# Project 2: Social Network Mining

William Nafack UID : 405725778

Lea Alcantara UID : 005872120

Gaurav Singh UID : 305353434

September 25, 2022

## 1 Part 1: Facebook Network

### 1.1 Question 1

A first look at the facebook Network. We first plot the undirected facebook Network as shown below in figure 1

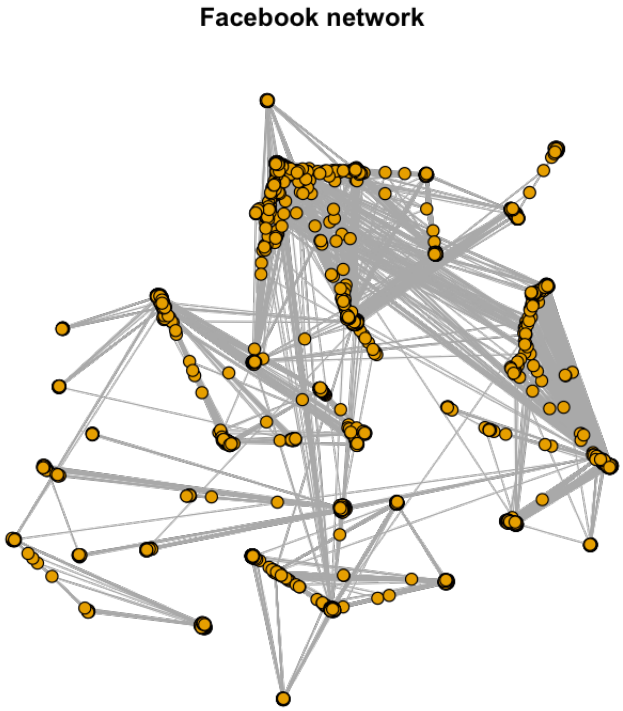



Figure 1: Facebook Network graph

### 1.1.1 Question 1.1

The generated network has **4039** node and **88234** edges.

### 1.1.2 Question 1.2

The Facebook Network is connected. This means that there are no isolated nodes in the network.

## 1.2 Question 2

The diameter of the network is **8**

## 1.3 Question 3

The average degree of the Facebook Network is **522.5**. The plot of the degree distribution can be seen below on figure 2

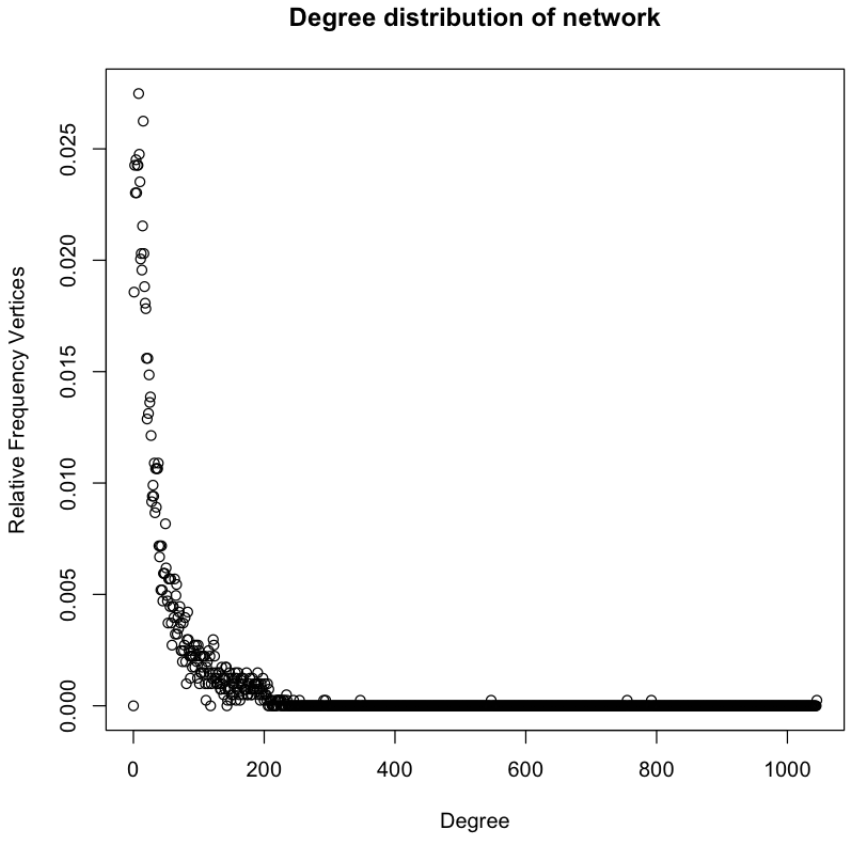



Figure 2: Degree Distribution of the Facebook Network

## 1.4 Question 4

The plot on figure 3 shows the degree distribution in the log-log scale and an estimation of the slope of the line.

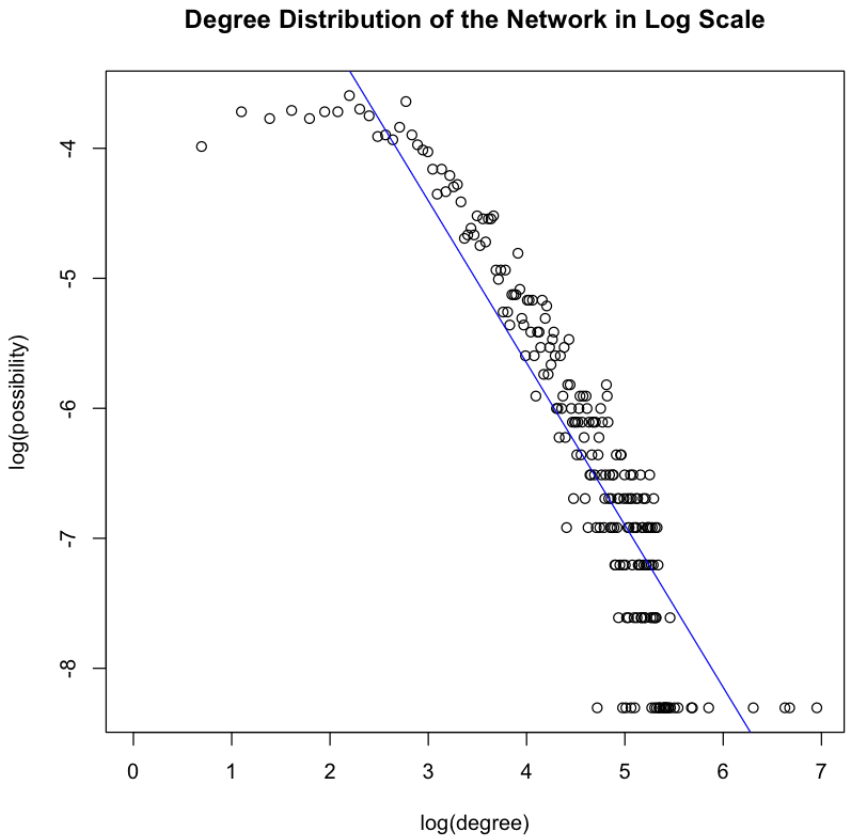



Figure 3: Fitted line to Degree distribution in the log-log scale

## 1.5 Question 5

The personalized network of user with ID 1 has **348** nodes and **2866** edges

## 1.6 Question 6

The diameter of the personalized network for user ID 1 is **2**. The trivial lower bound of is **1** and the trivial upper bound is **2**.

