Empirical Comparison of Information Spreading Algorithms in The Presence of 1-Whiskers

Kazem Jahanbakhsh
Computer Science Department
University of Victoria
Victoria, Canada
Email: jahan@cs.uvic.ca

Valerie King
Computer Science Department
University of Victoria
Victoria, Canada
Email: val@cs.uvic.ca

Gholamali C. Shoja
Computer Science Department
University of Victoria
Victoria, Canada
Email: gshoja@cs.uvic.ca

***Abstract*—Several information spreading algorithms for social networks have been proposed by researchers during the past few years. However, the main challenge is to find out the best algorithm with the lowest running time for a given communication graph. In this paper, we address the problem of information spreading in the context of social networks. We compare the running times of three well known information spreading algorithms in the field by using real data collected from the Facebook website. We prove the importance of 1-whisker communities for the speed of information spreading algorithms in social networks by comparing the performance of different spreading algorithms with and without 1-whiskers. Our results are important since they highlight the effect of 1-whiskers as the main communication bottlenecks for information spreading and the need for development of more efficient algorithms in this field.**

***Index Terms*—Information Spreading; Social Networks; Empirical Analysis; Conductance; 1-Whiskers;**

I. Introduction

Information spreading algorithms have several applications especially in social networking area such as advertising in cellphone communication network in which graph's vertices are people who carry cellphones. In this graph, there is an edge from node u to v if u has listed v in its contact list. For such a graph, designing an algorithm that efficiently broadcasts an advertisement message from a starting node to all other nodes in the network is very useful. Researchers have proposed several algorithms for information spreading. However, the main difficulty is to choose the fastest algorithm for a given communication graph. This decision strongly depends on the underlying structure of the communication network.

In this paper, we want to identify the best strategy for fast information spreading in social networks. To answer this question we study three mechanisms for information spreading in an undirected social graph. The first one is the well-known *random push-pull* [1], the second one is a variation of an algorithm proposed by Censor et al. [2], and the third one is an algorithm proposed by Doerr et al. [3]. For our empirical analysis, we assume a synchronized communication model where initially each node has a unique message. The goal of an information spreading algorithm is to spread all nodes' messages to all other nodes. In the random push-pull algorithm, in each round every node chooses one of its neighbors randomly to exchange its messages with. The Censor algorithm is a hybrid of the random push-pull and a deterministic approach. Finally, the Doerr algorithm is a simple variation of the random push-pull. For the rest of the paper, we refer to the algorithm proposed by Censor et al. as the *Censor* and the one proposed by Doerr et al. as *Doerr*.

In this paper, we compare the running times of the three mentioned algorithms on a social graph collected from the Facebook website. Interestingly, our empirical analysis shows that the Censor algorithm outperforms the other two mechanisms. We justify our empirical results by identifying a large number of *1-whiskers* in the Facebook graph as communities that are weakly connected to the rest of the graph. We prove that these 1-whiskers act as communication bottlenecks for information spreading and describe how the Censor algorithm spreads the information faster than both random push-pull and Doerr mechanisms.

Doerr et al. have shown that the random push-pull and Doerr algorithms spread information in logarithmic and sub-logarithmic number of rounds in graphs generated by preferential attachment model, respectively [3]. Censor et al. have also shown that if a graph has low conductance but large "weak conductance", their algorithm outperforms the random push-pull strategy. However, there has not been any performance comparison between Censor and Doerr algorithms in the context of social networks. In this paper, for the first time we compare the running times of Censor and Doerr algorithms. Interestingly, we show that the Censor algorithm performs even better than Doerr algorithm. We justify our performance results by showing that the Facebook graph has a core-periphery structure which cannot be explained by random or preferential attachment models.

The remainder of the paper is organized as follows: Section II reviews the recent work in the field. Section III defines the problem to be tackled. Section IV describes our empirical results for the performance of information spreading algorithms and justifies the performance results by identifying 1-whiskers as the main communication bottlenecks in the Facebook social graph. Finally, Section V concludes the paper.

II. Background And Related Work

Let us represent the underlying structure of our social network with graph $G = (V, E)$. For a given cut $(S, V \setminus S)$, we define the *conductance* φ of the set S as follows:

$$\varphi(S, V) = \frac{\sum_{i \in S, j \in V \setminus S} P_{ij}}{|S|}, \quad (1)$$

where P is the stochastic matrix associated with G [2]. The conductance of the graph G is then defined to be:

$$\Phi(G) = min_{S \subseteq V, |S| \leq \frac{n}{2}} \varphi(S, V) \quad (2)$$

Mosk-Ayoma and Shah [1] have shown that for a $\delta \in [0, 1]$, the random push-pull algorithm terminates in $O(\frac{\log n + \log \delta^{-1}}{\Phi(G)})$, with probability of at least $1 - \delta^{-1}$, where $|V| = n$ and $\Phi(G)$ is the conductance of G as defined in Equation 2. Conductance of a graph (i.e. $\Phi(G) \in [0, 1]$) determines how well-connected a graph is. While well-connected graphs (e.g. cliques) have large conductance values, the graphs that have many communication bottlenecks (e.g. paths) have low conductance. Here, the main obstacle to speeding up the information spreading is the size of graph conductance. If a given graph has a low conductance (e.g. $\Phi(G) = O(\frac{1}{n})$), the number of rounds needed to spread the messages to all other nodes is a polynomial function of n.

