Estimating Random Walk Centrality in Networks

Brad C. Johnson[a,*], Steve Kirkland[b]

[a] *Department of Statistics, University of Manitoba, Winnipeg, Manitoba, Canada*

[b] *Department of Mathematics, University of Manitoba, Winnipeg, Manitoba, Canada*

Abstract

In this paper the random walk centrality (equivalently, the accessibility index) for the states of a time-homogenous irreducible Markov chain on a finite state space is considered. It is known that the accessibility index for a particular state can be written in terms of the first and second moments of the first return time to that state. Based on that observation, the problem of estimating the random walk centrality of a state is approached by taking realizations of the Markov chain, and then statistically estimating the first two moments of the corresponding first return time. In addition to the estimate of the random walk centrality, this method also yields the standard error, the bias and a confidence interval for that estimate. For the case that the directed graph of the transition matrix for the Markov chain has a cut-point, an alternate strategy for computing the random walk centrality is outlined that may be of use when the centrality values are of interest for only some of the states. In order to illustrate the effectiveness of the results, estimates of the random walk centrality arising from random walks for several directed and undirected graphs are discussed.

Keywords: Random walk centrality; Network centrality; Accessibility index; Markov chains; Mean first passage times; Bootstrap

1. Introduction and preliminaries

Let $\{X_n : n \geq 0\}$ be a finite irreducible Markov chain on the state space $\Omega = \{1, \ldots, N\}$ with transition probability matrix $\mathbf{T}$. Define $\tau_i(k)$ by

$$\tau_i(k) := \inf\{n > 0 : X_n = k \mid X_0 = i\}, \qquad k \in \Omega,$$

as the first passage time from state i to state k. For each $i, k \in \Omega$, we denote the mean first passage time from state i to state k by $m_{i,k} = \mathbb{E}[\tau_i(k)]$. Letting $\boldsymbol{w}$ be the stationary distribution vector for the Markov chain, it is well–known that $m_{i,i} = 1/w_i$ for each $i \in \Omega$.

*Corresponding author: Department of Statistics, 318 Machray Hall, University of Manitoba, Winnipeg, Manitoba, R3T 2N2, Canada; E-mail: brad.johnson@umanitoba.ca

Fix a state $k \in \Omega$. The accessibility index for state k, denoted by α_k, is given by

$$\alpha_k = \sum_{i=1}^{N} w_i m_{i,k},$$

and measures the expected time to reach state k from a randomly chosen initial state (that is, randomly chosen according to the stationary distribution). Thus the α_ks provide a centrality measure for the states of the Markov chain – low values of α_k correspond to states that are easy to reach, while high values of α_k correspond to states that are not so easy to reach. Indeed, in Noh and Rieger (2004), the authors introduce a measure of vertex centrality for undirected graphs: they consider the simple random walk on a connected, finite, undirected, non–bipartite graph and, for each vertex k, introduce the so–called random walk centrality for vertex k, which we denote here by θ_k. In Kirkland (2016), it is shown that in Noh and Reiger's setting, $\theta_k = 1/\alpha_k$ for each vertex in the graph. Thus the accessibility index can be seen as an extension of the notion of random walk centrality to irreducible Markov chains.

How might one compute the α_ks? On the face of it, the computation might be costly and time–consuming, as both the mean first passage times and the entries in the stationary distribution would need to be found first. However, an observation in Kirkland (2016) suggests a statistical approach to estimating the accessibility indices. In order to outline that approach, we need a little more notation. Define $\tau(k;0)$ as the time of the first visit to state k and, for $j \in \mathbb{N}$,

$$\tau(k;j) := \inf\{n > \tau(k;j-1) : X_n = k\},$$

so that $\tau(k;j)$ is the time of the j–th return to state k. Finally, for $k \in \Omega$ and $j \in \mathbb{N}$, define

$$R_{k,j} = \tau(k;j) - \tau(k;j-1).$$

That is, $R_{k,j}$ is the j–th inter-arrival time between visits to state k. Given k, the $R_{k,j}$ are independent and identically distributed random variables and are denoted, generically, as R_k. Theorem 1.1 of Kirkland (2016) shows that the accessibility index α_k may be written as

$$\alpha_k = \frac{1}{2}\left(\frac{\mathbb{E}(R_k^2)}{\mathbb{E}(R_k)} - 1\right). \tag{1.1}$$

Equation (1.1) informs our approach to estimating the α_ks: by taking a realization of the Markov chain, we can produce estimates of both $\mathbb{E}(R_k)$ and $\mathbb{E}(R_k^2)$ in order to estimate α_k. We observe that this strategy may offer an advantage in the situation that the state space is large but one is only interested in the accessibility indices for a small number of states, or where explicit computation of the stationary distribution and/or the mean first passage times is prohibitively expensive or numerically unstable.

Given a sequence $R_{k,1}, \ldots, R_{k,M}$, a ratio type estimate of α_k is given by

$$\hat{\alpha}_k = \frac{1}{2}\left(\frac{\sum_{j=1}^{M} R_{k,j}^2}{\sum_{j=1}^{M} R_{k,j}} - 1\right). \tag{1.2}$$

For the corresponding random walk centrality measure $\theta_k = 1/\alpha_k$, the plug in estimate is simply

$$\hat{\theta}_k = \frac{1}{\hat{\alpha}_k}.$$

For a large integer L, let $\boldsymbol{x} = (x_1, \ldots, x_L)$ be a realization of a random walk of length L on $\{X_n : n \geq 0\}$ starting at an arbitrary state $X_0 = x_0 = \omega$. Then, for each $k \in \Omega$, we also obtain a vector of ($\nu_k \geq 0$) realizations of R_k (say $\boldsymbol{r}_k = (r_{k,1}, \ldots, r_{k,\nu_k})$). In *R* (R Core Team, 2016), this is easily accomplished by the command

$$\boldsymbol{r}_k = \text{diff}(\text{which}(\boldsymbol{x} == k)).$$

Provided that the number of visits ν_k to state k in $\boldsymbol{x}$ is reasonably large, a (consistent) estimate of α_k is given by

$$\hat{\alpha}_k = \frac{1}{2}\left\{\frac{\sum_{i=1}^{\nu_k} r_{k,i}^2}{\sum_{i=1}^{\nu_k} r_{k,i}} - 1\right\}.$$

Note that, since $\{X_n\}$ is recurrent, we have $\nu_k \to \infty$ as $L \to \infty$ for all k. Furthermore, since $\mathbb{E}(R_k^m)$ exists for all $m \in \mathbb{N}$ and $k \in \Omega$, we have, $\lim_{L\to\infty} \nu_k^{-1} \sum_{i=1}^{\nu_k} R_{k,i}^m \to \mathbb{E}(R_k^m) > 0$ almost surely and it follows by (for example) Slutsky's Theorem, that $\hat{\alpha}_k$ is consistent for α_k (and that $\hat{\theta}_k$ is consistent for θ_k).

