Many snowballs make light work: a technique for large networks

Alex Stivala[1] Peng Wang[1] Johan Koskinen[2]
Garry Robins[1] David Rolls[1]

[1]Melbourne School of Psychological Sciences, The University of Melbourne, Australia
[2]The Mitchell Centre for SNA and Social Statistics Discipline Area, School of Social Sciences, University of Manchester, UK

INSNA Sunbelt XXXIV, February 18–24 2014
St. Pete Beach FL

Introduction

- Exponential random graph models (ERGMs) are useful for analyzing social networks.
- But estimating parameters is computationally intensive.
- This restricts the size of the networks that can have an ERGM fitted, to a few thousand nodes at most.
- The Markov chain Monte Carlo (MCMC) methods often used are inherently sequential, limiting the use of high performance parallel computing.
- We will overcome this problem by taking multiple snowball samples, estimating the ERGM parameters for each in parallel, and combining the estimates with meta-analysis.

Exponential random graph models (ERGMs)

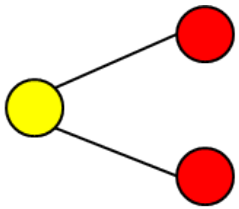
$$\Pr(X = x) = \frac{1}{\kappa} \exp\left(\sum_A \theta_A z_A(x)\right)$$

where

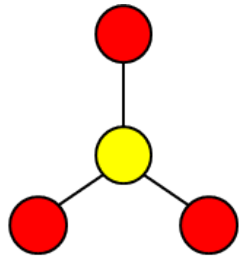
- $X = [X_{ij}]$ is a 0-1 matrix of random tie variables,
- x is a realization of X,
- A is a subgraph configuration,
- $z_A(x)$ is the network statistic for configuration A,
- θ_A is a model parameter corresponding to configuration A,
- κ is a normalizing constant to ensure a proper distribution.

Model configurations — structural

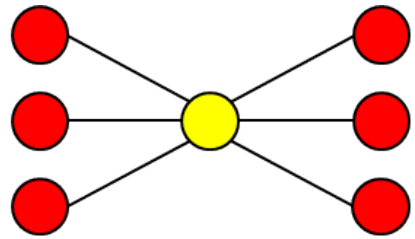
k-stars: useful for capturing degree distribution



Two-star

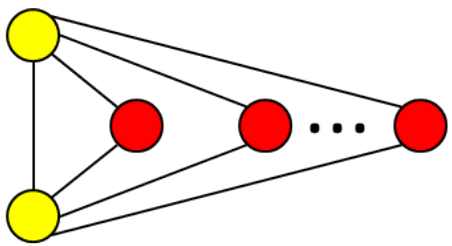


Three-star

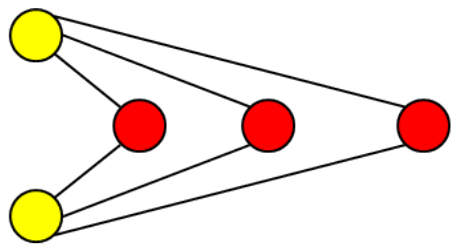


k-star

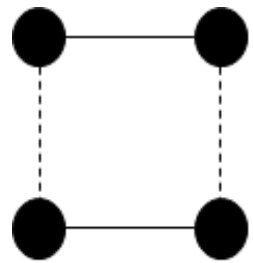
k-triangles (AKT), k-2-paths (A2P): useful for modelling social circuit dependence



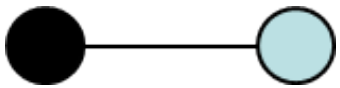
k-triangles



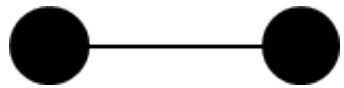
k-2-paths



Model configurations — binary actor attributes



Activity
`R_Attribute1`



Interaction
(homophily)
`Rb_Attribute1`

Actor with attribute

Actor with or without attribute

Prior work

Xu, B., Huang, Y., & Contractor, N. 2013, "Exploring Twitter networks in parallel computing environments", XSEDE '13

