Influence Maximization in Social Networks by injecting Memes

Niels Sluiter
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
n.i.sluiter@student.utwente.nl

ABSTRACT

The increasing importance of social networks opens up interesting discussions concerning the networks. This matter is extremely interesting in the marketing area, which influencer should companies approach for launching a new product, especially when there is limited budget? This poses several questions, which person in a social network should be chosen for initial injection of memes or a new product? To what extent are the node features crucial in predicting the most influential node? Can we, based on the results of the predicted node with the maximum influence, machine-learn a model to predict the most influential node in the future? Data sets for the influence and values for node centralities are calculated using the independent cascade model and the *networkx* library respectively. Both linear and non-linear regressors are trained and tested using the data sets, resulting in R^2 scores that define the accuracy of the regressor. Knowing which features maximize the R^2 scores is useful for accurately predicting the influence in the future using the machine-learned model. Current Flow Closeness is the dominant feature for maximizing the accuracy as a single feature and in combinations of multiple features. Degree and Closeness give the worst accuracy.

Keywords

social networks, independent cascade model, machine learning, influence maximization

1. INTRODUCTION

The increasing popularity of online social media such as YouTube and Facebook provides interesting new marketing strategies such as viral marketing [5] or word-of-mouth [8]. Businesses tend to minimize expenses at all cost, also in terms of marketing. Either an effectively small group of people is chosen for the initial injection of a new product or a larger substantial amount. The latter involves higher costs and therefore a smaller group of people is preferred.

Social networks are graphs of individuals including their relationships. Individuals are generally, explicitly or implicitly, influenced by their social contacts when deciding on embracing new trends, ideas, news or information in general. To effectively determine which individuals contribute remarkably to the adoption of new trends, such that influencing them will lead to a large cascade, we need to be able to characterize which types of individuals in social networks are the ”trendsetters”. In a more formal way, *influence maximization problem* is described as follows. Given a probabilistic model of a social network, determine the set V of n individuals that turns out to provide the largest expected cascade.

In this paper, we will not try to find the set of nodes that maximize the influence in a social network. However, we will machine-learn a model, using node centralities as features, that is able to accurately predict what the influence is in the network. Or in other words, given a social network, which node should be chosen for initial insertion of information to maximize influence in the network? To what extent are node centralities predictive of the final influence in a social network? And can we machine-learn a model that is able to predict the most influential node in a social network using these node features?

To answer these questions, we generate all connected graphs of size 5, 6, 7 and 1000 graphs for each graph of size 10, 20 and 30 (this choice will be substantiated in section 4.3). From these graphs we will generate data sets for both node centrality values and the influence by the *networkx* library and the independent cascade model respectively, the amount of records for a node centrality or the influence is shown in table 1. Using part of the data sets for training and the rest for testing, we will machine-learn a regression model which results in R^2 scores. Maximizing the R^2 scores results in the most accurate model and which features contributed to this accuracy.

Using a single feature, the Current Flow Closeness centrality is dominant when it comes to the accuracy of predicting the influence using the model. Eigenvector, PageRank and Katz centrality are second-best and Closeness and Degree are the worst. Using multiple features, combinations of good features maximize the accuracy, any combination of Current Flow Closeness and the second-best centralities result in nearly perfect predictions with 99% being the highest accuracy for the data sets.

2. RESEARCH QUESTIONS

- Given a social network which node should be chosen for initial insertion of information to maximize influence in the network?
- To what extent are node centralities predictive of the final influence in a social network?
- Can we machine-learn a model that is able to predict the most influential node in a social network by using node features?

31th Twente Student Conference on IT Febr. 2nd, 2018, Enschede, The Netherlands.

3. RELATED WORK

In this section, the important existing methods and results will be underlined and the added value of this paper on the research project will be substantiated.

Pal et al. [10] proposed two centrality measures, *Diffusion Degree* and *Maximum Influence Degree*, to determine a set of top-k influential nodes in a given social network. Pal et al. have conducted extensive experiments using five large scale real life directed social networks. Using the Diffusion Degree Heuristic and Maximum Influence Degree Heuristic for finding the top-k nodes resulted in superior influence compared to top-k nodes obtained by high degree heuristics and other variants of set covering greedy algorithms.

