A Distributed and Privacy Preserving Algorithm for Identifying Information Hubs in Social Networks

Muhammad U. Ilyas
Department of ECE
Michigan State University
East Lansing, MI 48824
ilyasmuh@egr.msu.edu

M. Zubair Shafiq
Department of CSE
Michigan State University
East Lansing, MI 48824
shafiqmu@cse.msu.edu

Alex X. Liu
Department of CSE
Michigan State University
East Lansing, MI 48824
alexliu@cse.msu.edu

Hayder Radha
Department of ECE
Michigan State University
East Lansing, MI 48824
radha@egr.msu.edu

***Abstract*—This paper addresses the problem of identifying the top-k information hubs in a social network. Identifying top-k information hubs is crucial for many applications such as advertising in social networks where advertisers are interested in identifying hubs to whom free samples can be given. Existing solutions are centralized and require time stamped information about pair-wise user interactions and can only be used by social network owners as only they have access to such data. Existing distributed and privacy preserving algorithms suffer from poor accuracy. In this paper, we propose a new algorithm to identify information hubs that preserves user privacy. The intuition is that highly connected users tend to have more interactions with their neighbors than less connected users. Our method can identify hubs without requiring a central entity to access the complete friendship graph. We achieve this by fully distributing the computation using the Kempe-McSherry algorithm to address user privacy concerns. To the best of our knowledge, the proposed algorithm represents an arguably first attempt that (1) uses friendship graphs (instead of interaction graphs), (2) employs a truly distributed method over friendship graphs, and (3) maintains user privacy by not requiring them to disclose their friend associations and interactions, for identifying information hubs in social networks. We evaluate the effectiveness of our proposed technique using a real-world Facebook data set containing about 3.1 million users and more than 23 million friendship links. The results of our experiments show that our algorithm is 50% more accurate than existing distributed algorithms. Results also show that the proposed algorithm can estimate the rank of the top-k information hubs users more accurately than existing approaches.**

I. Introduction

A. Background and Motivation

In a social network, a user that has a large number of interactions with other users is defined as an *information hub* (or simply a *hub*) [8]. An interaction refers to the transmission of information by one user to another user. For example, an interaction from user A to user B in online social networks may be the action when user A posts a message or comment on user B's profile. Hubs play important roles in the spread or subversion of propaganda, ideologies, or gossips in social networks. Taking the advertising industry as an example, instead of giving free product samples to random people, to improve the effectiveness of word of mouth advertising and increase recommendation based product adoption, they may want to give free samples to hubs only [12]. Furthermore, observing adoption of products or trends at hubs helps to predict the eventual total sale of a product [12]. Due to limited advertisement budget (*e.g.*, free product samples), advertisers want to identify the top-k nodes in a social network. Therefore, identifying top-k information hubs in social networks is an important problem.

B. Limitations of Prior Art

Prior methods for computing top-k information hubs (*e.g.*, [22] and [13]), are mostly centralized assuming the availability of either interaction or friendship graphs. The interaction graph of a social network is a directed multigraph [4] whose nodes represent users and directed links represent the existence of a directed pair-wise interaction. Each link is labeled with a time stamp that indicates when the interaction occurred. The friendship graph of a social network consists of nodes representing users and undirected links representing the friend relationship among users. However, centralized computation of top-k information hubs is mostly unrealistic for parties such as advertisers because online social networking companies are reluctant to share their interaction or friendship graphs due to privacy concerns, regulations [1] and usage limitations in the terms of service [3].

C. Proposed Solution

In this paper, we propose a distributed and privacy preserving algorithm for computing top-k information hubs in social networks. Privacy preservation is a requirement because users are typically hesitant to disclose explicit information about their friendship links or interaction information [31]. To preserve the privacy of user interactions, our algorithm is distributed and does not require the advertiser to know users' friendship associations or interactions. There are three technical challenges in designing such an algorithm. First, the problem of inferring a user's salience (whose ground truth resides in the interaction graph) from the corresponding friendship graph is inherently difficult because an interaction graph has more information than its corresponding friendship graph. Furthermore, a friendship graph is undirected and unweighted, whereas the interaction graph is a directed multigraph. Second, the complete friendship graph itself may not be available to the parties interested in identifying hubs. Third, preserving users' privacy in this computation is difficult as any information exchange involved in this computation should not contain any personal information.

