Project Milestone Report

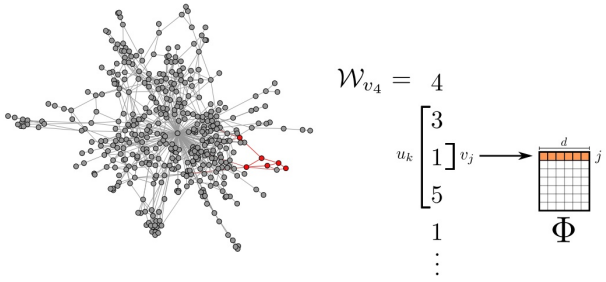
Yunjui Hsu, Chelsea Luo, Yang Zhang, Jiadong Lou

1 Abstract

With the rapid growth of online social networks and IoT networks, mining valuable knowledge from the graph data become important. For example, people have been using graph mining strategies to enhance the performance of personal recommendation, price prediction, communication anomaly detection. With the support of machine learning, online websites such as Amazon, eBay, Etsy are able to generate profitable recommendations for user using collaborative filtering methods. Often when people finished their purchase they can see the line: "User who bought this also bought.." and start buying more things from the website. Yet, there are around 100 million users around the world using Amazon. Even if they have one of the most precise algorithm that can predict what people wants, it is still very challenging to extract the network features from the data they conducted. The crazy amount of raw data generated everyday make the computation cost very high while making predictions. In our work, we are typically interested in improving the computational cost of one of the most powerful algorithm used recently from user (graph) predictions, graph embedding. For most previous works, computation cost is often not a consideration, and therefore we want to explore what we can do if our computation resource is constricted yet we still want to preserve a high precision.

2 Background

As mentioned in Sec. 1, our work aims to tackle the computation resource problem we might face in graph embedding methods. Up-to-date, the research Deepwalk seems to one of the most robust methods used for graph embedding. Briefly, as shown in Fig.1, the algorithm starts with sampling random walks for each node from the original graph as input to the model. For each sampled walk, it applies the skip-gram model (same as the skip-gram from NLP, as shown in Fig.2) to train and generate a embedding presentation (vector) for each node in the graph. Ideally, with enough sampling, two nodes with similar features or identity should have high cosine similarity in the embedding space generated from the model. As in later Sec.4, we show that given more resource of samples (for example, increasing the walk length of sampling) may significantly improve the performance when making predictions. Yet, model such as skip-gram has non-linear complexity, the computation cost increases dramatically even when increasing the walk length by one. The problem we want to solve is that if today we are running a startup, unlike Amazon or eBay that has literally unlimited computation resource, are we able to train a low require computation model by only sacrificing a small number of precision. Intuitively, we are planning to tackle this problem by using one of the most used feature in graph modeling: centrality. As we discover that in Deepwalk, all nodes are given the same amount of sampling resources while training. Intuitively, this is not a correct thing to do. For example in NLP researches, we often will take TF-IDF into consideration to decide what words are important and more representative before deciding rather or not we should account this word as feature. Similarly for graph sampling, we should also consider the fact that some nodes in the graph might be more important than others. We believe that if we reallocate the sampling resource using node centrality, given the same amount of sampling resource, we might be able to construct a more informative presentation for the nodes inside the graph. Since comparing to skip-gram models, the computation cost of computing centrality in graphs is much lower, therefore, with less amount of resource, we might be able to preserve the preciseness of our prediction model.



(a) Random walk generation. (b) Representation mapping.