Narayanam et al. [7] describes two problems. First, they intend to find the top-k nodes that maximizes the number of nodes being influenced in a social network. Second, they try to find the minimal size of a set of key nodes in a network such that λ, which denotes the percentage of influenced nodes, of the network is influenced. Using the ShaPley value-based Influential Nodes (SPINs) algorithms on four synthetically generated random graphs and six real life data sets to solve the above described problems, this resulted in a more powerful and computationally efficient approach compared to the well known algorithms in literature.

Kempe et al. [3] tackled the problem of influence maximization by conducting a research on node centrality measures as *distance centrality* and *high-degree*. The research was not only applied to the linear threshold model, but also to weighted cascade model and independent cascade model. These results were significantly worse compared to the greedy hill-climbing strategy [9]. Reasons for the results were high degree often implies a cluster, targeting all nodes in a cluster is unnecessary and also the both metrics do not take into account any network effects.

Acemoglu et al. [1] focused on applying the linear threshold model on deterministic topologies and heterogeneous threshold values. This resulted in a completely characterized set of final adopters. Following this, the relationship between the metrics and clustering of the network was explored. This resulted in highly clustered networks not necessarily being more advantageous over less clustered networks. This is due to clusters being hard to penetrate when they are not targeted.

Kundu et al. [4] investigated a new centrality measure which is based on both degree centrality and diffusion probability. The experiment used Monte-Carlo simulations of the independent cascade model for sufficient amount of times to accurately approximate the final influence of spread. The results of the experiment had a significant improvement over other existing centrality based heuristics.

There are several aspects with which this research differentiates itself from the others. The primary aspect is the machine-learned model, all of before-mentioned researches use an algorithm to calculate the final influence in a network. In this research, we machine-learn a model that is able to predict the influence for us, based on the values for the node centralities. Additionally, we have applied the following features to machine learning; Degree, PageRank, Current Flow Closeness, Katz, Closeness, Eigenvector, Betweenness and Current Flow Betweenness, where previous papers have only looked into the Degree or their own centrality.

4. METHOD

A social network is a graph $G = (V, E)$ where a vertex $v \in V$ is a person and an edge (v, u) is an undirected relationship, i.e., v is a friend of person u and the other way around. Each node in the graph has properties called node centralities. Each node centrality has a value $k \in \{0.0 \ldots 1.0\}$ for a vertex $v \in V$. In this research the following node centralities have been chosen for evaluation: Degree, Eigenvector, PageRank, Closeness, Flow-Closeness and Katz.

4.1 Prerequisites

We first obtain all prerequisites to start researching. This consists of gathering packages that are suitable and required for collecting data, analyzing data and making a predictive model using machine learning.

Networkx is the main library used for accessing graphs and provides functions to calculate each of the aforementioned node centralities. Also, *Networkx* provides functions for reading and writing graphs

The independent cascade model [6] is used as a tool to represent the way influence is spread in social networks. The code for the independent cascade algorithm is shown below and is deduced from the paper of Li et al. G is the graph in which we will spread new information from the seed node *seed*, and *probability* is the probability with which the remaining nodes will adopt new information.

Algorithm 1 The Independent Cascade Model

```
procedure INDEPENDENT CASCADE(G, seed, p)
    $active \leftarrow \emptyset$
    $target \leftarrow$ empty list
    append seed to target
    while length of target $> 0$ do
        node $\leftarrow$ last element of target
        for each neighbor $n$ of node do
            randnum $\leftarrow$ randomly generated number
            if $n$ not in active then
                if $randnum \leq p$ then
                    append $n$ to target
    return length of active
```

scikit − learn provides us with all the machine learning models of which some will be used later on. For displaying the results that come from machine learning, both *seaborn* and *pyplot* from *matplotlib* are used. *seaborn* is based on *matplotlib* and creates heatmaps of the acquired results, *pyplot* further handles the displaying of the heatmaps.