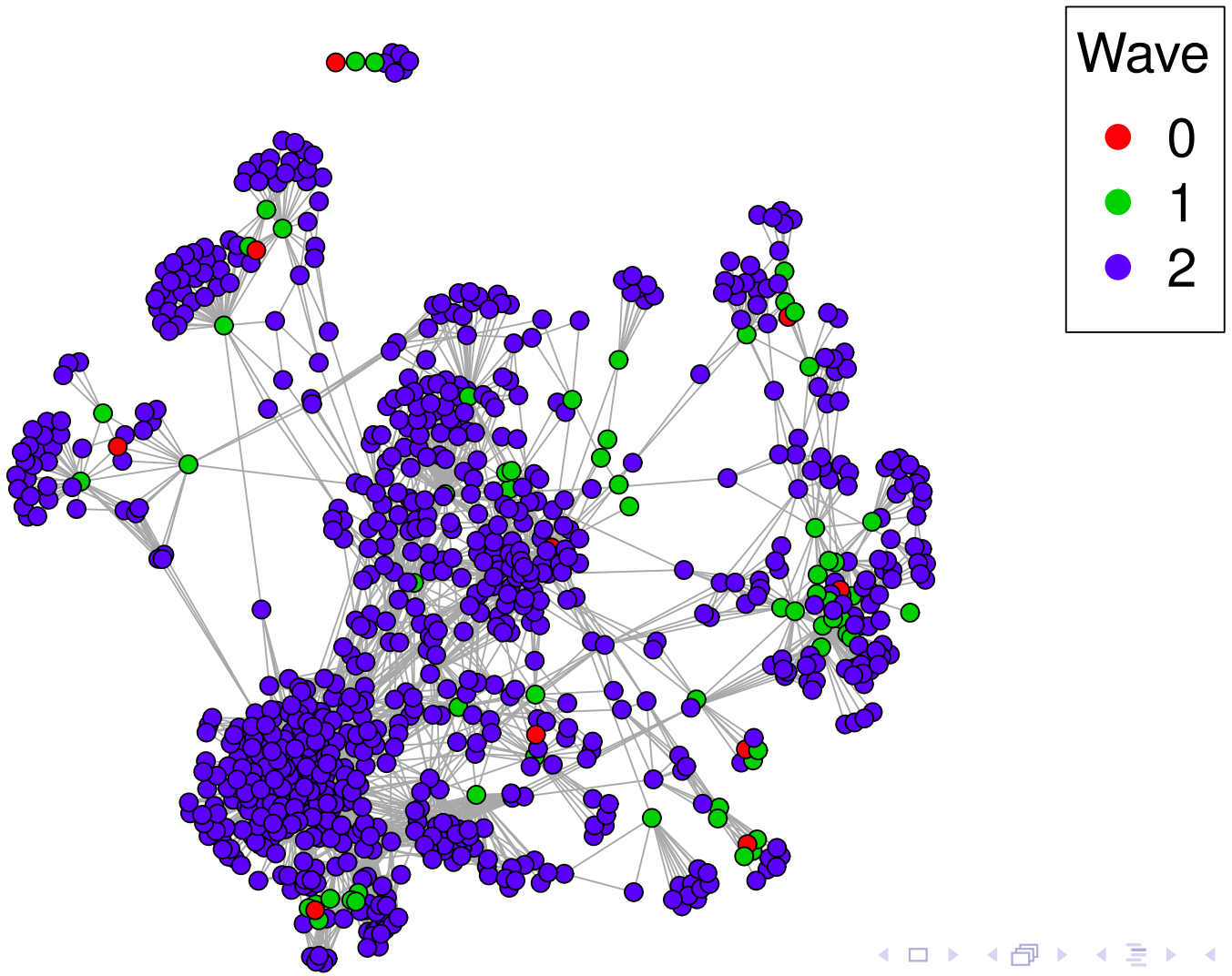
- Snowball sampling in a Twitter "unfollow" network
  - 211263 nodes
  - 1 wave, 1 seed
  - 394 samples
- Special data-intensive supercomputer
  - sample from large data set,
  - run statnet on each sample in parallel to estimate ERGM parameters.
- Combines estimates with meta-analysis, the same one we will use (Snijders and Baerveldt 2003).
- But:
  - does not account for snowball sampling structure,
  - applied only to the Twitter unfollow network; no testing of errors, bias, etc. by testing on known networks.