The estimator $\hat{\alpha}_k$ suffers from two sources of bias. The first source is from the fact that $\hat{\alpha}_k$ is a non-linear ratio type estimator. This bias can, at least, be estimated using either the common jackknife or bootstrap methods (see Efron and Tibshirani, 1994, Chapters 12–14, for example). The second source of bias is due to the fact that, by using random walks, we are essentially sampling from a truncated distribution of R_k because inter-arrival times longer than the walk length are impossible. Longer random walks should mitigate this bias. In practice, we perform a moderate number, say N_w, of independent random walks, each of length L, with random starting points.

In this paper we focus on strongly connected directed and undirected graphs, with the object of estimating the accessibility indices (equivalently, the random walk centralities) for the vertices using the technique described above. Section 2 presents some numerical results that illustrate our technique. In Section 3 we prove a theoretical result for graphs with a cut–point that allows one to compute mean first passage times, stationary distribution entries and accessibility indices by working with the corresponding quantities for Markov chains on a smaller state space. Again this result may offer some advantage in the setting where one is only interested in these quantities for a subset of states.

2. Numerical Examples

To illustrate some of the results presented in this paper we make use of three commonly available networks. In practice, identifying the most important states is of particular interest and, to this end, for each of these networks, we performed the following steps:

Step 1: $N_W = 10$ random walks are performed on the network, from starting points chosen uniformly at random. Each random walk is extended in increments of 10^4 steps until a given proportion (say p_{nodes}) of the nodes (states) in the network are visited at least n_{visits} times.

Step 2: For each state k that was visited at least n_{visits} times, the inter-arrival times were determined as $\boldsymbol{r}_k = (r_{k,1}, \ldots, r_{k,\nu_k})$, where ν_k is the total number of inter-arrival times for state k, and point estimates of θ_k were determined. The set S of the 100 nodes with the largest values of $\hat{\theta}_k$ were identified.

Step 3: For each state $k \in S$, a bootstrap procedure with 1000 replicates was performed to estimate θ_k ($\hat{\theta}_k$) and w_k ($\hat{w}_k = \nu_k / \sum_{j=1}^{\nu_k} r_{k,j}$), along with estimates of the standard errors, relative biases and coefficients of variation for these estimates.

In what follows, timings are reported as well as information regarding each step for each network analyzed. Timings are for a Mac Mini with a 2.6GHz Intel Core i7 processor and 8 GB memory using R (R Core Team, 2016) and the `igraph` package (Csardi and Nepusz, 2006).

2.1. The Gnutella peer-to-peer network of August 30, 2012

The Gnutella peer-to-peer network from August 30, 2012 (`Gnutella30`) can be found at the Stanford Large Network Dataset Collection (SNAP) (Leskovec and Krevl, 2014). It consists of consists of 36,682 nodes and 88,328 edges. The largest strongly connected component consists of 8490 nodes and 31706 edges and we restrict our attention to this component. Figure 2.1 shows the matrix plot of the adjacency matrix. The tick marks denote the 20 most important states according to the true θ_k, where darker ticks represent more important states (some tick marks are coincident due to the large number of states).

Other common measures of centrality include $w_k = 1/E(R_k)$, closeness (C_s), betweenness (C_b), in-degree (D_i) and weighted in-degree (D_w), which is the column sum of the transition probability matrix. Kendall's correlation coefficients between the θ_k and these measures are

	w_k	C_s	C_b	D_i	D_w
$\tau(\theta_k, \cdot)$	0.999	0.340	0.479	0.534	0.538

.

Figure 2.2 illustrates the number of random walk steps (in increments of 10^4 steps) and the time required to visit (at least) a given proportion of nodes in the network at least 50 times, based

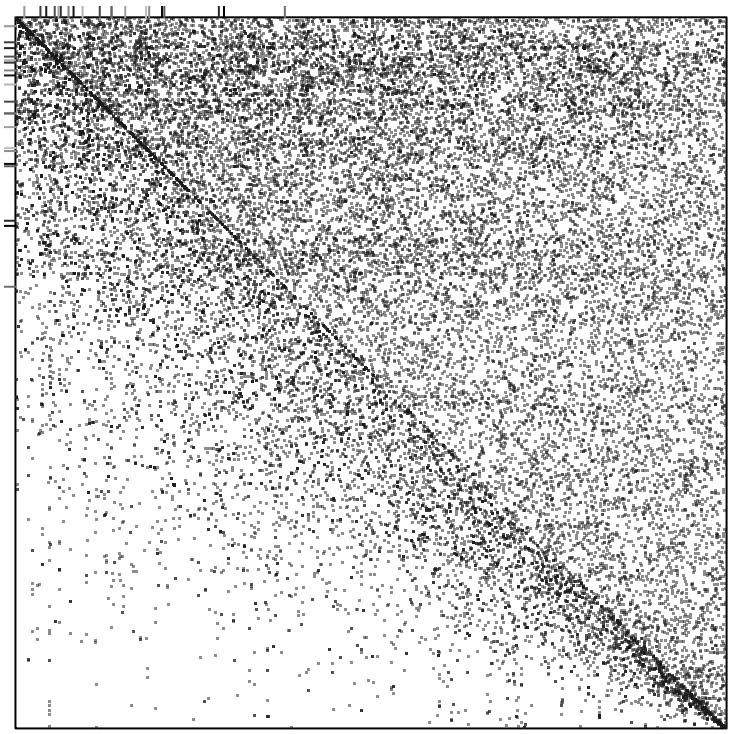

Figure 2.1: Plot of the adjacency matrix for the `Gnutella30` network.

