RUAN, JUNJUN, M.S. Efficient Link Cuts in Online Social Networks. (2015)
Directed by Dr. Jing Deng. 39 pp.

Due to the huge popularity of online social networks, many researchers focus on adding links, e.g., link prediction to help friend recommendation. So far, no research has been performed on link cuts. However, the spread of malware and misinformation can cause havoc and hence it is interesting to see how to cut links such that malware and misinformation will not run rampant. In fact, many online social networks can be modelled as undirected graphs with nodes represents users and edges stands for relationships between users. In this paper, we investigate different strategies to cut links among different users in undirected graphs so that the speed of virus and misinformation spread can be slowed down the most or even cut off.

Our algorithm is very flexible and can be applied to other networks. For example, it can be applied to email networks to stop the spread of viruses and spam emails; it can also be used in neural networks to stop the diffusion of worms and diseases. Two measures are chosen to evaluate the performance of these strategies: Average Inverse of Shortest Path Length (AIPL) and Rumor Saturation Rate (RSR). AIPL measures the communication efficiency of the whole graph while RSR checks the percentage of users receiving information within a certain time interval. Compared to AIPL, RSR is an even better measure as it concentrates on some specific rumors' spread in online networks. Our experiments are performed on both synthetic data and Facebook data. According to the evaluation on the two measures, it turns out that our algorithm performs better than random cuts and different strategies can have better performance in their suitable situations.

EFFICIENT LINK CUTS IN ONLINE SOCIAL NETWORKS

by

Junjun Ruan

A Thesis Submitted to
the Faculty of the Graduate School at
The University of North Carolina at Greensboro
in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Greensboro
2015

Approved by

Committee Chair

To my parents.

APPROVAL PAGE

This thesis written by Junjun Ruan has been approved by the following committee of the Faculty of The Graduate School at The University of North Carolina at Greensboro.

Committee Chair _____
Jing Deng

Committee Members _____
Shan Suthaharan

_____
Lixin Fu

_____
Date of Acceptance by Committee

_____
Date of Final Oral Examination

ACKNOWLEDGMENTS

In full gratitude I would like to acknowledge all the following individuals who encouraged, assisted and supported me on the way to pursue my graduate degree.

First and foremost I wish to express my sincere thanks to my advisor Dr. Jing Deng for his continuous support in my study and research. I have enjoyed the opportunity to learn from his knowledge and experience. His guidance helped me through all the hardest time of the research and writing of this thesis. Without his encouragement and assistance, I can hardly finish this work.

Besides, I would like to thank the rest of my thesis committee: Dr. Shan Suthaharan and Dr. Fu, who came to my thesis defense and gave me so much important advice.

Furthermore, I would like to thank my classmates Rui Da, Jie Hou and Kun Chen for their support over the last few years and the happy time of learning and playing together.

In addition, I would like to thank my parents for their love and encouragement. Thanks to my husband for his moral support and advice.

Last but not the least, I take this opportunity to express gratitude to all of the Department faculty members for their help and support.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

1.1 Online Social Networks

One of the biggest innovations in the 20th century is the computer network, which allows computers to exchange data. In computer networks, data is transferred in the form of packets among remote computers along network links. The construction of computer networks brings numerous benefits, such as file sharing, internet sharing and resource sharing, etc. Along with the fast development of computer networks, there are different types of topologies. The simplest one is called point-to-point network: two endpoints communicate with each other through a permanent link [8]. However, the number of hosts allowed to communication is limited. The other topology is a bus: all the computers or services are connected together to a long cable. In this way, they can communicate with each other. When each host is directly connected to a central controller named hub or switch, a new topology called star is formed [8]. Another topology is a ring: each device is connected to other two devices with one in each side, and finally formed a shape of ring. The data can only be transferred around the ring in one direction, each device is used to strength the signal. However, it is very vulnerable, since the destroy of one link can interrupt the transmission of the whole graph. A more complicate network is called mesh network, which includes fully connected network and partially connected [8]. In a fully connected network, each node has a link with every other node in the network. In a partially connected network, some nodes has more than one link with other nodes in the network.

Along with the widespread of networks, our communication evolution has spawned new tools such as online social networks, which change the communication way among people such as young generations. In the past, people can only interact face-to-face or through phone calls or text messages, which limit people's circle of friends. Now online social networks build the platforms for individuals to create a public profile and a list of users with whom to share interests and activities. In this way, the friend circles of people are significantly broaden. A good example of online social network would be Twitter. Once a user has registered a Twitter account, he or she can add other users to his or her friend list, post and read others' messages. These messages are called "tweets" [15] and can be sent through the web wherever there is a computer and network. Also, with Twitter, a user can interact with other users no matter how far they are. For example, Twitter can make a user close with famous people such as president Obama. A registered user can read Obama's Twitter updates and even make comments. This can hardly be imagined in the old days without social networks. In addition, due to the popularity of smart phones and tablets, mobile applications (Apps) like Facebook and Twitter can be easily downloaded from platforms such as Apple App Store and Google Play, which change the way of online social networking from web-based communication to mobile phone-based communication. Also, the convenience of these handy devices improves the frequency of online communications in ways that are unimaginable to pre-smartphone days.

1.2 Graph Theory

Online social networks can be represented by graphs. So many terms in graph theory should be clarified first. A graph can be represented by a formula $G = (V, E)$, where V stands for nodes or vertices and E stands for edges or links. An edge can

be directed or undirected depending on whether it has a specific direction or not. In addition, an edge can be assigned a number as a weight, which can be nonnegative, integral or positive, etc. A node of a graph can also have different graph properties. The degree of a node, which is defined as the number of edges incident to that node. For a node, the number of edges with directions pointed to a node is called the indegree of the node and the number of edges with directions pointed to other nodes from a node is its outdegree [11]. Moreover, a node can has one edge or multiple edges. A path is defined as several edges connecting a series of vertices, the number of such edges is called a path length. If a path has the same starting and ending vertex, which is called a closed path; if a path visit any vertex only once it is a simple path; if a path is both simple and closed, then it is called a cycle [11].

Another important term is graph connectivity, which is correlated to the spread of information including rumor in online social networks. There are many measures of connectivities that can be chosen. The density of a graph can be one of the measures, which is defined as the number of actual edges divided by the maximum number of edges that a graph can reach. It enables us to deeply understand the information transmission speed among the nodes or which node has the potential to add more links. The density of a graph can be easily computed according to its definition. In an undirected graph, the density is $\frac{2|E|}{|V|(|V|-1)}$. In a directed graph, the density is $\frac{|E|}{|V|(|V|-1)}$. The diameter of a graph can also be another measure, which is defined as the longest shortest path between any pair of vertices in the graph. It gives us a insight on the minimum numbers of steps that are needed for any node to visit all other nodes in the graph. One of such connectivities checks the minimum number of elements (nodes or edges) that need to be removed to disconnect the remaining nodes

from each other thus splitting the graph [7]. Average shortest path length can also be a measure, which is defined as the average distance between any pair of nodes of the whole graph [9]. It measures the efficiency of information diffusion in a graph. Clustering coefficient is also a good candidate measure. The clustering coefficient of a node can be calculated as: the number of existing edges connecting that node's neighbors to each other divided by the maximum possible numbers of such edges [7]. The global clustering coefficient is measured as the number of closed triplets over the total numbers of triplets (both closed and opened) [7]. In all, clustering coefficient is a ratio with a value between 0 and 1. It gives us an indication of the probability that friends of a user are friends of each other.

The connections in online social networks can be modelled as graphs consisting of many nodes (users) together with a set of edges. These graphs can be either directed or undirected. For example, Twitter is a directed graph while Facebook is an undirected graph. Whenever two users become friends in Facebook, there is an edge between them. A graph is connected when there is a path between every pair of nodes [7].

There are many types of graphs. A graph can be either directed or undirected, depending on whether an edge has a specific direction or not. A graph can also be weighted or unweighted depending on whether weights are assigned to it. A simple graph is a graph with no self-loops and no multiple edges, while a multigraph is a graph that multiple edges are allowed to connected to a pair of nodes [11]. A complete (or full mesh) graph means every node is connected to other nodes. It can also be defined as: every node has degree N-1 in a graph with N number of nodes. A connected, simple graph without cycles is called a tree [11].

There are many problems related to graphs is worth to analyze. One of them is the shortest path problem. The single-source shortest paths problem, in which we need to find the shortest paths from a given source vertex to another vertex (or several vertices) in the graph [6]. One simplest and famous algorithm is bread-first search (BFS). It starts at the tree root and visits its neighbors first before moving to the next level of nodes. It can be applied to both directed graph and undirected graph with time complexity $O(E)$ [6]. In this paper, we use BFS as one of our measure. The detail information of BFS can be found in out later section. However, BFS algorithm cannot be applied to weighted graphs. Dijkstra's algorithm makes up for the deficiency of bread-first search algorithm with running time $O((V + E) \lg V)$ [6]. It is widely used in network routing protocols. Another algorithm in solving single-source shortest paths problem is called Bellman-Ford algorithm. Compared to Dijkstra's algorithm, it is slower since it runs in $O(|V| \cdot |E|)$ time [6], but more versatile, as it can work on graphs with negative edges. The all-pair shortest paths problem, is to find a shortest path for every pair of vertices in the graph [6]. Floyd-Warshall algorithm can be applied to both undirected and directed (no negative cycles) graph with time complexity $O(V^3)$ [6].