## 1.7 Question 7

The diameter of a graph is defined as the largest distance between any pair of nodes (greatest shortest path between any two nodes). If the diameter of a personalized network is 2, it means that there exists nodes who are not connected directly in the sub-graph, but are connected through a mutual node hence not all nodes in the network are connected directly.

On the other hand, if the diameter of the personalized network is 1, then all nodes in the network are connected directly and there exist direct edges between all the nodes in the network.

## 1.8 Question 8

There are 40 core nodes in the Facebook Network and the average degree of the core node is 279.375.

## 1.9 Question 9

Looking into the modularity scores for each of the three algorithms: Fast-Greedy, Edge-Betweenness, and Infomap, we can see that there are varying modularity scores per each of the algorithms. We can see these scores in 4 through 18.

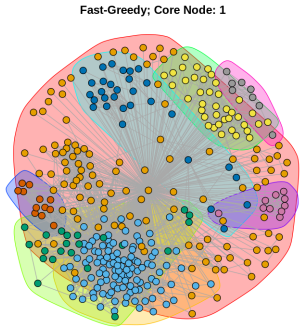



Figure 4: Fast-Greedy Node 1, **Modularity: 0.413101372834235**

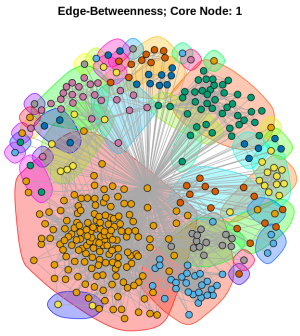



Figure 5: Edge Betweenness Node 1, **Modularity: 0.353302172546335**

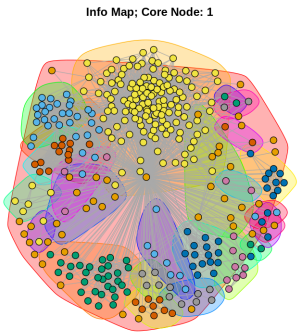



Figure 6: Info Map Node 1, **Modularity: 0.389118471050977**

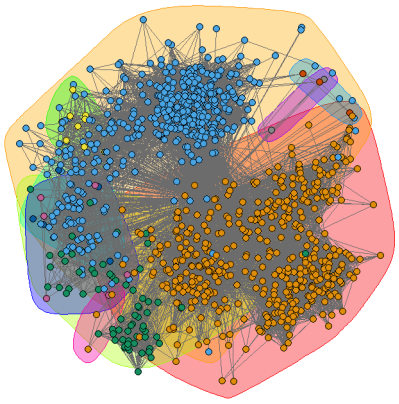


Figure 7: Fast-Greedy Node 108, **Modularity: 0.435958134882439**

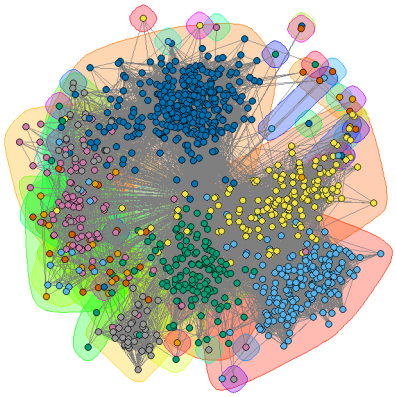


Figure 8: Edge Betweenness Node 108, Modularity: 0.506754916538902

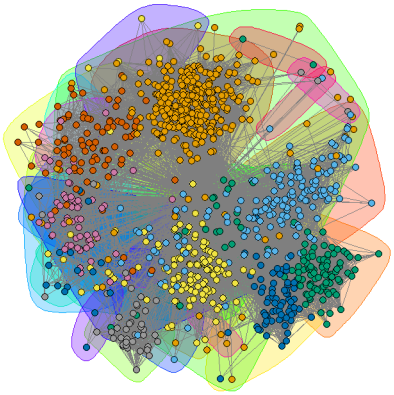


Figure 9: Info Map Node 108, Modularity: 0.508249171320581

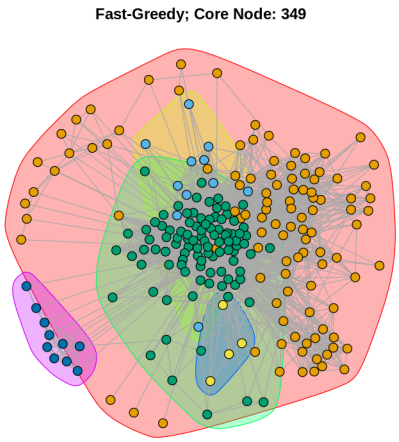



Figure 10: Fast-Greedy Node 349, Modularity: 0.251714858543331

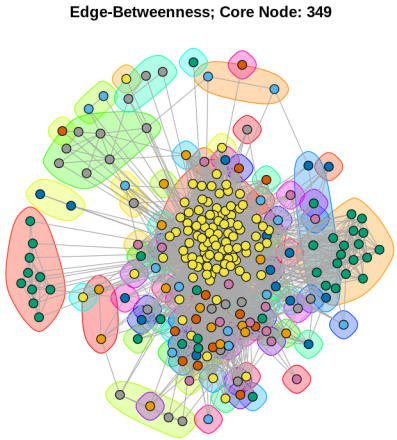



Figure 11: Edge Betweenness Node 349, Modularity: 0.133528021370078

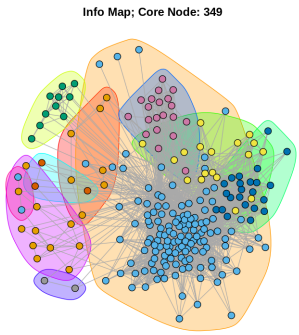



Figure 12: Info Map Node 349, Modularity:0.203752997872299

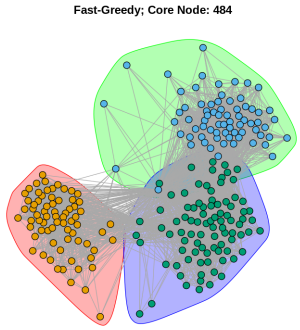



Figure 13: Fast-Greedy Node 484, Modularity: 0.507001642196514

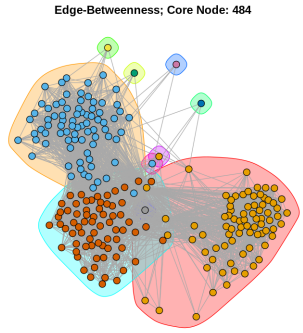



Figure 14: Edge Betweenness Node 484, Modularity: 0.489095180244803

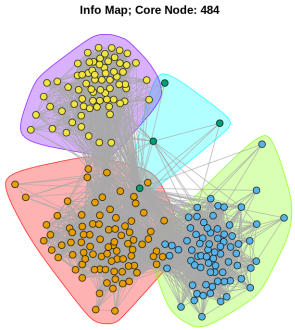



Figure 15: Info Map Node 484, Modularity:0.515278752174842

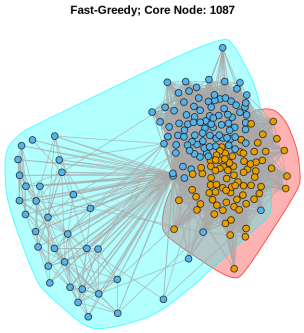



Figure 16: Fast-Greedy Node 1087, Modularity: 0.145531499565493

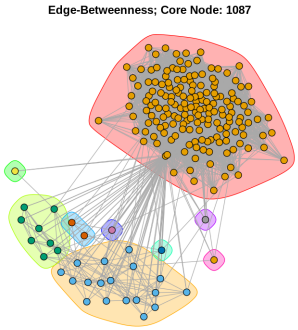



Figure 17: Edge Betweenness Node 1087, Modularity:0.0276237723884639

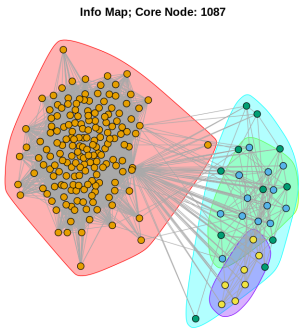



Figure 18: Info Map Node 1087, Modularity:0.0269066172233356

## 1.10 Question 10

Included in figures 19 through 33.