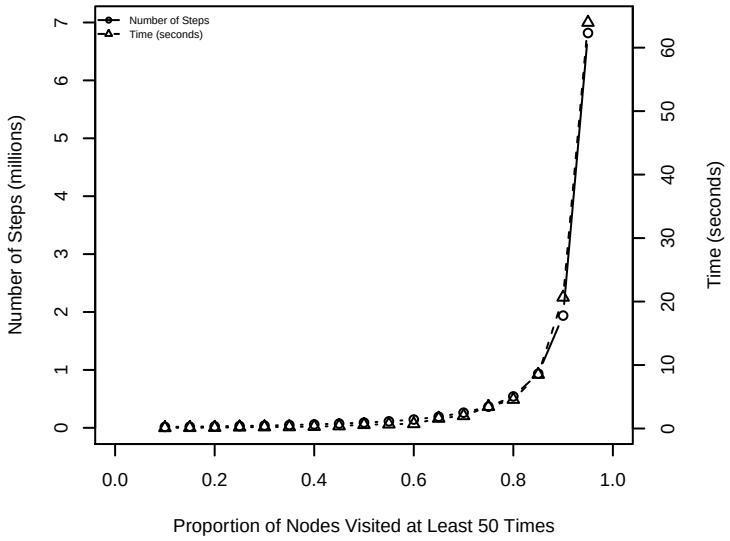

Figure 2.2: The number of random walk steps (millions) and time (seconds) required to visit a given proportion of states $(0.1, 0.15, \ldots, 0.95)$ at least 50 times in the `Gnutella30` network.

Table 2.1: Top 15 nodes in the `Gnutella30` Network with $N_W = 10$, $p_{nodes} = 0.5$ and $n_{visits} = 50$ according to θ_k with the estimated $\hat{\theta}_k$, estimated coefficients of variation and estimated relative bias.

Node ID	$\theta_k \times 10^4$	Rank(θ_k)	$\hat{\theta}_k \times 10^4$	Rank($\hat{\theta}_k$)	$\widehat{\text{cv}}(\hat{\theta}_k)$	$\widehat{\text{rel-bias}}(\hat{\theta}_k)$
1877	25.410	1	25.520	1	0.0091	0.0139
5383	16.087	2	15.858	2	0.0119	0.0177
876	15.396	3	15.496	3	0.0122	0.0184
9645	13.690	4	13.506	4	0.0121	0.0186
1423	13.172	5	13.217	5	0.0117	0.0182
3373	12.426	6	12.355	6	0.0141	0.0208
1504	12.370	7	12.235	7	0.0130	0.0200
6076	12.070	8	12.084	8	0.0136	0.0204
10869	11.769	9	11.846	10	0.0145	0.0211
12792	11.738	10	11.903	9	0.0125	0.0195
4714	11.724	11	11.595	11	0.0142	0.0211
2297	11.641	12	11.518	12	0.0143	0.0216
1120	11.485	13	11.456	13	0.0148	0.0221
4066	11.141	14	11.062	15	0.0135	0.0204
1474	10.995	15	11.085	14	0.0134	0.0206

on an average of 20 replicates starting at randomly chosen nodes. It can be seen that the time required to visit a moderate proportion of nodes at least 50 times is not onerous.

An experiment was conducted with $N_W = 10$, $p_{nodes} = 0.5$ and $n_{visits} = 50$. For this network, Step 1 required 6.1 seconds, Step 2 required 570 seconds and Step 3 required 56.6 seconds. The average walk length for the $N_W = 10$ walks was 899,910 steps.

Table 2.1 shows the 15 nodes in the network with the highest θ_k, the associated estimates $\hat{\theta}_k$, their ranks, estimated coefficients of variation and relative biases (as estimated by the bootstrap procedure). The top 15 nodes were correctly identified with the top 8 nodes in their correct ordering. The estimated coefficients of variation are relatively small and the estimated relative biases (as measured by the bootstrap procedure) are also small. For the top 100 nodes identified in Step 3, the Kendall's τ correlation coefficient between θ_k and $\hat{\theta}_k$ was $\tau(\theta_k, \hat{\theta}_k) = 0.936$.

Figure 2.3 shows the estimates $\hat{\theta}_k$ relative to the θ_k as well as the estimates $\hat{w}_k$ relative to w_k and plots of their coefficients of variation for the top 100 $\hat{\theta}_k$. Most of the $\hat{\theta}_k$ are within 2.5% of the actual θ_k (dashed line), as are the $\hat{w}_k$.

This experiment was repeated with $p_{nodes} = 0.25$ and $n_{visits} = 50$, where the timings (in seconds) were 3.59, 372 and 35.8 for Steps 1, 2 and 3 respectively. The rankings of the top 15 nodes are given in Table 2.2. The top 13 nodes were identified in the correct order. As expected, the estimated coefficients of variation and relative biases are slightly higher in this case but still acceptable.

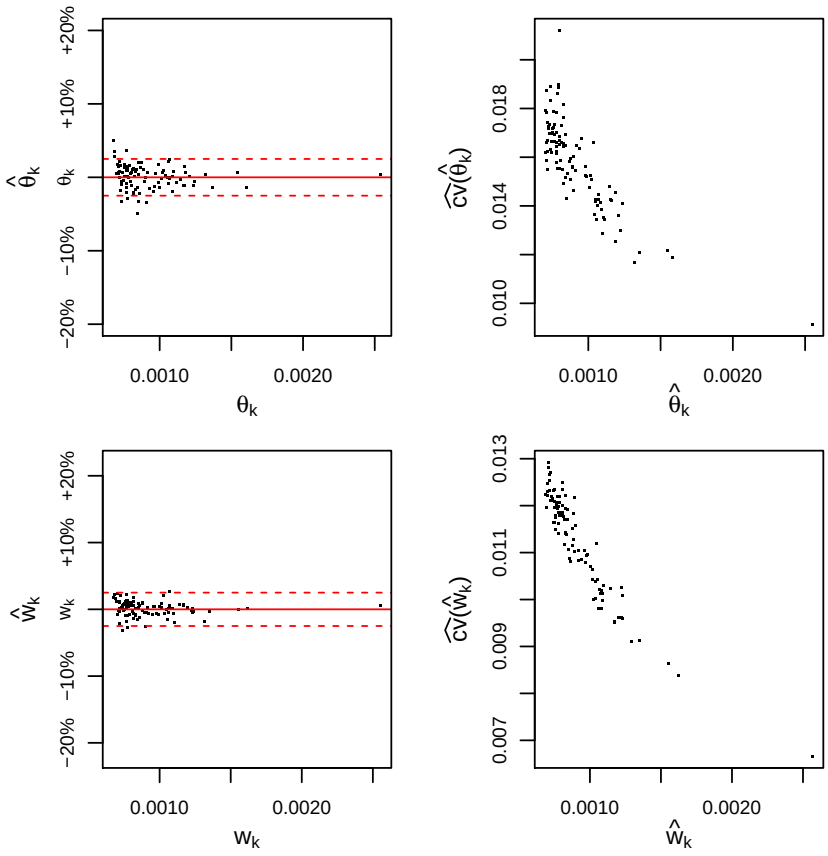

Figure 2.3: Summary of the estimates and coefficients of variation for $\hat{\theta}_k$ and $\hat{w}_k$ (relative to θ_k and w_k) for the top 100 nodes in the `Gnutella30` network. The left panels plot the parameter estimates relative to the true parameter values, where the horizontal dashed lines represent $\pm 2.5\%$. The right panels plot the estimated coefficients of variation of the parameter estimates.