1.3 Rumor Issues in Online Social Networks

Online social networks such as Facebook and Twitter make news dissemination faster than ever. However, virus, malwares, misinformation (such as rumors) will also propagate quickly in such networks. Such propagation speed is hard to imagine in the old days of fliers and/or floppy-drive virus infections.

In general, more edges in a graph will likely lead to higher speed of (mis)information spread, and vice versa. A natural solution to slow down the spread of misinformation

or rumor is to cut some edges. The question is which of the edges to cut so that the spread of misinformation or rumor slows down the most.

In this work, we focus on the problem of how to choose edges to cut. A cut is defined as the removal of an edge. With the removal of some edges, the new graph will be less connected and thus it will take longer time for rumor to reach nodes. More specifically, a rumor will reach fewer nodes within a certain delay, limiting its impact. And the goal is to lower network connectivity the most with efficient cuts.

We designed an algorithm named CDegree Cut, which cuts nodes' edges depending on the choice of degree and gets the best result. It is easy to understand by a given situation as shown in Figure 1: two popular nodes A and B have lots of friends in the graph and there are no connections among all of their friends. It is obviously that cutting the edge between node A and B can make the graph disconnected: split into two sub-graphs. As there is no path between node A and B, the rumor cannot be transferred between the two sub-graphs. In this case, we choose edges of nodes with the highest degree to cut.

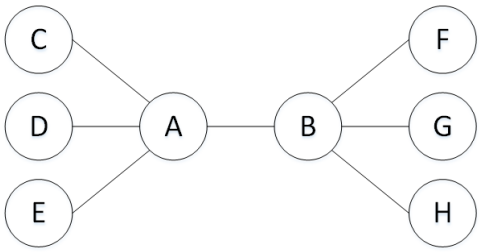

Figure 1. A Simple Network.

Note that the above graph and discussion can be misleading. Usually there are other paths, other than the link between these two popular nodes, connecting such

subgraphs. Therefore, cutting just one link may not disconnect the graph. Still, it is possible that cutting some of these links may help disconnect them.

1.4 Document Organization

The following paper is divided into four chapters followed by the references and the appendix.

- Chapter II introduces the related work.
- Chapter III explains our schema and solution in detail.
- Chapter IV shows the simulations results to evaluate our work.
- Chapter V summarizes our work and discuss future works.

CHAPTER II

RELATED WORK

Our work deals with link cuts to reduce rumor diffusion in online social networks, which is the opposite of link prediction. So those designs of link prediction algorithms and analysis of rumor and network features can give us some hints. In this chapter, we discuss several related work in the following.

2.1 Background in Online Social Networks

In the real world, the online social networks seems very complex, but some studies showed it is actually a "small world". In the 1960s, Milgram et al. considered a question: randomly choose two people from a large populations, through how many intermediate ones they can know each other. That is where the famous "small world" problem came [18]. In order to solve this problem, Milgram et al. set up several experiments to study how some start persons forward documents to target persons through some intermediaries in three distinct subgroups. Here they defined a complete chain as the number of intermediaries, through whom the start persons can forward documents to target ones successfully. Finally they found the relationship between the number of intermediaries and the number of chains, also the relationships between the number of incomplete chains and remove from start at which termination occurred [18]. Their study helped us realize the world is not as "big" as we ever thought and laid the groundwork for further studies in interconnectedness in large social networks.

In 1998, a first small world model was generated by Watts and Strogatz, also they found that networks can be classified by two structural features: clustering coefficient, and characteristic path length [19]. According to the two classifiers, the small world is between regular graph and random graph, it is highly clustered and with short characteristic path length, in which diseases can be transmitted easier than in regular lattices. Three empirical examples of small world networks investigated by them are the collaboration network of film actors, the electrical power grid of the western United States and the neural network of the worm C. elegans [19]. Finally they found that a small world can be constructed by rewiring a regular network. In 1999, Watts and Strogatz published another paper, in which they named their small world model as Watts-Strogatz model [3]. Now the small world network is still a popular topic which is studied by a lot of researchers.

2.2 Link Prediction Problems

In researches related to link changes, many researchers have investigated the link prediction problem. Backstrom and Leskovec [2] assigned weights to the edges and identified the heavy weights indicating the occurrence of the new links. Based on such an observation, they developed an algorithm to predict potential links based on Supervised Random Walks.

In addition, due to the fast growing of social networks, links may be missing quickly. Many existing algorithms are unable to deal with this situation. Fire et al. developed an algorithm based on the extraction of graph topological features; at the same time, they pointed out a new topological feature named friends-measure, which worked better than the traditional common-friends. Their algorithm can widely be

used in indicating missing links and helping users to discover new friends in real online social networks.

Link prediction problem also exists in recommender systems for online shopping website like Amazon. Much work has been reported recently in this field. Sarwar et al. proposed an item-based collaborative filtering recommendation algorithm by identifying relationships between different items, it outperformed traditional user-based algorithm in producing higher quality recommendations and more recommendations per second for millions of users [16].

In addition, cold start problem is also related to recommender systems. In order to recommend those never or rarely rated items, Schein et al. developed three strategies according to specific real world conditions by combining content and collaborative data. Also, they gave a new measure named CROC curve to evaluate the performance of different algorithms [17]. Finally, it turned out that their algorithm performed better than a naive Bayes classifier.

Another paper relevant to this research was published by Huang. Unlike Sarwar et al. who only focused on the linkage information itself [16], Huang exploited the connection between link prediction and graph topological structure, analyzed generalized clustering coefficients and finally designed a cycle formation model [9], which made a big progress on solving link prediction problem.

2.3 Rumors' Influence in Online Networks

These studies on link prediction problem gave us some hints on solving link cuts problem. In addition, many research studies have been carried out on analyzing the behavior and damage of the rumor/misinformation in different networks, which motivates us to design an algorithm to stop rumor's propagation.

In online social networks, reducing connectivity can slow down or even stop rumor spreading. Chierichetti et al. studied the performance of rumor spreading in the classic preferential attachment model of Bollobás et al, compare the efficiency of disseminating information among different strategies: the standard PUSH-PULL strategy, PUSH and PULL strategy [5]. These strategies have been insightful in the development of our strategies in this work.

In email networks, viruses can be transmitted quickly through attachments. An email network is a graph with email address books as sources and edges representing communication. Newman et al. presented techniques to prevent virus infection by analyzing how they spread [13].

In citation networks, nodes represent papers, edges represent citations. Therefore, if one paper cites another, one directional edge would be added between these two. Hummon and Doreian developed a new algorithm to analyze a citation network describing the development of DNA theory, the selected papers are identified through their structural connectivity in the network [10].

Recently a study on the analysis of connectivity damage to a graph was done by Cartledge and Nelson. The motivation for them to work on this paper is the traditional methods like using the size of the largest connected component can not reflect the damage to a graph especially when it is disconnected, instead they gave a new measure: average inverse path lengths (AIPL) [4]. This measure can even be used on the measure the influence caused by adding new edges in the graph. It offered an idea for us on our work later.

CHAPTER III

SCHEME DESIGN

In this section, we proposed an algorithm to reduce network connectivity, from which 16 strategies can be chosen.

3.1 CDegree Cut

In general, the selection of edges to cut can be separated into two steps: deciding which node's links to cut and deciding which link from the chosen node to cut. While there are many different selection criterion in making these two decisions, we focus on a natural node property: node degree, defined as the number of neighbors that each node has. In the first decision, we can see that there are four different strategies: high-degree, medium-degree, low-degree, random. High-degree selection is to choose the node with the highest node degree. Similarly, medium-degree and low-degree selection are based on node degree being medium/lowest among all nodes. Random selection is just randomly picking one node. The second decision again can be made with four different strategies: high-degree, medium-degree, low-degree, random.

Combining these two decision, we would have 16 different strategies. Two examples are high-high and low-low selections. In the high-high selection, we choose the node with highest degree and then sort all neighbors of the node based on the neighbors' degree from high to low. Edges will be chosen from the list in the same order. In the low-low selection, the selection is basically the opposite. The node with the lowest degree will be chosen first. Then all neighbors of the node will be sorted based on their degrees from low to high. And cuts are performed from the sorted list.

There is another parameter that will impact computation overhead, called L. L is the number of edges to cut before sorting is performed again. Since cutting the edges will change node degrees, sorting is needed in order to ensure that all further selections are made accurately. Therefore, L is the "knob" to tune how strictly the chosen algorithm is followed. Two extreme cases are $L = 1$ and $L = \infty$. When $L = 1$, sorting is performed after every cut, rendering high overhead. When $L = \infty$, no sorting will be performed. All edges from a chosen node will be cut until K cuts are made. In fact, when L is greater than the maximum degree of all nodes, these 16 strategies collapse to 4 as all links from a chosen node will be cut before L links are exhausted.

Note that it may seem that such K cuts are sequential, i.e., cutting one edge after another. In fact, these are all cut at once and we are interested in finding the best set of edges to cut so that rumor will spread slowest in the new graph.

Suppose we are given an undirected graph G and need to find K edges to cut in order to reduce G's connectivity. We define two functions: $f()$ for the choice of which node's link to cut and $g()$ for the choice of which links of a chosen node to cut. The two functions can be executed according to an input strategy and the updated graph. For example, if we choose high-random cuts this time, function $f()$ will rank nodes' degrees from high to low and choose the highest one's index each time. Then the second highest-degree node is chosen, etc., until L links are cut, at which time the list is updated based on the new degrees. Similarly, function $g()$ in high-random cuts will choose edges from the chosen node randomly.

