Predicting influence spread in Online Social Networks using combinations of node centralities

Justin Praas
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
j.w.praas@student.utwente.nl

ABSTRACT

The extent and magnitude of information spread through an online social network is of great interest to marketing strategists and social scientists. The spread of information within a network has a strong correlation with the network statistics of the participating nodes. While the effects of individual node centralities on the influence spread has been researched by many, the predictive power of these centralities (or combinations thereof) have not.

In this paper we show how well individual and combinations of those network statistics can predict the total estimated information spread (influence spread). We look at all non-isomorphic graphs of size $N \leq 9$ nodes and small-to-medium graphs of $N \in \{50, 100, 200, 400\}$ nodes, randomly generated using the Barabási-Albert random graph generation model. Next, the Independent Cascade (IC) and Weighted Cascade (WC) spread models are used on each individual seed node in each graph to simulate the spread of information. Finally, the Machine Learning techniques Random Forest Regression and k-Nearest Neighbors regression are used to predict the total information spread.

We find that the WC spread model results in higher R^2 scores than the IC model. The combination of the centralities *degree and PageRank and betweenness* are particularly predictive of the influence spread, both for IC and WC.

Keywords

Online Social Networks, Machine Learning, Influence Prediction, Node Centralities

1. INTRODUCTION

The rise of Twitter, Facebook and other social media has caused a paradigm shift in the way we spread news, fake news and memes. Equally important, these Online Social Networks (OSN) have become a platform for excessive marketing [17]. The promotion of services, products and ideas is mainstream. Behind the scenes are the marketing strategists. While being on a budget, their task is to reach as many potential clients as possible by injecting an advert or piece of information into the network. The total spread of the advert or information (influence spread) are thought to correlate with the network statistics of the seed nodes. Marketing strategists use Influence Maximization techniques to determine the most influential nodes within a network for the purpose of maximizing the total influence spread.

In this research, we aim to find the combination of network statistics that best predicts the estimated total information spread. The results and parts of the method can be applied to Influence Maximization, the field in which nodes are ranked based on their predicted influence spread.

In the context of epidemiology, Bucur et al. applied Machine Learning techniques to calculate the expected epidemic outbreak size, albeit on small networks, given any seed node. They find that certain combinations of two or more statistics are much more predictive of outbreak sizes than others [6]. Oo et al. verify that PageRank can detect influential spreaders more than other centralities [15].

Research aim and scope.

In this research we expand on the findings of Bucur et al. Our aim is to similarly predict the total influence spread (see Section 2.3). However, we improve upon their work by considering not only small, but small-to-medium networks. Moreover, we shift the context from epidemiology to Online Social Networks. This entails that we employ different spread models: where Bucur et al. applied the *Susceptible-Infectious-Recovered* spread model, we will apply both the IC and WC spread models and compare the results. These spread models are chosen because they are the most simplistic models and sufficiently approximate the way information spreads in Online Social Networks [11].

We retrieve all non-isomorphic graphs of size $N \leq 9$ and generate sufficiently many graphs of some small-to-medium graphs ($N \in \{50, 100, 200, 400\}$) using the Barabási-Albert random graph generation model [1]. The Barabási-Albert model is an algorithm that generates graphs approximating certain natural and artificial systems, such as social networks and the world wide web [1]. The algorithm employs a preferential attachment mechanism which simulates a network that follows a scale-free power-law distribution. An example of such a network can be found in Figure 1.

The different network statistics that we cover are mentioned and briefly explained in Section 2.2. Two supervised statistical learning techniques will be used after applying the spread models: Random Forest regression and k-Nearest Neighbors regression.

32nd Twente Student Conference on IT Jan. 31st, 2020, Enschede, The Netherlands.

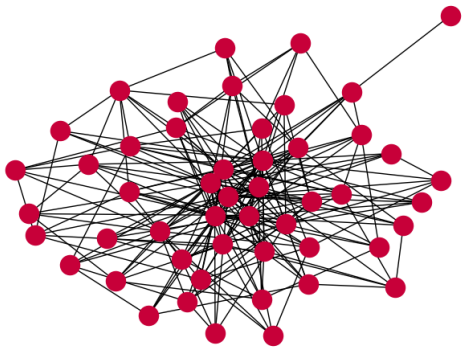


Figure 1. A network generated using the Barabási-Albert random graph generation model, with $N = 50, m = 5$.

Limitations.

For the purpose of this research we must make some assumptions about the networks and methods used. Firstly, Online Social Networks are changing every second; we will assume the networks that we retrieve and generate to be static. Secondly, certain Online Social Networks are represented by directed graphs (Twitter), whereas others are undirected (Facebook [7]). To make this project feasible, we will solely work with undirected graphs. Finally, we admit that no graph generation model will generate graphs that perfectly resemble a real OSN. Therefore, using the acclaimed Barabási-Albert model will suffice for the purpose of this research.

2. BACKGROUND

This section will contain brief descriptions of certain concepts, tools, and terminology.

2.1 Models of information spread in networks

The two spread models, IC and WC will be used on each seed node to calculate the respective total spread of information.

The Independent Cascade (IC) model attaches to each neighbour-pair (u, v) in a graph a chosen propagation probability p. If either one of the nodes of the neighbour-pair is informed, it has p chance of propagating the information to the neighbour. However, if the node fails to inform its neighbour node, it will no longer try to inform said neighbour in future iterations of the algorithm. The process terminates once no more nodes are informed [11]. The pseudocode can be found in Algorithm 1.

Algorithm 1 The IC algorithm

```
1: procedure INDEPENDENT CASCADE(G, seed, p)
2:     active ← ∅
3:     target ← [seed]
4:     while length of target > 0 do
5:         node ← pop last element of target
6:         add node to active
7:         for each neighbor n of node do
8:             if n not in active then
9:                 append n to target with probability p
10:    return the length of active
```

The algorithm is illustrated in Figure 2. Each iteration i is one iteration in the while loop.