Fig. 1: Deepwalk

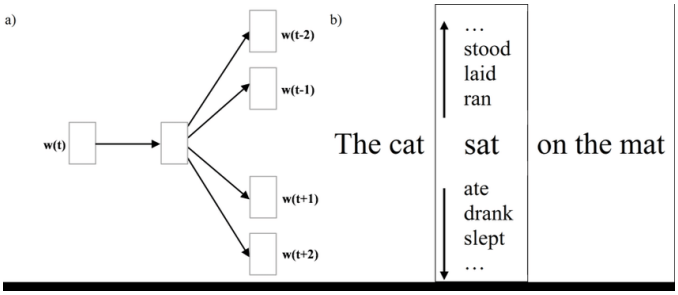


Fig. 2: Skipgram

3 Dataset

In this work, we used topical Twitter dataset for graph analyzation http://mlg.ucd.ie/aggregation/. For each dataset, it contains firstly the ground truth labels that labels which set of communities a user belongs to, secondly, tweet and retweet graph, follower links,

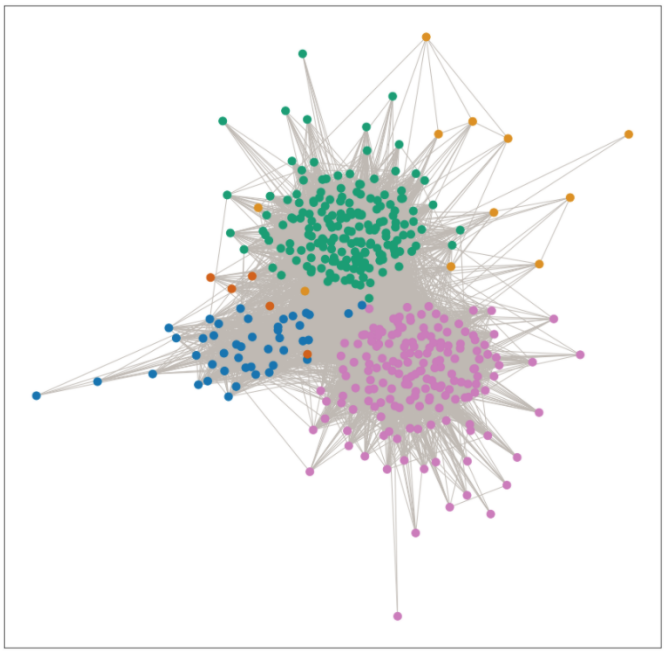


Fig. 3: Ground truth data for Politics-uk

and lastly, twitter tag features. The data we collected are all in pre-processed edge list format, without any raw tweets or any full context. In the dataset Politics-uk, there are 419 members that belongs to 5 political party affiliation. Coming along in the dataset we also have the information of 540k tweet and retweet, 27k follower links, and also 3k of some twitter tag features. Similarly, in the other dataset Olympics, there are 464 athletes assigned to 28 communities, following with 726k tweets, 11k follower links, and also 4k of some twitter tag features.

4 Milestones/Analysis

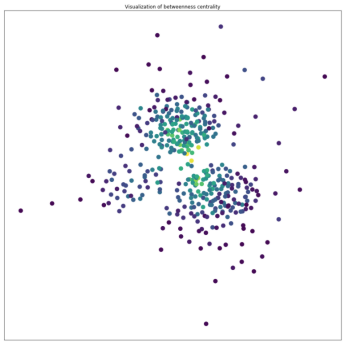
Up-to-now, we have start examine the use of Deepwalk model and try to make some initial predictions for our collected datasets. Things we have done can be briefly break into three parts. First, we have finished extracting the input and output labels from original datasets, as we decided to use twitter tag features as input graph. Secondly, we have also come up with some visualization on the datasets to provide a brief picture of what the dataset is like. And lastly, we have done some initial experiences with Deepwalk to test the performance under different circumstances.