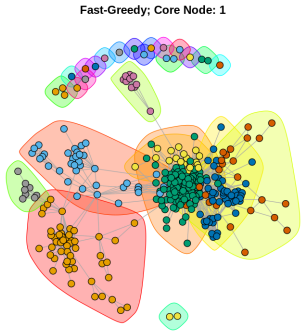



Figure 19: Fast-Greedy Node 1, Modularity: 0.44185326886839

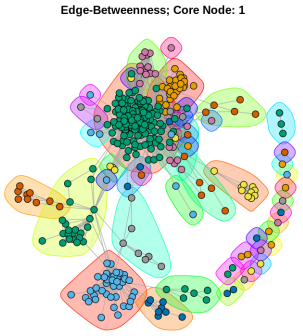



Figure 20: Edge Betweenness Node 1, Modularity:0.41614614203983

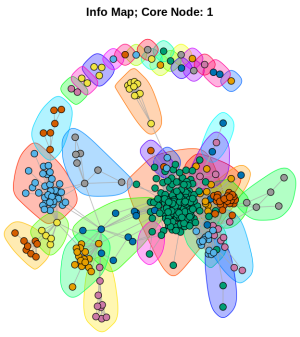



Figure 21: Info Map Node 1, Modularity: 0.418007659453891

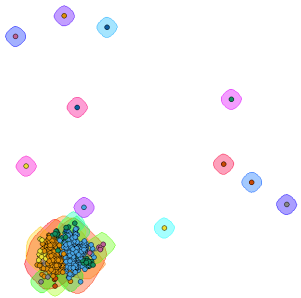


Figure 22: Fast-Greedy Node 108, Modularity: 0.458127093719977

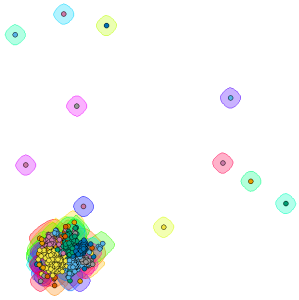


Figure 23: Edge Betweenness Node 108, Modularity:0.521321576382217

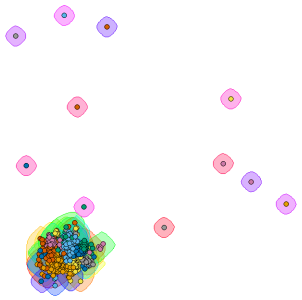


Figure 24: Info Map Node 108, Modularity: 0.520209473665822

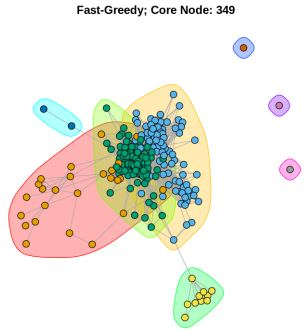



Figure 25: Fast-Greedy Node 349, Modularity: 0.245691795942674

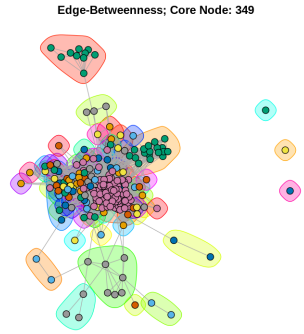



Figure 26: Edge Betweenness Node 349, Modularity: 0.150566340187559

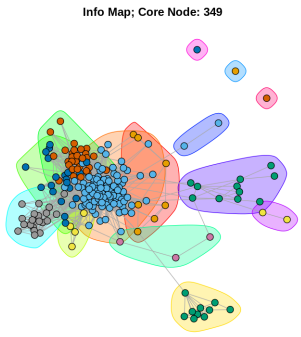



Figure 27: Info Map Node 349, Modularity: 0.246578492623397

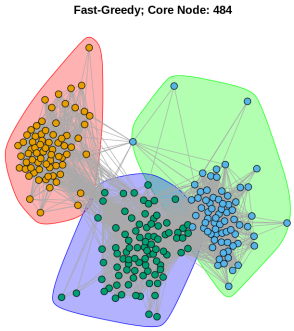



Figure 28: Fast-Greedy Node 484, Modularity: 0.534214154606172

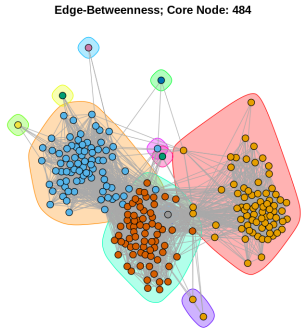



Figure 29: Edge Betweenness Node 484, Modularity:0.515441277123504

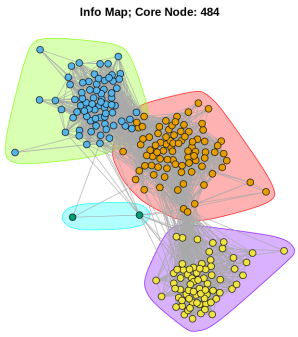



Figure 30: Info Map Node 484, Modularity:0.543443679279522

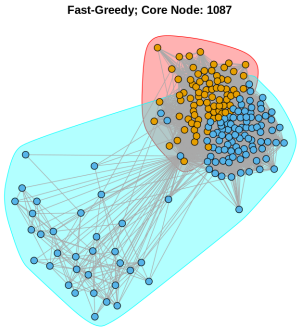



Figure 31: Fast-Greedy Node 1087, Modularity:0.148195631953499

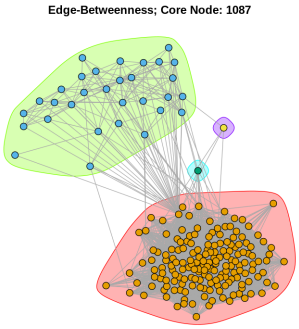



Figure 32: Edge Betweenness Node 1087, Modularity: 0.0324952980499141

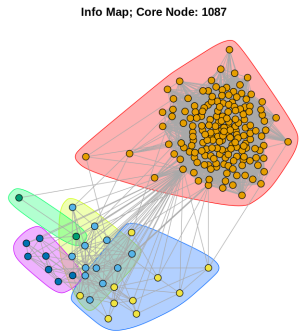



Figure 33: Info Map Node 1087, Modularity: 0.0273715944871147

## 1.11 Question 11

Write an expression relating the Embeddedness between the core node and a non-core node to the degree of the non-core node in the personalized network of the core node. **See code**