Finally, the Python library *statistics* has been used to provide values for both the mean and standard deviation which will be used in calculating the influence within a graph.

4.2 Data acquisition

The first step to data analysis is acquiring the actual data, i.e., graphs, node centrality values etc. This is done in two steps.

Firstly, we have to generate the graphs which we will use later on for the second part of data acquisition and machine learning. To have a generalized view on how influence is spread in smaller graphs, all distinct connected graphs of size 5, 6 and 7 have been collected using the *nauty − geng* library, which generates all graphs of a specified class. Using smaller graphs has a big advantage. We can evaluate **all** graphs of smaller sizes, i.e., if we take all connected graphs of size 9 we have 261.080 graphs, this results in files with 2.349.720 records whereas there are 11.117 graphs of size 8 which results in 88.936 records. The amount of graphs increases by a lot when we go from size 8 to 9 and results in lots of records. A big amount of records in machine learning is computationally intensive, i.e., evaluating **all** graphs of size 20 is impossible on normal computers. To see how the effects would change whenever larger graph sizes are chosen, of each of the graph sizes ten, twenty and thirty, a thousand graphs have been generated according to the Barabasi-Albert preferential attachment model [2] (Functions for this are provided in the *Networkx* library). This model generates graphs by preferential attachment, i.e., the more connected a node is, the more likely it is to receive a link. If a new node enters a social network, it is more likely to be acquainted with more visible people rather than relatively unknown. The specifically chosen amount of graphs for each of the larger sizes will be elaborated upon in the next section.

Secondly, *Networkx* has been used for gathering the values for each of the evaluated node centralities. Additionally, we want to determine an accurate influence of each of the graph sizes with respect to the real world. To acquire this, we apply fixed probabilities to the independent cascade model. As fixed probabilities we have taken p = 0.01, p = 0.08 and p = 0.15. Smaller probabilities are chosen intentionally since higher probabilities may cause for a cascade in the network, meaning all nodes get activated and from that, you can not make meaningful conclusions.

To calculate the influence within a graph G, we go through each of the probabilities $p \in \{0.01, 0.08, 0.15\}$ and through each of the nodes $N \in G$ and calculate the influence in the graph using the independent cascade model. Calculating this influence is based on probabilities, therefore running the algorithm for the independent cascade model once does not give sufficient data. For this we have chosen to calculate the mean of 100 iterations using the *statistics* library. Not only the statistical mean is used in determining these values, also the standard deviation is used in determining relevant data. Given a confidence-interval of 90%, any values outside (based on the formula for confidence intervals $\overline{X} \pm Z\frac{s}{\sqrt{n}}$, where $\overline{X}$ is the mean, Z is $1,645$ for a confidence interval of 90%, s is the standard deviation and n is the number of observations) of this interval are neglected to increase accuracy. Determining whether a score is outside of the confidence interval we use the formula for confidence intervals where the value must be in the boundaries of $\overline{X} \pm Z\frac{s}{\sqrt{n}}$, where $\overline{X}$ is the mean, Z is $1,645$ for a confidence interval of 90%, s is the standard deviation and n is the number of observations.

Similar to saving the values for node centralities, values for influence are saved as i.e. *influence*_6_015 where 6 denotes the graph size and 015 denotes the probability of 15%

4.3 Machine Learning

Now that all data is acquired, we can start training/testing a machine learning model. Generally, machine learning requires a training set for training the model, and a testing set to provide an unbiased evaluation of a final model fit on the training set. There are several types of machine learning (classification, regression etc.), we want to apply different features on machine learning to see how predictive this is for the data, the suitable type of machine learning for this is regression. There are two types of regression (linear and non-linear), both have been used in research and will be explained further on. Regression makes use of the coefficient of determination (explained in section 4.4) and this is the value that we ought to maximize based on our features.

