Friend Recommendation with a Target User in Social Networking Services

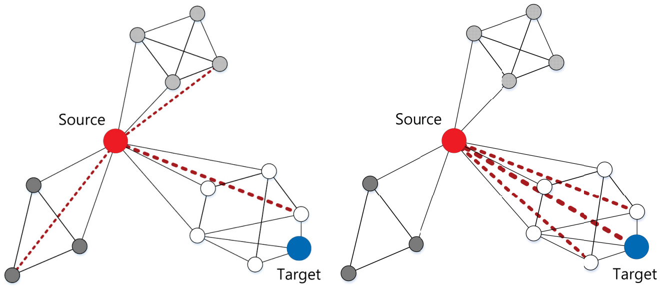
Sundong Kim
supervised by Jae-Gil Lee
Expected graduation date: Feb, 2018
Korea Advanced Institute of Science and Technology, Daejeon, Korea
{sundong.kim, jaegil}@kaist.ac.kr

Abstract—**Friend recommendation is one of the primary functions in social networking services. Suggesting friends has been done by calculating node-to-node similarity based on topological location in a network or contents on a user's profile. However, this recommendation does not reflect the interest of the user. In this paper, we propose a friend recommendation problem in which the source user wants to get more attention from a special target. The goal of our friend recommendation is finding a set of nodes, which maximizes user's influence on the target. To deliver this problem, we introduce information propagation model on online social networks and define two measures: influence and reluctance. Based on the model, we suggest an IKA(Incremental Katz Approximation) algorithm to effectively recommend relevant users. Our method is compared with topology-based friend recommendation method on synthetic graph datasets, and we show interesting friend recommendation behaviors depending on the topological location of users.**

I. Introduction

People use social networking services such as Facebook, Twitter, and Instagram extensively. Almost a billion users are active on such applications, on a daily basis. Several of these users send friend requests to others. One of the most important features of any social networking services is the ability to suggest friends [1], [2]. Each application uses its own algorithm to suggest probable friends to a user [3]. For example, an application might suggest people belonging to a group recently joined by the user, or people related to the user's current job or school. Although these recommendations help the user to form connections within their groups, or to suggest people that he/she probably knows, the results are not meant to help the connection between the user and his/her special target. In this paper, we propose a friend recommendation algorithm for scenarios in which the user wants to get more attention from a special target. Therefore, we intend to suggest friends who can facilitate information flow from the source to the target. For example, if a source node and a target node are not directly connected, suggesting intermediate nodes facilitate communication between them. Additionally, by suggesting the target's neighboring nodes to the source node, we can increase the possibility of a direct connection. Figure 1(b) shows the probable changes to friend suggestions where a system catches a user's willingness to get more attention from the target.

Before tackling the recommendation problem, we model the process of information diffusion through social networks in which an article is shared with direct neighbors of an author, and propagates over the corresponding two-hop neighbors by sharing activities. Among the posts that the target node receives, there might be some posts that originated from the source node, which we define as the source node's influence over the target node. And we formulate a node suggestion problem in online social networks. The main objective is to maximize influence gain by establishing new connections and each recommendation should meet the reluctance threshold.



(a) Recommendation without considering a target user
(b) Recommendation considering a target user

Fig. 1. Two concepts of friend recommendation

To solve this problem, we propose an algorithm called Incremental Katz Approximation(IKA). In this algorithm, we first decrease the candidate set size, and then apply Monte-Carlo simulation to approximate the influence value. Using the approximation result, we update the random diffusion and calculate the influence by taking into account the effect of a new edge. Through experiments, we measure the performance of the proposed algorithm against non-incremental greedy algorithms and topology-based recommendation algorithms using synthetic networks. Additionally, we interpret the recommendation result by changing the topological location of the source and target node.

We summarize our contribution as follows:

- The problem: We suggest a friend recommendation problem in online social networks in which a user wants to maximize his/her influence over a specific target user.
- Design a new measure: We define an influence measure and analyze the effect of having new connections
- Performance: We design the IKA algorithm, which incrementally approximates Katz centrality and proves its performance over topology-based recommendation algorithms.

978-1-4799-8442-8/15/$31.00 © 2015 IEEE

- Discovery: We experiment with diverse settings and interpret the characteristics of recommended nodes.