To address the first challenge, we apply principal component centrality (PCC), a new measure of centrality we introduced in [14], to the friendship graphs. The intuition behind PCC is that a user who is connected to well-connected users (even if the user himself is poorly connected) has a more central status. Unlike other measures of user influence (*e.g.*, eigenvector centrality [5], [6]), PCC takes into consideration the fact that social networks can be multi-polar consisting of multiple communities with few connections between them.

To address the second challenge of friendship graph data availability, we distribute the computation of PCC among users and therefore do not require a central entity to access the friendship graph. Advertisers can utilize existing functionality in popular online social networks (such as *groups* and *pages* in Facebook [2]) to implement our proposed distributed method. Motivation for user participation in decentralized PCC computation may range from tangible incentives such as receiving free samples from advertisers (*e.g.* [11]), to intangible incentives such as bragging rights about one's popularity (*e.g.* [27]). We decentralize the PCC computation using the Kempe-McSherry (KM) algorithm [16]. This iterative algorithms computes eigenvalues and eigenvectors that are essential for computing nodes' PCCs. Our decentralized algorithm restricts the set of users that a particular user has to communicate with to its immediate friends. Furthermore, the memory requirement at each user of this algorithm grows only linearly with the number of friends. Hence, one of the contributions of this work is extending the original centralized PCC approach to a more practical distributed PCC form.

Finally, to address the issue of user privacy, only real numbers representing PCC intermediate scores are exchanged between users. It is impossible to reverse-engineer users' friendship associations from these intermediate scores.

D. Experimental Results and Findings

We evaluated the effectiveness of our proposed technique using a real-world Facebook data set [30] containing about 3.1 million users and more than 23 million friendship links. We have four major findings from our experimental results. First, there is indeed close correlation between the PCC of nodes in the friendship graph and corresponding dynamic user interaction data. We envision that this correlation can be exploited for other purposes as well. Second, the computation of PCC can be effectively distributed across individual users in a social graph without compromising its accuracy. This eliminates the requirement of a central authority for identifying hubs. Third, the accuracy of PCC improves as we use more eigenvectors in its computation. Further, the appropriate number of eigenvectors required in the computation of PCC for real-world social networks is around 10. Fourth, the accuracy (in terms of number of correctly identified top-k users and their estimated rank) of PCC improves as the duration of interaction data used for comparison is increased from 1 month, to 6 months to more than a year. This essentially shows that PCC scores reflect the flow of information between users of a social network over long time periods.

E. Key Contributions

We make four key contributions in this paper. First, we propose a novel method to infer information lying in the interaction graph (*i.e.* hub identification) from the friendship graph in social networks without using the interaction data. Earlier works are limited to solving this problem using complete interaction graph data. Second, our proposed method, first of its kind, allows third parties (other than social network owners) to solve this problem. We use a distributed method to overcome the requirement of a central authority. Third, our proposed method preserves the privacy of users, *i.e.*, users do not release any personal information to other users. We achieve this objective by letting each user share only some real numbers that cannot be reverse-engineered. Finally, we evaluate the effectiveness of our proposed technique using a real-world Facebook data set that is publicly available. The results of our experiments show that the proposed approach improves the accuracy of finding the top-k user set by approximately 50% over existing measures. Furthermore, the proposed technique accurately estimates the rank of individual users.

The rest of the paper proceeds as follows. In Section II, we present an overview of related work. We provide the details of our proposed approach in Section III. We then provide the detailed results of our evaluation in the rest of Section IV. Finally, we conclude the paper in Section V.