## 1.12 Question 12

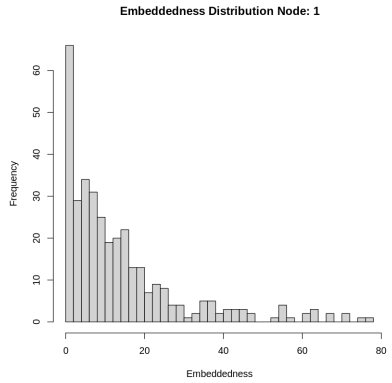



Figure 34: Embeddedness Distribution Node 1

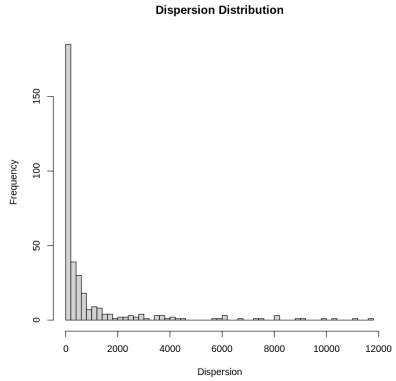



Figure 35: Dispersion Distribution Node 1

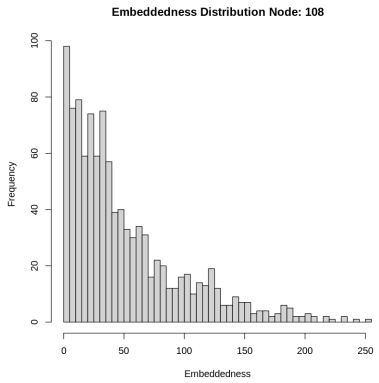



Figure 36: Embeddedness Distribution Node 108

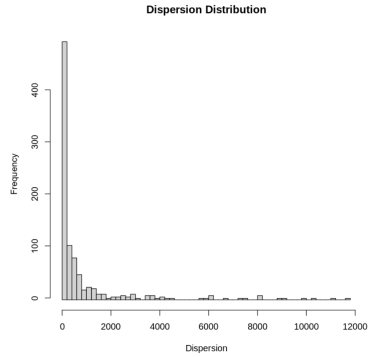



Figure 37: Dispersion Distribution Node 108

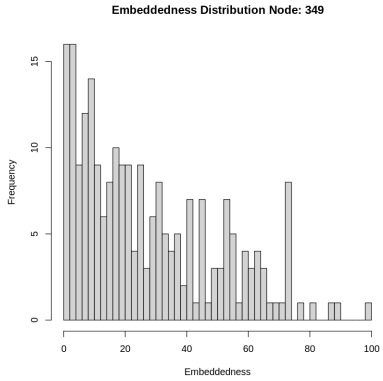



Figure 38: Embeddedness Distribution Node 349

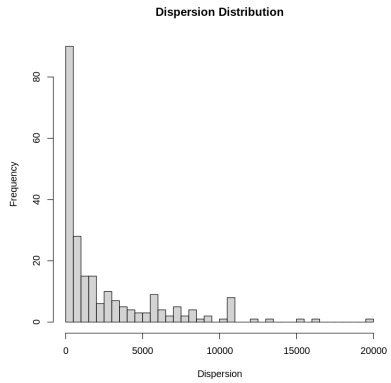



Figure 39: Dispersion Distribution Node 349

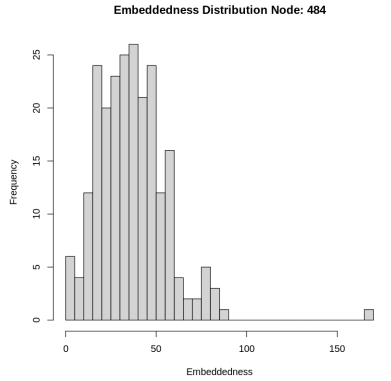



Figure 40: Embeddedness Distribution Node 484

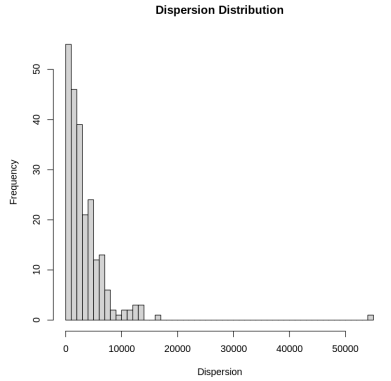



Figure 41: Dispersion Distribution Node 484

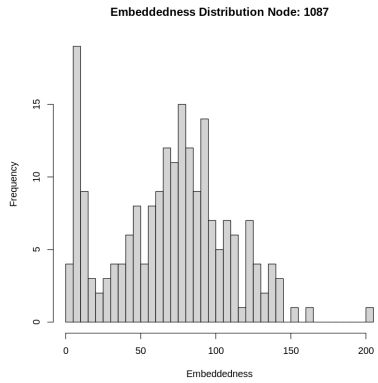



Figure 42: Embeddedness Distribution Node 1087

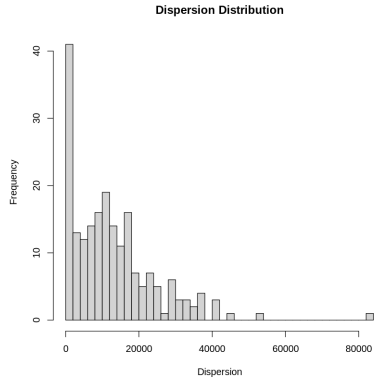



Figure 43: Dispersion Distribution Node 1087

## 1.13 Question 13

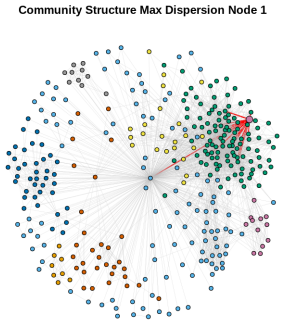



Figure 44: Max Dispersion Node 1;node: 56; 11712”

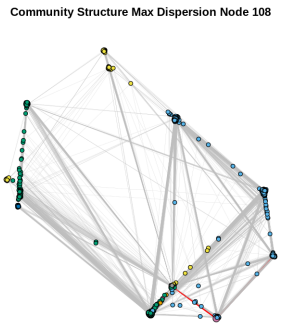



Figure 45: Max Dispersion Node 108; node: 1888; 127520

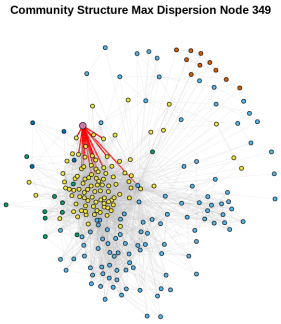



Figure 46: Max Dispersion Node 349; node: 376; 19808

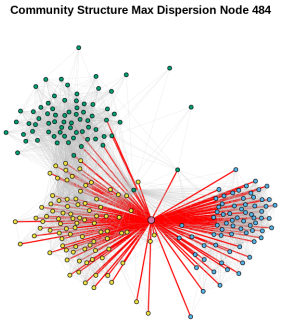



Figure 47: Max Dispersion Node 484; node:107; 54788

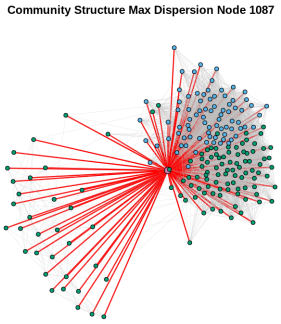
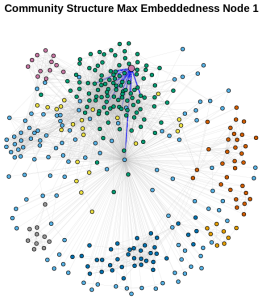
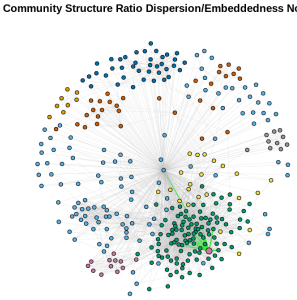