II. Related Work

A. Friend Recommendation

There are two main friend recommendation approaches, a topology-based approach and a content-based approach. A topology-based approach exploits properties from the network structure and calculates node-to-node similarities. The recommendation will be the node with the highest similarity. Jaccard [4] and SimRank [5] are well-known node-to-node similarity measures. Zhao et al [6] proposed P-Rank, which is the general version of structural similarity. And Leicht et al [7] proposed a similarity measure viewed as a weighted count of the number of paths having possible length between two vertices. A content-based approach tries to recommend items similar to those a given user has liked before. Collaborative filtering is widely used in the content-based approach. Comprehensive knowledge of the content-based approach and collaborative filtering is covered in [8], [9]. Different types of recommendation algorithms are used in different contexts. Lo et al [10] developed a topology-based model to estimate relationship strength by exploiting real message interaction and Armentano et al [11] developed an unsupervised model in a Twitter environment to identify users who can be considered as good information sources. Yang et al [12] suggested an active friending concept and developed an algorithm to maximize acceptance probability.

B. Approximation and Incremental Algorithm for Centrality

Kas [13] dealt with incremental algorithms on closeness, betweenness, and k-centrality. In order to compute those incrementally, the only information needed is the all-pair shortest path. Okamoto et al [14] combined existing methods on calculating exact value and approximate value of close centrality and efficiently find top-k vertices. However, methods in [13], [14] cannot be adapted to the Katz centrality since it is a variant of eigenvector centrality. As a result, the calculation requires computing all centrality values for all vertices, although we only need top-k values. However, Bahmani et al [15] analyzed the efficiency of Monte-Carlo methods for incremental computation of PageRank [16], personalized PageRank [17] on evolving social networks.

III. Problem Definition

In this section, we introduce our social network model and formulate a k-node suggestion problem to derive the best recommendation result. First, we introduce our information propagation model in online social networks and define influence and reluctance between two vertices. Second, we formulate the k-node suggestion problem. The main goal of this problem is to find the set of nodes, which maximizes influence on the target node.

A. Information Propagation Model

We focus on online social network environments wherein each individual has his/her own feed that displays recent articles of its neighbors. We observed the following four information propagation principles. First, an article uploaded by a source node appears on its neighboring node's web feed without any action. Second, an article can be propagated to friends of friends through sharing actions. Third, people can receive an article several times due to the multiple sharing actions by different neighbors. Last, every node can act as a source node and an intermediate node at the same time. With these principles we define the influence measure.

Definition 1: (*Influence*) Let r_{st} is the number of articles that n_t received from the network with a single uploading node n_s. By considering that every node has an uploading behavior, then the probability of n_s's articles covered in n_t's feed is:

$$I_{st} = \frac{r_{st}}{\sum_s r_{st}} \tag{1}$$

Lemma 1: Assume that the only individual who upload its post is the source node n_s. Let S is the set of all walks from n_s to target node n_t, and $length_w$ is a length of each walk w. By having fixed sharing probability p_s, then the expected number of articles that n_t received is:

$$r_{st} = \sum\nolimits_{w \in S} p_s^{length_w - 1} \tag{2}$$

Theorem 1: By considering p_s as an attenuation factor of the Katz centrality, we can represent influence I_{st} by using Katz centrality and personalized Katz centrality.

$$I_{st} = \frac{C_{PKatz}(t)}{C_{Katz}(t)} \tag{3}$$

Then, we model the reluctance between two individuals. The concept of reluctance is awkwardness between two nodes. If two users have high reluctance each other, friend request might not be accepted. To sum up this idea, we define reluctance as negative exponential to Adamic-Adar similarity [18].

Definition 2: (*Reluctance*) Let ρ_{ij} is the reluctance between the two nodes n_i and n_j, and $\Gamma(i)$ is a set of neighbors of node n_i. Then ρ_{ij} is defined as:

$$\rho_{ij} = e^{-sim(i,j)} \tag{4}$$

where

$$sim(i,j) = \sum\nolimits_{n \in \Gamma(i) \cap \Gamma(j)} \frac{1}{\log|\Gamma(n)|} \tag{5}$$