II. Related Work

Several algorithms have been proposed for identifying top-k influential users in social networks [10], [15], [17], [18], [29], [32]. The objective function for this influence maximization problem is to maximize the number of users that are part of information flows initiated by the top-k influential users. In contrast, our method uses friendship graphs, is fully distributed, and is privacy-preserving, while such work uses user interaction data and is centralized and is not privacy-preserving. Algorithms that forgo using interaction data use structural information like the friendship graph. They are based on centrality measures computed from friendship graph topologies [19], [21], [23], [26], [28]. However, as Borgatti showed in [7], degree, closeness and betweenness centrality are inappropriate measures of centrality for influence processes.

III. Our Proposed Solution

A. Motivation for Principal Component Centrality

We model the information flow as an influence process. The underlying rationale for doing so is rooted in the assumption that in social networks people (nodes) with more friends (connections) send and receive more messages. Furthermore, people will receive more messages from friends that send/receive a lot of traffic than from those that send/receive fewer messages. This information flow can be modeled as an influence process. According to Borgatti's two dimensional taxonomy of node centrality measures in [7], the appropriate measure to quantify nodes' influence is eigenvector centrality (EVC) [5], [6]. As we demonstrated in [14], in networks of multiple communities with sparse connectivity between communities, EVC assigns centrality scores to nodes according to their location with respect to the most dominant community. When applied to large networks, EVC fails to assign significant scores to a large fraction of nodes. The principal eigenvector is "pulled" in the direction of the largest community. The motivation for using PCC as a measure of node influence may be understood by looking at EVC in the context of principal component analysis (PCA) [9].

B. Definition of PCC

While EVC assigns centrality to nodes according to their location with respect to the most dominant community in a graph G, PCC takes into consideration additional communities. Let $\mathbf{A}$ denote the adjacency matrix of a graph $G(V, E)$ consisting of the set of nodes $V = \{v_1, v_2, v_3, \ldots, v_N\}$ of size N and set of undirected edges E. When a link is present between two nodes v_i and v_j, both $A_{i,j}$ and $A_{j,i}$ are set to 1 and set to 0 otherwise. Here PCA is used to find the eigenvectors $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_N$ and eigenvalues $\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_N$ of the graph G's adjacency matrix $\mathbf{A}$. We define the PCC of a node in a graph as its Euclidean distance/ℓ^2 norm from the origin in the P-dimensional eigenspace. The basis vectors of that eigenspace are the P most significant eigenvectors of the adjacency matrix A of the graph G under consideration. For a graph G, its N eigenvalues $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_N|$ correspond to the normalized eigenvectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$, respectively. The eigenvector/eigenvalue pairs are indexed in descending order of magnitude of eigenvalues. When $P = 1$, PCC equals a scaled version of EVC. The parameter P in PCC can be used as a tuning parameter to adjust the number of eigenvectors included in PCC.

Let $\mathbf{X}$ denote the $N \times N$ matrix of concatenated eigenvectors $\mathbf{X} = [\mathbf{x}_1 \mathbf{x}_2 \ldots \mathbf{x}_N]$ and let $\mathit{\Lambda} = [\lambda_1 \lambda_2 \ldots \lambda_N]^T$ be the vector of eigenvalues. Furthermore, if $P < N$ (typically $P \ll N$) and if matrix $\mathbf{X}$ has dimensions $N \times N$, then $\mathbf{X}_{N \times P}$ will denote the submatrix of $\mathbf{X}$ consisting of the first N rows and first P columns. Then PCC can be expressed in matrix form as:

$$\mathbf{C}_P = \sqrt{\left(\left(\mathbf{A}\mathbf{X}_{N \times P}\right) \odot \left(\mathbf{A}\mathbf{X}_{N \times P}\right)\right) \mathbf{1}_{P \times 1}} \tag{1}$$

The '$\odot$' operator is the Hadamard (or entrywise product) operator and $\mathbf{1}_{P \times 1}$ is a vector of 1s of length P. Equation 1 can also be expressed in terms of the eigenvalue and eigenvector matrices $\mathit{\Lambda}$ and $\mathbf{X}$, of the adjacency matrix $\mathbf{A}$:

$$\mathbf{C}_P = \sqrt{\left(\mathbf{X}_{N \times P} \odot \mathbf{X}_{N \times P}\right) \left(\mathit{\Lambda}_{P \times 1} \odot \mathit{\Lambda}_{P \times 1}\right)}. \tag{2}$$

Node PCCs, as defined in equations 1 and 2, are not normalized. To allow interpretation of centrality scores, Ruhnau advocated in [25] that they should be normalized by either the Euclidean norm (ℓ_2 norm) or the maximum norm (ℓ_∞ *i.e.* the maximum centrality score) of the centrality vector. For the remainder of this paper the PCC vector will be normalized by the ℓ_∞ norm, thereby restricting all entries to the range $[0, 1]$.

