LETTER

Localized Ranking in Social and Information Networks*

Joyce Jiyoung WHANG[†a)], *Member and* Yunseob SHIN[††], *Nonmember*

SUMMARY In social and information network analysis, ranking has been considered to be one of the most fundamental and important tasks where the goal is to rank the nodes of a given graph according to their importance. For example, the PageRank and the HITS algorithms are well-known ranking methods. While these traditional ranking methods focus only on the structure of the entire network, we propose to incorporate a local view into node ranking by exploiting the clustering structure of real-world networks. We develop localized ranking mechanisms by partitioning the graphs into a set of tightly-knit groups and extracting each of the groups where the localized ranking is computed. Experimental results show that our localized ranking methods rank the nodes quite differently from the traditional global ranking methods, which indicates that our methods provide new insights and meaningful viewpoints for network analysis.

key words: *ranking, PageRank, HITS, clustering, network analysis*

1. Introduction

Social and information networks can be modeled as graphs where a node represents an individual or a web page, and an edge represents a social relationship between the individuals or a hyperlink between the web pages. It has been known that each node in a graph plays a different role (or a different level of importance) depending on the node's position in the network [1]. To measure the importance of nodes in a network, a number of centrality measures (e.g., the betweenness centrality [2]) and ranking algorithms have been proposed. Among those, the HITS algorithm [3] and the PageRank method [4] have been considered to be most successful methods. These methods rank the nodes in a graph by assigning a particular value to each node and repeatedly updating the values until convergence by considering the structure of the entire network.

While this *global* view is one that should be considered to evaluate the importance of the nodes, we can have deeper understanding of node centrality by also considering a *local* structure of the network. Based on the fact that real-world social networks and information networks can be decomposed into a set of densely connected subgraphs, we propose the localized ranking algorithms. By partitioning the graphs using graph clustering algorithms, we extract a set of tightly-knit groups from a network and compute *localized ranking* by only focusing on each of the extracted group. The experimental results show that the localized ranking is quite different from the global ranking, which indicates that our localized ranking mechanisms provide meaningful viewpoints for network analysis.

There have been several studies which incorporate the clustering structure of networks into centrality computation or ranking methods. In [5], the local and the community centrality methods have been proposed, and we note that the local centrality is closely related to our localized ranking even though [5] considers the closeness and the betweenness centrality measures whereas we focus on extending node ranking algorithms such as the HITS and the PageRank methods. Also, we consider the node ranking problem for bipartite graphs while [5] focuses on computing the shortest-path-based centrality measures on unipartite graphs. On the other hand, [6] utilizes the local and global social context for recommender systems. In [7], the HITS algorithm is modified to produce customized authority lists by incorporating users' feedback while a context-sensitive PageRank algorithm has been proposed in [8]. While these methods propose to extend the traditional ranking algorithms by augmenting additional information, our method exploits the inherent clustering structure of real-world networks.

We develop our own web crawlers to collect real-world data from the web. We describe our datasets in Sect. 2. Our main algorithms are described in Sect. 3 and the experimental results are shown in Sect. 4. We present our conclusions and future work in Sect. 5.

2. Web Crawling

We develop web crawlers to construct our own datasets by collecting link information among objects from the web. First, we consider a social network by crawling data from Facebook (`www.facebook.com`). On Facebook, users can make friendship relationships with each other, and the users can follow a *page* which is a small online community that contains information (or advertisements) on a specific topic or product, e.g., an entertainer's *page*. We can model the Facebook data as a bipartite graph where two different types of nodes exist – one for an individual and the other for *page*.

Manuscript received August 10, 2017.
Manuscript revised October 23, 2017.
Manuscript publicized November 22, 2017.

[†]The author is with the Department of Computer Science and Engineering, Sungkyunkwan University (SKKU), Korea.

[††]The author is with the Department of Electrical and Computer Engineering, Sungkyunkwan University (SKKU), Korea.

[*]This research was supported by Basic Science Research Program through the NRF of Korea funded by MOE (2016R1D1A1B03934766 and NRF-2010-0020210) and by the National Program for Excellence in SW supervised by the IITP (2015-0-00914).

a) E-mail: jjwhang@skku.edu (Corresponding author)

DOI: 10.1587/transinf.2017EDL8178

By looking at the friendship relationships among the users, we can construct a social network. Note that the edges between the users are undirected, i.e., the friendship relationship is symmetric. Given an ego node, we conduct a breadth-first search up to two-hop distant nodes from the ego node. As a result, we get 128,821 individuals and 4,333,884 edges between the individuals in our dataset. Let $G_s = (\mathcal{V}_s, \mathcal{E}_s)$ denote this graph where $\mathcal{V}_s$ denotes the set of individuals and $\mathcal{E}_s$ denotes the set of edges between them.