B. Node Suggestion Problem

In this paper, we consider the network with a specific target in which the source node n_s wants to maximize its influence over a specific target n_t. To solve this issue, we suggest relevant friend recommendations to increase the information flow from n_s to n_t. More specifically, we want our suggestions to naturally control the portion of a source node's articles that the target received. Increasing the influence by recommending nodes is not a trivial problem. Intuitively, we can think that I_{st} increases by having another connections from n_s to intermediate node n_t. However, information flows from other nodes to n_t also increase by having those connections. And there is another issue. If we only consider influence maximization, then the friend recommendation algorithm might suggests n_t directly or only suggests the nodes which are located next to n_t. However, those nodes are not close with n_s. So, we have to consider that the suggestions are at least relevant to the source node, which means that we need to consider the reluctance between n_s and our suggested nodes. Our node suggestion problem aims to maximize the influence by having k connections. And each recommendation would not exceed the reluctance threshold. In this preliminary case, we set the reluctance value as 1, which means that two users must have at least a single mutual friend. The basic formulation of the k-node suggestion problem can be represented as follows:

$$\begin{array}{lll} \text{maximize} & I_{st}(G') - I_{st}(G) & \\ \text{subject to} & \rho_{si} < 1 & i = 1, ..., k \end{array} \tag{6}$$

Symbols	Description
G	Undirected Network, $G = (V, E)$
G'	Network after adding edges, $G' = (V, E + E_S)$
S	Ordered Set of k suggested nodes, $S = \{n_1, n_2, ..., n_k\}$
E_S	Ordered Set of new connections by S, $E_S = \{e(n_s, n_1), e(n_s, n_2), ..., e(n_s, n_k)\}$
I_{st}(G)	Influence of n_s over n_t in graph G
ρ_{si}	Reluctance between n_s and n_i in graph G

TABLE I. SYMBOLS FOR THE PROBLEM

IV. PROPOSED ALGORITHM

In this section, we discuss the algorithm designed for searching nodes, which maximize the source node's influence over the target node. Our algorithm *Incremental Katz Approximation*(IKA) follows a greedy approach which selects nodes sequentially. In this algorithm we apply the novel candidate reduction technique and approximate the influence value incrementally.

A. Candidate Reduction

Our greedy approach suggests nodes sequentially, which maximize the influence of each step. The algorithm halts if no suggestion exists, which would increase the influence. In order to find k suggestion sequentially, we need to compute the influence value k times for all unconnected candidate nodes at each round. As experiment network size becomes larger, candidate size also grows proportional to the size of the network. To reduce the problem size, we need to shrink the size of candidates in order to lessen the total running time of the algorithm. First, we have set the candidate set as two-hop neighbors of the source node, since those nodes have at least one mutual friend with the source node. This reduces the size of the searching set from $O(n)$ to $O(d^2)$. Second, we applied gradual candidate reduction after each recommendation by removing candidate nodes that do not increase ΔI_{st}. Since we only require the best node on each recommendation, there is little possibility that those non-beneficial nodes are selected at the next recommendation among all candidates.

B. Influence Approximation

Since the influence measure can be represented using a Katz centrality, we need matrix inversion to calculate I_{st}. So we approximate this value by Monte-Carlo simulation to avoid the complexity. We found the possibility of Monte-Carlo simulation on Katz centrality from the paper which analyzes the efficiency of approximating personalized PageRank on an evolving graph[15]. Unlike PageRank, articles can spread or disappear to multiple nodes in our settings. So we simulate information diffusion according to our propagation model. For the example of Katz centrality, we initialize R_1 articles starting from each node of the network. And we count X_t which is the total number of articles that pass through the target until the simulation ends. We can approximate the Katz centrality or information transfer rate from all nodes to target with:

$$\sum\nolimits_s \tilde{r}_{st} = X_t / nR_1 \tag{7}$$

Then, we approximate the average number of source node's articles that the target node received. This is done by initializing articles solely from the source node. Unlike the previous case, a sufficiently large number of articles should be initialized from the source node to approximate the value. By doing this, we are able to get the personalized Katz centrality. For the target node n_t, Y_t is the total number of articles pass through. If R_2 articles are initialized from the n_s, we can approximate r_{st} with:

$$\tilde{r}_{st} = Y_t / R_2 \tag{8}$$

Our measure for recommendation, influence of source node over target, I_{st} can be approximated by using (7) and (8).