Figure 48: Max Dispersion Node 1087; node: 107; 82832

## 1.14 Question 14

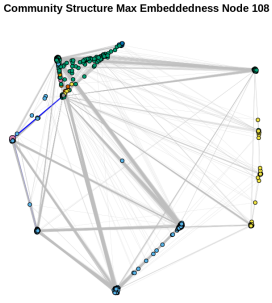



(a) Max Embeddeddness Node 1; node: 56; 77

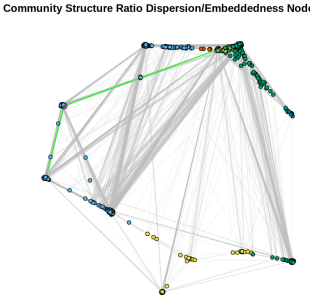



(b) Max Ratio Node 1; node: 56 ;152.103896103896

Figure 49

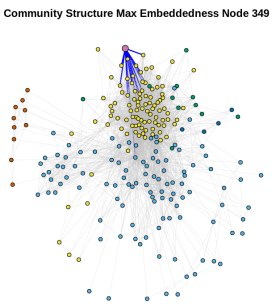



(a) Max Embeddeddness Node 108; node: 1888; 253

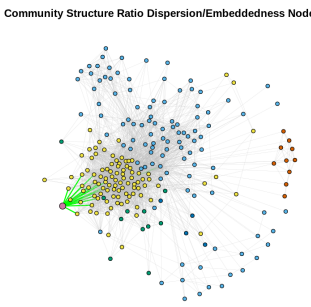



(b) Max Ratio Node 108; node: 1888 ; 504.03162055336

Figure 50



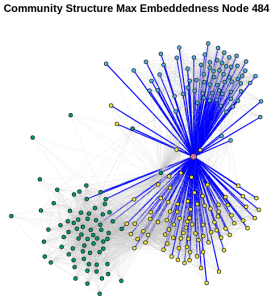

(a) Max Embeddeddness Node 349; node 376; 100



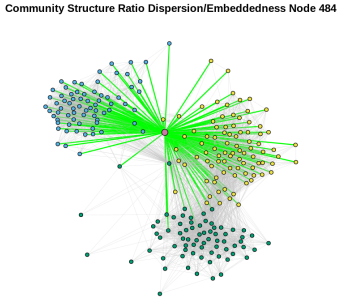

0.3

(b) Max Ratio Node 349; node 376; 198.08

Figure 51

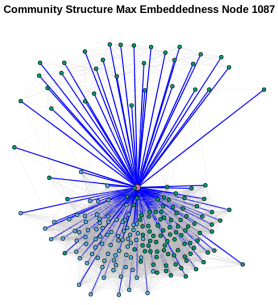



(a) Max Embeddeddness Node 484; node 107; 166

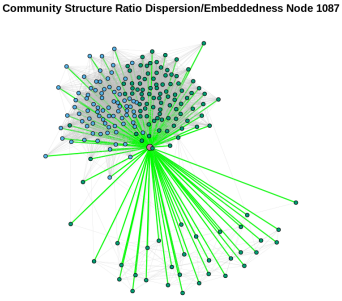



(b) Max Ratio Node 484; node 107; 330.048192771084

Figure 52




(a) Max Embeddeddness Node 1087; node 107; 204




(b) Max Dispersion/Embeddedness Raio Node 1087; node 107; 406.039215686275

Figure 53

### 1.15 Question 15

From question 13 and 14, we can see the difference in the community structures of each of the core node's personalized networks. From question 13, the red highlighted edges are the edges that are incident to the node with the maximum dispersion. The maximum dispersion node is indicated by a pink dot.

In question 14, we plotted both the nodes with the maximum embeddedness and another plotted community structure (per each core node) to highlight the nodes with the maximum ratio of dispersion over embeddedness. You can see on the left side of the plot, the max embeddedness community structure plot, the node and adjacent edges with the max embeddedness are highlighted with a color other than grey and the node itself is pink. On the right-hand side, the node and corresponding edges of the maximum ratio of dispersion over embeddedness is indicated with a green line and a pink dot. Here we can see that for all the core nodes, we get an ID range of max embedded nodes of 1:56, 108:1888, 349:376, 484:107, 1087:107. Max dispersion nodes similarly have these node IDs for the corresponding max nodes.

Looking at the ratio of max dispersion over embeddedness, we can get the best measure. This is because you can see that dispersion improves upon embedded in the network structure because it considers mutual friends in addition to the structure of their network. However, combining the two together is overall the best to additionally combine the emphasis on mutual friends and the similar structures in a normalized matter that will produce the most significant results.