Graph size	Amount of records
5	105
6	672
7	5971
10	10000
20	20000
30	30000

Table 1. Amount of records in the data set

As mentioned, regression takes features to train and evaluate a machine learning model. We want to see how well both a single feature and multiple features are at predicting the influence.

For the before-mentioned number of graphs generated for larger sizes, we have decided to take 1000 graphs of each of the sizes. This number has been achieved by experimenting with the amount of graphs. Firstly, we had taken 10 graphs of each size and used those graphs to generate the plots for R^2 scores. This resulted in a plot where across the x-axis for the corresponding graph size, there was little to no correlation in the R^2 scores, i.e., for graph size 20 and p = 0.01, p = 0.08 and p = 0.15 the R^2 scores were 0.84, 0.65 and 0.89 respectively, where some correlation is expected. Secondly, we had taken 100 graphs which had similar results as 10 graphs, no correlation. Taking 1000 graphs resulted in the plots shown in the figures. To verify that these results were accurate, we had also taken 900 graphs which resulted in a 0.01 for some of the cells which is negligible. Adding more graphs than a 1000 would give similar results but with a slight increase of accuracy. This slight increase of accuracy is not worth the additional computational power

Besides the amount of graphs used for machine learning, there is also the parameter of how much of the data is used for training and how much is used for testing. We want to provide an adequate amount for training so the model is able to predict the influence in the future on additional data, yet also leave part of the data to test how accurate the model is. Table 1 shows the total amount of records for each of the graph sizes. Graphs of size 5 have the smallest amounts of records, therefore we want to use a lot of this data for the actual training and less for testing. For the other graphs we did the same, the more data we use for training, the more accurate it is in predicting new data. This also has its downside since it will take longer to actually train the data. Therefore to have a generalized separation of training and testing data, 75% of the data is used for training the model and the remaining 25% is used for testing.

4.3.1 Linear Regression

scikit − learn offers a variety of linear regression models which is a perfect way to start machine learning. In essence, linear regression looks for linear relationships between dependent and independent variables.

LinearRegression is one of the simplest models that fits a linear model with coefficients w such that the sum of squares between targets in the data sets, and the targets predicted by linear approximation is minimized.

We used this to see if the target value is expected to be a linear combination of the features.

4.3.2 Nonlinear regression

For nonlinear regression there is also a variety of models in $scikit-learn$, several of these have been evaluated and one is chosen as final model for the results. The two main models used are Gaussian Process Regression model and Random Forests model. Both models have their advantages and disadvantages.

$GaussianProcessRegressor$ (GPR) implements Gaussian processes for regression purposes. The main advantage of GPR is that the prediction interpolates the observations however a big disadvantage of GPR is that it is not sparse, i.e., the whole samples/features information is used to perform the prediction. On smaller datasets this is not as big of a problem, but when datasets tend to become bigger, the efficiency is decreasing extremely and takes too long for GPR to be worth it. Though the results are accurate, other models are more advantageous. As an alternative, $RandomForestsTree$ has been chosen. Especially in this research where data is fairly large and efficiency is a must, $RandomForestTree$ definitely beats GPR.

As opposed to linear regression and as the word would assume, nonlinear regression looks for nonlinear combinations of the model parameters and depends on one or more independent variables.

4.4 Evaluation

From previous steps, we have yet obtained a series of numbers as data called coefficients of determination for each probability and each graph size. Representing and understanding the results is the next step to drawing a conclusion. The numbers imply nothing until it has been given a meaning. In short, the coefficient of determination, denoted R^2, is described as the proportion of the variance in the dependent variable that is predictable from the independent variable(s). Values normally range from 0 to 1 with 1 being the best possible score. Of course there are exceptions to the rule in which the R^2 score could be negative because the model can be arbitrarily worse, also it can get a score of 0.0 if a constant model always predicts the expected value of y, disregarding the input features. A more formal definition of the R^2 score is described below.