The pseudo-code of the procedure is shown in Algorithm 1 from step 2 to 3. We regard the undirected graph as a directed graph with one edge in each direction in our

connectivity matrix. Therefore, cutting one edge in the undirected G means cutting two edges in both directions at the same time. After each cut is performed, we add 1 to the count of cuts that have been made. We apply steps 2 - 3 before resorting degrees of the updated graph unless L numbers of edges from current node are cut or K edges have been cut or all edges of the currently selected node have been cut.

Algorithm 1 CDegree Cut

input : G: an $N \times N$ symmetric matrix represents a graph with N nodes
L: number of edges to cut before re-sorting;
K: total number of edges to cut;
schema: choose which strategy to apply (high, medium, low, random);
output: $\mathcal{C}$: set of edges to cut

```
1  countK ← 0;
   while countK < K do                                   // not enough K cuts
2      countL ← 0;
        nodeLeft ← f(G, schema);
       connList ← g(G, schema, nodeLeft);
       for i ← 1 to length(connList) do
3          nodeRight ← connList(i);
           G(nodeLeft, nodeRight) ← 0,
           G(nodeRight, nodeLeft) ← 0;
           C ← (nodeLeft, nodeRight);
           countK ← countK + 1;
           countL ← countL + 1;
            if countL == L then                          // L cuts?
4              break;
5          end
6          if countK == K then                           // K cuts?
7              break;
8          end
9      end
10 end
```

3.2 Discussions on Schema Selection

Here we discuss our schema selection in $f()$ and $g()$ in the following.

Intuitively, in order to reduce network connectivity quickly, the graph should be cut to be as sparse as possible. Cutting the links of the most popular nodes can be a good choice, but we need to be careful of how many links should be cut. If cutting the link of two popular nodes can split the graph into two subgraph, then cut it would be helpful. However, online social networks are usually highly connected [1] and it is difficult to split the graph. Then randomly cutting some of the edges or even all the edges of a popular node would be useful, as it will cost all other nodes more steps to transmit information among themselves. This method is called high-random cuts. On the other hand, if we want to isolate a few nodes quickly, i.e., with a relatively small number of cuts, from the graph, cutting the links of least popular nodes can be a good choice. This is because of their small number of edges. The method is called low-low cuts.

The intrinsic question is which method would be the most efficient. If we want to see how many nodes can be affected by the source rumor information in just a few steps (or delays, if we model the propagation of misinformation on each link as one unit time), high-random cuts would be a good choice because such hubs can be quickly dismantled. However, if a longer delay is allowed, low-low cuts would be more efficient. The reason is the following: with a large allowable delay, misinformation will most likely reach throughout the network except those isolated nodes. Therefore, the best solution is to isolate some nodes with low-low cuts.

CHAPTER IV

EXPERIMENTATION AND EVALUATION

4.1 Performance Evaluation

In this section, we evaluate different strategies in CDegree Cut with different L values. Our evaluations are based on a subgraph of Facebook snapshot, obtained from SNAP [12]. It is an undirected graph and consists of 4,039 nodes and 88,234 edges. Each node in the graph represents a user and each edge stands for a relationship between two nodes. The network diameter (maximum undirected shortest path length) of Facebook graph is 8, the average shortest-path length (APL) value is 3.7, and AIPL value is 0.3066. The highest node degree is 1,045, the top-10 highest and lowest degrees are shown on Tables 1 and 2. The maximum number of edges to cut is set to $K = 6,000$, about 7% of the number of edges on the graph. We use a baseline algorithm called Random Cuts, which simply picks edges randomly to cut.

Table 1. Top Ten Highest Degrees of Facebook Graph.

Ranking	1	2	3	4	5
Degree	1045	792	755	547	347
Ranking	6	7	8	9	10
Degree	294	291	254	245	235

Table 2. Top Ten Non-zero Lowest Degrees of Facebook Graph.

Ranking	1	2	3	4	5
Degree	1	1	1	1	1
Ranking	6	7	8	9	10
Degree	1	1	1	1	1

Before we present our results, we first introduce performance measures.

We mainly focused on two performance measures: Average Inverse of Shortest Path Length and Rumor Saturation Rate.

Average Inverse of Shortest Path Length (AIPL) In graph theory, the shortest path between any two nodes is an interesting and well-investigated problem. We choose breadth-first search (BFS) to compute the shortest path from any node to other nodes in the undirected graph, add all these path lengths for all nodes in the graph. Then the sum is divided by the number of nodes. The result is usually called average shortest-path length (APL), defined as the average number of steps along the shortest paths for all possible vertex pairs on the graph [21]. However, the APL measurement cannot handle partitioned graphs, on which some pairs of nodes are infinity distance from each other [20]. For this consideration, we use the inverse of the distance, since $1/\infty$ is simply 0. The average of all such inverse path lengths is called average inverse of shortest path length (AIPL). It can be proven that the range of AIPL is (0,1), where 0 means there is no edge in the graph and 1 means that the graph is fully connected. Therefore, the goal of efficient cut in our investigation is to lower AIPL.

Rumor Saturation Rate (RSR) AIPL gives us the mean value of the inverse lengths of the shortest paths between all possible pairs of vertices in G, but it still does not tell us how quickly (mis)information can spread. Therefore, we look at a new performance measure called rumor saturation rate (RSR), which can be obtained through experimental settings. In order to find out RSR, S number of nodes are chosen as the sources of the same misinformation (these are called "rumor sources"). In each unit time, rumor is spread from all those nodes carrying it so far to all their neighbors. Such a procedure continues until D unit times. RSR is defined as the number of nodes who have seen the rumor D unit times later divided by the total number of nodes N. Because of the random selections, RSR needs to be measured through repeated Monte Carlo random experiments and it is a function of (S, D).

4.2 Experiments on Synthetic Data

Before testing on real data, we generate an undirected synthetic graph first: it consists of 4039 nodes and 8823 edges, which is exactly the same number of nodes as Facebook graph, but 10% edges of Facebook graph. All the edges in this graph is generated randomly. We generate it because it is less complex than real Facebook graph and easy to start our analytic. The graph's average shortest path length is 5.6 and the highest degree is 13. In order to have a good understanding of the graph, the top ten highest and lowest non-zero degrees are listed in Table 3 and 4 separately. Before starting our experiment, there is one question to think about: how many total edges should be cut from the graph? We consider all cuts should be up to 10% of the number of the original links. Otherwise, the experiment will be meaningless as the graph becomes very sparse and dysfunctional. So for this synthetic graph, total 600 cuts will be done later.

Table 3. Top Ten Highest Degrees of Synthetic Graph.

Ranking	1	2	3	4	5
Degree	13	13	12	12	12
Index	1753	2382	3	742	1754
Ranking	6	7	8	9	10
Degree	12	11	11	11	11
Index	3080	2789	2978	3353	4016

Table 4. Top Ten Non-zero Lowest Degrees of Synthetic Graph.

Ranking	1	2	3	4	5
Degree	1	1	1	1	1
Index	22	25	43	129	167
Ranking	6	7	8	9	10
Degree	1	1	1	1	1
Index	183	201	215	246	294

During designing our algorithm, there is one question we need to confirm the answer: when pick up the highest or lowest degree node, is it better to cut all of its connected edges or just cut few of its links before re-sorting? In order to address this issue, we add a variable L to our algorithm, which means cutting L links of selected highest or lowest degree node every time. Let the number of total cuts K be 600, try CDegree Cut in different L's (L = 1-13), we compare the AIPL results in each

method (high-random cuts and low-low cuts) with different L's. It turns out there is no big difference between the curves of different L.

Next, we need to compare the RSR results in our algorithm with different L's. We choose some groups of (S, D) values for test: (2, 2), (4, 2), (2, 7), (3, 8), also compute the 95% confidence interval. We find out that, compared to the 95% confidence interval, the difference between L's curves is very small, which means we can almost ignore the difference. In all, L can be ignored in our later test, or we can choose L be equal with or greater than the highest degree, that is, select the highest degree node each time, then cut all its links.

Next, we leave out L and use our algorithm to test AIPL and RSR. Figure 2 shows the AIPL values with different methods. Figure 3 show RSR results on several different methods. When D is a small value, high-random cuts should perform better than low-low cuts, since the rumor information can only be transmitted to a small range with a small D, but with low-low cuts, we cut the link between the lowest degree nodes, these nodes are in the border of the graph that cannot be affected by rumor information. Thus it will not influence RSR value a lot; When D is a big value, low-low cuts should perform better than high-random cuts when the number of cuts is less than half of the original edges. As with a bigger D value, rumor information can be transmitted through the most part of the graph, with low-low cuts, most lowest degree nodes can be isolated, thus will reduce RSR a lot, but with high-random cuts, nodes are not easy to be isolated, thus it can influence RSR a little.

Finally, we compare CDegree Cut with random cuts in the performance of AIPL and RSR. In random cuts, we randomly choose one edge to cut each time until reaching K = 600 cuts. In order to reducing errors from random number, we simulate

it 20 times to get a lower standard deviation. From Figure 3, it is easy to see no matter how many edges to be cut, CDegree Cut always performs better than random cuts.