On the other hand, we can also add edges between the users and the *pages* by looking at which user follows which *page*. These edges are directed edges because only users can follow the *pages*, whereas the *pages* are not allowed to follow the users on Facebook. We create nodes for the *pages* that receive at least one link from the users included in G_s. As a result, we get 1,367,333 nodes for the *pages* and there are 7,649,773 edges between the users and the *pages*.

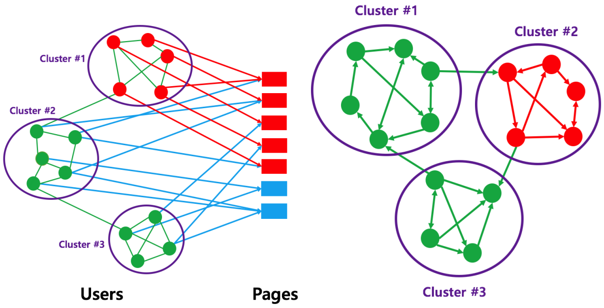
Second, we consider an information network by parsing the link structure between web pages inside Namuwiki (`www.namu.wiki`) which can be considered as a Korean version of Wikipedia. In the Namuwiki web site, we extract a subset of the web pages and construct a subgraph induced by the extracted web pages by parsing the hyperlink information between the web pages. In our dataset, we have 303,221 web pages and there are 8,392,018 edges between them. This can be modeled as a directed graph where each web page corresponds to a node and a hyperlink from a page to another is represented as a directed edge.

3. Localized Ranking

We can decompose a graph into smaller dense groups by graph clustering. If the underlying graph has a modular structure, each cluster might be separable from the rest of the graph. Within each of these separable groups, the nodes can be differently ranked from the case where we compute the ranking by considering the entire network. Based on this idea, we define two localized ranking methods.

3.1 Graph Partitioning

It has been known that social networks and information networks have clustering structures [1]. That is, these networks can be decomposed into a set of smaller cohesive subgroups where the nodes inside each group are densely connected with each other. Such a subgroup is also called as a *cluster* in a graph. By applying graph clustering algorithms such as Graclus [9] or GEM [10] for larger networks, we can partition a given graph into a set of clusters. Formally, given a graph $G = (\mathcal{V}, \mathcal{E})$, a set of k clusters for the graph can be represented as $\mathcal{V}_1, \mathcal{V}_2, \cdots, \mathcal{V}_k$ such that $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2 \cup \cdots \cup \mathcal{V}_k$ and $\forall i \neq j$ $V_i \cap V_j = \emptyset$ where k is the number of clusters. When we consider clustering a social network, each cluster corresponds to a tightly-knit group which can be interpreted as a *community* or a social circle [11].



(a) Localized hubs and authorities (b) Localized PageRank

Fig. 1 Localized ranking.

3.2 Localized Hubs and Authorities

The well-known HITS algorithm [3] is a traditional link analysis algorithm for ranking web pages. The main idea of the algorithm is that if a web page x includes a link to a web page y, then it is considered that the page x confers 'authority' on y. Two scores are computed within the algorithm – *hubs* and *authorities*. A hub score of a node x is computed by adding all the authority scores of the pages that the page x points to, whereas an authority score of a node y is computed by adding the hub scores of the pages that point to y. If a node has a high hub score, it indicates that the node points to the pages that are also pointed by many other pages. On the other hand, if a node has a high authority score, it indicates that the node receives links from many good hubs.

Let us consider our Facebook dataset which we described in Sect. 2. When we focus on the edges between the users and the *pages* on Facebook, we can compute the hub scores for the users and the authority scores for the *pages*. Here, we assume that the users confer authorities on the *pages* by following the *pages*. Then, the *pages* with high authorities can be interpreted as the *pages* that are most popular and reliable *pages*, and thus we might want to recommend those *pages* to the Facebook users. However, these *global* hubs and authorities, i.e., the scores that are computed based on the structure of the entire network, might fail to capture a *local* structure around a user, and thus cannot provide the users with the optimal recommendations. When we consider a *personalized* recommendation for each user, an individual might want to follow a *page* that his or her close friends also follow. Based on this intuition, we compute the localized hubs and authorities by extracting a subgraph for each cluster. First, we partition the social network G_s into k clusters by applying the Graclus graph partitioning method [9]. Then, for each cluster, we extract the *pages* that receive at least one link from the users inside the cluster. Finally, we compute the hubs and authorities on this subgraph. Figure 1 (a) shows a toy example of a subgraph induced by a cluster.