Table 2.2: Top 15 nodes in the `Gnutella30` Network with $N_W = 10$, $p_{nodes} = 0.25$ and $n_{visits} = 50$ according to θ_k with the estimated $\hat{\theta}_k$, estimated coefficients of variation and estimated relative bias.

Node ID	$\theta_k \times 10^4$	Rank(θ_k)	$\hat{\theta}_k \times 10^4$	Rank($\hat{\theta}_k$)	$\widehat{\text{cv}}(\hat{\theta}_k)$	$\widehat{\text{rel-bias}}(\hat{\theta}_k)$
1877	25.410	1	25.525	1	0.0155	0.0241
5383	16.087	2	15.643	2	0.0194	0.0295
876	15.396	3	15.464	3	0.0201	0.0304
9645	13.690	4	13.537	4	0.0202	0.0312
1423	13.172	5	13.033	5	0.0194	0.0303
3373	12.426	6	12.345	6	0.0241	0.0350
1504	12.370	7	12.186	7	0.0221	0.0346
6076	12.070	8	12.087	8	0.0247	0.0361
10869	11.769	9	11.970	9	0.0207	0.0320
12792	11.738	10	11.867	10	0.0220	0.0342
4714	11.724	11	11.495	11	0.0251	0.0365
2297	11.641	12	11.401	12	0.0246	0.0359
1120	11.485	13	11.347	13	0.0236	0.0357
4066	11.141	14	10.764	21	0.0248	0.0378
1474	10.995	15	11.172	14	0.0218	0.0343

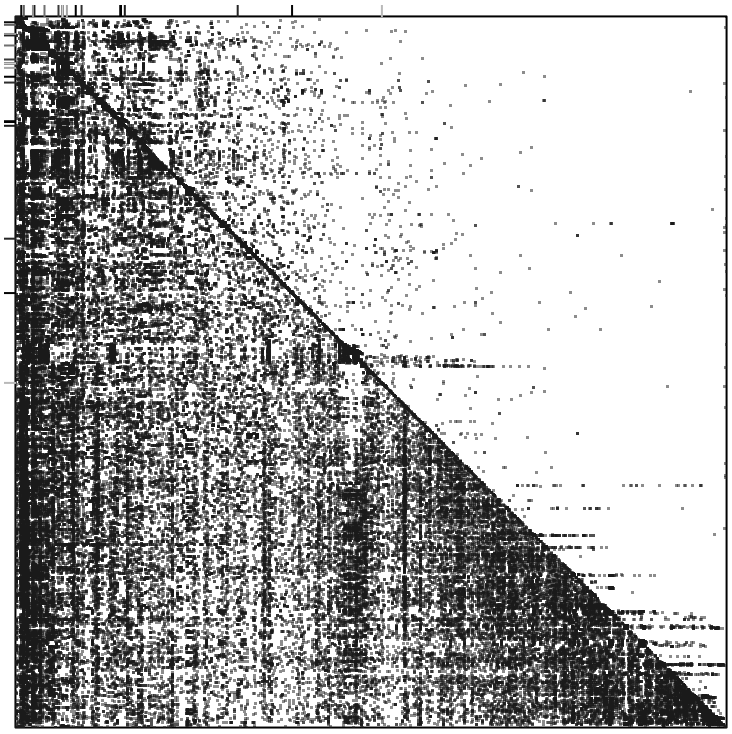

Figure 2.4: Adjacency matrix plot for the `HEP-TH` network.

2.2. High-energy physics theory citation Network

The arXiv (`arXiv.org`) high-energy physics theory citation network (`HEP-TH`) consists of 27,770 nodes and 352,807 edges covering papers in the period from January 1993 to April 2003 (Leskovec and Krevl, 2014). The largest strongly connected component consists of 7,464 nodes and 116,268 edges and we restrict our attention to this component. Figure 2.4 plots the adjacency matrix for this network. The tick marks denote the 20 most important states according to the true θ_k, where darker ticks represent more important states (some tick marks are coincident due to the large number of states) This network is difficult to search via random walks due to the temporal nature of citations and the length of walks required to visit a moderate number of nodes a moderate number of times can exhaust computer memory. For this network we performed $N_W = 10$ random walks such that each walk visited at least 1,000 nodes ($p_{nodes} = 1000/7464$) at least twice ($n_{visits} = 2$). Step 1 took 6.57 seconds, Step 2 took 781 seconds and Step 3 took 473 seconds so that the entire analysis took approximately 21 minutes. The mean walk length required for Step 1 was 762,924. The Kendall's correlation coefficients between the θ_k and other centrality measures for this network are

	w_k	C_s	C_b	D_i	D_w
$\tau(\theta_k, \cdot)$	0.999	0.208	0.183	0.462	0.441

.