5 Initial results

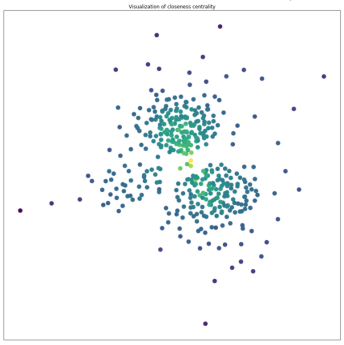
As mentioned in Sec.4, we have done some visualization on the raw data. Firstly is the visualization of the ground truth data related to each users, an example is shown at Fig.3. Each color in the graph presents the community that node belongs to. Afterwards, while we haven't implemented the centrality feature to our model yet, we have done plotting and showing what the centrality should be for each node presented using heatmap, as in Fig.4. In the research of graph theories, betweenness, closeness, degree, and eigenvector are often used to preserve different kinds of importance for nodes. Typically, degree centrality captures the count of neighbor for each node. Betweenness centrality captures the number of times a node acts as a bridge along the shortest path between two other nodes. Closeness centrality captures the importance by the average length of the shortest path between the node and all other nodes in the graph. And eigenvector centrality measures the influence of a node inside the network by the contribution interaction between nodes. Lastly, for our initial experiment with Deepwalk, we've tried different walk lengths when sampling random walks with the length of 3, 8, 10, and 20. For representation size we use 128, number of walks as 10 for each node and window size as 2. We applied Deepwalk for node representation and use it as input for liblinear model prediction. For each experiment, we tried using from 10% to 90% on nodes as training data. We used Micro-F1 to evaluate to result of our experiment. A sample result is shown as Fig.5

6 Next steps

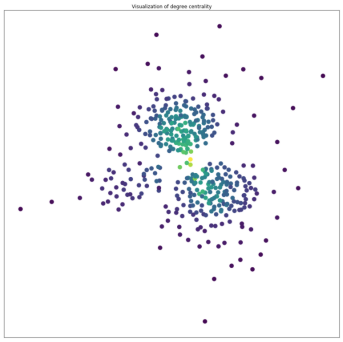
For our next steps, first we want to figure out how we can modify the released GitHub code from Deepwalk in order to reallocate the sampling resource by centrality of the nodes. Next for each type of centrality, we are going to run and experiment rather or not our hypothesis works. And lastly, find out the reason if or if not our hypothesis works. If there is more time, we will also expand this work from node classification to a more complete collaborative filtering process. The expect time of finishing these work will be around one month and we will figure out what to do next after we get our results.



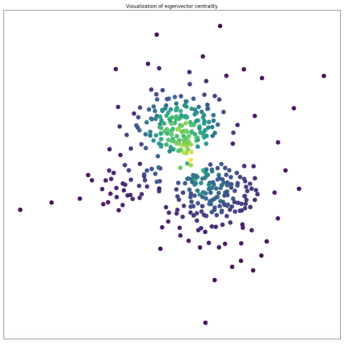
(a) Betweenness centrality



(b) Closeness centrality



(c) Degree centrality



(d) Eigenvector centrality

Fig. 4: Centrality graphs for Politics-uk

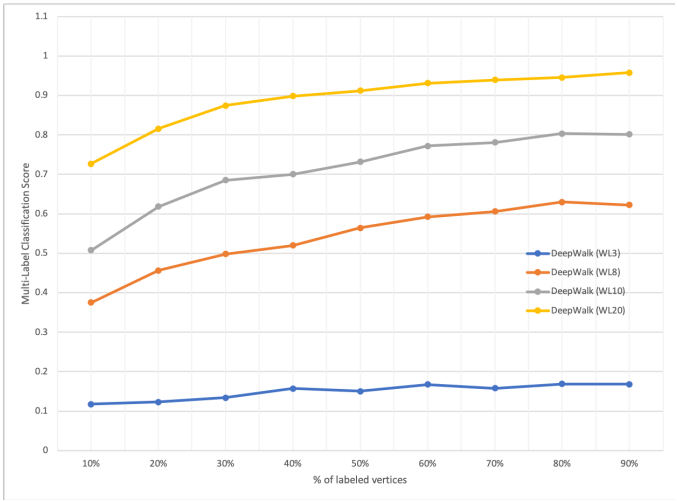


Fig. 5: Micro-F1 score for Olympics