C. Selection of Number of Eigenvectors

The cost of computing an eigenvector can be significant for large matrices, favoring the use of as few eigenvectors for PCC as are necessary. To determine appropriate number of eigenvectors (P_{app}), we consider the phase angle ϕ as a function of P. The phase angle $\phi(P)$ of a PCC vector $\mathbf{C}_P$ is defined as its angle with the EVC vector $\mathbf{C}_E$ and is defined mathematically in equation 3.

$$\phi(P) = \arccos\left(\frac{\mathbf{C}_P}{|\mathbf{C}_P|} \cdot \frac{\mathbf{C}_E}{|\mathbf{C}_E|}\right) \tag{3}$$

When the phase angle function is plotted for a range of P, the value of P at which ϕ begins approaching its final steady value is used for that particular graph (P_{app}) [14]. The selection of P_{app} can be made as,

$$P_{app} = min\{\phi(P+1) - \phi(P)\} \quad \in \ [-\epsilon, \epsilon], \forall \ [P, N], \tag{4}$$

where ϵ is a small real number. It is our observation that the value of P_{app} is close to the number of well-connected communities in a social graph.

D. Decentralized Eigendecomposition Algorithm

The massive sizes of social networks require a method whose space and time complexity scales well with the number of nodes and links between them. According to Equation 1, for a node to compute its own PCC score it needs to know its corresponding entries in the first P eigenvectors $\mathbf{x}_1$, $\mathbf{x}_2$ ... $\mathbf{x}_P$ of the adjacency matrix $\mathbf{A}$, as well as who its neighbors are, i.e. the entries in its corresponding row of A. Although many decentralized algorithms for computing eigenvectors of a matrix exist, many of them, like the power iteration [20], are not designed to minimize the communication overhead between participating nodes. In [16] Kempe and McSherry developed a decentralized algorithm for the computation of the first P most significant eigenvectors. Their approach differs from other algorithms in that each node is only required to communicate with neighbor nodes.This means that the computational complexity of the algorithm at every node, and the volume of messages exchanged by each node scales only linearly with the number of its neighbors and linearly with the number of eigenvectors that are computed. Furthermore, the time for the algorithm to converge is $O(\tau_{mix} \log N)$, where N is the total number of nodes in the graph and τ_{mix} is the mixing time of the Markov chain with a transition matrix that is the row-normalized version of $\mathbf{A}$. Kempe reported the error of the ℓ_2 norm of the space spanned by R_P, the projection of P most significant eigenvectors on $\mathbf{A}$, and $R_{P'}$, the projection of P most significant eigenvectors by KM-algorithm onto $\mathbf{A}$, with high probability as follows.

$$\|\mathbf{R}_P - \mathbf{R}_{P'}\|_2 \leq O\left(\left|\frac{\lambda_{P+1}}{\lambda_P}\right|^t \cdot N\right) + 3\epsilon^{4t} \tag{5}$$

Here, t is the number of iterations for which the KM algorithm executes. Clearly, since $\lambda_{P+1} < \lambda_P$, the fractional term decreases with t at a geometric rate. For a detailed coverage of the KM algorithm we refer the reader to [16]. From the above discussions of the KM algorithm, we conclude the following:

- The KM algorithm never requires a node to communicate beyond its immediate neighbors. This implies that the

communication overhead of the KM algorithm scales linearly with the number of computed eigenvectors.
- Kempe *et al.* reported near perfect convergence for their algorithm.