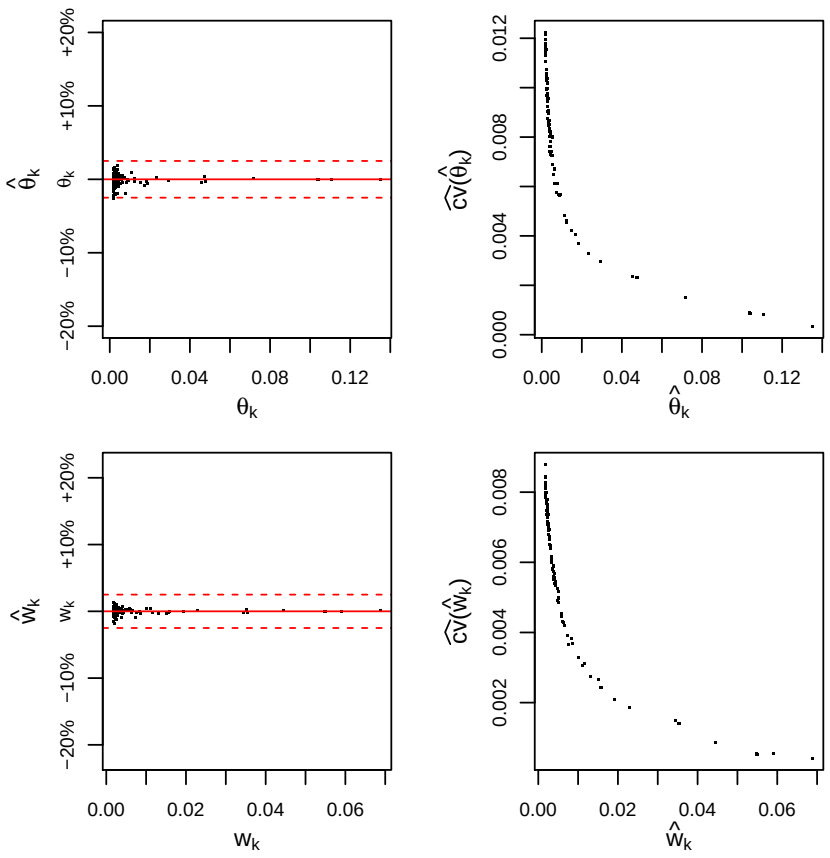

Figure 2.5: Summary of the estimates and coefficients of variation for $\hat{\theta}_k$ and $\hat{w}_k$ (relative to θ_k and w_k) for the top 100 nodes in the `HEP-TH` network. The left panels plot the parameter estimates relative to the true parameter values, where the horizontal dashed lines represent $\pm 2.5\%$. The right panels plot the estimated coefficients of variation of the parameter estimates.

Figure 2.5 shows the estimates $\hat{\theta}_k$ relative to the θ_k as well as the estimates $\hat{w}_k$ relative to w_k and plots of their coefficients of variation for the top 100 $\hat{\theta}_k$. Most of the $\hat{\theta}_k$ are within 2.5% of the actual θ_k (dashed line), as are the $\hat{w}_k$.

Table 2.3 shows the 15 most important nodes in the `HEP-TH` network, ranked according to the θ_k. It also shows the estimates $\hat{\theta}_k$ for these nodes as well as the estimated coefficients of variation and relative bias. Each of the nodes in this table were visited over 100,000 times. It can be seen that the estimates $\hat{\theta}_k$, in this case, correctly identified these top nodes in the correct order. The coefficients of variation and the relative biases of these estimates are quite small.

Table 2.3: Top 15 nodes in the `HEP-TH` Network with $N_W = 10$, $p_{nodes} = 1000/7464$ and $n_{visits} = 2$ with the estimated $\hat{\theta}_k$, estimated coefficients of variation and estimated relative bias.

Node ID	$\theta_k \times 10^4$	Rank(θ_k)	$\hat{\theta}_k \times 10^4$	Rank($\hat{\theta}_k$)	$\widehat{\text{cv}}(\hat{\theta}_k)$	$\widehat{\text{rel-bias}}(\hat{\theta}_k)$
9509140	1351.832	1	1352.155	1	0.0003	0.0007
9605009	1105.776	2	1104.918	2	0.0008	0.0013
9703196	1040.768	3	1040.206	3	0.0009	0.0013
9611132	1036.664	4	1036.153	4	0.0009	0.0014
9612215	1036.238	5	1035.754	5	0.0009	0.0013
9701025	717.530	6	718.407	6	0.0015	0.0022
9601023	478.574	7	477.234	7	0.0023	0.0034
9907085	471.740	8	473.616	8	0.0023	0.0034
9912210	456.933	9	455.225	9	0.0023	0.0035
9702163	295.403	10	295.037	10	0.0030	0.0044
9701125	235.248	11	235.780	11	0.0033	0.0049
9701151	186.426	12	185.268	12	0.0037	0.0055
9702101	184.217	13	183.602	13	0.0037	0.0055
9711200	172.097	14	170.676	14	0.0041	0.0061
9703040	150.900	15	150.274	15	0.0042	0.0063

Table 2.4: Result summary for top 15 identified nodes in the `FacebookNO` network with $N_W = 10$, $p_{nodes} = 2000/59691$ and $n_{visits} = 100$ with the estimated $\hat{\theta}_k$, estimated coefficients of variation and estimated relative bias.

Node ID	$\hat{\theta}_k \times 10^4$	$\widehat{\text{cv}}(\hat{\theta}_k)$	$\widehat{\text{rel-bias}}(\hat{\theta}_k)$
2146	6.613	0.0145	0.0220
430	5.975	0.0149	0.0227
508	5.554	0.0166	0.0254
2136	4.913	0.0180	0.0267
22	4.774	0.0177	0.0269
411	4.768	0.0172	0.0258
9162	4.598	0.0169	0.0262
1338	3.801	0.0205	0.0311
3627	3.794	0.0211	0.0310
2174	3.651	0.0210	0.0315
1751	3.501	0.0210	0.0310
79	3.445	0.0192	0.0294
257	3.435	0.0197	0.0304
1002	3.409	0.0214	0.0330
4695	3.393	0.0205	0.0317

2.3. Facebook New Orleans Network

The Facebook New Orleans Network (`FacebookNO`) (Viswanath et al., 2009) has 63,731 nodes and 1,545,686 edges (available at `http://socialnetworks.mpi-sws.org/data-wosn2009.html`). The largest strongly connected component has 59,691 and 1,456,818 edges and we restrict our attention to this component. As in Viswanath et al. (2009), we treat the network as directed even though it is undirected. Due to the large number of nodes in this network, calculation of the true θ_k and w_k was not possible. Using $p_{nodes} = 2000/59691$ and $n_{visits} = 100$, so that, in each walk, at least 2,000 states are visited at least 100 times, the timings were 22.6, 188.4 and 27.313 seconds for steps 1, 2 and 3 respectively. The average walk length for this example was 1,282,872 steps. The 15 nodes with the largest $\hat{\theta}_k$ are given in Table 2.4 along with the estimated coefficients of variation and relative biases, which are small.

The above experiment was repeated using $p_{nodes} = 0.75$ and $n_{visits} = 50$. The average walk length was 21,772,822. The timings for this experiment were 332, 5258 and 389 seconds for steps 1, 2 and 3 respectively (approx 99 minutes total). The results for the top 15 nodes as determined by $\hat{\theta}_k$, are given in Table 2.5. The top 15 identified nodes in Tables 2.4 and 2.5 are the same but the order has changed in the smaller experiment.