Censor et al. have identified a class of graphs that have low conductance but large weak conductance [2]. While conductance measures the connectivity of the whole graph G, weak conductance of a graph G (i.e. $\Phi_c(G)$) measures the *best* connectivity among subsets that include each node. The size of subsets depends on a c value. We refer readers to [2] for the formal definition of weak conductance. One example for a graph with low conductance but large weak conductance is a path of c cliques where each of them has $\frac{n}{c}$ nodes. While the path of c cliques has low conductance (i.e. $\Phi(G) = O(\frac{1}{n})$), each clique is a very well-connected component (i.e. $\Phi_c(G) = O(1)$).

To speed up the running time of spreading algorithms for graphs with many communication bottlenecks one needs to identify them. Censor et al. have used a hybrid approach to identify the bottlenecks. In their algorithm, every node has a bottleneck list which initially contains all of its neighbors. In even rounds each node chooses a random neighbor to contact, but in odd rounds each node chooses the next node to be contacted from the top of its bottleneck list. In the Censor algorithm, each node v will permanently keep one of its neighbors such as u in its bottleneck list if v contacts u at round r and receives u's message (i.e. $m(u)$) for the first time directly from u. If v receives $m(u)$ in any other ways, v removes u from its bottleneck list. If v receives $m(u)$ for the first time from node u itself when v contacts u directly, this implies that node v's community is probably weakly connected to node u's community; otherwise, it could have received $m(u)$ earlier from one of its other neighbors. Censor et al. have shown that their algorithm improves the running time from polynomial to poly-logarithmic time for graphs with low conductance but large weak conductance [2].

Very recently Doerr et al. have shown that the random push-pull spreads a message starting from a node to all other nodes in the graph within $\Theta(\log n)$ rounds with a high probability in graphs with power-law node degree [3]. They have also proposed a new algorithm (i.e. Doerr) that works differently from the random push-pull in that in each round each node contacts one of its neighbors uniformly at random except the one that contacted in the previous round [3]. They have shown that their new algorithm just requires $\Theta(\frac{\log n}{\log \log n})$ rounds to spread a message to all nodes in the networks with power-law node degree. However, our empirical analysis shows that the Censor algorithm beats the other two spreading algorithms in the field.

III. Problem Definition

Information Spreading: For a given communication graph $G = (V, E)$, suppose every node $u \in V$ has a unique message $m(u)$ that is identified by its node *id*. In an information spreading algorithm, every node u chooses one of its neighbors (e.g. $v \in N(u)$) in each round r according to the communication model and exchanges its current messages with the selected neighbor. The algorithm terminates when all nodes receive all other nodes' messages (i.e. by the time when every node has n unique messages in its queue where $|V| = n$). We list the computation and communication constraints that every node has to obey as follows:

- The only information that is known to each node is the set of its neighbors and the number of nodes in the graph (i.e. $|V| = n$). In other words, the structure of the graph is unknown to all nodes.
- The communication is synchronous where in each round r every node u contacts one of its neighbors to exchange its messages with (i.e. push-pull model).
- In each round, every node contacts *one* of its neighbors to exchange its messages with that neighbor. However, in each round a node can be contacted by multiple nodes.

We assume that in each round, the required computation for node u to exchange its message with other nodes (i.e. including the one that u contacts and the neighbors that contact u by themselves) as well as other required internal computations for u take constant number of steps. Therefore, the cost of the spreading algorithm is the number of required rounds until all nodes receive all messages of the network.

IV. Information Spreading And Community Structure

In this section, first we compare the running times of random push-pull, Censor, and Doerr algorithms by using our real data from the Facebook website. Then, we justify the performance results of the three information spreading algorithms by analyzing the underlying structure of the Facebook graph.