The Weighted Cascade (WC) model counters a property of the IC model, which is the following. In IC, a node

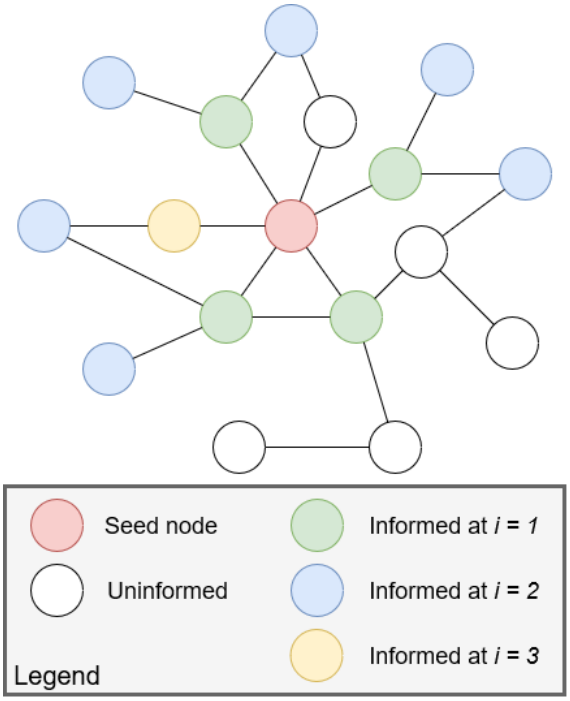


Figure 2. A simulation of information spread through a network visualized. In this example IC spread probability $p = \frac{2}{3}$.

that has a high degree not only has an increased chance to influence other nodes, but is also more likely to be influenced at some point by one of those neighbour nodes [11]. This property of IC is or is not desirable depending on the application.

Using WC, a node u has propagation probability $1/d_v$ towards node v, where d_v is the in-degree of node v. In other words, the chance of propagating information over an edge depends on the degree of the node on the other side of the edge [11].

To obtain the pseudocode of WC, substitute $1/d_v$ for p in line 9 of Algorithm 1.

2.2 Node centralities

The node centralities (previously referred to as network statistics) are network characteristics of each individual node, like the number of incoming and outgoing edges. The centralities used in this research are listed and briefly explained below. We chose these centralities because they are either frequently used or easy to interpret and not hard to implement.

- **Degree** The number of edges of a node.
- **Closeness** The average distance to all other nodes [16].
- **Betweenness** The number of times a node acts as a bridge along the shortest path between all other nodes [9].
- **Eigenvector** Each node starts with a relative score, after which the scores iteratively increase depending on how high-scoring a node's neighbours are [4].
- **Katz** Variant of Eigenvector, incorporating an attenuation factor [10].
- **PageRank** Variant of Eigenvector, incorporating a damping factor [5].

2.3 Machine Learning

For the prediction of total influence spread, we will be using two regression algorithms: *Random Forest* and *k-Nearest Neighbors* regression. A brief description of the two algorithms follows.

The Random Forest (RF) algorithm makes use of a technique called bagging [2], which is short for random sampling with replacement. Bagging is used to reduce variance of certain algorithms, like decision trees. Another advantage of bagging is that each model can be run in parallel, after which the outputs of each are aggregated. A Random Forest is an estimator that uses many decision trees, each being attached to random sub-samples of the training data. Once each tree has computed its output, the average is taken over all trees, resulting in the prediction.

k-Nearest Neighbors (KNN) is a non-parametric prediction method, which essentially means that no randomness is involved in calculating the prediction [3]. Considering the context of graphs, pick a centrality feature c. The algorithm will then calculate the difference between the value of c of a query node and those of the nodes in the training data. Next, it determines the k closest samples from the training data, after which, for regression, it evaluates the outcome by averaging the labels from the k closest training samples. The distance is computed using the Minkowski metric, which is a generalization of the Manhattan and Euclidean distance.

2.4 The data set

For each graph size N we will generate a data set. The data set will contain a row for each seed node in each graph of a certain size. For example, a data set for generated graphs will contain $N * M$[1] rows each. The following columns will be included in the data set: values for each centrality $c \in C$, where C is the set of node centralities covered in this research (see Section 2.2); and the estimated total influence spread per spread model configuration (see Section 5).

2.5 Tools

All programming work will be done using *Python 3.7*. To keep installation of tools and plugins simple, we will be using *JetBrains Pycharm, professional edition with the Anaconda plugin*. The Anaconda plugin eliminates the overhead of finding compatible scientific plugins, since most are already included.

scikit-learn, networkx, matplotlib and seaborn are the python libraries that we depend on. *scikit-learn* will be used for the Machine Learning matters. *networkx* will aid us in generating and processing (network) graphs. *matplotlib* and *seaborn* are plug-and-play libraries for generating plots and heatmaps.

3. RELATED WORK

Other related work that are of particular interest are mentioned below.

Bucur et al. succeed in predicting the outbreak size of an epidemic using Random Forest and k-Nearest Neighbor regression [6]. Remarkably, they manage to achieve worst-case R^2 scores of 0.92 for two predictors and 0.96 for three predictors. Interestingly, they find that using the combination of PageRank, Katz and any measure sensitive to the edge density (degree or edge density itself) results in the highest prediction precision. What this research lacks is the analysis of these centralities on larger networks. Furthermore, their focus lies on an epidemiology context only.

Erkol et al. perform a systematic test regarding the performance of several heuristics for the identification of influential spreaders [8]. They use the results of greedy optimization as a baseline of identification performance for the tested heuristics and conclude that relatively simple network metrics, such as closeness centralities, can achieve performances close enough to this baseline. More interestingly, they achieve an increased performance of 2 to 5% when combining topological metrics. However, they have only shown that the results hold for small networks. In our research, we will try to achieve the same performances on larger networks. The node centralities looked at in the context of our research are listed in section 2.2.

Oo et al. apply Social Network Analysis to extract knowledge from social media data, like Twitter [15]. They create a trending topic network graph related to an event, after which they use centrality measurements and link analysis to find influential users. They verify that PageRank is the most important centrality measure that can detect more influential spreaders than any other centrality.

4. RESEARCH QUESTIONS

The main questions that will be answered by this research are stated as follows:

1. How well can various combinations of node centralities predict the spread of information through an OSN?
2. What is the most predictive combination of node centralities?
3. Is the most predictive combination different per spread model?