$$I_{st} = \frac{r_{st}}{\sum_s r_{st}} \approx \frac{nY_tR_1}{X_tR_2} \tag{9}$$

C. Incremental Update

In order to recommend a node, we have to recalculate an influence value for all candidates considering that a new edge is added to the network. As a result, substantial amount of calculation overlap occurs if we recalculate Katz centrality from the beginning. Here we refer to the previous random diffusion result, and only calculate the extra part that is affected by a new edge. Having another edge means that information from the source node can flow to new neighbors if the sharing condition is met. Therefore at each simulation step, we check articles located at the end of the new edge. If the sharing condition is met, another diffusion begins from the two endpoints. The first part of the algorithm initializes the new diffusion if there are existing articles available on the two nodes affected by the new edge. The second part is generating additional transmission for articles, which arrives at two nodes at each iteration. And the last part is continuing the random diffusion initiated by the effect of the new edge. Using this incremental update, we can significantly reduce the computation burden. In the same way, we can update the influence value while there are existing beneficial users to recommend. Algorithm 1 describes an overview of our method. Since we adopt an approximation rather than the matrix inversion for computing Katz centrality, and incrementally update this value. We achieve our algorithm works on large networks. Here, V_C, $N_G(n_s)$, ${N_G}^2[n_s]$ represents the candidate set, close neighborhood of n_s, and two-hop neighbor of n_s respectively.

V. EXPERIMENTS

In our experiments we aim to answer the following questions:

Q1. How well can IKA solve our problem?

Q2. How do we determine the adequate simulation size to achieve accuracy and speed at the same time?

Q3. How do recommendation results vary by changing the topological location of source and target user and initial connectedness?

Algorithm 1 INCREMENTAL KATZ APPROXIMATION(IKA)

```
Input: Graph $G = (V, E)$, $n_s$, $n_t$, $p_s$, $R_1$, $R_2$, $nIter$
Output: Set of nodes $S = \{n_{i_1}, n_{i_2}, ..., n_{i_k}\}$
Approximate $C_{Katz}(t)$, $C_{PKatz}(t)$
Calculate $I_{st}$ using $C_{Katz}(t)$ and $C_{PKatz}(t)$
$S = \{\}$
$V_R = \{\}$
$V_C = {N_G}^2[n_s] - N_G[n_s]$
while $\Delta I_{st} > 0$ do
    $V_R = \{n_c | \Delta I_{st} < 0\}$
    Find $n_m = \underset{n_c \in V_C}{\text{argmax}}\, \Delta I_{st}$ by updating $I_{st}$ for each new connection
    if $\Delta I_{st} > 0$ then
        $G = G + e(n_s, n_c)$
        $S = S + n_c$
        $V_C = V_C - V_R + {N_G}^2[n_s] - N_G[n_s]$
        Update $I_{st}$ and its diffusion process
    end if
end while
return S
```

A. Implementation Details

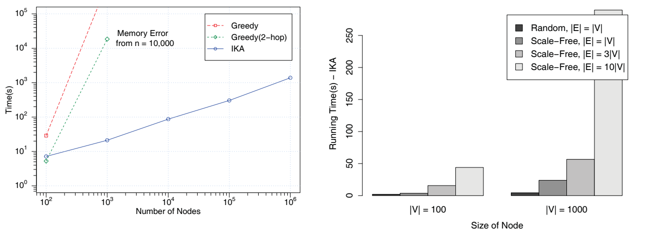
The code for IKA has been written in Python NetworkX module [19]. All experiment were performed on a PC with Intel Core i5-4670 3.4GHz processor, 8 GB of main memory and a 128GB SSD drive. Networks we use in the experiments along with their descriptions are summarized in Table II.

Topology	Node	Edges	Description
Scale-Free [20]	100 1,000 10,000 100,000 1,000,000	99 999 9,999 99,999 999,999	m=1 (m : number of edges to attach from a new node to existing nodes)
	200	398	m=2
	100 1,000	297 2,997	m=3
	100 1,000	990 9,990	m=10
Erdős-Rényi Random [21]	100 1,000	100 1,000	degree = 1

TABLE II. NETWORK DESCRIPTION

B. Time Comparison

In order to compare the performance between exact influence calculation and Monte-Carlo approximation, we measured the time for recommending ten consecutive nodes. We setup the first five synthetic graphs in Table II, and set the source node as a leaf node, and the target node as a hub node. We set $R_1 = 1$, $R_2 = 10,000)$ regardless of the network size. Based on the results, we found our algorithm IKA to be fast and scalable compared to the greedy algorithms with exact influence calculation. Figure 2(a) shows the time comparison between IKA and two greedy algorithms. Red line is a greedy algorithm without candidate reduction, influence approximation and incremental update. And green line is a greedy algorithm with candidate reduction. Then, we tried our algorithm on graphs that have different densities. The results are shown in Figure 2(b).



