edges in the YouTube related video graph, specifically via using supervised learning, rather than traditional proximity score based approaches.

Future work includes exploring more features to use in the PropFlow algorithm and hand-tuning parameters for each feature to yield better results. This PropFlow score can also be utilized as another feature for supervised learning methods to more accurately predict edges. Because video uploader seems to be a useful feature in supervised random walks and supervised learning, another potential direction to explore using more user/uploader data as features (which were not present in our data set), as for a network like YouTube, an uploader is likely to upload videos about related topics.

9 References

1. C.W. Chen et al. (Eds.): Intel. Multimedia Communication: Tech. and Appli., SCI 280, pp. 367–402.

2. M. Wattenhofer, R. Wattenhofer and Z. Zhu. The YouTube Social Network. In ICWSM, 2012.

3. F. Benevenuto, F. Duarte, T. Rodrigues, V. Almeida, J. Almeida and K. Ross. Understanding Video Interactions in YouTube. In MM '08 Proceedings of the 16th ACM International Conference on Multimedia, 671-764, 2008.

4. "YouTube Dataset", Netsg.cs.sfu.ca. [Online]. Available: http://netsg.cs.sfu.ca/youtubedata/. [Accessed: 19- Oct- 2016].

5. J. Leskovec, Assortativity, Resilience, and Link Prediction, 1st ed. 2015, pp. 35-53.

6. Mislove, Alan, et al. "Measurement and analysis of online social networks." Proceedings of the 7th ACM SIGCOMM conference on Internet measurement. ACM, 2007.

7. Liben-Nowell, David, and Jon Kleinberg. "The link-prediction problem for social networks." Journal of the American society for information science and technology 58.7 (2007): 1019-1031.

8. Lichtenwalter, Ryan N., Jake T. Lussier, and Nitesh V. Chawla. "New perspectives and methods in link prediction." Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2010.

9. Backstrom, Lars, and Jure Leskovec. "Supervised random walks: predicting and recommending links in social networks." Proceedings of the fourth ACM international conference on Web search and data mining. ACM, 2011.

10. Adamic, Lada A., and Eytan Adar. "Friends and neighbors on the web." Social networks 25.3 (2003): 211-230.

10 Teamwork Distribution

We all did equal work.