3.3 Localized PageRank

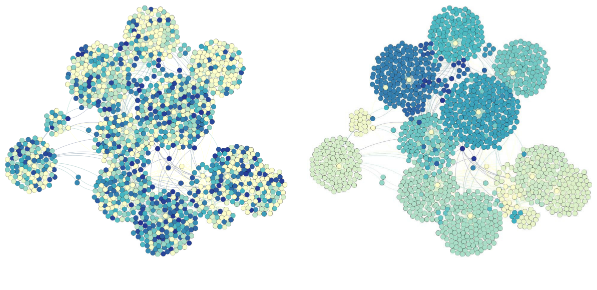
While the HITS algorithm computes two different scores for the nodes, the PageRank algorithm [4] computes one score for every node in a graph. The PageRank value of each web page indicates the importance of the corresponding web page. Starting with the uniform PageRank values for all the nodes in a graph, it is assumed that the PageRank value of a node is evenly divided into each of its out-going links and the new PageRank of a node is computed by adding all the PageRank values that its incoming neighbors confer. By appropriate scaling and normalization, it can be shown that the PageRank vector converges to the left eigenvector of the Google matrix with eigenvalue one [4].

Let us consider our Namuwiki dataset described in Sect. 2. When we simply compute the PageRank on this dataset, we are able to compute the global PageRank ranking which considers the link structure of the entire hyperlink graph. Now, if we recall the concept of clusters, we note that we can group the web pages such that each group contains a set of highly-correlated web pages. Then, one might be more interested in ranking nodes within each group to identify a set of highly ranked web pages within each cluster that might consist of web pages on similar topics. To compute this localized PageRank, we first partition the Namuwiki graph into several clusters, and extract an induced subgraph for each cluster. Then, we compute the localized PageRank on the subgraph for each cluster. Figure 1 (b) shows an example of the induced subgraph for a cluster.

4. Comparison with the Global Ranking

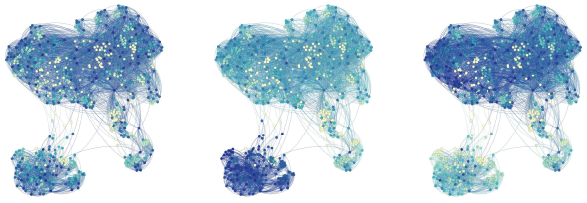
We define two different localized ranking schemes in Sect. 3. If the scores or the ranking derived by the localized ranking methods are *different* from the global scores or the global ranking, it implies that the localized ranking provides us with meaningful information that the traditional global ranking methods fail to capture. Thus, we conduct experiments with our real-world datasets introduced in Sect. 2 to investigate whether the traditional global ranking and the localized ranking are *different* or not. Since we compute the localized scores for each cluster, we compare these scores with the normalized global scores. That is, for each cluster, we extract the global scores for the nodes that belong to the cluster, and normalize the global scores for those nodes so that the sum of within-cluster global scores becomes to one. Also note that the sum of the localized authorities (or the localized PageRank values) is also one.

We visualize the global authority and the localized authority rankings for cluster#55 on the Facebook dataset in Fig. 2[†]. A darker color indicates a higher rank. In the extracted cluster, there are 18 users and 1381 *pages*, and we represent the users using the lightest color in the figure since those nodes do not have the authority scores. In Fig. 2, it is interesting to see that when we rank the nodes by the localized authority scores, similarly ranked nodes are closely located with each other in Fig. 2 (b) whereas we cannot see this pattern in the global authority ranking in Fig. 2 (a).



(a) Global authority ranking (b) Localized authority ranking

Fig. 2 Global and localized authorities on Facebook.