IV. Experimental Results

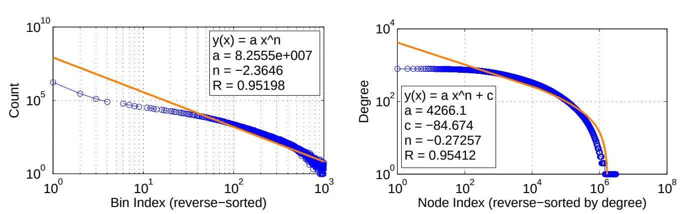
A. Data Set

In our study, we use the data set collected by Wilson *et al.* from Facebook [30]. For detailed information about the data collection procedure, coverage completeness, justification and verification we refer the reader to [30]. The data set further consists of two types of graphs. First, we have an undirected friendship graph where the nodes represent users and links represent the friendship between two users. Second, we have a directed pair-wise user interaction graph where the nodes represent users and the directed links represent the interaction from one user to another. The interaction data spans a time duration of one year. Note that we use the interaction data only to evaluate the ground truth. Table I provides the basic statistics of the friendship graphs analyzed in this study.

TABLE I: Basic statistics of the friendship graph analyzed in this study

Property	Value
# Users	3097165
# Friendship Links	23667394
Average Clustering Coefficient	0.0979
# Cliques	28889110
Transitivity	0.0477

Figure 1 shows the plots of degree distributions for friendship graph. In Figure 1a, we plot the histogram of one thousand bins in which we have grouped users. Although the distribution does not follow a power-law exactly, it fits it relatively well as shown by straightness on log-log scale and verified by high goodness-of-fit (R^2) values for the data set. This observation is in accordance with the result of recent studies that have shown that the degree distribution of many online social networks is power-law [24]. An equivalent representation is shown in Figure 1b where users are reverse-sorted by their degree.



(a) Power law (b) Zipf-like

Fig. 1: Degree distribution of friendship graph for Facebook data set A

B. Selection of PCC Parameter

We can compute the PCC vector $\mathbf{C}_P$ for a range of number of eigenvectors P. Note that at $P = 1$ the PCC $\mathbf{C}_1$ is the EVC $\mathbf{C}_E$, which serves as the measure of baseline comparison, as mentioned in [28] and [23]. Although we will be comparing PCC with EVC for a range of values of P in some of our subsequent analysis, we will try to determine the "appropriate" number of eigenvectors for PCC (denoted by P_{app}). We do this by means of plotting the phase angle function defined in Equation 3. Figure 2 plots the phase angle functions of the Facebook data set for the range of $P = 1$ to 100. The phase angle function rises quickly initially until $P = 6$ and rises only very slowly thereafter. Using Equation 4, the P_{app} value is 10. In other words, fewer number of eigenvectors are enough to approximate the steady-state PCC value.

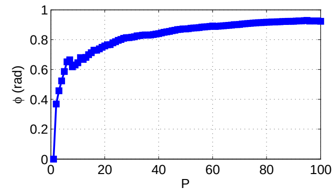


Fig. 2: Plot of the phase angle $\phi(P)$ between PCC vectors $\mathbf{C}_P$ and EVC vector $\mathbf{C}_E$ plotted against number of feature vectors P.

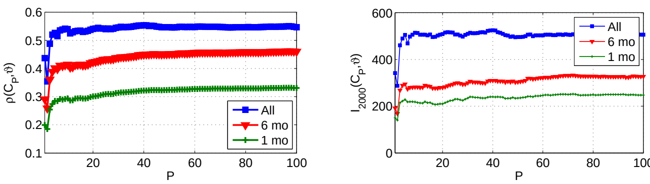
C. Comparison With Ground Truth

1) Verification of Optimal PCC Parameter: The remaining section is devoted to evaluating PCC's accuracy by comparing its performance with the ground truth, *i.e.* interaction data. We compare the PCC of nodes against their actual flows over various time periods to get a sense of the time period over which PCC best predicts the flow. We have used a symmetric measure called Pearson's product-moment coefficient to quantify the similarity between the output of PCC and the ground truth from interaction data. The Pearson's product-moment coefficient ρ is defined in Equation 6. Here E is the expectation operator, σ the standard-deviation, and μ the mean value.