Snowball sampling

- Start with N_0 seed nodes (wave 0).
- Follow their ties to get a further set of nodes (wave 1).
- In general, follow the ties from nodes in wave $k-1$ to get the nodes in wave k.
- There is a picture on the next slide...
- Snowball sampling generates approximately well-separated conditionally independent subsets of the network, so that we can use new conditional estimation procedures (Pattison *et al.* 2013).

Snowball sampling example from Nexus condmatcollab2005

Snowball sample ($n = 907$) from condensed matter collaborations network $N = 40421$, 2 waves, 10 seeds.



Conditional estimation (basics only; see paper for details)

Pattison, Robins, Snijders and Wang, 2013, "Conditional estimation of exponential random graph models from snowball sampling designs" *J. Math. Psychol.* 57(6):284–296.

- Do conditional estimation, respecting the snowball sampling structure.
- Conditional probability model of ties in waves $0, \ldots, k-1$ has the same parameters as the ERGM for the whole network,
- but can be estimated conditionally from the ties between a node in wave $k-1$ and a node in wave k, and ties between nodes in wave k, and the composition of the node sets in waves $0, \ldots, k$.
- A tie between nodes in wave i is conditionally independent of ties between nodes not in wave i, $i-1$ or $i+1$.

Estimates of samples are approximate estimators of ERGM

- Snijders, 2010, "Conditional Marginalization for Exponential Random Graph Models" *J. Math. Sociol.* 34(4):239–252:
  - A "component independent" ERGM can be estimated using the usual stochastic approximation methods.
  - Completely separated components are conditionally independent.
- Snowball samples can be used to generate completely separated regions.
- We relax this assumption, but for large N these samples are approximately well-separated,
- so the estimates for each are approximately i.i.d. estimators of the same ERGM.
- This allows us to pool the estimates with meta-analysis.

Meta-analysis

Weighted least squares estimator, as used in Snijders and Baerveldt, 2003 *J. Math. Sociol.* 27:123–151:

$$\hat{\mu}_\theta^{\mathrm{WLS}} = \frac{\sum_j \left(\hat{\theta}_j / (\hat{\sigma}_\theta^2 + s_j^2) \right)}{\sum_j \left(1 / (\hat{\sigma}_\theta^2 + s_j^2) \right)}$$

where

- $j \in 1, \ldots, N_s$ are the N_s snowball samples,
- $\hat{\theta}_j$ is the estimate for sample j,
- $\hat{\sigma}_\theta^2 = 0$ is the estimated between-sample variance, zero by assumption,
- s_j is the estimated standard error for sample j.

Estimating confidence intervals with bootstrap method

- The standard error of the WLS estimator can be calculated as per Snijders & Baerveldt (2003) with $\text{s.e.}(\hat{\mu}_\theta^{\text{WLS}}) = \frac{1}{\sqrt{\sum_j 1/(\sigma_\theta^2 + s_j^2)}}$.
- But this assumes that θ_j and s_j^2 are independent across samples,
- and we found that confidence intervals were too small.
- So instead we use the non-parametric bootstrap adjusted percentile (BCa) method to estimate the confidence interval:
  - with R bootstrap replicates of our estimator $\hat{\mu}^*_{\theta(i)}, i \in 1 \ldots R$, in nondecreasing order, the basic percentile C.I. is $(\hat{\mu}^*_{\theta((R+1)\alpha)}, \hat{\mu}^*_{\theta((R+1)(1-\alpha))})$
  - The BCa method (Efron 1987) adjusts for bias and skewness in the bootstrap distribution, using the estimated std. errors s_j.
  - We use the R `boot` package to do this.