We have n values in our data set marked $y_1 \, \ldots y_n$ associated with a predicted value $f_1 \, \ldots f_n$. The mean of the observed data is $\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. From this mean we can calculate the sum of residuals (SS_{res}) and the total sum of squares (SS_{tot}) as follows:

$$SS_{\text{res}} = \sum_i \, (f_i - \overline{y})^2$$

$$SS_{\text{tot}} = \sum_i \, (y_i - \overline{y})^2$$

Then the general formula for the coefficient of determination is the following

$$R^2 \equiv 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

5. RESULTS

Above-mentioned method provided us with the results of our research. To determine how well features are at predicting the influence, the R^2 is evaluated and displayed in heatmaps. To be able to fully answer the research questions we have used combinations of 3 features on the machine learning. With these results we should be able to answer which node should be chosen for initial insertion, to what extent are node features predictive of the final influence and can we machine-learn a model to predict the most influential nodes in a social network.

Ideally, nodes that have the highest values for node centralities that maximize the R^2 score are preferred to be chosen for initial injection of information. Which features maximize this R^2 score are elaborated upon in the following sections.

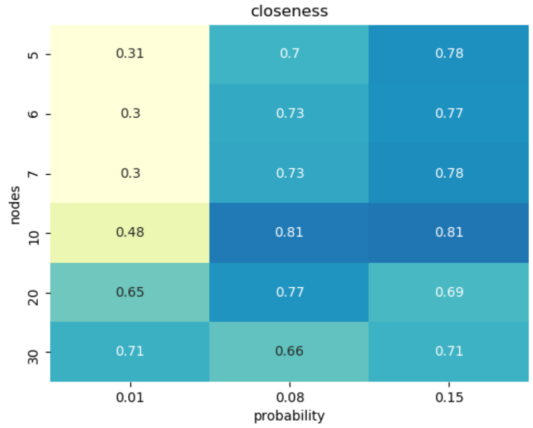

Figure 1. Closeness centrality

Trivially, we predict that the higher the probability, the higher the R^2 score. This can be explained by the total influence in graphs. Using the independent cascade model, the higher the probability, the more likely a node adopts new information from another node. The higher the R^2 scores, the higher the predictive power of the model, the easier it is to predict which node has the highest influence.

Following features are correlated and displayed in sets: $\{Eigenvector, PageRank, Katz\}$, $\{Degree, Closeness\}$

5.1 Single node centrality

A single node centrality applied to machine learning tends to show more diverging results, contrary to multiple node centralities. Some of the researched centralities are intercorrelated, i.e., closeness and degree, for this reason, only distinct plots will be shown to avoid redundancy.

Figures 1, 2 and 3 display the R^2 score of Closeness, Eigenvector and Current Flow Closeness respectively. From the figures we can conclude several things.

The closeness tends to predict the influence the worst, comparing the closeness with current flow closeness we see that none of the values for closeness reach higher than that of current flow closeness. Based on this observation, closeness is redundant as a single feature.

Contrarily, comparing Current Flow Closeness with Eigenvector show interesting results. At first sight we could immediately conclude that Current Flow Closeness is best fit at predicting the influence for a single feature. However, looking deeper into both graphs we see that indeed most values of Current Flow Closeness exceed that of Eigenvector, but interestingly enough, at a smaller probability for graphs of smaller sizes, we see that Eigenvector dominates both graphs of size 6 and 7 and equalizes for size 10, 20 and 30. Therefore in smaller networks in which the probability of someone adopting information is lower, the Eigenvector centrality might be preferred, however, generally Current Flow Closeness is preferred.

5.2 Dual node centralities

Similar to the single node features, plots of combinations of correlated features, i.e., Eigenvector and Katz vs PageRank and Eigenvector since Katz and PageRank are correlated. To avoid redundancy yet again, only interesting