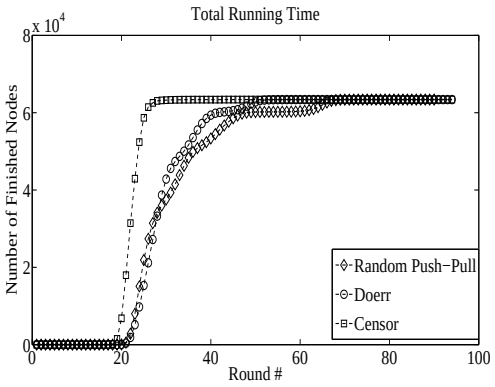

Fig. 1. Running times of the random push-pull, Doerr, and Censor algorithms

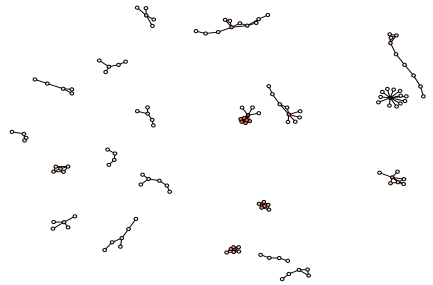

Fig. 2. A sample of detected 1-whiskers in Facebook graph

A. Real Data Description

In this paper, we use the social network (i.e. $G = (V, E)$) of *New Orlean* regional network in Facebook collected by Viswanath et al. [5]. Our total dataset has 63731 nodes and 817090 edges. Since the original graph is not connected, we focus on the largest component of the original dataset that has $|V| = 63392$ nodes and $|E| = 816886$ edges for the rest of our analysis.

B. Empirical Analysis of Information Spreading Algorithms

For the first part of our analysis, we run the random push-pull, Censor, and Doerr algorithms by using the Facebook data to analyze their running times. We assume that each node initially has one message identified by its node id. We run each spreading algorithms twenty times and compute the average number of nodes that have received all the messages for each round number. The results for the algorithms' progress versus round number are shown in Figure 1. As we can see, the running time of Censor is interestingly better than both random push-pull and Doerr algorithms.

We can make several observations from Figure 1. First, we have found that 95% of nodes have terminated in 25, 43, and 57 number of rounds in Censor, Doerr, and random push-pull algorithms respectively. This shows that in Censor algorithm 95% of nodes receive all messages at least twice faster than the random push-pull. Moreover, we can see that after a few number of rounds, the number of nodes that terminate start growing exponentially as a function of round number. This can be justified if we assume that the underlying communication graph has an expanding structure. Our graph analysis result shows that our social network has a core-periphery structure where the core of the network has an expanding structure.

C. Finding 1-Whiskers as Main Communication Bottlenecks

As the speed of information spreading in any graph is directly connected to its underlying structure, identifying the community structure of the given graph allows us to better understand the running times of information spreading algorithms. Communities can be considered as subsets of nodes that are internally well-connected; however, they are loosely connected to the rest of network. This informal definition for communities is related to the conductance definition where a set S is considered as a good community if we can cut it easily from the rest of the graph G (i.e. S has a low conductance). Finding communities play an important role in information spreading as nodes from the same community receive each others messages faster than messages of nodes from different communities.

For identifying communities inside the Facebook graph, we employ the Tarjan algorithm to find all "bridges" of the graph [6]. In a graph G if there is a path from a vertex u to a vertex v but every path from u to v contains edge e, then we say e is a *bridge* of G. Identifying bridges is important as we show that they play an essential role on the running times of information spreading algorithms. Let us define *1-whiskers* to be maximal subgraphs that can be detached from the rest of the network by removing a *single edge* [4]. These maximal subgraphs are communities that have low conductance. By removing all identified bridges, we can find the largest 2-edge connected component (i.e. the *core* of G). Having the core of the graph and all bridges, we can identify the 1-whiskers by only removing those bridges that have one node in the core. Identifying 1-whiskers is crucial because they are small components that are weakly connected to the core. Figure 2 shows a subset of identified 1-whiskers in Facebook friendship graph. As we can see, 1-whiskers have a variety of structures although they are connected to the rest of the network by a single edge.

We have shown the number of 1-whiskers as a function of their sizes in Figure 3. We have found that around 7% of 1-whiskers have more than 2 nodes. In other words, we have identified 7736 number of 1-whiskers where 590 of them have at least two nodes. Among these 590 1-whiskers we have found a variety of structures. While some of them are paths, there are some 1-whiskers that have a rich internal structure as it is shown in Figure 2. Next, we remove all 1-whiskers from the Facebook graph to show the importance of 1-whiskers on the performance of the spreading algorithms. The running times of the random push-pull, Doerr, and Censor algorithms without 1-whiskers are shown in Figure 4. Comparing Figures