Simulated networks — methods

By using simulated networks, we can measure errors in the estimation. By simulating networks, each with a single parameter set to zero, we can measure Type I error rates in inference.

N	Attr	Edge	Alt.k-Star	AKT	A2P	R	Rb
5000	None	-4.0	0.2	1.0	-0.2		
5000	50/50	-4.0	0.2	1.0	-0.2	0.2	0.5
5000	50/50 †	-4.0	0.2	1.0	-0.2	-0.25	0.5
10000	None	-4.0	0.2	1.0	-0.2		

†We refer to this network as "balanced" since homophily and interaction are balanced (there is no "differential homophily" since Rb $= -2$R).

Estimating the 5000 node network (no attributes) takes 26 days with PNet on an Intel Core i7 PC (3.40 GHz).

Snowball estimation — methods

We take 100 sample networks for each simulated network and do parallel snowball PNet estimations using 20 parallel processes for each with parameters:

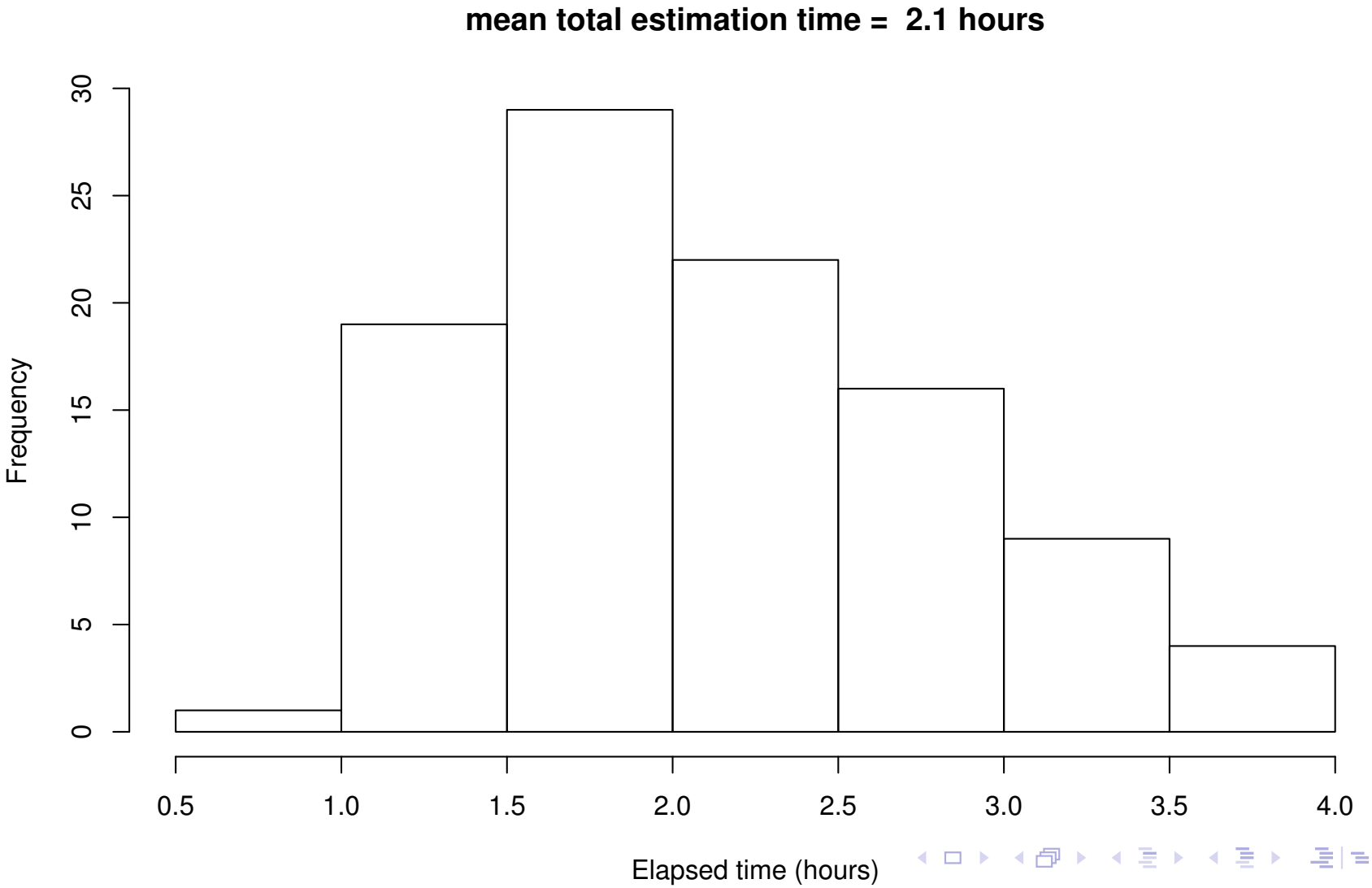
- 2 waves
- 10 seeds
- 20 seed sets, i.e. snowball samples (so one per parallel process)