Repeating this experiment again with a different random number seed identified the same top 15 nodes in the same order. Since the maximum absolute differences between the $\hat{\theta}_k$'s is 8.8×10^{-6} and the maximum absolute differences between the coefficients of variation is 7.7×10^{-4}, we can be fairly confident that we have identified the important nodes in this network. The smaller experiment in Table 2.4 illustrates the inherent problems with shorter random walks (and sample sizes) in terms

Table 2.5: Result summary for top 15 identified nodes in the `FacebookNO` network with $N_W = 10$, $p_{nodes} = 0.75$ and $n_{visits} = 50$ with the estimated $\hat{\theta}_k$, estimated coefficients of variation and estimated relative bias.

Node ID	$\hat{\theta}_k \times 10^4$	$\widehat{\text{cv}}(\hat{\theta}_k)$	$\widehat{\text{rel-bias}}(\hat{\theta}_k)$
2146	6.676	0.0037	0.0055
430	5.679	0.0040	0.0059
508	5.522	0.0041	0.0063
2136	4.924	0.0046	0.0069
411	4.689	0.0044	0.0067
22	4.670	0.0044	0.0066
9162	4.549	0.0047	0.0070
1338	3.785	0.0048	0.0072
3627	3.756	0.0050	0.0074
2174	3.720	0.0049	0.0075
1751	3.505	0.0054	0.0081
79	3.448	0.0051	0.0076
1002	3.403	0.0054	0.0081
4695	3.387	0.0052	0.0078
257	3.383	0.0051	0.0076

of variation and relative bias.

3. Random walk centrality in the presence of a cut-point

Suppose that we have a strongly connected directed graph D on nodes labelled $1, \ldots, N$. Fix a node $j = 1, \ldots, N$, and let $D \setminus \{j\}$ denote the directed graph formed by deleting node j and all edges incident with it. We say that node j is a *cut–point* if $D \setminus \{j\}$ has $k \geq 2$ strongly connected components such that there are no paths between nodes in different strongly connected components. While it is certainly not the case that every connected graph contains a cut-point, graphs with cut-points certainly arise in applied settings. For instance there is an interest in the existence and location of cut-points in wireless networks, and there are a number of algorithms for identifying cut-points in such networks (see, for example Dagdeviren and Akram, 2014; Stojmenovic et al., 2011; Xiong and Li, 2010). Similarly, connected scale-free networks contain large numbers of low degree vertices, and are likely to have many vertices of degree 1, which are necessarily adjacent to cut-points. For a graph that contains a cut-point, the following algorithm shows how we can decouple the graph into a number of graphs of smaller order so as to compute the random walk centralities of the corresponding vertices.

If we have an $N \times N$ irreducible stochastic matrix $\mathbf{T}$ whose directed graph has node N (say)

as a cut–point, then we may write $\mathbf{T}$ in the following form:

$$\mathbf{T} = \left[\begin{array}{ccccc|c} \mathbf{T}_{1,1} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{T}_{1,k+1} \\ \mathbf{0} & \mathbf{T}_{2,2} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{T}_{2,k+1} \\ \vdots & & \ddots & & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{T}_{k,k} & \mathbf{T}_{k,k+1} \\ \hline \mathbf{T}_{k+1,1} & \mathbf{T}_{k+1,2} & \dots & & \mathbf{T}_{k+1,k} & t_{k+1,k+1} \end{array}\right], \tag{3.1}$$

where, for each $j = 1, \dots, k$, $\mathbf{T}_{j,j}$ is a square stochastic matrix of order m_j whose largest eigenvalue is real and less than 1 and $\mathbf{T}_{j,k+1}$ and $\mathbf{T}_{k+1,j}$ are column and row vectors (respectively) of order m_j. Note that $t_{k+1,k+1}$ is a scalar, while $\mathbf{0}$ represents a zero matrix whose dimensions are clear from the context.

Let $\boldsymbol{w}$ denote the stationary distribution vector for $\mathbf{T}$, partitioned conformally with (3.1) as

$$\boldsymbol{w}^\intercal = \left[\begin{array}{cccc|c} \boldsymbol{w}_1^\intercal & \boldsymbol{w}_2^\intercal & \dots & \boldsymbol{w}_k^\intercal & w_{k+1} \end{array}\right]. \tag{3.2}$$

For each $j = 1, \dots, k$, let $r_j = \frac{w_{k+1}}{1 - \boldsymbol{w}_j^\intercal \boldsymbol{e}}$, where $\boldsymbol{e}$ is an all–ones vector of the appropriate order, and define $T_{(j)}$ by

$$\mathbf{T}_{(j)} = \left[\begin{array}{c|c} \mathbf{T}_{j,j} & \mathbf{T}_{j,k+1} \\ \hline r_j \mathbf{T}_{k+1,j} & 1 - r_j \mathbf{T}_{k+1,j} \boldsymbol{e} \end{array}\right]. \tag{3.3}$$

We find readily that each $\mathbf{T}_{(j)}$ is irreducible, stochastic, $(m_j + 1) \times (m_j + 1)$ and has stationary vector given by

$$\boldsymbol{v}_j^\intercal = \left[\begin{array}{c|c} \boldsymbol{w}_j^\intercal & 1 - \boldsymbol{w}_j^\intercal \boldsymbol{e} \end{array}\right]. \tag{3.4}$$

Define $\mathbf{M}_\mathbf{T}$ to be the matrix of mean first passage times associated with $\mathbf{T}$, where we take the convention that the diagonal entries of $\mathbf{M}_\mathbf{T}$ are zero; define $\mathbf{M}_{\mathbf{T}_{(j)}}$ analogously, for each $j = 1, \dots, k$. Partition each $\mathbf{M}_{\mathbf{T}_{(j)}}$ conformally with (3.3) as

$$\mathbf{M}_{\mathbf{T}_{(j)}} = \left[\begin{array}{c|c} \mathbf{A}_j & \boldsymbol{a}_j \\ \hline \boldsymbol{b}_j^\intercal & 0 \end{array}\right]. \tag{3.5}$$