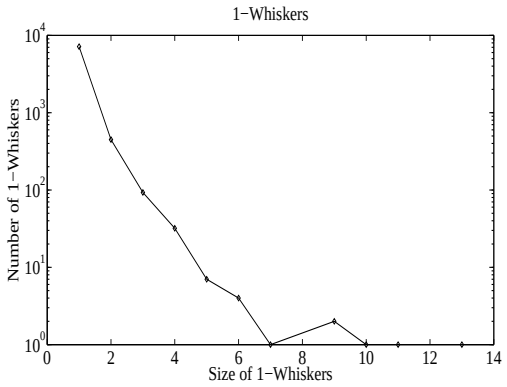

Fig. 3. The number of 1-whiskers as a function of their sizes

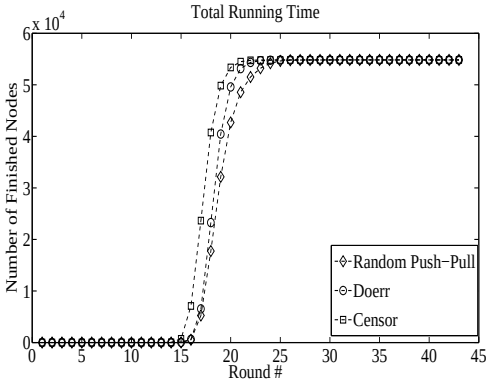

Fig. 4. Running times of random push-pull, Doerr, and Censor algorithms without 1-whiskers

1 and 4, we can see that removing 1-whiskers decreases the running times of random push-pull and Doerr algorithms by at least a factor of two. Our results show that there are a significant number of bridges in social network that act as the main communication bottlenecks for information spreading.

D. Discussions

Leskovec et al. have studied the statistical properties of communities in undirected graphs collected from several large social networks [4]. They have identified communities by using graph conductance definition. They have found that the best communities have relatively small sizes that are in the order of 100 nodes (i.e. 1-whiskers). Most importantly, they have found that large social networks have a *core-periphery*. While the periphery of the graph is composed of small components that are weakly connected to the core of the graph, the core itself has an expanding structure. Our empirical results also reveal the same core-periphery structure for the Facebook graph as shown in Figure 5. While the core of the graph is well-connected, there are a large number of 1-whisker components that are connected just with a single edge to the core. Our empirical results show that these 1-whiskers play a fundamental role in the speed of information spreading algorithms.

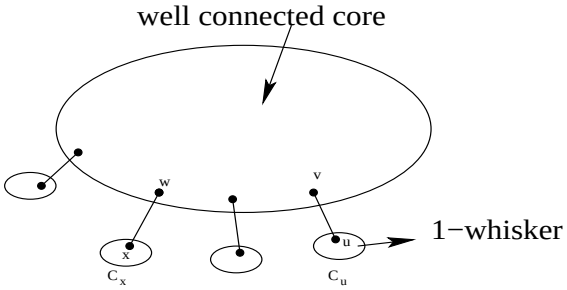

Fig. 5. Well connected core and weakly connected periphery

Although authors of [3] have claimed a sub-logarithmic algorithm for information spreading in social networks, we have shown that Censor algorithm interestingly performs even better than Doerr algorithm. This is mainly because Censor works well in the presence of communication bottlenecks while the Doerr strategy does not. In other words, Censor algorithm can efficiently identify the communication bottlenecks and choose them with a higher probability than the other two algorithms. This strategy helps Censor algorithm to perform better than other algorithms in the presence of 1-whiskers.

V. Conclusions and Future Work

In this paper, we have studied the problem of information spreading in social networks. We have analyzed the running times of random push-pull, Doerr, and Censor algorithms by using empirical data from the Facebook website. Our empirical results have shown that the Censor algorithm outperforms the random push-pull and even the Doerr algorithm. We have justified our performance results by identifying the 1-whiskers as communities which play an essential role in speed of information spreading in social networks. Our results are important because they highlight the importance of 1-whiskers in the performance of information spreading algorithms. For future work, we plan to study how we can improve the performance of Censor algorithm in the presence of 1-whiskers.

References

[1] D. Mosk-Aoyama and D. Shah, "Computing separable functions via gossip," in *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, ser. PODC '06. New York, NY, USA: ACM, 2006, pp. 113–122.

[2] K. Censor-Hillel and H. Shachnai, "Fast information spreading in graphs with large weak conductance." in *SODA*. SIAM, 2011, pp. 440–448.

[3] B. Doerr, M. Fouz, and T. Friedrich, "Social networks spread rumors in sublogarithmic time," in *Proceedings of the 43rd annual ACM symposium on Theory of computing*, ser. STOC '11. New York, NY, USA: ACM, 2011, pp. 21–30. [Online]. Available: http://doi.acm.org/10.1145/1993636.1993640

[4] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Statistical properties of community structure in large social and information networks," in *Proceeding of the 17th international conference on World Wide Web*, ser. WWW '08. New York, NY, USA: ACM, 2008, pp. 695–704.

[5] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *Proceedings of the 2nd ACM workshop on Online social networks*, ser. WOSN '09. New York, NY, USA: ACM, 2009, pp. 37–42.

[6] R. E. Tarjan, "A note on finding the bridges of a graph." *Inf. Process. Lett.*, pp. 160–161, 1974.