Parallel snowball PNet has been implemented using both MPI (for clusters) and OpenMP (for multicore PCs).

The cluster system we used is an SGI Altix XE Cluster, 1088 Intel Nehalem cores (8 per node) 2.66 GHz, CentOS 5, OpenMPI.

Simulated networks — elapsed time

5000 node network, no attributes.



Simulated networks — results and Type II error rate

N	Attributes	Fixed density	Effect	Bias	RMSE	False negative rate (%)	Std. dev. estimate
5000	50/50 balanced	Y	A2P-T(2.00)	0.007509	0.01493	0	0.01297
5000	50/50 balanced	Y	AKT-T(2.00)	-0.01895	0.02522	0	0.01672
5000	50/50 balanced	Y	K-Star(2.00)	0.1198	0.1951	66	0.1548
5000	50/50 balanced	Y	R Attribute1	-0.01076	0.03981	0	0.03852
5000	50/50 balanced	Y	Rb Attribute1	0.00414	0.04926	0	0.04933
5000	50/50	Y	A2P-T(2.00)	0.01934	0.02167	0	0.00983
5000	50/50	Y	AKT-T(2.00)	-0.02282	0.02627	0	0.01308
5000	50/50	Y	K-Star(2.00)	0.1075	0.1607	50	0.1201
5000	50/50	Y	R Attribute1	-0.07279	0.08115	16	0.03605
5000	50/50	Y	Rb Attribute1	-0.001384	0.03803	0	0.03819
10000	None	Y	A2P-T(2.00)	0.003446	0.01464	0	0.0143
10000	None	Y	AKT-T(2.00)	-0.007439	0.02183	0	0.02063
10000	None	Y	K-Star(2.00)	0.1554	0.238	66	0.1812

Error rate calculated with 3 std. error confidence interval.