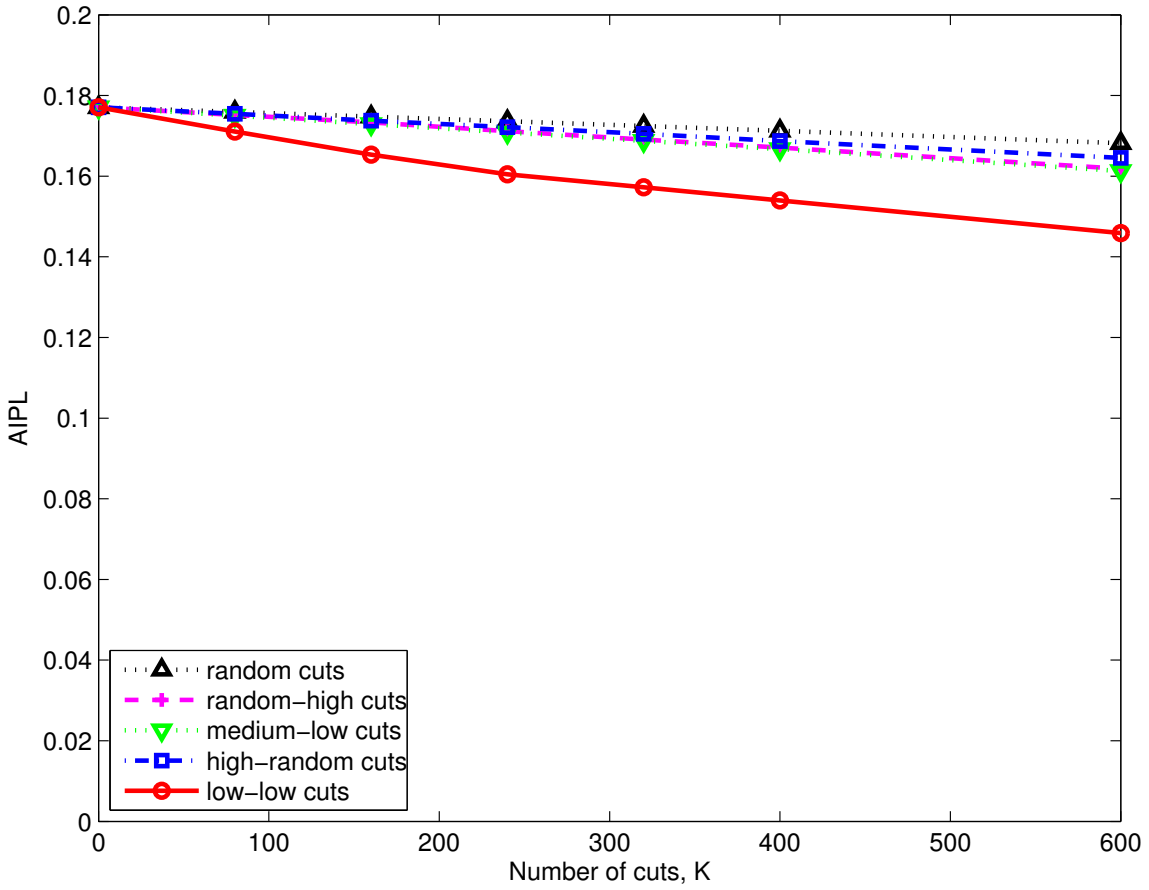

Figure 2. AIPL of Different Cut Methods on Synthetic Graph. Low-low Cuts Seem to Be The Best Method among All Shown.

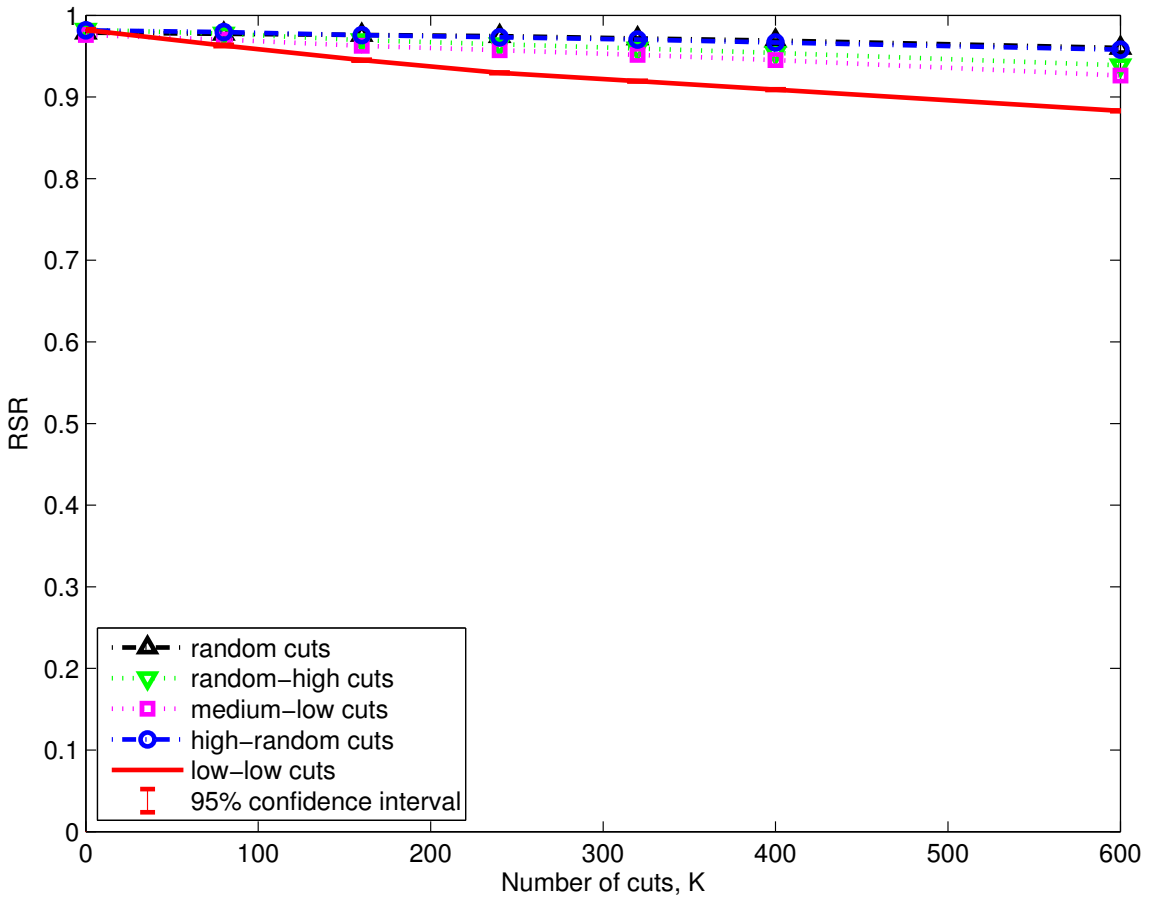

Figure 3. RSR Comparisons for (S, D) = (2, 7) on Synthetic Graph. S Represents The Number of Rumor Sources and D is The Delay Before RSR is Measured. Low-low Cuts Method is The Best in This Graph.

4.3 Experiments on Facebook Data

We first investigated the effects of L. With $K = 6,000$, we tested high-high cuts for different L values and presented the results in Figure 4. As can be seen from Figure 4, AIPL decreases as K increases, first rather quickly and then the rate of AIPL decrease slowing down. Comparing the results of different L's, we observe that $L = 1$ and 20 have rather similar performance. $L = 500$ and 1,045are similarly better

than $L = 1$ and 20. Therefore, a large L actually will result into a better performance, with lower overhead (of re-sorting the new degrees). Similar comparison results have been observed for other schemes for both AIPL and RSR. Hence, we conclude that L can be chosen to a relatively large value such that the overhead of re-sorting is low. In the following experiments, we chose $L = 1,045$. In this case, the methods in each strategy should have similar results, for example, high-random cuts, high-medium cuts, high-high cuts and high-low cuts will finally cut the same edges, the only difference is the order of edges to be cut and the final round of cut.