(a) Running time comparison between IKA and exact algorithms according to the network size

(b) Running time of IKA according to the density of networks

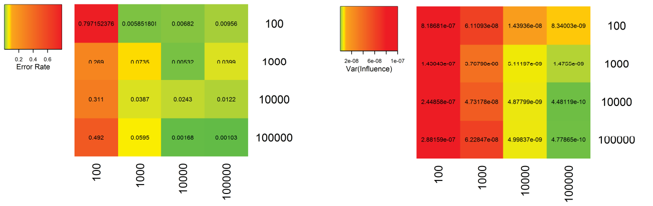
Fig. 2. Time comparison

C. Error Analysis

Second we found that our approximation of $\Delta \tilde{I}_{st}$ is accurate by comparing it with the exact calculation of ΔI_{st}. Here we measure the influence gain for the first recommendation in a scale-free graph by changing the number of initializing articles for the Monte-Carlo simulation. We used $nR_1 = R_2 = \{100, 1000, 10000, 100000\}$. And the relative error is measured as

$$\frac{|\Delta I_{st} - E[\Delta \tilde{I}_{st}]|}{\Delta I_{st}}$$

Figure 3(a) describes the relative error by increasing R_2(x-axis) and nR_1(y-axis), and Figure 3(b) describes $Var(\Delta \tilde{I}_{st})$ by increasing R_1 and R_2.



(a) Accurate result with enough large number of R_2(x-axis)

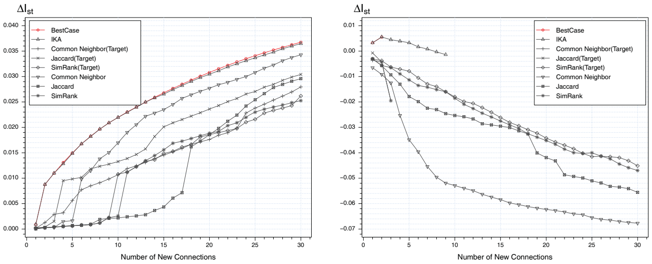
(b) Stable result by increasing both R_1 and R_2

Fig. 3. Error analysis

From error analysis, we found that our algorithm guarantees accurate friend recommendation with a sufficiently large R_2. We also found that the approximations work well even for the small size of R_1. This experimentally shows that findings of [15], [22] are valid in the case of Katz centrality and influence. Selecting adequate size of R_1 and R_2 is important since running time also increases proportional to the number of initializing articles. To achieve fast running time with accuracy and stability, we select $R_1 = 1, R_2 = 10,000$ for the following experiments.

D. Performance Comparison and Interpretation

We compared the performance of our algorithm with topology-based friend recommendation with node-to-node similarity measures. The performance measure of these experiments is cumulative influence gain by having new connections. For the comparison, we implemented variation of node similarity measures such as common neighbor, Jaccard [4], SimRank [5]. For example, for the Jaccard(Target) measure, we recommend nodes sequentially, which have the highest Jaccard similarity with n_t from the candidate set. For this experiment, we used a scale-free graph with $|V| = 200, |E| = 396$ with various settings. Figure 4(a) and 4(b) show the performance of IKA and topology-based recommendation results with node-to-node similarity measures. Figure 4(a) is the case in which n_s(leaf node) and n_t(hub node) are initially unconnected, and Figure 4(b) is the case in which n_s(leaf node) and n_t(leaf node) are initially connected.



(a) Friend recommendation result (n_s, n_t = not connected, n_s = leaf, n_t = center

(b) Friend recommendation result (n_s, n_t = connected, n_s = leaf, n_t = leaf

Fig. 4. Performance of IKA over other algorithms

There are several aspects to check in these figures. To begin, how cumulative influence changes by the first few recommendations must be checked. More specifically, how many steps are needed to achieve a the large leap on influence gain must be checked. In the case of Figure 4(a), IKA connects to the most influential node in the first few steps. As we know that the leap occurs by connecting to the n_t directly, we can understand that recommendation before the big leap is the process of indirectly exposing oneself to the target. And the recommendation result after the big leap shows the behavior of connecting to the target node's neighbors. Figure 4(b) is a special case in which the recommendation is refrained. Since n_s and n_t are both leaf nodes and they are already connected, generating more edges around the two nodes adversely affects their information transfer. In contrast to the other experiment, IKA stops after recommending a few nodes. Most other algorithms completely fail in terms of increasing the influence of the source node.