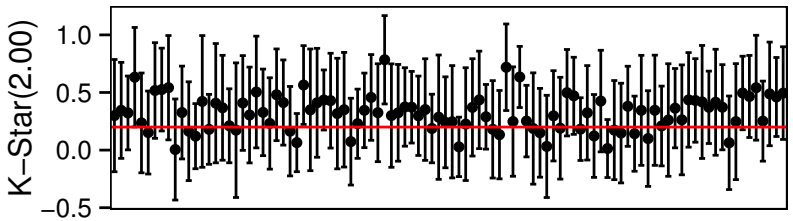
Simulated networks — Type I error rate

N	Attributes	Fixed density	Effect	Bias	RMSE	False positive rate (%)	Std. dev. estimate
5000	50/50 balanced	Y	AKT-T(2.00)	0.04861	0.0723	25	0.05379
5000	50/50 balanced	Y	K-Star(2.00)	0.2493	0.3944	7	0.3072
5000	50/50 balanced	Y	R Attribute1	-0.03099	0.05074	9	0.04038
5000	50/50 balanced	Y	Rb Attribute1	0.002953	0.05535	4	0.05555
5000	50/50	Y	AKT-T(2.00)	0.0286	0.04753	7	0.03816
5000	50/50	Y	K-Star(2.00)	0.2349	0.2827	10	0.1581
5000	50/50	Y	R Attribute1	-0.03099	0.05074	9	0.04038
5000	50/50	Y	Rb Attribute1	0.01091	0.05292	7	0.05205

Error rate calculated with 3 std. error confidence interval.

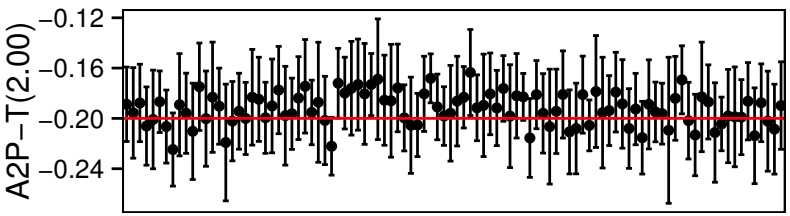
Results: 5000 node balanced 50/50 network