### 1.16 Question 16

The personalised network into consideration for this part is of node ID 415. The set of Nodes ($N_r$) which need friend recommendation is chosen by filtering nodes in the network with degree 24.
The length of list $N_r = 11$.

### 1.17 Question 17

In this question we studied how well a friend recommendation algorithm is performing on the given set of nodes $N_r$. Three algorithms are considered for this namely, Common Neighbors, Jaccard, Adamic Adar. The accuracy of each recommendation algorithm is measured by following process:
Each edge of node i at random was removed with probability 0.25. The list of nodes deleted is in set $R_i$. Then each of the algorithms is used to recommend $|R_i|$ friends. The accuracy is the intersection of the two sets: nodes deleted and nodes recommended by the algorithm. An average of 10 trials of the above process for each user in $N_r$ gave the accuracy for each algorithm.

| Algorithm | Accuracy |
|---|---|
| Common Neighbors | 0.8542 |
| Jaccard | 0.8292 |
| Adamic Adar | 0.8506 |

We observe that all three algorithms are performing quite comparably with Common Neighbors and Adamic Adar having mean acccuracy of 0.85 and Jaccard have the lowest accuracy of 0.829 among the three. Theoretically, Adamic Adar should be performing better among all since it gives more importance to nodes with few neighbors. One of the reasons for the similar performance of all three algorithms can be, the network in consideration is a small personalized networks with relatively low nodes and edges as compared to large social networks.

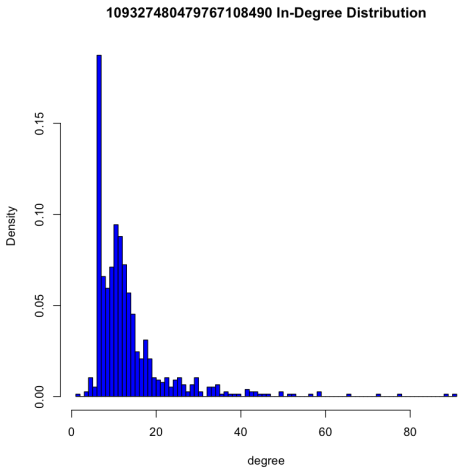
## 2 Part 2: Google+ Network

### 2.1 Question 18

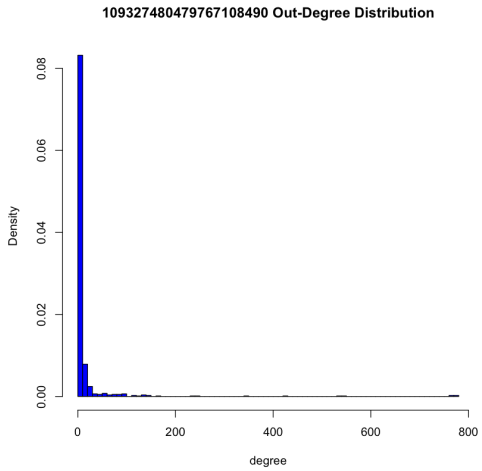
There are **57** personal networks for users with more than two circle out of **132** networks.

## 2.2 Question 19

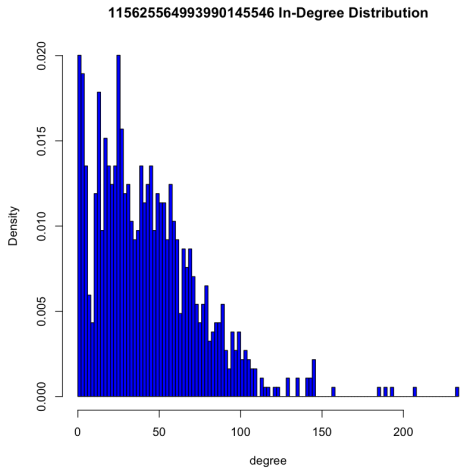
The plots for the in-degree and out-degree distributions of the users are shown below in 54. From 54 we see that the out-degree distributions for all three networks are similar to each other following the power-law distribution. The in-degree distributions are different across the three networks.Among each network, the in-degree and out-degree distribution differs.