$$\rho(\mathbf{C}_P, \vartheta) = \frac{E[(\mathbf{C}_P - \mu_{\mathbf{C}_P})(\vartheta - \mu_\vartheta)]}{\sigma_{\mathbf{C}_P}\sigma_\vartheta} \quad (6)$$

Figure 3a shows the plots of correlation coefficients $\rho(\mathbf{C}_P, \vartheta)$ as a function of number of eigenvectors for the range $1 \le P \le 100$ for flows collected over 1 month, 6 months and the entire collection time period (labeled 'All'). We make two major observations from these plots. First, we note that the value of ρ generally increases with increasing number of eigenvectors P for computing PCC. ρ reaches close to its steady-state value at around 10 eigenvectors. Note that the steady-state values for ρ are reached at P_{app} values selected in the previous subsection. This observation verifies the merit of using phase angle for selection of appropriate value of P for PCC computation. Second, we note that the correlation coefficients are higher for interaction data collected over longer periods of time. This observation follows our intuition that the trends in short-term interaction data can deviate from our expectations in steady-state friendship graph; however, the trends in long-term interaction data show greater similarity with the underlying friendship graph.

2) Accuracy of PCC in Predicting Top-2000 Users: Now we analyze the overlap between the set of top-2000 users by PCC (denoted by $S_{2000}(\mathbf{C}_P)$) and the ground truth. Note that the choice of 2000 nodes in the following analysis is purely arbitrary. The results of our analysis for different set sizes are qualitatively similar. Let the cardinality of the intersection set of the first k nodes by PCC and the first k nodes by flow ϑ



(a) Correlation coefficients

(b) Size of intersection set

Fig. 3: (a) Correlation coefficients ρ of PCC $\mathbf{C}_P$ and flow count of Facebook data set (ϑ). The correlation coefficients are plotted as functions of the number of eigenvectors P for each interaction graph. (b) Size of the intersection set in Facebook data set for varying number of eigenvectors used for PCC.

be denoted by $I_k(\mathbf{C}_P, \vartheta)$ and defined in Equation 7 below.

$$I_k(\mathbf{C}_P, \vartheta) = \frac{|S_k(\mathbf{C}_P) \cap S_k(\vartheta)|}{k} \tag{7}$$

Figure 3b plots I_{2000} for interaction data of different durations. As expected, the cardinality of the intersection set increases with the number of eigenvectors used for PCC. In the figure, the data points at $P = 1$ represent the baseline for our comparison, *i.e.* EVC. The cardinality of the intersection set of the top-2000 nodes by EVC and top-2000 nodes by flow ϑ, the cardinality of the intersection set $I_{2000}(\mathbf{C}_E, \vartheta)$ is 342. At $P = 10$, $I_{2000}(\mathbf{C}_{10}, \vartheta) = 513$. These represent increases of 50.0%. For the remainder of this section, we fix the values of P at $P_{app} = 10$. We see greater agreement between the list of nodes generated by PCC score with flow data collected over a longer durations.

V. Conclusions

Information hubs in a social network play important roles in the speed and depth of information diffusion. Identifying hubs helps us to harness their power to pursue social good at local, national, and global levels. In this paper, we propose the first friendship graph based, fully distributed, and privacy-preserving method for identifying hubs. Unlike prior work, our method can be used to identify hubs by parties other than social network owners as we do not require any entity to access interaction or friendship graphs. We conducted experiments using data collected from the Facebook social network. The data set used in this study was collected over the period of more than a year and contain data from about 3.1 million users. The results of our experiments using this data showed that our proposed protocol accurately (in terms of number of correctly identified top-k nodes and their estimated rank) identifies the top-k information hubs in a social network.