no Edge parameter



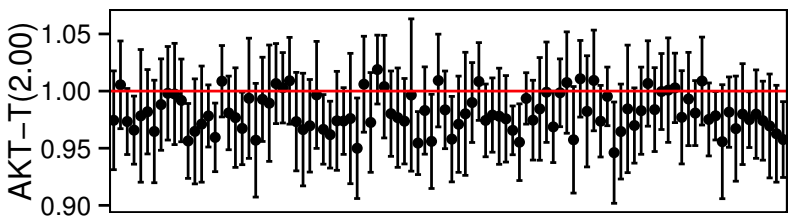
bias = 0.1198, RMSE = 0.1951

% In CI = 95, FNR% = 66



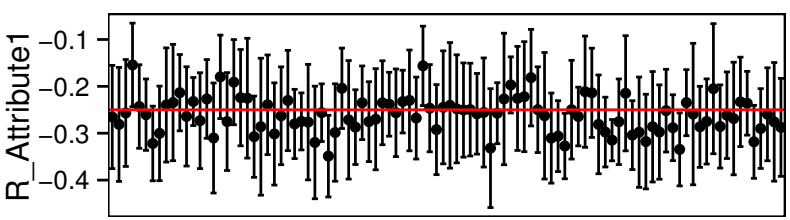
bias = 0.007509, RMSE = 0.01493

% In CI = 95, FNR% = 0



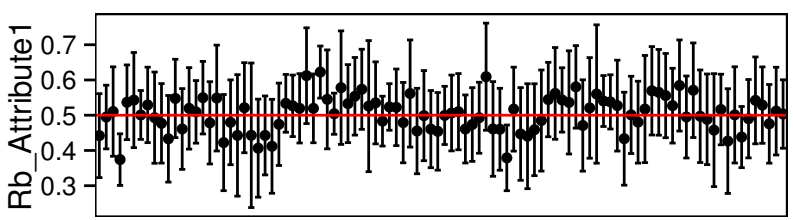
bias = −0.01895, RMSE = 0.02522

% In CI = 83, FNR% = 0



bias = −0.01076, RMSE = 0.03981

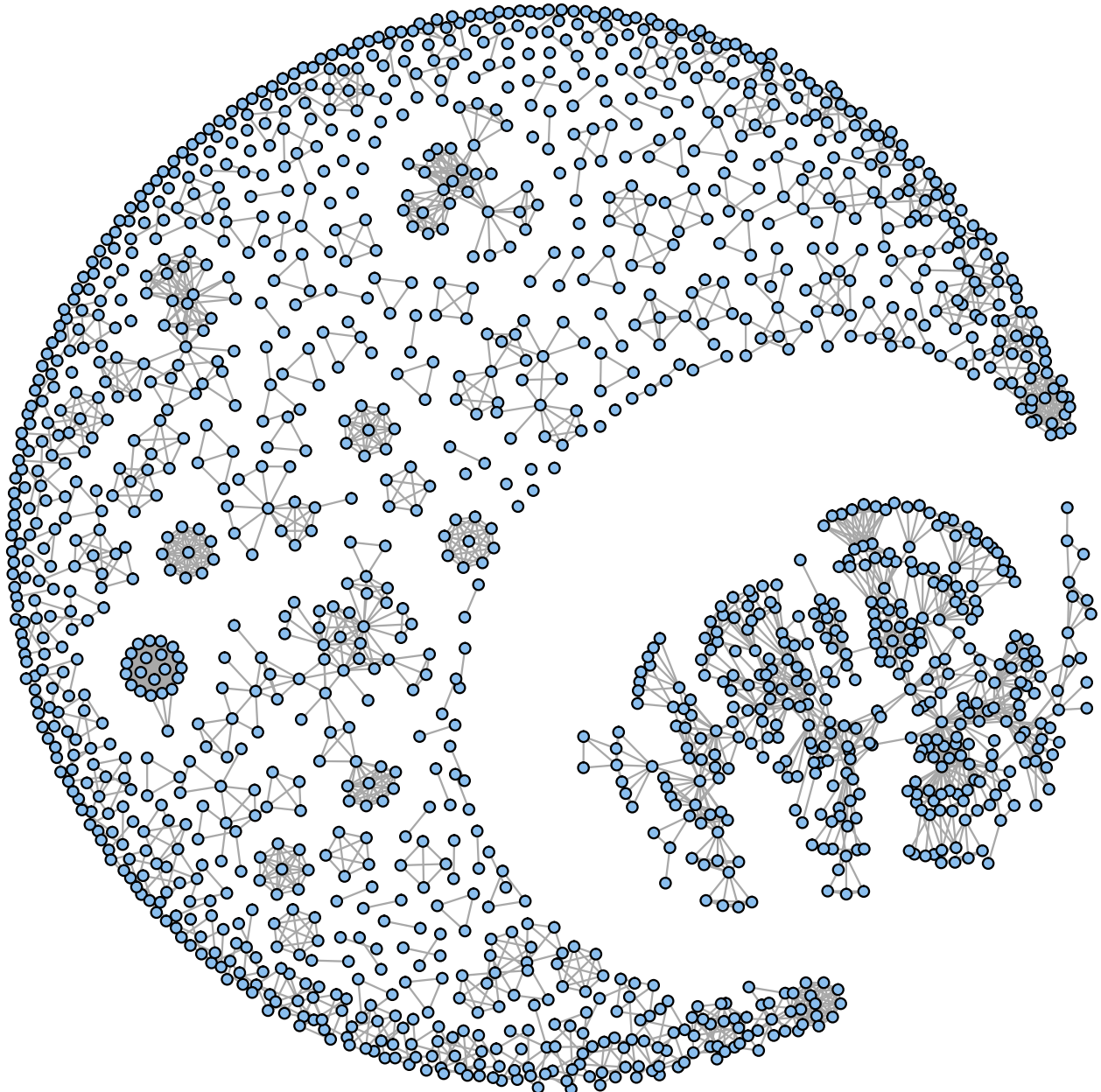
% In CI = 94, FNR% = 0



bias = 0.00414, RMSE = 0.04926

% In CI = 97, FNR% = 0

Network science collaboration network $N = 1589$



Results for network science collaboration network

Effect	Estimate	std. errror	t-ratio	
A2P-T(2.00)	-0.0241	0.0099	0.0914	*
AKT-T(2.00)	3.9059	0.0668	0.0491	*
Edge	-7.3861	0.1388	0.0410	*
K-Star(2.00)	-0.7518	0.0551	-0.0557	*

Standard PNet (full network), 7 days.