VI. Future Work and Conclusions

In this paper, we propose the friend recommendation problem and an algorithm on the social network environment in which the user has a specific target node to maximize its influence. We define the influence as how much effect the source node's post has on a target node's feed. Additionally, we formulate our problem by suggesting nodes one-by-one, which maximize the influence score. From our model, we find the influence value matches with a concept of Katz centrality, and designed an algorithm to incrementally approximate it. We test our algorithm on various networks and show that our algorithm is able to suggest relevant friends in terms of promoting information flow. We want to extend our idea to more realistic settings in which all users in the network have different posting behaviors and sharing probabilities, and enable our algorithm to work on the problem having multiple target users.

References

[1] M. Moricz, Y. Dosbayev, and M. Berlyant, "PYMK: friend recommendation at myspace," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 999–1002.

[2] M. Roth, A. Ben-David, D. Deutscher, G. Flysher, I. Horn, A. Leichtberg, N. Leiser, Y. Matias, and R. Merom, "Suggesting friends using the implicit social graph," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 233–242.

[3] S. A. Golder and S. Yardi, "Structural predictors of tie formation in twitter: Transitivity and mutuality," in *Social Computing (SocialCom), 2010 IEEE Second International Conference on*. IEEE, 2010, pp. 88–95.

[4] G. Salton and M. J. McGill, "Introduction to modern information retrieval," 1983.

[5] G. Jeh and J. Widom, "SimRank: a measure of structural-context similarity," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 538–543.

[6] P. Zhao, J. Han, and Y. Sun, "P-Rank: a comprehensive structural similarity measure over information networks," in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 553–562.

[7] E. Leicht, P. Holme, and M. E. Newman, "Vertex similarity in networks," *SIAM Journal on Numerical Analysis*, vol. 45, no. 2, pp. 890–904, 2006.

[8] P. Lops, M. De Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender systems handbook*. Springer, 2011, pp. 73–105.

[9] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, p. 4, 2009.

[10] S. Lo and C. Lin, "WMR–a graph-based algorithm for friend recommendation," in *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*. IEEE Computer Society, 2006, pp. 121–128.

[11] M. G. Armentano, D. Godoy, and A. Amandi, "Topology-based recommendation of users in micro-blogging communities," *Journal of Computer Science and Technology*, vol. 27, no. 3, pp. 624–634, 2012.

[12] D.-N. Yang, H.-J. Hung, W.-C. Lee, and W. Chen, "Maximizing acceptance probability for active friending in online social networks," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 713–721.

[13] M. Kas, "Incremental centrality algorithms for dynamic network analysis," Ph.D. dissertation, Citeseer, 2013.

[14] K. Okamoto, W. Chen, and X.-Y. Li, "Ranking of closeness centrality for large-scale social networks," in *Frontiers in Algorithmics*. Springer, 2008, pp. 186–195.

[15] B. Bahmani, A. Chowdhury, and A. Goel, "Fast incremental and personalized pagerank," *Proceedings of the VLDB Endowment*, vol. 4, no. 3, pp. 173–184, 2010.

[16] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." 1999.

[17] G. Jeh and J. Widom, "Scaling personalized web search," in *Proceedings of the 12th international conference on World Wide Web*. ACM, 2003, pp. 271–279.

[18] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social networks*, vol. 25, no. 3, pp. 211–230, 2003.

[19] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Laboratory (LANL), Tech. Rep., 2008.

[20] P. Holme and B. J. Kim, "Growing scale-free networks with tunable clustering," *Physical review E*, vol. 65, no. 2, p. 026107, 2002.

[21] P. Erdős and A. Rényi, "On random graphs I." *Publ. Math. Debrecen*, vol. 6, pp. 290–297, 1959.

[22] K. Avrachenkov, N. Litvak, D. Nemirovsky, and N. Osipova, "Monte carlo methods in PageRank computation: when one iteration is sufficient," *SIAM Journal of Numerical Analysis*, vol. 45, no. 2, pp. 890–904, 2007.