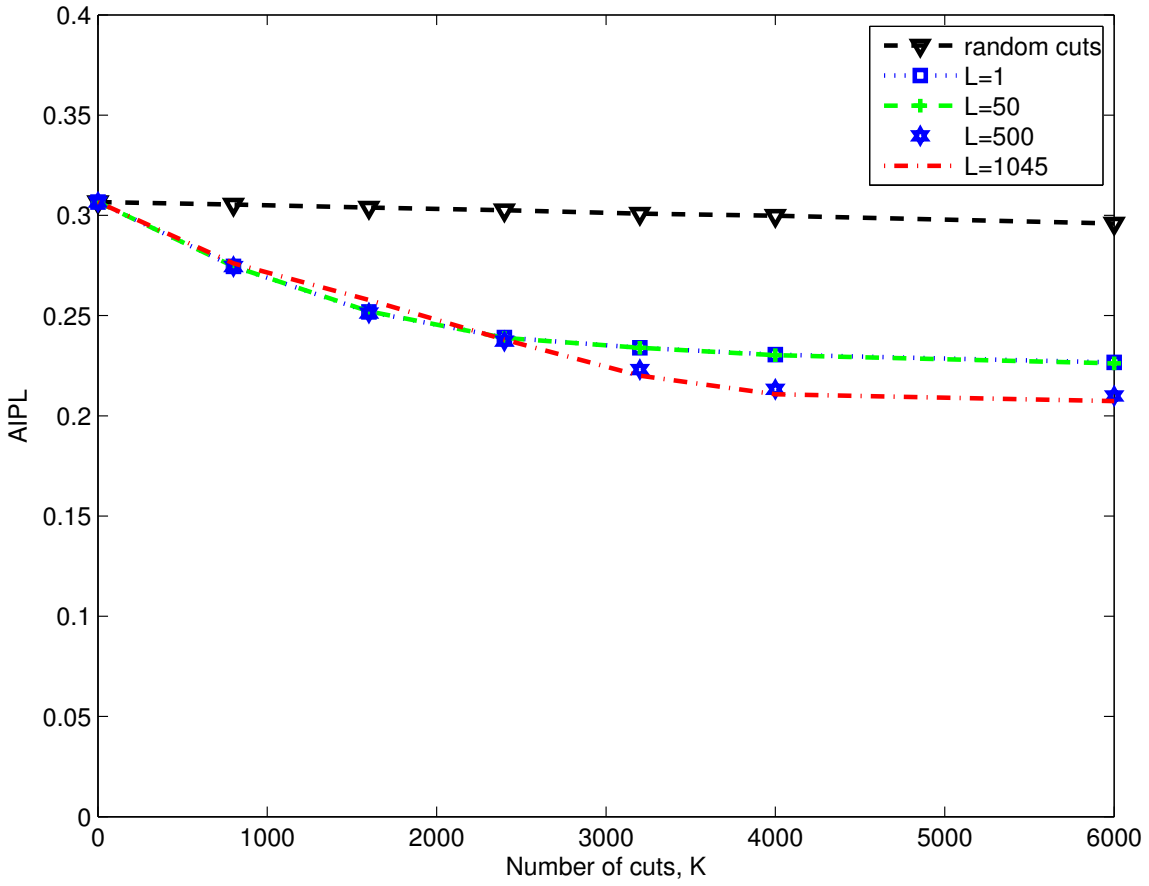

Figure 4. AIPL with High-high Cuts in Different L Values on Facebook. Performance is Better with Larger L's, Which Also Have Lower Overhead. Random Cuts Results are There for Comparison Purpose.

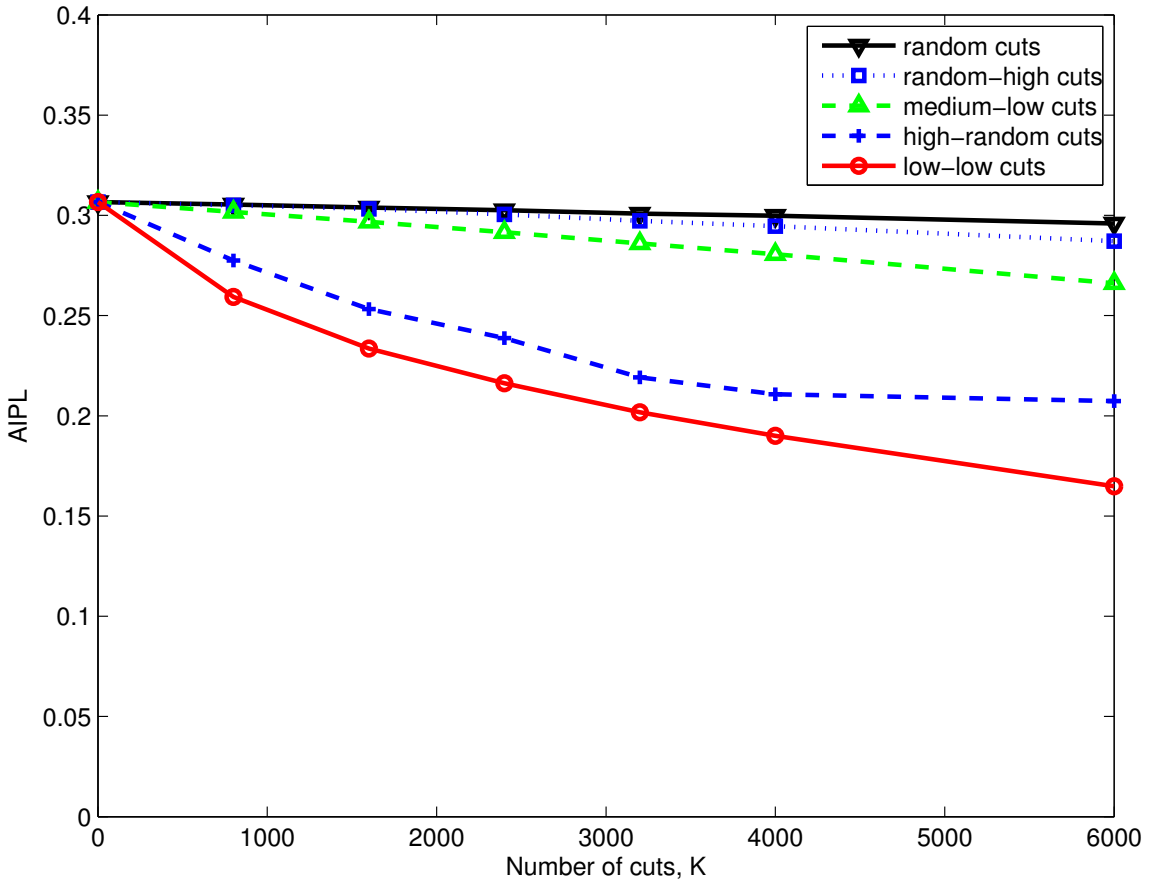

Figure 5. AIPL Comparisons of Different Cut Methods on Facebook. Low-low Cuts Seem to be The Best Method among All Shown.

In Figure 5, we showed the AIPL values of different cut methods for different K values. The two random methods, random cuts and random-high cuts, ended with similar results of very slow AIPL decrease. Medium-low cuts and high-random cuts are better, but the best performance belongs to low-low cuts. Therefore, if we were to conclude based solely on AIPL comparisons, low-low cuts would be a clear winner. Interestingly, as to be demonstrated with RSR results, such is not always the case.

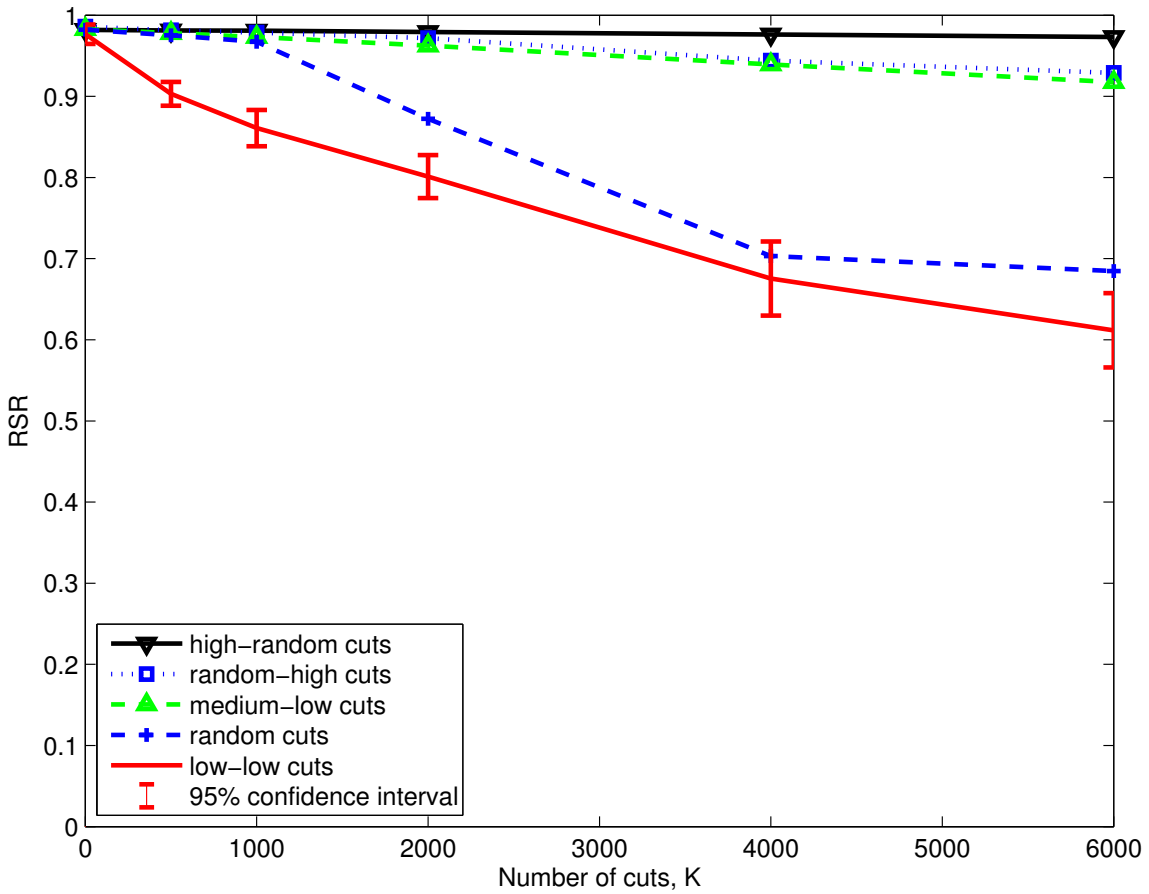

Figure 6. RSR Comparisons for (S, D) = (2, 5) on Facebook. S represents The Number of Rumor Sources and D is The Delay Before RSR is Measured. Low-low Cuts Method is The Best in This Graph.