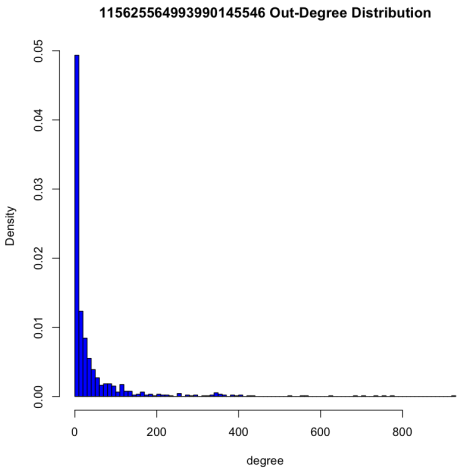

(a) In-degree distribution for user ID 109327480479767108490



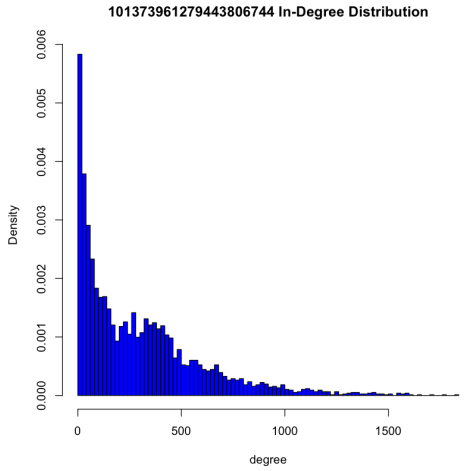

(b) Out-degree distribution for user ID 109327480479767108490



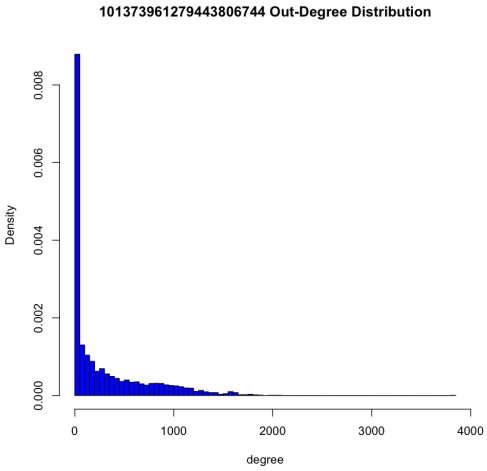

(c) In-degree distribution for user ID 115625564993990145546




(d) Out-degree distribution for user ID 115625564993990145546




(e) In-degree distribution for user ID 101373961279443806744

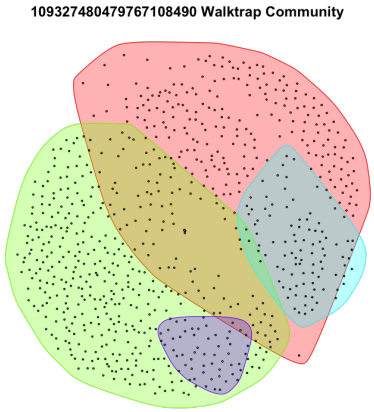



(f) Out-degree distribution for user ID 101373961279443806744

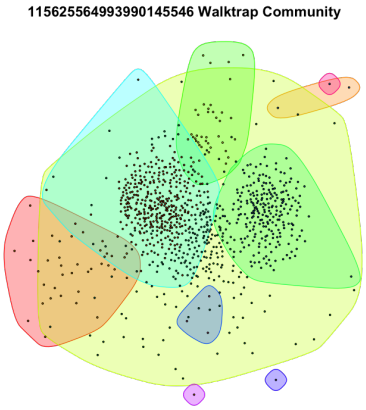
Figure 54: In and out-degree distributions for the 3 personalized networks.

## 2.3 Question 20

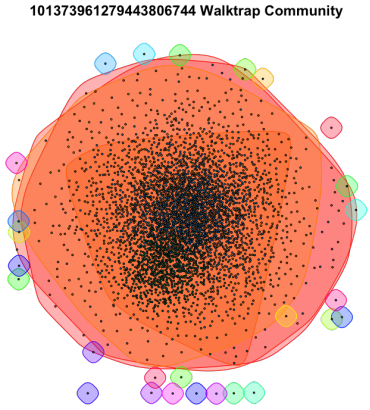
The community structures of the 3 personalized networks are shown on figure 55 below. User ID 109327480479767108490 has a modularity score of **0.252780646085739**. User ID 115625564993990145546 has a score of **0.319473803043365** and user ID 101373961279443806744 a score of **0.191093370318413**.User ID 101373961279443806744 has the lowest modularity score, showing that the network is least capable of being divided into densely packed vertices with strong connections and sparse interconnections among the communities. This is also visible from 55 where we observe that most of the nodes in the personalized network for user ID 101373961279443806744 are grouped in a single chunk in the centre of the plot. User ID 115625564993990145546 has the highest modularity score. From Figure 55, we see dense communities or clusters being formed in the community structure plot, with sparse inter-connectivity among these clusters.




(a) Community structure for user ID 109327480479767108490




(b) Community structure for user ID 115625564993990145546




(c) Community structure for user ID 101373961279443806744

Figure 55: Community structures for the 3 personalized networks using Walktrap Community

## 2.4 Question 21

Homogeneity refers to the amount of variation in each community. A greater homogeneity score is achieved when all members of a community share the same circle. Low homogeneity score is achieved when members

of a community are evenly distributed throughout all feasible circles.

The completeness measure examines each circle and determines how much variety there is in the community assignment of each circle member. A circle has a high completeness score if all of its members are projected to belong to the same community. A circle's completeness score is low if the distribution of the circle members in the community is uniform.

### 2.5 Question 22

The homogeneity and completeness values for the 3 networks are shown below on table 1. From 1 node ID 109327480479767108490 has the highest homogeneity and completeness score. This shows that most of its users belong to the same circle given a community. The low completeness score shows that some of the users belonging to one circle are not all members of one community. Node ID 115625564993990145546 has a lower homogeneity score as compared to the previous node. This indicates that not all of its users belong to the same circle. Users of different circles are classified to the same community.The completeness score is negative and could indicate a large mismatch between the number of circles and the number of communities . Node ID 101373961279443806744 has the lowest homogeneity score indicating that most of the users are from different circles and are being classified in the same community. The completeness score being negative follows the same explanation above.

| Node ID | Homogeneity | Completeness |
|---|---|---|
| 109327480479767108490 | 0.85188512 | 0.32987391 |
| 115625564993990145546 | 0.45189030 | -3.42396235 |
| 101373961279443806744 | 0.00386671 | -1.50423839 |

Table 1: Homogeneity and Completeness score of the 3 personalized networks

## 3 Part 3: Cora Dataset

### 3.1 Question 23: Idea 1

A Graph Convolutional Network, or GCN, is an approach for semi-supervised learning on graph-structured data. It is based on an efficient variant of convolutional neural networks which operate directly on graphs [2]. After hyperparameter tuning, our best model ended up with the following parameters: Learning rate = 0.005, GCN layers = [32, 32], Dropout = 0.7, to achieve a validation accuracy of = 0.852.
The test accuracy for the above model : 0.8259.