5. METHOD

This section will present our methodology in roughly chronological order. Important details and semantics are mentioned, but most other code specifics can be found on the project's GitHub page[2].

Graph retrieval and generation.

We retrieve all non-isomorphic graphs of size $6 \leq N \leq 9$ from an enumeration algorithm that is described in [13]. The algorithm is implemented as *geng* in McKay's graph isomorphism checker *nauty*.

Small-to-medium graphs will be generated using the Barabási-Albert model, which is implemented in the *networkx* library. The parameter m, which represents the number of edges a new node will make between existing nodes is chosen to be $\frac{N}{10}$, where N is the size of the graph. M graphs of sizes $50, 100, 200$ and 400 will be generated using this model. The value for M is determined using Learning Curves (discussed in Section 6.2).

Each set of N sized graphs will be used for a separate data set, as explained in Section 2.4.

Deployment of the spread models.

We obtain the estimated total (information) spread of each graph in a data set using the IC and WC spread model (see Section 2.1). The spread probabilities for IC are

[1]The number of graphs we generate for graph size N. The value M is determined later in Section 5.1

[2]`https://github.com/JustinPraas/ResearchProject`

$p \in \{0.01, 0.05, 0.1, 0.15\}$. These probabilities are seemingly low, but higher probabilities will result in complete cascades[3] quicker. Besides that, the probabilities $p \in \{0.01, 0.1\}$ were used in previously peer-reviewed research [11].

For each node contained in each graph, both spread models will be applied I times, such that the estimated total spread is the average of all I repetitions. The constant I must be chosen carefully, since it can impact both the prediction performance and time consumption of the program tremendously. A higher I means greater performance, but long computation times, especially on larger graphs. To make matters worse, large graphs actually require a higher I for a sufficiently good performance, because there are many more ways the information can spread. The coefficient of determination (R^2) is the chosen metric for prediction performance. This metric ranges from 0 to 1, where 1 is the best value.

Choosing combinations of node centralities.

First we will look at the prediction performance of each individual centrality. Next, we form combinations of two centralities. There are $\binom{6}{2} = 15$ different combinations of two centralities, from which the the most predictive will be discussed. Finally, combinations of 3 centralities will be evaluated and discussed.

Calculation of node centrality dictionaries.

For each graph in a data set, the required node centralities for the combination are computed using the *networkx* library. Some important notes:

- All centrality values are normalized.
- The parameter k as specified in the documentation of the *betweenness* centrality is set to the number of nodes in the graph. A higher k results in a better approximation. k is capped at the total number of nodes in the graph.
- Katz centrality uses the following default parameters: $alpha = 0.1$ and $beta = 1.0$.
- PageRank centrality uses the following default parameter: $alpha = 0.85$.

5.1 Data Analysis and Machine Learning

We generate simple but helpful scatter plots once the data sets have been generated. This way, we can inspect the correlation of the estimated total spread with the other node centralities.

Next, we will split the data in training (75%) and testing (25%) sets. Once split, we can fit the RF model on the training data, score the test set and do the same for KNN. Both RF and KNN regression models will return the coefficient of determination as score metric.

We tune the hyperparameters for the machine learning models while using 10-fold cross-validation to evaluate those models. This diminishes bias and overfitting. *GridSearchCV*[4] is used for this purpose for both RF and KNN regression models.

How many samples we need in a training data set can be discovered by plotting a learning curve. The training size is plotted against the cross-validation score and training score. From the resulting plots we can confirm that we have sufficient samples for meaningful R^2 score and we can determine whether our learning methods are proper (not overfitting or biased).

[3]The phenomenon where each node in the network is ultimately informed

[4]GridSearchCV is part of the *sklearn* library

6. RESULTS

This section is divided in subsection per graph context. Within these subsections, we look at the prediction results from different combinations of node centralities, starting with one node centrality and finishing with combinations of three centralities. We show and analyse the prediction results of Random Forest Regression in this section. While we also show prediction results of k-Nearest Neighbours, those results will be put in the appendix. At the end of this section we will briefly discuss the differences between RF and KNN.

The figures shown in this research should be interpreted using Figure 3, which depicts the color bars used in the coming figures.

6.1 All small non-isomorphic graphs

The results in this subsection are based on data sets for all non-isomorphic graphs. These data sets have been generated with $I \in \{200, 1000\}$, where I is the number of spread repetitions.

Data analysis.

From the retrieved data sets we can confirm that there is some correlation between the centralities and the information spread, as expected. This is especially evident in the *katz*, *pagerank* and *closeness* scatterplots in Figure 4. Focusing on the top-left scatter plot, we can see that a higher PageRank of a seed node results in an increased total spread. This essentially means that higher PageRank nodes will influence more nodes in the network. It is also good to note how the hue shifts from red to blue while the PageRank increases. This also shows that there is some correlation between degree and information spread.

Prediction performance.

We have generated hundreds of results, each of which shows the R^2 scores. Only the most noteworthy results will be displayed in this research. However, you can find more on the GitHub page. Furthermore, we will only show results that take $I = 1000$ spread repetitions into account. Increasing the spread repetitions will result in greater performance, but longer computation times. It is notable that we got an average R^2 increase of 0.1 up to 0.2 when going from $I = 200$ to $I = 1000$ repetitions.

RFR - Single centrality

The predictive performance of single node centralities in small graphs is displayed in Figure 5. It is clear that *degree and closeness* outperform the other centralities, especially when the IC spread model is used. Remarkably, however, for degree and closeness the WC spread model

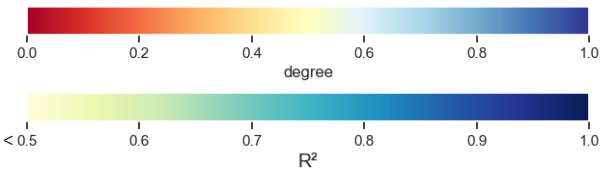


Figure 3. Color bars used for the scatter plots and heatmaps respectively.