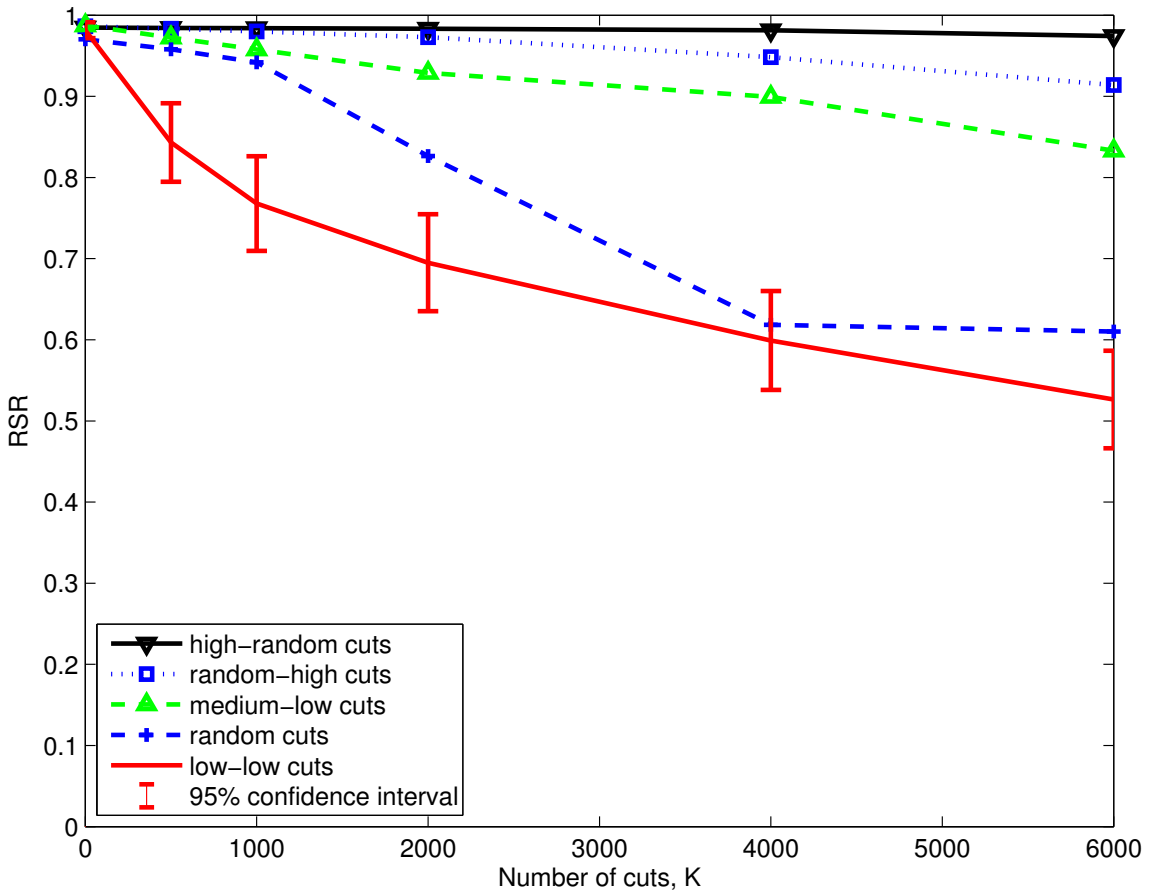

Figure 7. RSR Comparisons for (S, D) = (1, 6) on Facebook. S Represents The Number of Rumor Sources and D is The Delay Before RSR is Measured. Low-low Cuts Method is The Best in This Graph.

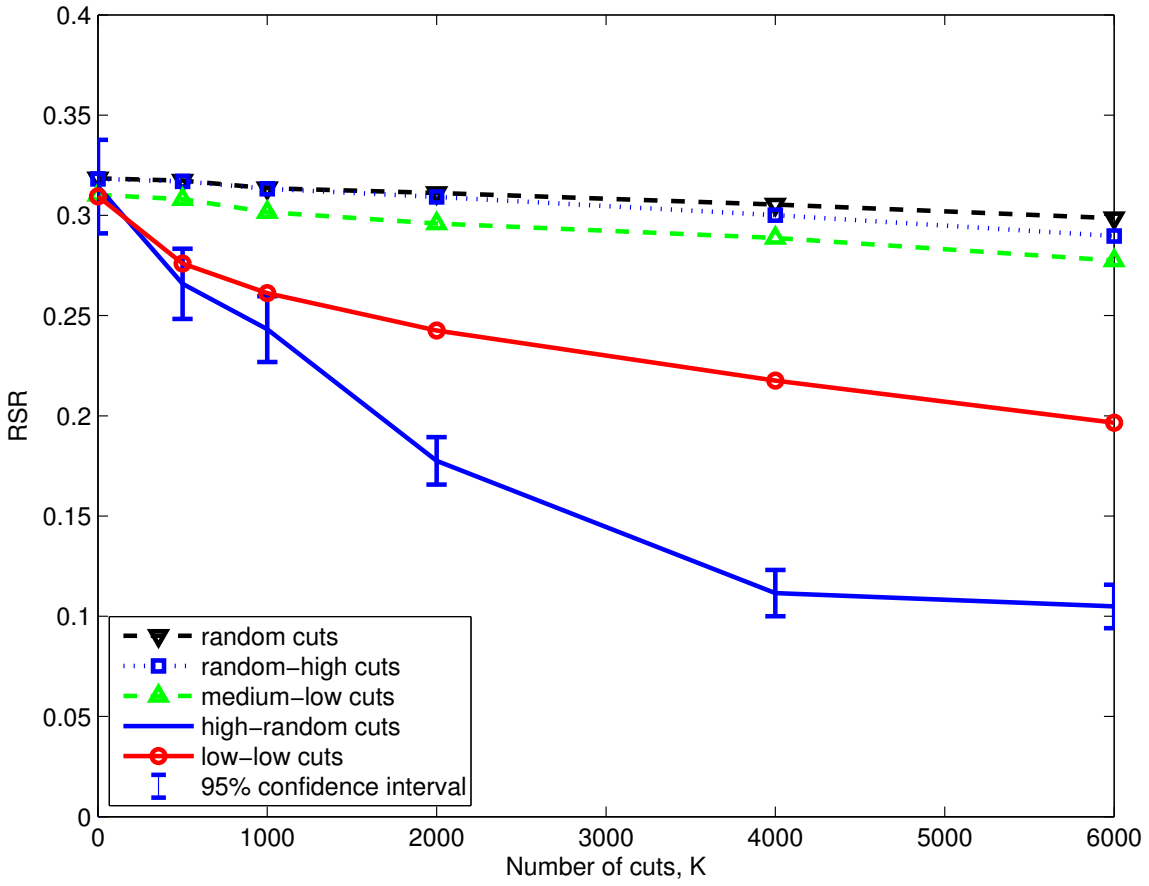

Figure 8. RSR Comparisons for (S, D) = (2, 2) on Facebook. S Represents The Number of Rumor Sources and D is the Delay Before RSR is Measured. High-random Cuts Method is The Best in This Graph.

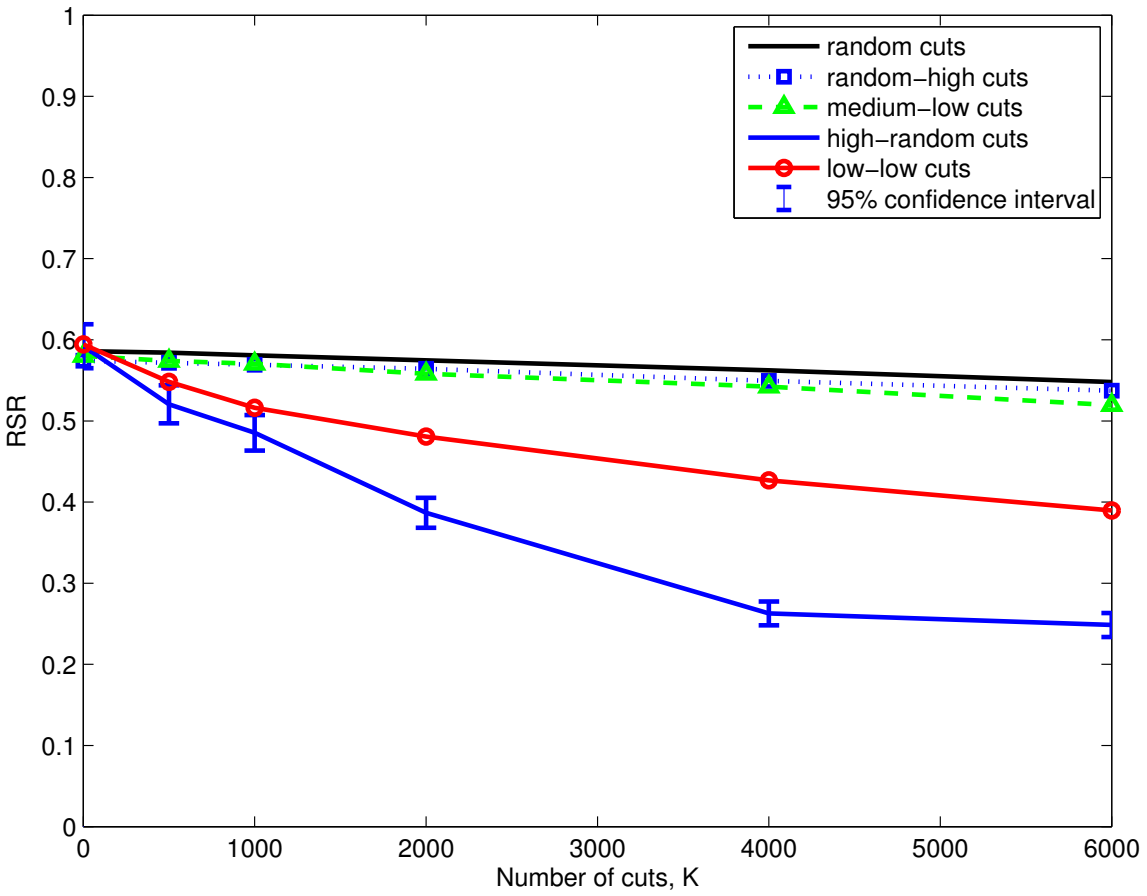

Figure 9. RSR Comparisons for (S, D) = (5, 2) on Facebook. S Represents The Number of Rumor Sources and D is The Delay Before RSR is Measured. High-random Cuts Method is the Best in This Graph.