Effect	N_s	Estimate	C.I.		
			lower	upper	
A2P-T(2.00)	13	-0.0112	-0.0669	0.0444	
AKT-T(2.00)	13	3.6320	3.1220	4.1420	*
Edge	13	-7.1954	-8.2884	-6.1025	*
K-Star(2.00)	13	-0.5316	-0.9687	-0.0945	*

Snowball PNet: 2 waves, 10 seeds, 20 processors, 1.5 hours.

Results for condensed matter collaboration network

Nexus condmatcollab2005, $N = 40421$, snowball sampling 2 waves, 10 seeds, 20 seed sets.

Effect	N_s	Estimate	C.I.		
			lower	upper	
A2P-T(2.00)	10	-0.0037	-0.0091	0.0016	
AKT-T(2.00)	10	4.8141	4.2843	5.3439	*
Edge	10	-9.4337	-10.6413	-8.2260	*
K-Star(2.00)	10	-0.6272	-0.8349	-0.4196	*

Acknowledgments

- Co-authors: Peng Wang, Johan Koskinen, Garry Robins, David Rolls
- Pip Pattison
- University of Melbourne ITS High Performance Computing
- This research was supported by Victorian Life Sciences Computation Initiative (VLSCI) grant numbers VR0261 and VR0297 on its Peak Computing Facility at the University of Melbourne, an initiative of the Victorian Government, Australia.

Hidden bonus slide — differential homophily

	0	1
0	θ_L	$\theta_L + \rho$
1	$\theta_L + \rho$	$\theta_L + 2\rho + \rho_B$

- θ_L is the edge (density) parameter,
- ρ is the activity parameter,
- ρ_B is the interaction (homophily) parameter.
- If $\rho_B = -2\rho$ then there is no "differential homophily", and we say it is "balanced".
- If there is differential homophily, to estimate homophily (ρ_B), interaction (ρ) must be included to control for bias in density from sampling, if density is fixed in estimation.
- We can test for differential homophily by checking proportion of 1 nodes in sample versus whole network.

Hidden bonus slide: Results for astrophysics collaboration network

Nexus astrocollab, $N = 16706$, snowball sampling 2 waves, 5 seeds, 20 seed sets.

Effect	N_s	Estimate	C.I.		
			lower	upper	
A2P-T(2.00)	7	-0.0025	-0.0130	0.0080	
AKT-T(2.00)	7	4.5364	3.3327	5.7400	*
Edge	7	-13.4567	-23.1142	-3.7992	*
K-Star(2.00)	7	0.1497	-2.6886	2.9880	

Hidden bonus slide — statnet "Stepping" results for network science network ($N = 1589$)

Hummel, Hunter and Handcock, 2012, "Improving Simulation-Based Algorithms for Fitting ERGMs" *J. Comp. Graph. Stat.* 21(4):920–939

Effect	Estimate	std. errror	p-value	
A2P-T(2.00) [gwdsp(ln(2))]	-0.0683	0.0259	0.00827	**
AKT-T(2.00) [gwesp(ln(2))]	4.1113	0.0843	$<$ 1e-04	***
Edge	-6.5116	0.1719	$<$ 1e-04	***
K-Star(2.00)	-0.9442	0.0991	$<$ 1e-04	***

Statnet "stepping" algorithm (full netscience network), 1.9 hours.