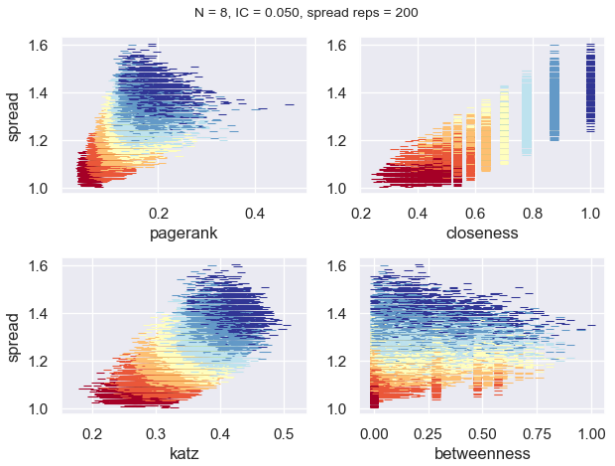


Figure 4. Scatter plots of small graphs of size 8. IC is applied on each node, with spread probability $p = 0.05$ and $I = 200$ repetitions. A correlation between the total estimated spread and the centralities is especially evident in the top and bottom-left plots.

is less predictive than the IC model, whereas in the other node centralities it is the other way around.

RFR - combination of 2 centralities
In Figure 6, heatmaps are displaying the prediction performance of the most noteworthy 2-combinations of node centralities. The most predictive combination with both the IC and WC spread models is *degree and PageRank*, achieving great prediction performance as high as $R^2 = 0.983$. The worst predictive combination in terms of IC but best in terms of WC is *PageRank and betweenness*. The largest difference in prediction performance between IC and WC is apparent when analysing the *PageRank and betweenness* combination.

RFR - combination of 3 centralities
The two most notable heatmaps that were generated from using combinations of 3 node centralities are shown in Figure 7. From the left heatmap we can see that increasing p of the spread model in small graphs increases the prediction performance. Furthermore, we observe that the WC spread model in combination with 3 node centralities predicts the total information spread best. In fact, in the right heatmap of Figure 7 you can see we achieve an R^2 score of 0.985 for $N = 7$ and spread model WC.

6.2 Small-to-medium preferential attachment graphs

The results in this section are based on data sets for graphs generated with preferential attachment. These data sets have been generated with $I \in \{200, 500, 1000\}$ spread repetitions and $M \in \{200, 100, 50, 25\}$ mapped on graph size $N \in \{50, 100, 200, 400\}$ (e.g. 200 graphs of 50 nodes).

The values for M have been chosen such that the total samples in the data sets are equal across the data sets ($M * N = 10000$). The number 10000 was chosen because capping the maximum number of data samples considerably reduced computation time on these larger graphs (most notably when applying the spread models). Furthermore, in the following section we will show that no more than 10000 data samples are necessary in order to obtain usable prediction performance data.

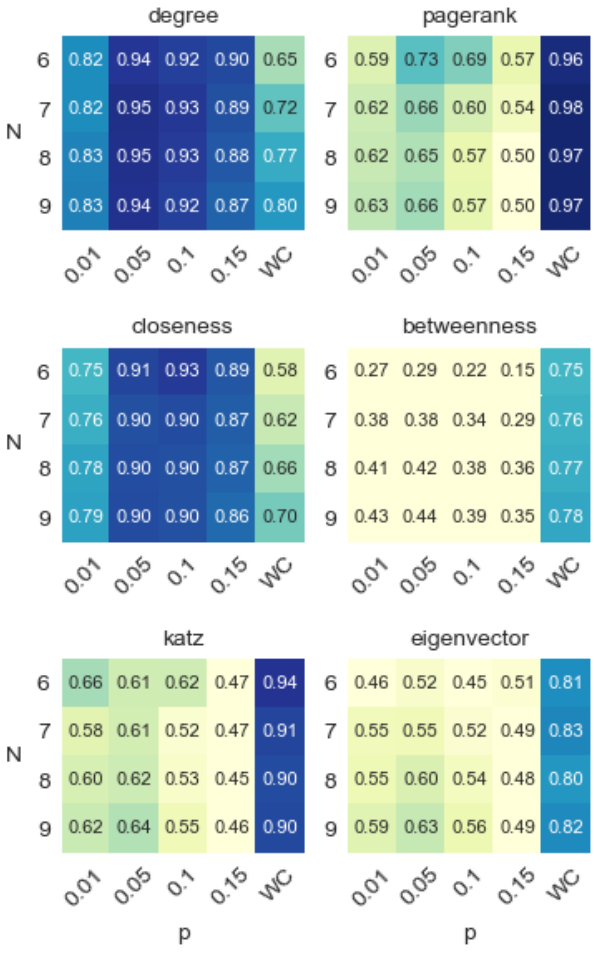


Figure 5. Heatmaps showing the prediction performance of single node centralities in small graphs. Parameters: $I = 1000$, IC probabilities: $p \in \{0.01, 0.05, 0.1, 0.15\}$, $N \in \{6, 7, 8, 9\}$

Data analysis.

Similarly to the small graphs, we can confirm from the retrieved data sets that there is again a correlation between the centralities and the information spread (see Figure 8). We see a few differences when we compare these scatter plots to those in Figure 4. The first thing that catches the eye is how the Katz centrality loses its correlation with estimated spread. Why this happens is explained later. Moreover, the other scatter plots have become more dense and have a more slim and curvy shape.

Furthermore, within the context of (randomly) generated large graphs, we plotted learning curves to estimate the necessary number of data samples. In the long term, this number will save us time as we can limit the necessary computations according to this number. In Figure 12 in the appendix we show the learning curves for each node centrality, each on the same 50 graphs of size 200, with an IC spread probability of 5%. From this figure we can see that increasing the training set size does not increase the cross-validation score at around 2000-3000 samples. Nevertheless, all the heatmaps in the following section are generated using 10000 samples per data set. The poor cross-validation score for Katz is explained later.