Figures 6, 7, 8, and 9 demonstrated different behaviors of different cut schemes under various S and D values. Note that S represents the number of rumor sources and D is the delay before RSR is measured. We choose these D values based on the APL value of the graph. In Figures 6 and 7, i.e., when D is relatively large, low-low cuts remain the best among all schemes. However, as shown in Figures 8 and 9, i.e., when D is rather small, high-random cuts outperforms other schemes. An intuitive explanation is in order: when D approaches APL, rumors will reach a majority of the

nodes in the graph and it would be better to cut away some nodes so that they are isolated; However, when D is smaller than APL, it would be better to cut of some of the major connectors of the graph (those high degree nodes) in order to lower the rumor spread rate.

Comparing the results from Figures 5, 6, 7, 8, and 9, we can see that AIPL measure has its limits in measuring network connectivity. In fact, sometimes, AIPL would provide misleading indication of network connectivity. While RSR requires repeated Monte Carlo experiments, it provides interesting insights into how fast information could spread.

As we discussed, low-high, low-medium, low-low, low-random schemes work similarly for large L because of the small number of links on such nodes. One would naturally wonder which of the high-* schemes performs best. Figure 10 shows the comparison of these schemes. From Figure 10, we can conclude that these schemes are similar in performance. Considering computational overhead, high-random scheme works fine as no sorting is needed.

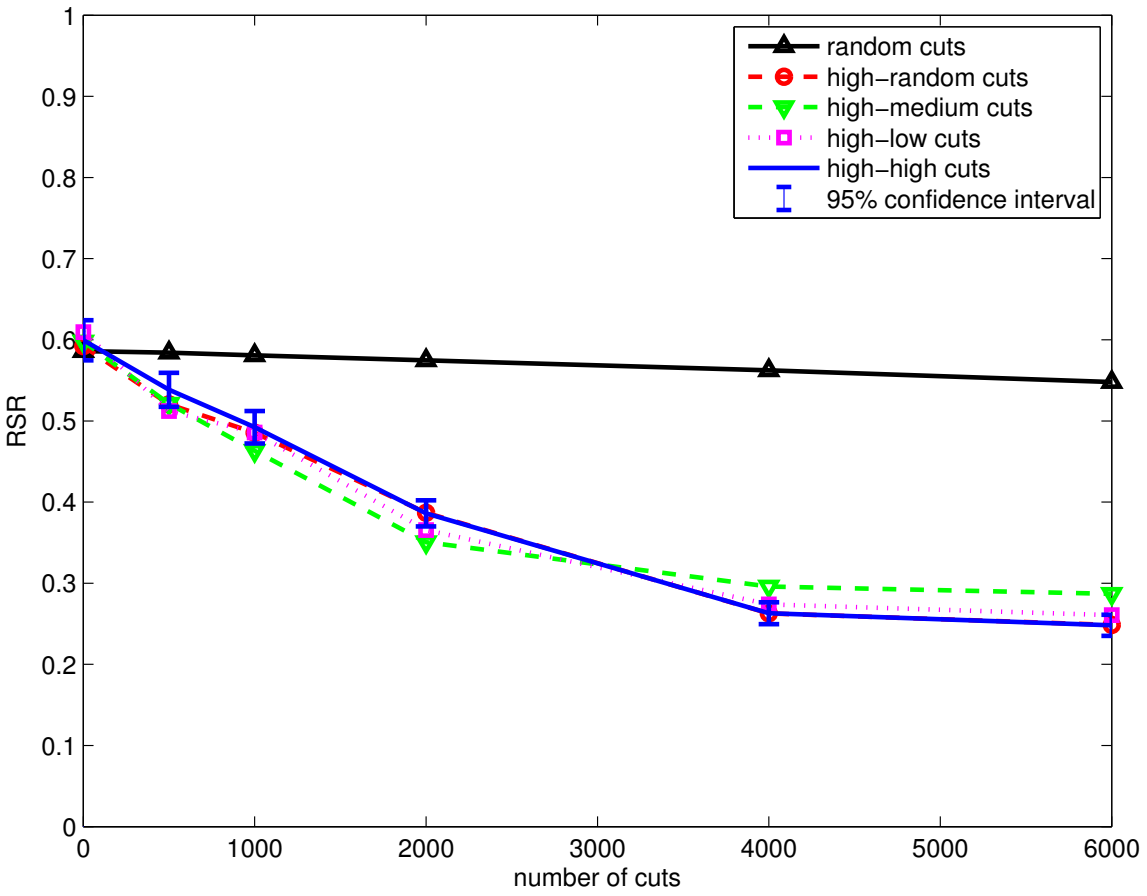

Figure 10. RSR for (S, D) = (5, 2), Different High-* Schemes on Facebook. The RSR Results only Differ Slightly among Different High-* Schemes.

4.4 Rumor Propagation Models on Facebook Data

Probability theory deals with conditions in which the outcomes occur randomly. It is is one of the most important theories in mathematics as it can tell us how likely an event will happen in our real lives. Firstly, we need to clarify some terms in probability language. The conditions here are called experiments. The set of all possible outcomes or results from an experiment is called a sample space, which is represented by a symbol Ω [14]. A subset of a sample space is called an event [14].

For example, when we flip a coin, there will be two possibly outcomes: one is a head, the other one is a tail. Of course, the two outcomes cannot happen at the same time, We use a number 0 to represent head and number 1 as tail , so $\Omega = \{0,1\}$. If the coin stops at tail side, this event can be denoted by $A = \{1\}$.

A probability can be denoted by P [14], which is a ratio between zero and one. If an event has probability of 1, this event is certain to happen; if its probability is equal to 0, the event will never happen. The probability of all the outcomes Ω is $P(\Omega) = 1$. If an event will happen with a probability of $P(A)$, then the probability that event will not happen is $1 - P(A)$. In the above coin flip example, either head or tail can happen with an evenly probability of 50%. So the probability of the event that the coin stops at head side can be: $P(A) = 0.5$.

Based on probability theory, we use a probability model for rumor propagation. The model is designed as follows: let the probability is 25%. In the first step, each source node, which contains rumor, flips a biased coin to decide whether it will forward the rumor which it received so far toward all its neighbors. We realize this by applying a simple rand() function in our algorithm: each source node makes a random selection based on this function. If a node gets a value which is less than 0.25, it will be active and forward the rumor; if the value is greater than 0.25, it will be inactive and not forward the rumor. So here each node has a 25% chance of forwarding the rumor. Suppose we have 40 nodes with the rumors, $40 \times 0.25 = 10$. So on an average, 10 of them will forward rumors to their neighbors. In the second step, for each selected source nodes each of its links flips a biased coin to decide whether the link will forward the rumor. Similar to the first step, we apply a rand() function to each link, a return value will decide whether it will be active or not. Thus, each link will have 25% chance

of forwarding the rumor. The probability of forwarding a rumor by two sequential nodes of the same path would be $0.25 \times 0.25 = 0.0625$. We apply this model to different cut schemes to check their performance.

We choose a lower D value ($D = 2$) to run the simulations and finally get the results that are shown in Figure 11 and 12. It is easy to see: RSR values are much lower compared to results in Section 4.3 and High-random cuts also perform best among all the methods. This result is closer to reality, since not all the users on Facebook will forward the rumors whenever they receive them, especially for those inactive users; and not every user will forward the rumors to all of his friends, especially to whom they rarely contact with.