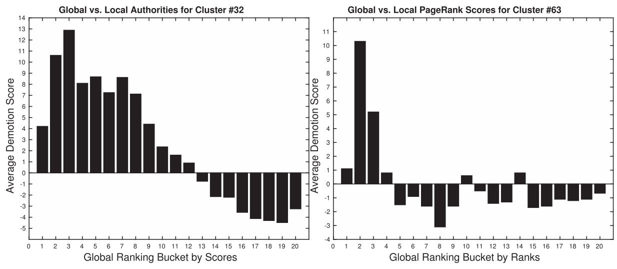
(a) Localized PageRank (b) PPR with node#332 (c) PPR with node#443

Fig. 3 Localized and personalized PageRank on Namuwiki.

We compare our localized PageRank with a personalized PageRank (PPR) [8]. In Fig. 3 (a), we visualize the ranking of the nodes in cluster#23 on the Namuwiki dataset according to the localized PageRank scores (a darker color indicates a higher rank). The localized PageRank represents the importance of each web page by only considering the link structure among the closely related web pages. While the localized PageRank method computes the PageRank scores within each cluster, the PPR method computes a biased PageRank vector for each node. Thus, in PPR, we should specify a particular personalization vector, and a different personalization vector yields a different ranking as shown in Fig. 3 (b) and Fig. 3 (c). Without any prior knowledge, it is hard to choose an appropriate personalization vector for the PPR computation. Even though one can repeatedly compute the PPR with different personalization vectors, how to aggregate and interpret all the different results is not clear. On the other hand, our localized ranking method computes the ranking per cluster instead of per node, which provides an efficient and effective way to conduct a cluster-level analysis.

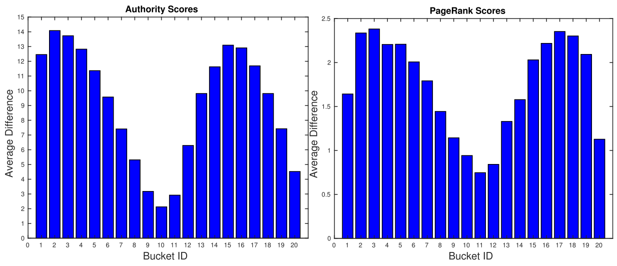
To compare the differences between the global and the localized ranking methods more systematically, we use the notion of *demotion* which has been introduced in [12]. The high-level idea is that we divide the nodes into 20 buckets based on the global ranking, and measure how many changes occur if we redivide the nodes according to the localized ranking. For each cluster, we sort the nodes according to their global scores in descending order. In the first

[†] We draw Fig. 2 and Fig. 3 using Gephi (https://gephi.org/).



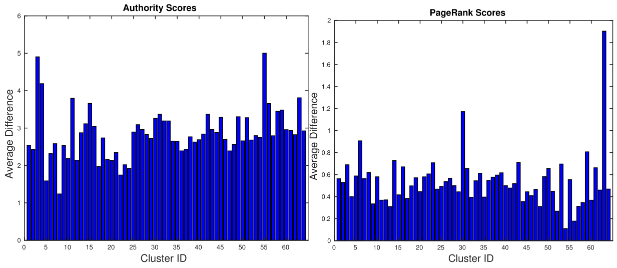
(a) Facebook dataset (b) Namuwiki dataset

Fig. 4 Average demotion scores.



(a) Facebook dataset (b) Namuwiki dataset

Fig. 5 Average difference per bucket.



(a) Facebook dataset (b) Namuwiki dataset

Fig. 6 Average difference per cluster.

bucket, we assign nodes whose global scores sum up to 5% of the total global score. Similarly, by considering the rest of the nodes, we make the second bucket by assigning nodes whose global scores sum up to 10% of the total global score. In this way, we can divide the nodes into 20 buckets by their global scores. Then, we consider a situation where we divide the nodes based on the *localized* scores. For each node, we compare the bucket number by its global score and its localized score. For example, if a node is assigned to the first bucket by the global score, but the node is assigned to the 5th bucket by the localized score, then the node gets demoted four buckets. In this case, the *demotion score* of that node is four. On the other hand, if a node is assigned to the 10th bucket by the global score, and the node is assigned to the 7th bucket by the localized score, the node gets promoted three buckets, and thus, the demotion score is −3.