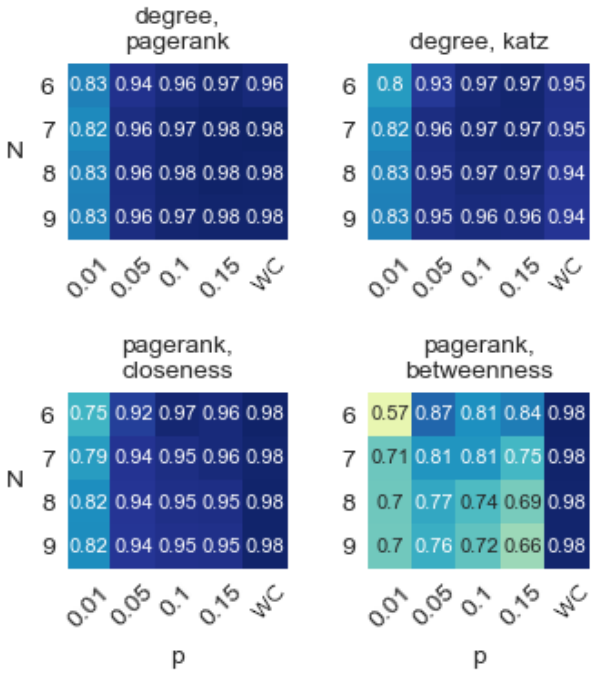


Figure 6. Heatmaps showing the prediction performance of the most noteworthy 2-combination of node centralities in small graphs. Parameters: $I = 1000$, IC probabilities: $p \in \{0.01, 0.05, 0.1, 0.15\}$, $N \in \{6, 7, 8, 9\}$

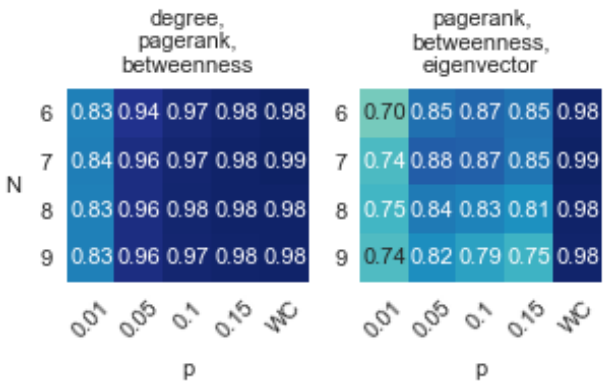


Figure 7. Heatmaps showing the prediction performance of three node centralities in small graphs. Parameters: $I = 1000$, IC probabilities: $p \in \{0.01, 0.05, 0.1, 0.15\}$, $N \in \{6, 7, 8, 9\}$

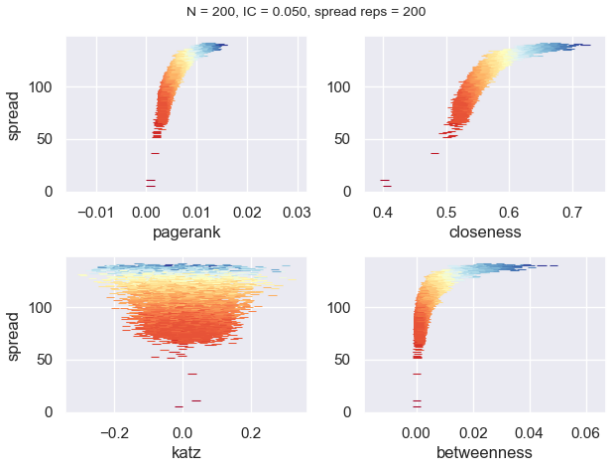


Figure 8. Scatter plots of 50 graphs of size 200. IC is applied on each node from all graphs with $I = 200$ repetitions. A correlation between the total estimated spread and the centralities is evident in these plots.

Prediction performance.

Switching context from small to medium size graphs, we similarly analyse the prediction performance of centralities and combinations thereof (see Section 6.1).

RFR - Single centrality

We first take a look at the R^2 scores of single node centralities. These numbers are depicted in Figure 9. There are a couple of things that we can quickly observe from these heatmaps. Firstly, the predictive performance of the Katz node centrality (given its method parameters) is only sufficient for graphs of size $N < 100$. The Katz values of each node are all equal when increasing the graph and thus no predictions can be made from that data. This trend may be due to the fact that Katz is more suitable for directed acyclic graphs [14].

Furthermore, we observe that for all other centralities (disregarding Katz) have very high highs in the context of most IC probabilities and, especially, WC. The IC spread model is less predictive of the total spread than WC as we can see from the figure.

Finally, we see a yellow or light-blue area when increasing N and p, indicating bad prediction performance. This is due to the fact that for each node, there was almost always a complete cascade or a near-complete cascade. This is very likely given the high probability and the number of possible neighbours in larger graphs. It is now that we can see the effect of the WC spread model, which counters the property of IC causing complete cascades (also explained in Section 2.1).

RFR - combination of 2 centralities

The heatmaps in Figure 10 that result from combining all centralities look very similar. The combination *degree and PageRank* performs best (again). Perhaps not surprising, the power of a combination of centralities does not help in increasing the prediction performance when (near) complete cascades are occurring.

RFR - combination of 3 centralities

Figure 11 shows the prediction performance heatmaps of the most two most noteworthy combinations. It is worth mentioning that the R^2 scores slightly increase, compared to when combinations of two centralities are used. Furthermore, the WC spread model again outperforms IC in most instances, especially when p and N increase. Finally, the combination as used on the right of Figure 11 shows improved prediction performance for instances where complete cascades occur (when p and N both increase). This is remarkable, because the only difference in combination is the Katz centrality versus degree and previously we saw that Katz did not predict well for large N.