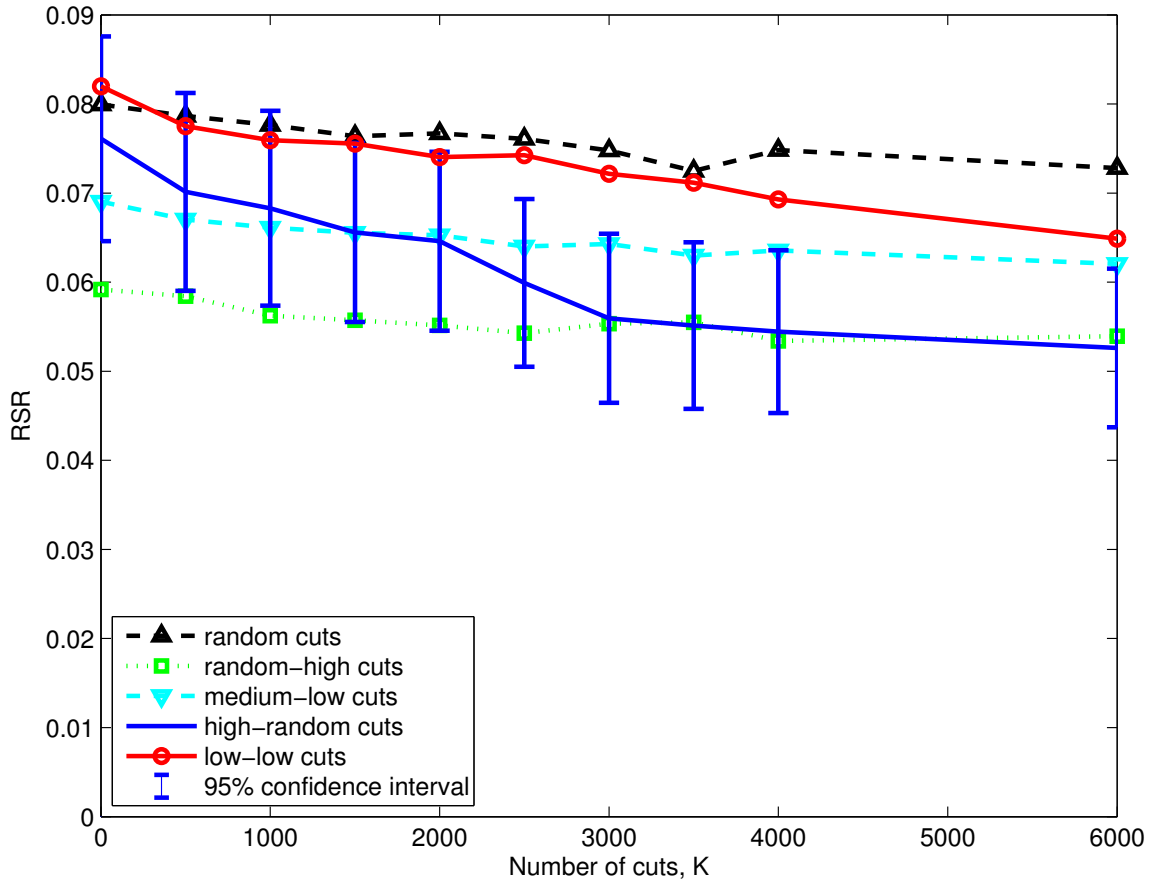

Figure 11. RSR for (S, D) = (10, 2) with The Probability Model. S Represents The Number of Rumor Sources and D is The Delay Before RSR is Measured. High-random Cuts Method is The Best in This Graph.

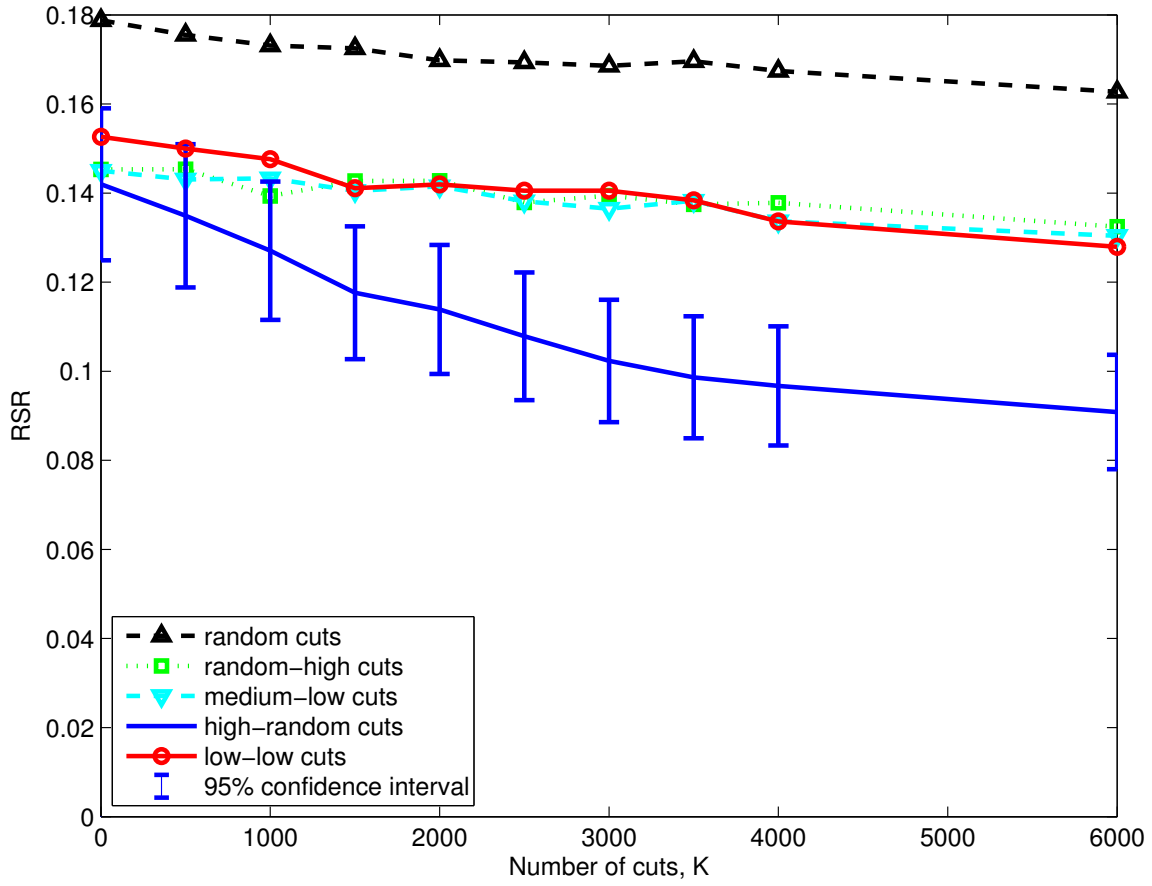

Figure 12. RSR for (S, D) = (20, 2) with The Probability Model. S Represents The Number of Rumor Sources and D is The Delay Before RSR is Measured. High-random Cuts Method is The Best in This Graph.

CHAPTER V

CONCLUSIONS AND FUTURE WORKS

In this paper, we have investigated the problem of efficient link cuts in large online social networks to lower the speed of misinformation spread, which is exactly opposite to the well-known link prediction problem. We have designed an algorithm called CDegree Cut to choose such links. In this algorithm, 16 strategies can be chosen based on the selection of degree to cut and a parameter L. CDegree Cut has gone through extensive experimental evaluations through synthetic data and real Facebook data with two different performance measures: AIPL and RSR. We have found that L cannot be a factor as curves of different L's in the same strategy do not differ a lot. So let L be the maximum degree of all nodes, the 16 strategies have been collapsed to 4: high-*, medium-*, random-*, low-*. Then we have compared the four strategies, the results from the two experiments coincide with each other, that is: when the delay is larger than APL, low-low cuts should be used; if the delay is shorter than APL, high-random cuts should be chosen. Also, we have found that our all strategies perform better than random cuts. Another interesting observation is the ambiguity of AIPL results. Instead, a more computation-intensive RSR measurement can help to provide better insights in the comparison of different schemes. In addition, we have applied rumor propagation models to our algorithm since that will be closer to users' behavior in online social networks. We have applied the updated algorithm to Facebook data and discovered that the results math the previous ones.

REFERENCES

[1] Lars Backstrom, Paolo Boldi, Marco Rosa, Johan Ugander, and Sebastiano Vigna. Four degrees of separation. In *Proceedings of the 4th Annual ACM Web Science Conference*, WebSci '12, pages 33–42, New York, NY, USA, 2012. ACM.

[2] Lars Backstrom and Jure Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 635–644. ACM, 2011.

[3] Marc Barthélémy and Luis A Nunes Amaral. Small-world networks: Evidence for a crossover picture. *Physical Review Letters*, 82(15):3180, 1999.

[4] Charles L Cartledge and Michael L Nelson. Connectivity damage to a graph by the removal of an edge or a vertex. *arXiv preprint arXiv:1103.3075*, 2011.

[5] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. Rumor spreading in social networks. In *Automata, Languages and Programming*, pages 375–386. Springer, 2009.

[6] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.

[7] Reinhard Diestel. *Graph Theory*. Number 173 in Graduate Texts in Mathematics. Springer, 1997.

[8] A Behrouz Forouzan. *Data Communications & Networking (sie)*. Tata McGraw-Hill Education, 2006.

[9] Zan Huang. Link prediction based on graph topology: The predictive value of generalized clustering coefficient. *Available at SSRN 1634014*, 2010.

[10] Norman P Hummon and Patrick Dereian. Connectivity in a citation network: The development of dna theory. *Social Networks*, 11(1):39–63, 1989.

[11] David Joyner, Minh Van Nguyen, and Nathann Cohen. Algorithmic graph theory. *Google Code*, 2010.

[12] Jure Leskovec and Julian J Mcauley. Learning to discover social circles in ego networks. In *Advances in neural information processing systems*, pages 539–547, 2012.

[13] Mark EJ Newman, Stephanie Forrest, and Justin Balthrop. Email networks and the spread of computer viruses. *Physical Review E*, 66(3):035101, 2002.

[14] John Rice. *Mathematical statistics and data analysis*. Cengage Learning, 2006.

[15] Jagan Sankaranarayanan, Hanan Samet, Benjamin E Teitler, Michael D Lieberman, and Jon Sperling. Twitterstand: news in tweets. In *Proceedings of the 17th acm sigspatial international conference on advances in geographic information systems*, pages 42–51. ACM, 2009.

[16] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[17] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM, 2002.

[18] Jeffrey Travers, Stanley Milgram, Jeffrey Travers, and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, 32:425–443, 1969.

[19] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world' networks. *nature*, 393(6684):440–442, 1998.

[20] Jameson Watts and Kenneth W Koput. Supple networks: Preferential attachment by diversity in nascent social graphs. *Available at SSRN 2420764*, 2014.

[21] Chia-Chen Yen, Mi-Yen Yeh, and Ming-Syan Chen. An efficient approach to updating closeness centrality and average path length in dynamic networks. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 867–876. IEEE, 2013.