The hyperparameters which were used in the search space are:
Learning Rates: [0.005, 0.01],
GCNLayers: [16], [32], [64], [128], [16, 16], [32, 32], [64, 64], [16, 16, 16], [32, 32, 32], [64, 64, 64]
Dropout: [0.2, 0.5, 0.7]
The max. number of epochs used is 200. This is set by looking at the loss history curves also Early stopping with a patience of 50 is used.

For graphical analysis of Part 2 of the project, we used python NetworkX package, along with StellarGraph and Tensorflow for building the GCN.

### 3.2 Question 24: Idea 2

Node2vec is an embedding method that transforms graphs (or networks) into numerical representations [1]. Node2vec generates numerical representations of nodes in the graph via 2nd order (biased) random walk. First order random walk is done by sampling nodes on the graph along the edges of the graph, and each step depends only on the current state. Second order random walk is a modified version of the first order random walk that depends on the current and previous state. A corpus of random walks is generated using

each node in the network as a starting point. This corpus is then fed through word2vec to generate final node embeddings . Table 2 below shows the performance using the different features with a SVM classifier. From 2, It indicates that using only the node features gives us the best accuracy while using only text features attain the worst accuracy. This could come as a result of the text features been very sparse as it is essentially a bag of word representation of the documents with **ones** and **zeros** only. On the other hand the node features gives a dense embedding representation of the documents hence able to achieve better results on classification. By combining the two features together the model could partly rely on the sparse features since their values could weigh more than the dense features for learning hence impacting the performance of the model which led to achieving a slightly lower accuracy than with node features only.

| Model | Validation Accuracy |
|---|---|
| Node Features + SVC | 76.1 |
| Text Features + SVC | 57.7 |
| Node Features + Text Features + SVC | 73.6 |

Table 2: Accuracy of SVC model using different embeddings features on the Cora dataset

### 3.3 Question 25: Idea 3

In this part the goal of predicting the label of each Node in the test set using the 20 seed Nodes (Papers) is achieved via random walks using PageRank. The idea here is that the random walk from a particular start node in the seed set will end at the node having similar features (label) in the train set.

Two methods are used to get the predictions, One is to use the adjacency matrix, for the GCC component of the given CORA network where visiting a node during a random walk from current node to its neighbors is same.

The second method is to use a custom transition matrix which gives more weight to transition to those neighbors which have similar node features (more number of common words in their feature space). The weight is calculated via the softmax formulation over the cosine similarity between each edge. Softmax formulation is used instead of just cosine similarity since, the total probability of transition from a given node to it's neighbors should add to 1.

$$\text{Cosine Similarity}(X_i, X_j) = \frac{X_i^T X_j}{|X_i||X_j|},$$

where $X_i$ and $X_j$ are text features

For both of the approaches, during a random walk the teleportation probability is varied from {0, 0.1, 0.2}. The seed set contains 20 nodes (documents) from each of the 7 classes. Thus when teleporting from current node, we are only teleporting to nodes which belong to the seed set and having the same class as the start node.

To generalize the results well, for each node in the seed set, 1000 random walks, each with 100 steps is performed. The node at which the random walk ended is assigned the label of the starting node (from seed set). A majority voting is done at the end of all iterations to predict the class label.

Mathematically, the personalised Pagerank with teleportation is given by:

$$\pi(i) = (1-\alpha)\sum_{j=1}^{|V|} \frac{1}{k_{out}(j)} A_{ji}\pi(j) + \alpha \sum_{j=1}^{|V|} \frac{\pi(j)}{|T|},$$

where $\alpha$ is the teleportation probability to nodes which have the same label as the start node in the seed set. $A$ is the adjacency matrix/ cosine similarity scores based on the approach used. V is the set of neighboring nodes for the current node i.

The results are summarised below:

| Teleportation Prob. | Accuracy | F1-Score | Unvisited Nodes |
|---|---|---|---|
| 0 | 0.3722 | 0.37 | 0 |
| 0.1 | 0.7364 | 0.74 | 34 |
| 0.2 | 0.7114 | 0.71 | 77 |

Table 3: Approach (A): With Naive transition matrix.

| Teleportation Prob. | Accuracy | F1-Score | Unvisited Nodes |
|---|---|---|---|
| 0 | 0.3774 | 0.38 | 0 |
| 0.1 | 0.7219 | 0.72 | 22 |
| 0.2 | 0.7086 | 0.71 | 71 |

Table 4: Approach (B): With Cosine Similarity based transition matrix.

From the above results, we can see that with no teleportation, the accuracy and F1 score for both the approaches is low compared to where there is teleportation. Accuracy and F1 scores are quite comparable in all of the results and are equivalently good estimates of the performance for this task since the classes are balanced. Accuracy gives the estimates of true positives and true negatives while F1 gives the estimates taking false positives and false negatives into consideration.

No teleportation leads to accessing all the nodes atleast once since we are traversing every node in the seed set 1000 times, and given we are considering only GCC component of the whole graph, the probability to visit each node is high as compared to teleportation trials which have a chance of selecting a node in seed set more than once in different trials and thus leading to some nodes as unvisited as the last node of random walk. This can be seen from the results where high teleportation probabilities lead to higher no. of nodes left as unvisited.

The accuracy for both the approaches with teleportation is quite similar, though transition matrix based on cosine similarity should have performed better given the fact that similar documents (having similar kind of features) tend to have more chances for selection and thus the same label set.

Teleportation helps in predicting labels as we can see teleportation based approach is have approx. (35 percentage) boost in performance. The reason might be that as we go further down in path with each step in random walk, the chances of a document X that needs to be of the same class as that of the start node are very less. This is the property of the social graphs in general. Like things usually stay together. This is why without teleportation, the accuracy is quite low given its hard to find a document of the same class 100 steps away, which is easier to find in a vicinity of a teleported node of the same class.

# References


[1] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[2] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17, 2017.