6.3 Differences between RFR and KNN

Now that we have seen the different centralities and their impact on prediction performance, it is time to see what difference a Machine Learning model can make. On the

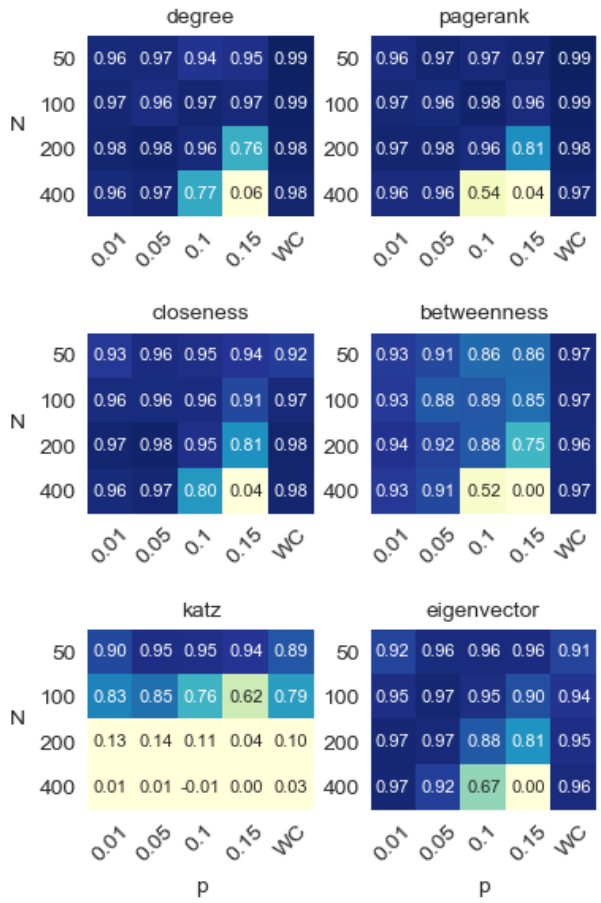


Figure 9. Heatmaps showing the prediction performance of the single node centralities in small-medium sized graphs. Parameters: $I = 1000$, IC probabilities: $p \in \{0.01, 0.05, 0.1, 0.15\}$, $N \in \{50, 100, 200, 400\}$, $M \in \{200, 100, 50, 25\}$

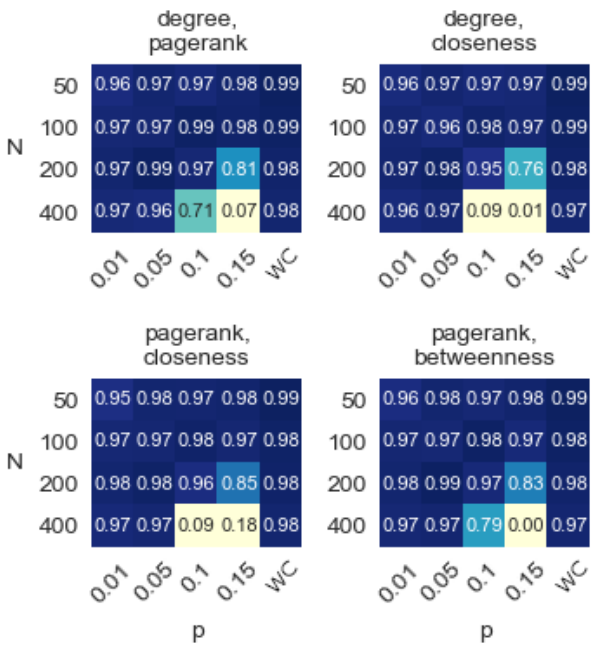


Figure 10. Heatmaps showing the prediction performance of the most notable combination of 2 node centralities in small-medium sized graphs. Parameters: $I = 1000$, IC probabilities: $p \in \{0.01, 0.05, 0.1, 0.15\}$, $N \in \{50, 100, 200, 400\}$, $M \in \{200, 100, 50, 25\}$

first page of the Appendix you can find heatmaps that are generated using KNN. We find that, while there are minor differences in R^2, the same conclusions for the centralities hold.

7. CONCLUSION

In this research we have shown which node centralities (or combinations thereof) are particularly predictive of the total estimated information spread and which are not. We have done this by employing the IC and WC spread models over each node in sufficiently many small-to-medium graphs generated using the Barabási-Albert random graph generation model and all non-isomorphic graphs of size $6 \leq N \leq 9$. Finally, the Random Forest regression learning method and k-Nearest Neighbors regression make predictions on the total estimated information spread using 10-fold Cross-Validation, resulting in R^2 scores, the chosen metric for prediction performance.

Small graphs.

We showed that the degree and closeness centralities work particularly well (individually) on small graphs, using IC. PageRank is a great predictor of influence spread in small graphs using WC. This is in line with the results of [15].

The best combinations of two centralities on small graphs are *degree & PageRank* and *PageRank & closeness*, for both IC and WC. A combination of three node centralities that predicts the influence spread best is *degree & PageRank & betweenness* for both IC and WC, whereas the combination *PageRank & betweenness & Eigenvector* performs worst for IC, but is tied best for WC.

Small-to-medium graphs.

In the context of small-to-medium graphs we showed that degree, pagerank and closeness perform best for IC. Furthermore, we observed that the Katz centrality is a bad predictor for larger graphs, as this centrality is more suitable for directed acyclic graphs. Equally important, the heatmaps depict poor performance when both N and IC probability p grow. This is due to the likeliness of a (near) complete cascade, causing insufficient correlation between the centralities and the influence spread.

Moving to combinations of two centralities, we showed that *degree & PageRank* again works best for both IC and WC. Combining the centralities *degree & PageRank & betweenness PageRank & betweenness & Katz* result in the best prediction performance for IC and WC.

IC versus WC.

The heatmaps show that WC outperforms IC in most cases. These spread models were solely used for comparison reasons; which spread model to use remains context specific.

The best predictors.

Generally speaking, using the combination *degree & PageR-*

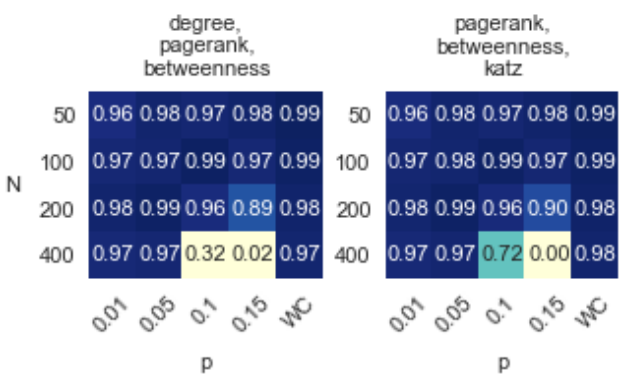


Figure 11. Heatmaps showing the prediction performance of the three node centralities in small-medium sized graphs. Parameters: $I = 1000$, IC probabilities: $p \in \{0.01, 0.05, 0.1, 0.15\}$, $N \in \{50, 100, 200, 400\}$, $M \in \{200, 100, 50, 25\}$

ank & betweenness as influence spread predictor will result in the best prediction performance. The highest R^2 score that we were able to obtain using this combination was for large graphs, using WC: 0.989

8. FUTURE IMPROVEMENTS

The size and number of graphs used in this research was limited due to the relatively short duration of the research. It might be interesting to apply the same method to larger graphs or perhaps even real-life graphs obtained and anonymized from Facebook. Data sets for these real graphs already exist [12].