For each cluster, we compute the average demotion score of each bucket by averaging the demotion scores of all the nodes assigned to the corresponding bucket. Figure 4 (a) shows the average demotion score of each bucket for the authority scores on our Facebook dataset. Since we cluster the graph into 64 clusters, we have 64 plots. Among those, we select the one with the maximum variance between the global and localized authorities. Note that if there is not much difference between the global and the localized authorities, the demotion score should be close to zero. However, as we can see in Fig. 4 (a), the demotion scores we got from our Facebook dataset are significantly greater than (i.e., the nodes get demoted) or significantly less than zero (i.e., the nodes get promoted), which indicates that there is a huge difference between the global authorities and the localized authorities. When we divide the nodes into 20 buckets, we can just focus on the ranks of the nodes instead of their scores. Now, we divide the nodes into 20 buckets based on the ranks such that the first bucket contains the top 5% of the nodes. Then, each bucket contains the same number of nodes (note that when we divide the nodes based on the *scores*, the number of nodes assigned to each bucket might vary since the scores tend to follow a power-law distribution). Figure 4 (b) shows the results for the PageRank scores on the Namuwiki dataset. We see that the demotion scores are not close to zero, which indicates that there is a discrepancy between the global ranking and the localized ranking.

We define the average *difference* of each bucket to be the average of the absolute values of demotion scores (of the nodes in each bucket) to measure the difference between the global ranking and the localized ranking. We take the absolute value of each demotion score because whether a node is demoted or promoted is not our main focus. Figure 5 shows the results. Similarly, we also define the average difference of each *cluster* by considering the average of the absolute values of demotion scores per cluster. Figure 6 shows the results. In Fig. 5 and Fig. 6, we note that the average differences are significantly greater than zero. Also, we observe that the difference between the global and localized ranking is more significant for the authority scores than the PageRank scores. Our experimental results indicate that the localized ranking provides a different view from the traditional global ranking methods, and thus can be utilized for various practical applications in network analysis.

5. Conclusion and Future Work

We develop the localized ranking methods by extending the traditional HITS and the PageRank algorithms. By exploiting the inherent clustering structure of real-world networks, we partition the networks into a set of densely connected subgraphs, and extract each of the subgraphs where we compute the localized ranking for the nodes included in the subgraph. Experimental results show that the localized ranking methods result in different orderings of the nodes from the traditional global ranking methods, and thus our methods provide new insights on node ranking in network analysis.

We plan to extend our methods and analysis by considering the non-exhaustive, overlapping clustering [13]. Also, we intend to incorporate our localized ranking methods into recommender systems [14].

References

[1] D. Easley and J. Kleinberg, Networks, Crowds, and Markets: Reasoning about a Highly Connected World, Cambridge University Press, 2010.

[2] J.-R. Liu, S.-Z. Guo, Z.-M. Lu, F.-X. Yu, and H. Li, "An approximate flow betweenness centrality measure for complex network," IEICE Trans. Inf. & Syst., vol.E96-D, no.3, pp.727–730, 2013.

[3] J.M. Kleinberg, "Authoritative sources in a hyperlinked environment," JACM, vol.46, no.5, pp.604–632, 1999.

[4] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," Computer Networks and ISDN Systems, vol.30, no.1-7, pp.107–117, 1998.

[5] S. Adalı, X. Lu, and M. Magdon-Ismail, "Local, community and global centrality methods for analyzing networks," SNAM, vol.4, no.1, p.210, 2014.

[6] J. Tang, X. Hu, H. Gao, and H. Liu, "Exploiting local and global social context for recommendation," IJCAI, pp.2712–2718, 2013.

[7] H. Chang, D. Cohn, and A. McCallum, "Learning to create customized authority lists," ICML, pp.127–134, 2000.

[8] T.H. Haveliwala, "Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search," IEEE Trans. Knowl. Data Eng., vol.15, no.4, pp.784–796, 2003.

[9] I.S. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors: A multilevel approach," IEEE Trans. Pattern Anal. Mach. Intell., vol.29, no.11, 2007.

[10] J.J. Whang, X. Sui, and I.S. Dhillon, "Scalable and memory-efficient clustering of large-scale social networks," ICDM, pp.705–714, 2012.

[11] J.J. Whang, D.F. Gleich, and I.S. Dhillon, "Overlapping community detection using neighborhood-inflated seed expansion," IEEE Trans. Knowl. Data Eng., vol.28, no.5, pp.1272–1284, 2016.

[12] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen, "Combating web spam with trustrank," VLDB, pp.576–587, 2004.

[13] J. Whang, D. Gleich, and I. Dhillon, "Non-exhaustive, overlapping *k*-means," SDM, pp.936–944, 2015.

[14] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," Computer, vol.42, no.8, pp.30–37, 2009.