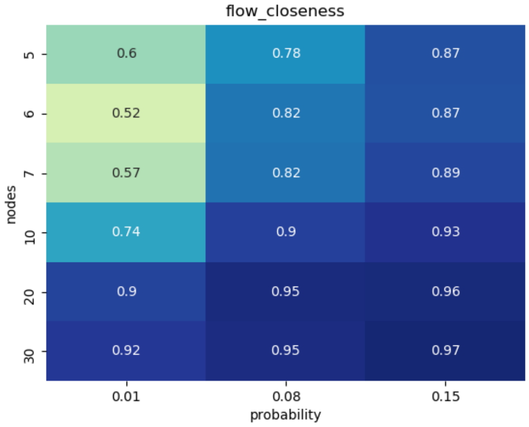

Figure 2. Current Flow Closeness centrality

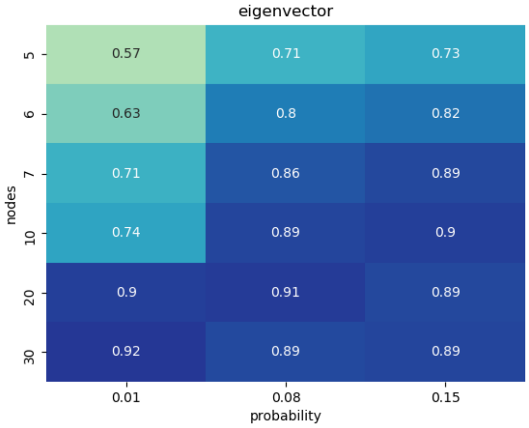

Figure 3. Eigenvector centrality

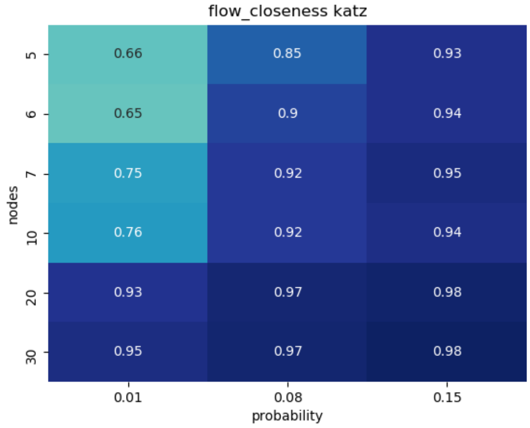

Figure 4. Current Flow Closeness and Katz centrality

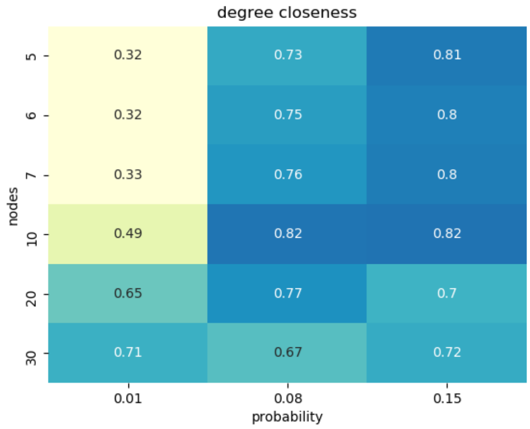

Figure 5. Degree and Closeness centrality

distinct plots will be shown.

From these plots we can actually make some interesting conclusions. As shown, we have combined multiple features. These plots consists of the following combinations: a good feature with a good feature (figure 4), a good feature with a bad feature (figure 6) and a bad feature with a bad feature (figure 5).

From this we can conclude that combinations of good features imply higher R^2 score, the worse the combination, the lower the R^2 scores. Generalizing this for triple node centralities, we assume that the combination of three good features maximizes the R^2 score and is best for maximizing the influence in a social network.

5.3 Triple node centralities

An example of three good features is displayed in figure 7, note that in this plot the minimum value for the graph has been increased to properly display the difference in R^2 scores amongst the plot. In previous amount of features this was no issue, but due to any combination of three features containing at least one good feature, the minimum R^2 scores are higher.