The approximation of real life scenarios could potentially be improved by not just considering single seed nodes, but more seed nodes at a time, as it is common for marketing strategists to inject an advert in more than one place within a network.

Additionally, it might be interesting to work with more, less trivial node centralities. Examples of such centralities are: k-shell, LocalRank, Collective Influence and Non-backtracking. A comparison and summary of such centralities is provided in [8].

9. ACKNOWLEDGEMENTS

Firstly, I want to thank my supervisor Doina Bucur for her support and enthusiasm during this research. Many of the obtained results would not have been made possible without her help.

Generating tens of data sets and hundreds of results would not have been possible with just my laptop. My sincere appreciation goes out to Geert Jan Laanstra for giving me access to the University's compute lab and assisting me with the environment setup.

10. REFERENCES

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

[2] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

[4] P. Bonacich. Factoring and weighting approaches to status scores and clique identification. *The Journal of Mathematical Sociology*, 2(1):113–120, 1972.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International Conference on World Wide Web 7*, WWW7, page 107âĂŞ117, NLD, 1998. Elsevier Science Publishers B. V.

[6] D. Bucur and P. Holme. Beyond ranking nodes: Predicting epidemic outbreak sizes by network centralities, 2019.

[7] S. Catanese, P. De Meo, E. Ferrara, and G. Fiumara. Analyzing the facebook friendship graph. volume 685, 01 2010.

[8] Å. Erkol, C. Castellano, and F. Radicchi. Systematic comparison between methods for the detection of influential spreaders in complex networks. *Scientific Reports*, 9(1), Oct 2019.

[9] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[10] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, Mar 1953.

[11] 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.

[12] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[13] B. D. McKay. Applications of a technique for labelled enumeration. *Congressus Numerantium*, 40:207–221, 1983.

[14] M. E. J. Newman. *Networks an introduction*. Oxford University Press, 2018.

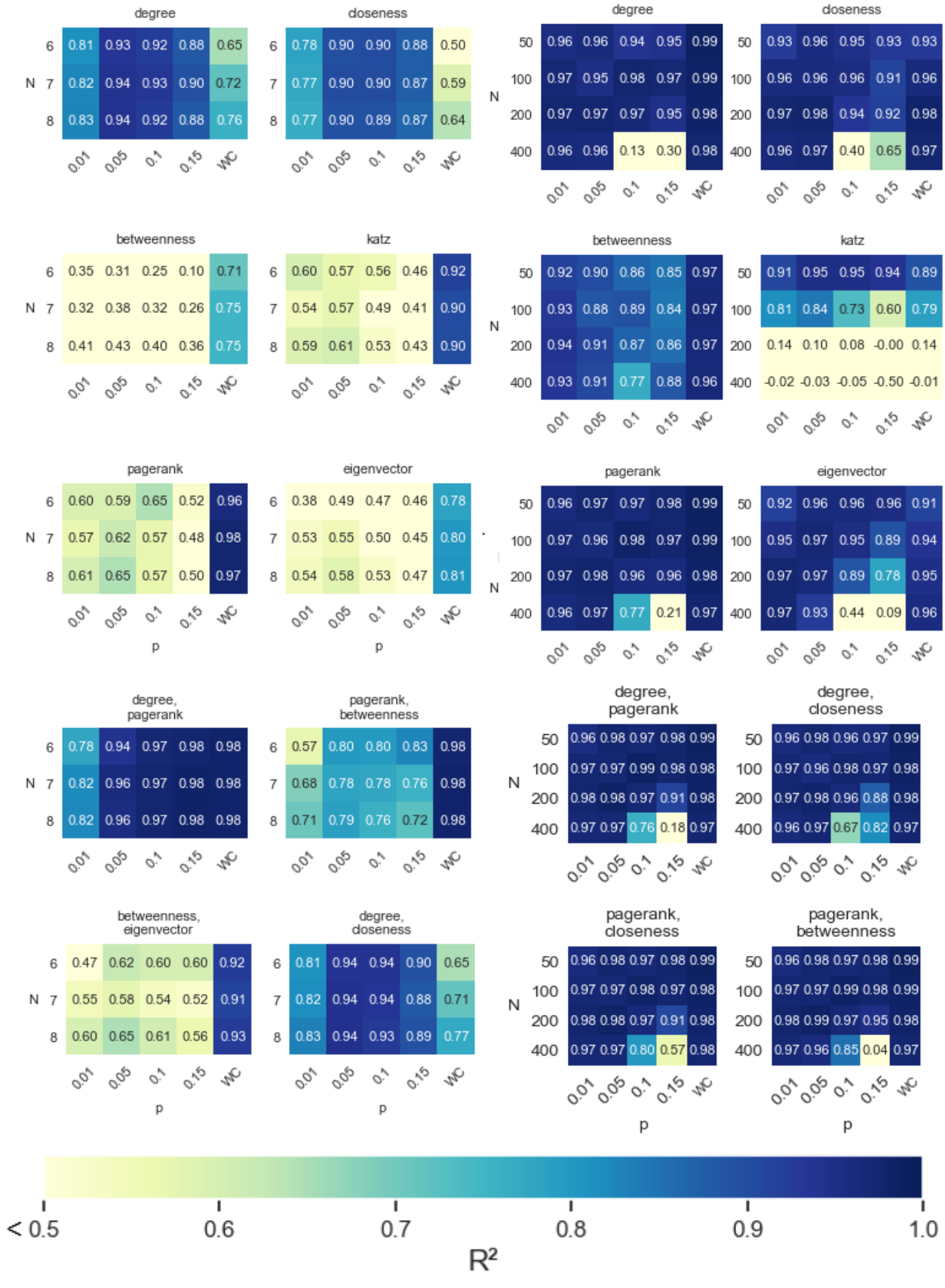
[15] M. M. Oo and M. T. Lwin. *Analysis of online social network after an event*, volume 849 of *Studies in Computational Intelligence*. 2020.