Acknowledgement

This material is based in part upon work supported by the National Science Foundation under Grant Numbers IIS-0968495. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] *Facebook Advertising, http://www.facebook.com/advertising/*, 2010.
[2] *Facebook Pages, http://www.facebook.com/advertising/?pages*, 2010.
[3] Statement of rights and responsibilities. *http://www.facebook.com/terms.php*, 2010.
[4] B. Bollobas. *Modern graph theory*. Springer Verlag, 1998.
[5] P. Bonacich. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, 2(1):113–120, 1972.
[6] P. Bonacich. Technique for analyzing overlapping memberships. *Sociological Methodology*, 4:176–185, 1972.
[7] S. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55–71, 2005.
[8] G. Caldarelli. *Scale-Free Networks*. Oxford University Press, 2007.
[9] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley New York, 2nd edition, 2001.
[10] P. A. Estevez, P. Vera, and K. Saito. Selecting the most influential nodes in social networks. In *proceedings of IJCNN*, 2007.
[11] D. Geere. Samsung offers free phones to frustrated iPhone users. *CNN Tech*, http://www.cnn.com/2010/TECH/mobile/07/24/ samsung.replacing.iphones/, 2010.
[12] J. Goldenberg, S. Han, D. R. Lehmann, and J. W. Hong. The role of hubs in the adoption processes. *Journal of Marketing, American Marketing Association*, 2008.
[13] A. Goyal, F. Bonchi, and L. Lakshmanan. Discovering leaders from community actions. In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 499–508. ACM, 2008.
[14] M. U. Ilyas and H. Radha. A KLT-inspired node centrality for identifying influential neighborhoods in graphs. In *Conference on Information Sciences and Systems*, Princeton, NJ, 2010. Princeton University.
[15] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *proceedings of KDD*, 2003.
[16] D. Kempe and F. McSherry. A decentralized algorithm for spectral analysis. *Journal of Computer and System Sciences*, 74(1):70–83, 2008.
[17] M. Kimura, K. Saito, and R. Nakano. Extracting influential nodes for information diffusion on a social network. In *AAAI*, 2007.
[18] M. Kimura, K. Saito, R. Nakano, and H. Motoda. Finding influential nodes in a social network from information diffusion data. *Springer Social Computing and Behavioral Modeling*, 2009.
[19] C. Kohlschütter, P. Chirita, and W. Nejdl. Efficient parallel computation of pagerank. *Lecture Notes in Computer Science*, 3936:241, 2006.
[20] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Nat. Bur. Standards*, 45(4):255–282, 1950.
[21] A. Langville, C. Meyer, and P. FernÁndez. Googles pagerank and beyond: the science of search engine rankings. *The Mathematical Intelligencer*, 30(1):68–69, 2008.
[22] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *proceedings of KDD*, 2007.
[23] P. Marsden. Egocentric and sociocentric measures of network centrality. *Social Networks*, 24(4):407–422, 2002.
[24] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of IMC*, San Diego, CA, October 2007.
[25] B. Ruhnau. Eigenvector-centrality–a node-centrality? *Social networks*, 22(4):357–365, 2000.
[26] K. Sankaralingam, S. Sethumadhavan, and J. Browne. Distributed pagerank for p2p systems. In *12th IEEE International Symposium on High Performance Distributed Computing*, pages 58–68, 2003.
[27] R. Shahbaz. how high is your popularity? *http://www.facebook.com/apps/application.php?id=174042725891*, 2010. 12120 monthly users.
[28] X. Shi, M. Bonner, L. Adamic, and A. Gilbert. The very small world of the well-connected. In *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, pages 61–70. ACM, 2008.
[29] N. R. Suri and Y. Narahari. Determining the top-k nodes in social networks using the shapley value. In *proceedings of AAMAS*, 2008.
[30] C. Wilson, B. Boe, A. Sala, K. Puttaswamy, and B. Zhao. User interactions in social networks and their implications. In *Proceedings of ACM European Conference on Computer Systems*, 2009.
[31] E. Zheleva and L. Getoor. To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles. In *18th International World Wide Web conference (WWW)*, 2009.
[32] F. Zou, Z. Zhang, and W. Wu. Latency-bounded minimum influential node selection in social networks. In *proceedings of WASA*, 2009.