Then according to Theorem 6.4.12 in Kirkland and Neumann (2013), we can write $\mathbf{M}_\mathbf{T}$, partitioned conformally with (3.1) as follows:

$$\mathbf{M}_\mathbf{T} = \left[\begin{array}{ccccc|c} \mathbf{A}_1 & \boldsymbol{a}_1\boldsymbol{e}^\intercal + \boldsymbol{e}\boldsymbol{b}_2^\intercal & \boldsymbol{a}_1\boldsymbol{e}^\intercal + \boldsymbol{e}\boldsymbol{b}_3^\intercal & \dots & \boldsymbol{a}_1\boldsymbol{e}^\intercal + \boldsymbol{e}\boldsymbol{b}_k^\intercal & \boldsymbol{a}_1 \\ \boldsymbol{a}_2\boldsymbol{e}^\intercal + \boldsymbol{e}\boldsymbol{b}_1^\intercal & \mathbf{A}_2 & \boldsymbol{a}_2\boldsymbol{e}^\intercal + \boldsymbol{e}\boldsymbol{b}_3^\intercal & \dots & \boldsymbol{a}_2\boldsymbol{e}^\intercal + \boldsymbol{e}\boldsymbol{b}_k^\intercal & \boldsymbol{a}_2 \\ \vdots & & \ddots & & \vdots & \vdots \\ \boldsymbol{a}_k\boldsymbol{e}^\intercal + \boldsymbol{e}\boldsymbol{b}_1^\intercal & \boldsymbol{a}_k\boldsymbol{e}^\intercal + \boldsymbol{e}\boldsymbol{b}_2^\intercal & \dots & \boldsymbol{a}_k\boldsymbol{e}^\intercal + \boldsymbol{e}\boldsymbol{b}_{k-1}^\intercal & \mathbf{A}_k & \boldsymbol{a}_k \\ \hline \boldsymbol{b}_1^\intercal & \boldsymbol{b}_2^\intercal & \dots & & \boldsymbol{b}_k^\intercal & 0 \end{array}\right]. \tag{3.6}$$

From (3.4) and (3.5), we find that

$$\boldsymbol{v}_j^{\intercal}\mathbf{M}_{\mathbf{T}_{(j)}} = \left[\ \boldsymbol{w}_j^{\intercal}\mathbf{A}_j + (1-\boldsymbol{w}_j^{\intercal}\boldsymbol{e})\boldsymbol{b}_j^{\intercal}\ \middle|\ \boldsymbol{w}_j^{\intercal}\boldsymbol{a}_j\ \right]. \tag{3.7}$$

Similarly, we find from (3.2) and (3.6) that the subvector of $\boldsymbol{w}^{\intercal}\mathbf{M}_{\mathbf{T}}$ corresponding to the j–th subset in the partitioning is given by

$$\boldsymbol{w}_j^{\intercal}\mathbf{A}_j + (1-\boldsymbol{w}_j^{\intercal}\boldsymbol{e})\boldsymbol{b}_j^{\intercal} + (\sum_{l=1,\ldots,k,l\neq j} \boldsymbol{w}_l\boldsymbol{a}_l)\boldsymbol{e}^{\intercal}, \tag{3.8}$$

while the last entry of $\boldsymbol{w}^{\intercal}\mathbf{M}_{\mathbf{T}}$, namely $\boldsymbol{w}^{\intercal}\mathbf{M}_{\mathbf{T}}\boldsymbol{e}_N$, is given by

$$\boldsymbol{w}^{\intercal}\mathbf{M}_{\mathbf{T}}\boldsymbol{e}_N = \sum_{l=1,\ldots,k} \boldsymbol{w}_l\boldsymbol{a}_l. \tag{3.9}$$

The observations above lead to a straightforward "divide and conquer" approach to finding the vector of accessibility indices for an irreducible stochastic matrix whose directed graph contains a cut–point. Without loss of generality we take the transition matrix to have the form (3.1). We may proceed as follows:

1. Compute the stationary vector $\boldsymbol{w}$.
2. For each $j = 1, \ldots, k$, compute $r_j = \frac{w_{k+1}}{1-\boldsymbol{w}_j^{\intercal}\boldsymbol{e}}$.
3. For each $j = 1, \ldots, k$, form $\mathbf{T}_{(j)}$ as in (3.3).
4. For each $j = 1, \ldots, k$, compute the vector of accessibility indices corresponding to $\mathbf{T}_{(j)}$.
5. From step 4, we now have the vectors $\boldsymbol{w}_j^{\intercal}\mathbf{A}_j + (1-\boldsymbol{w}_j^{\intercal}\boldsymbol{e})\boldsymbol{b}_j^{\intercal}, j = 1, \ldots, k$, and the scalars $\boldsymbol{w}_j^{\intercal}\boldsymbol{a}_j, j = 1, \ldots, k$. We can now assemble the vector of accessibility indices corresponding to $\mathbf{T}$ from (3.8) and (3.9).

We note in passing that once $\boldsymbol{w}$ has been found in step 1, steps 2–4 may be computed independently and in parallel. We observe further that the vector of accessibility indices corresponding to $\boldsymbol{v}_j^{\intercal}\mathbf{M}_{\mathbf{T}_{(j)}}$ is, up to the addition of a scalar multiple of the all–ones vector, the same as the subvector of $\boldsymbol{w}^{\intercal}\mathbf{M}_{\mathbf{T}}$ that corresponds to the j–th subset in the partitioning along with state N. In particular, if we are only interested in the ordering of the accessibility indices in that subvector, it suffices to compute $\boldsymbol{v}_j^{\intercal}\mathbf{M}_{\mathbf{T}_{(j)}}$.

For an irreducible stochastic matrix $\mathbf{T}$ of the form (3.1), there is an approach to step 1 that can also be implemented in a distributed fashion by considering several stochastic matrices of smaller order. For each $j = 1, \ldots, k$, consider the irreducible stochastic matrix $\mathbf{S}_j$ given by

$$\mathbf{S}_j = \left[\begin{array}{c|c} \mathbf{T}_{j,j} & \mathbf{T}_{j,k+1} \\ \hline \mathbf{T}_{k+1,j} & 1-\mathbf{T}_{k+1,j}\boldsymbol{e} \end{array}\right]. \tag{3.10}$$

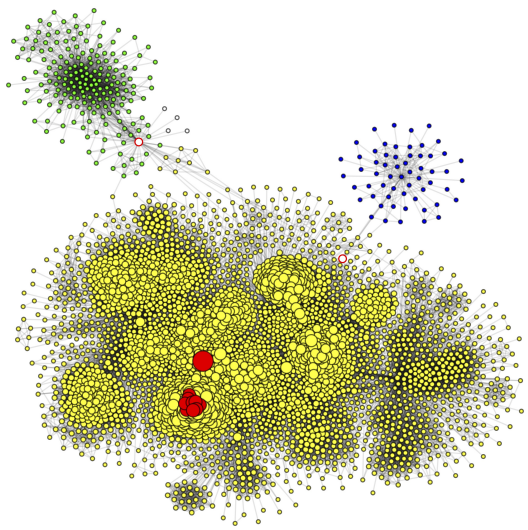

Figure 3.1: Order 3 neighbourhood of the nodes with the 10 largest θ_k (in red) along with the two cut nodes (white with red border) in the `ego-Facebook` Network.