5.4 Discussion

As mentioned before, influence maximization in social networks is of high importance in many practical applications. Identifying which features contribute most to the influ-

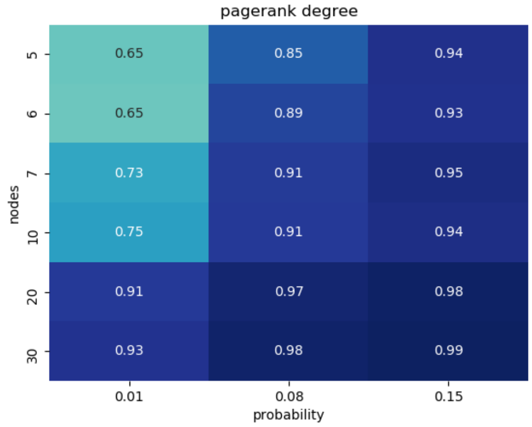

Figure 6. PageRank and Degree centrality

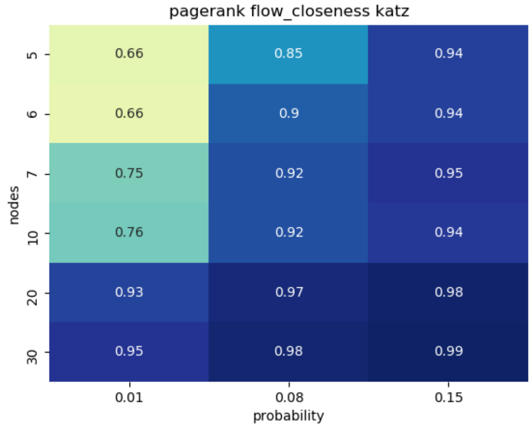

Figure 7. PageRank, Flow Closeness and Katz centrality

encing of other nodes is key to finding the most influential nodes. As we have seen in previous sections, Current Flow-Closeness is generally the best node centrality when it comes to predicting the influence, combining this centrality with either of the following centralities in the set $\{PageRank, Eigenvector, Katz\}$ results in almost perfect scores for a probabilty of 15%. With this information, we are able to accurately predict the influence based on

Due to the use of Machine Learning and the extensive data set used for both training and testing, we can generalize this for both smaller and bigger graphs, since the trained model is able to predict future data points with high accuracy given the selected features.

6. CONCLUSIONS

At first, 8 node centralities had been chosen for this research, those are before-mentioned including Betweenness and Flow Betweenness. The Betweenness and Flow Betweenness have been neglected due to their bad performance on R^2 scores. Current Flow Closeness showed great accuracy in predicting the influence in social networks, when combined with other good node centralities as PageRank, Katz and Eigenvector, it nearly maximizes the R^2 score.

An improvement to this research could be the extension of having additional node centralities, as there are many node centralities yet to analyze. Furthermore it could be interesting to see what the results would be for all graphs of size 8, 9 and 10 however this might also be a waste of resources seeing as there is a relationship between the sizes of graphs. Additionally, real life social networks can be used as data to see how the research questions apply to those networks.

7. REFERENCES

[1] D. Acemoglu, A. Ozdaglar, and E. Yildiz. Diffusion of innovations in social networks. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 2329–2334. IEEE, 2011.

[2] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[3] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[4] S. Kundu, C. Murthy, and S. K. Pal. A new centrality measure for influence maximization in social networks. In *International Conference on Pattern Recognition and Machine Intelligence*, pages 242–247. Springer, 2011.

[5] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5, 2007.

[6] Y. Li, J. Fan, Y. Wang, and K.-L. Tan. Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1852–1872, 2018.

[7] R. Narayanam and Y. Narahari. A shapley value-based approach to discover influential nodes in social networks. *IEEE Transactions on Automation Science and Engineering*, 8(1):130–147, 2011.

[8] F. Naz. Word of mouth and its impact on marketing. 2013.

[9] T. Ohashi, Z. Aghbari, and A. Makinouchi. Hill-climbing algorithm for efficient color-based

image segmentation. In *IASTED International Conference on Signal Processing, Pattern Recognition, and Applications*, pages 17–22, 2003.

[10] S. K. Pal, S. Kundu, and C. Murthy. Centrality measures, upper bound, and influence maximization in large scale directed social networks. *Fundamenta Informaticae*, 130(3):317–342, 2014.