[16] G. Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, 1966.

[17] M. Saravanakumar and T. SuganthaLakshmi. Social media marketing. *Life Science Journal*, 9(4):4444–4451, 2012.

APPENDIX

Heatmaps generated using the KNN regression method, and 1000 repetitions are used for the spread models.



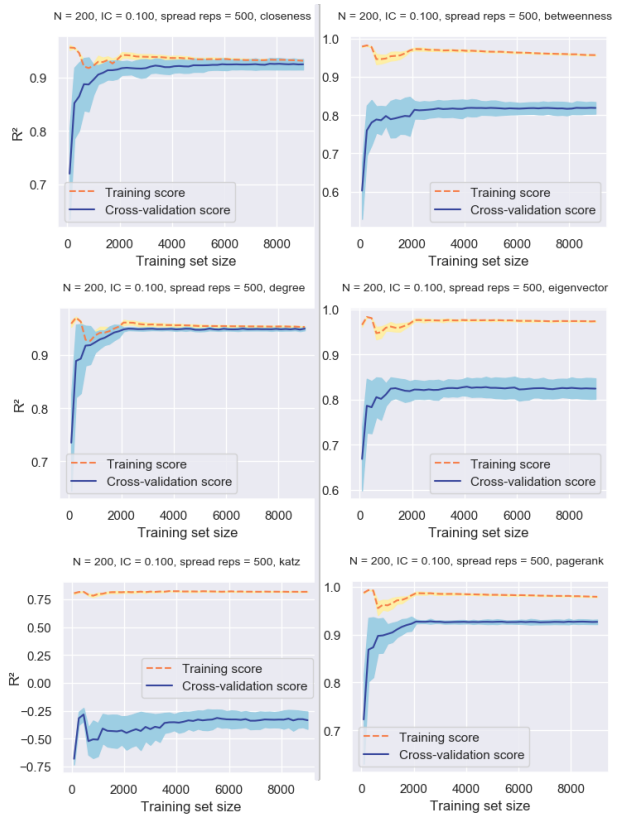


Figure 12. Learning curves for each node centrality , each on the same 50 graphs of size 200, with a IC spread probability of 5%. From top to bottom, left to right: closeness, betweenness, degree, EigenVector, Katz, PageRank

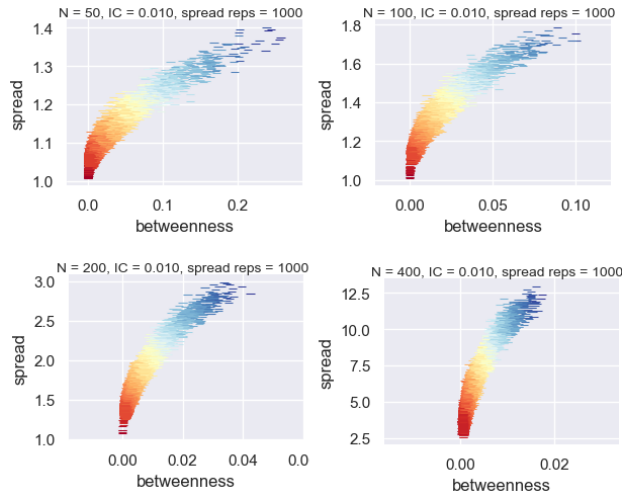


Figure 13. Scatter plots showing the change in spread with graph size N as variable

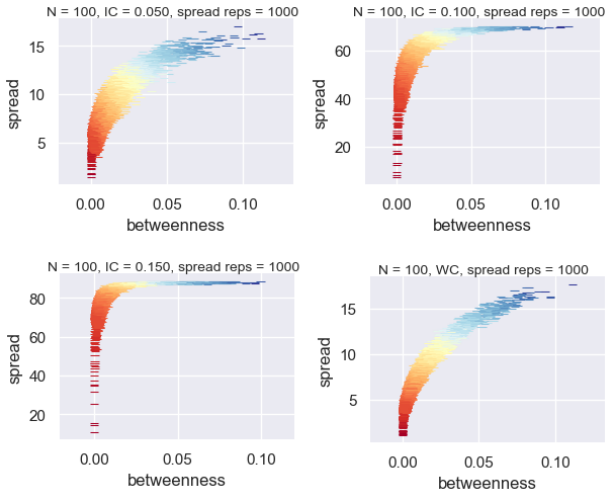


Figure 14. Scatter plots (betweenness on the horizontal axis and degree as hue) showing the change in spread with spread probability (and model) as variable. The bottom right plot shows the Weighted Cascade model.

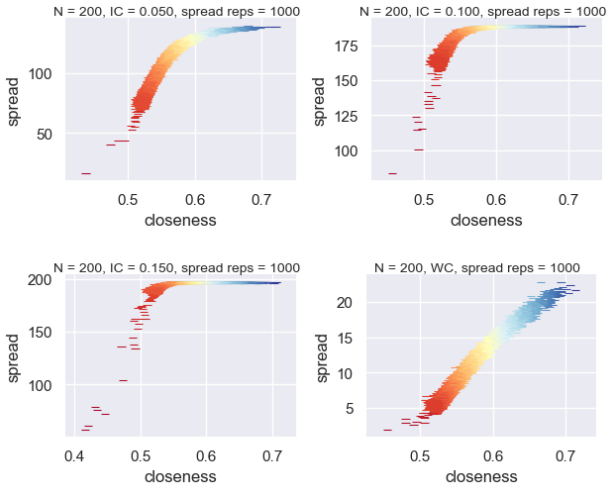


Figure 15. Scatter plots (closeness on the horizontal axis and degree as hue) showing the change in spread with spread probability (and model) as variable. The bottom right plot shows the Weighted Cascade model.