It is readily verified that $\mathbf{S}_j$ is the stochastic complement (Meyer, 1989) arising from $\mathbf{T}$ by keeping the rows and columns corresponding to the j–th subset of the partition as well as row and column N, and censoring the remaining rows and columns. Write the stationary vector of $\mathbf{S}_j$ in partitioned form as $\mathbf{y}_j^\intercal = \left[\begin{array}{c|c} \boldsymbol{u}_j^\intercal & 1 - \boldsymbol{u}_j^\intercal \boldsymbol{e} \end{array} \right]$. A result in Meyer (1989) now shows that the subvector of $\boldsymbol{w}$ corresponding to the j–th subset of the partition as well as state N is a scalar multiple of $\mathbf{y}_j$. Letting

$$\tilde{\boldsymbol{w}} = \left[\begin{array}{cccc|c} (\frac{1}{1-\boldsymbol{u}_1^\intercal \boldsymbol{e}})\boldsymbol{u}_1^\intercal & (\frac{1}{1-\boldsymbol{u}_2^\intercal \boldsymbol{e}})\boldsymbol{u}_2^\intercal & \dots & (\frac{1}{1-\boldsymbol{u}_k^\intercal \boldsymbol{e}})\boldsymbol{u}_k^\intercal & 1 \end{array} \right],$$

we find readily that $\boldsymbol{w} = \frac{1}{\tilde{\boldsymbol{w}}^\intercal \boldsymbol{e}} \tilde{\boldsymbol{w}}$. Thus we can compute the stationary vector of $\mathbf{T}$ by computing the stationary distributions of the lower order matrices $\mathbf{S}_1, \dots, \mathbf{S}_k$.

3.1. Facebook Ego Network

The undirected `ego-Facebook` network (Leskovec and Krevl, 2014) consists of a single strongly connected component with 4039 nodes and 88234 edges. It also contains two cut-points such that, when all edges are removed from these nodes, the network becomes separated into 9 strongly connected components of sizes 3782, 192, 59 plus 6 singletons. For each node k in this network, we calculated α_k, θ_k and w_k. Figure 3.1 shows the order-3 neighbourhood of the nodes with the 10 largest θ_k (in red) along with the two cut nodes (white with red border). The three largest groups are depicted in yellow, green and blue respectively and the singletons are grey. The nodes are sized according to the θ_k.

To illustrate the results of Section 3, we made use of the cut-node that divided the network into components of sizes 193 (including the cut node), which is $\mathbf{T}_{(j)}$, and the remaining $4039 - 193 =$

3846. Let us denote the resulting accessibility indices calculated on $\mathbf{T}_{(j)}$ as α'_j. As expected, numerical results provide an exact correspondence in the ordering of the accessibility indices α'_j and α_j using the entire network and using the methods of Section 3 respectively. One may also approximate the r_j for this component using $\hat{r}_j = \hat{w}_{k+1}(1 - \hat{\mathbf{w}}_j^\intercal \boldsymbol{e})^{-1}$, where the $\hat{w}_k$ are estimated as in Section 2, and then forming

$$\hat{\mathbf{T}}_{(j)} = \left[\begin{array}{c|c} \mathbf{T}_{j,j} & \mathbf{T}_{j,k+1} \\ \hline \hat{r}_j \mathbf{T}_{k+1,j} & 1 - \hat{r}_j \mathbf{T}_{k+1,j} \boldsymbol{e} \end{array}\right]. \tag{3.11}$$

Since this network has two cut-nodes, we could decompose the network recursively and obtain results for each component. We leave this to the reader to explore further.

4. Concluding Remarks

In this paper we examined the Markov chain accessibility index for strongly connected (un)directed networks and its connection to the so-called random walk centrality measure. We showed how these could be estimated using simple random walks and that standard errors and biases can be estimated using basic resampling methods. Results showed that identifying the "most important" nodes in the network (according to the θ_k) can be accomplished with relative ease and with fairly high confidence. This sets the accessibility indices (random walk centralities) apart from many other centrality measures which are far more complex to calculate and do not lend themselves to easy estimation. Other advantages to this approach are the facts that most of the procedures are easy to do in parallel or in a distributed fashion and data can be collected incrementally over time for networks that are relatively static.

5. Acknowledgements

The authors would like to thank Dr. Mohammad Jafari Jozani for his comments on an earlier version of this paper. The authors would also like to thank the editor, associate editor and two anonymous referees for their valuable comments on earlier versions of this paper. This work was supported by the Natural Sciences and Engineering Research Council of Canada under grant numbers RGPIN/05480-17 (Brad C. Johnson) and RGPIN/6123-2014 (Steve Kirkland).

References

Csardi, G. and T. Nepusz (2006). The igraph software package for complex network research. *InterJournal Complex Systems*, 1695.

Dagdeviren, O. and V. Akram (2014). An energy-efficient distributed cut vertex detection algorithm for wireless sensor networks. *Computer Journal 57*, 1852–1869.

Efron, B. and R. J. Tibshirani (1994). *An Introduction to the Bootstrap*. CRC Press.

Kirkland, S. (2016). Random walk centrality and a partition of Kemeny's constant. *Czechoslovak Mathematical Journal 66*(3), 757–775.

Kirkland, S. and M. Neumann (2013). *Group Inverses of M–matrices and their Applications*. CRC Press.

Leskovec, J. and A. Krevl (2014, June). SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`.

Meyer, C. (1989). Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems. *SIAM Review 31*, 240–272.

Noh, J. and H. Rieger (2004). Random walks on complex networks. *Physical Review Letters 92*, 118701.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Stojmenovic, I., D. Simplot-Ryl, and A. Nayak (2011). Toward scalable cut vertex and link detection with applications in wireless ad hoc networks. *IEEE Network 25*, 44–48.

Viswanath, B., A. Mislove, M. Cha, and K. P. Gummadi (2009, August). On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks (WOSN'09)*.

Xiong, S. and J. Li (2010). An efficient algorithm for cut vertex detection in wireless sensor networks. In *Proceedings of the 2010 IEEE 30th International Conference on Distributed Computing Systems*, ICDCS '10, Washington, DC, USA, pp. 368–377. IEEE Computer Society.