Modeling Large Social Networks via Snowball Samples

Alex Stivala[1] Johan Koskinen[2] Peng Wang[3]
Garry Robins[1] David Rolls[1] Alessandro Lomi[4]

[1]The University of Melbourne, Australia
[2]University of Manchester, UK
[3]Swinburne University of Technology, Australia
[4]University of Lugano, Switzerland

ICCSS 2015, June 8-11, Helsinki, Finland

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)

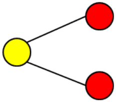
$$\Pr(X = x) = \frac{1}{\kappa} \exp\left(\sum_{A} \theta_A z_A(x)\right)$$

where

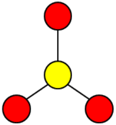
- $X = [X_{ij}]$ is a 0-1 matrix of random tie variables,
- x is a realization of X,
- A is a subgraph configuration,
- $z_A(x)$ is the network statistic for configuration A,
- θ_A is a model parameter corresponding to configuration A,
- κ is a normalizing constant to ensure a proper distribution.

Model configurations — structural

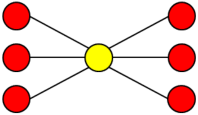
k-stars: useful for capturing degree distribution



Two-star

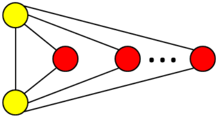


Three-star

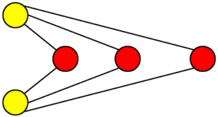


k-star

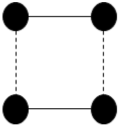
k-triangles (AKT), *k*-2-paths (A2P): useful for modelling social circuit dependence



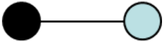
k-triangles



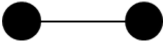
k-2-paths



Model configurations — binary actor attributes



Activity, ρ



Interaction (homophily), ρ_B

● Actor with attribute

○ Actor with or without attribute

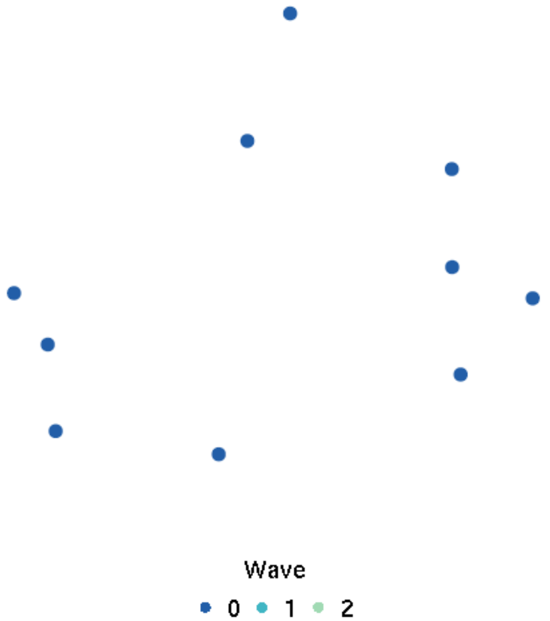
Scalability

- We can estimate parameters for “small” (up to a few thousand nodes) networks with MCMC methods.
- But the computational intractability means it becomes impractical to estimate larger networks.
- What can we do? One answer is sampling.
- But sampling in networks is not as straightforward as simple random sampling.
- And we can’t assume parameter estimates for subnetworks apply to the whole network, as they are specific to N and do *not* scale linearly.
- One solution is *snowball sampling* with *conditional estimation*.

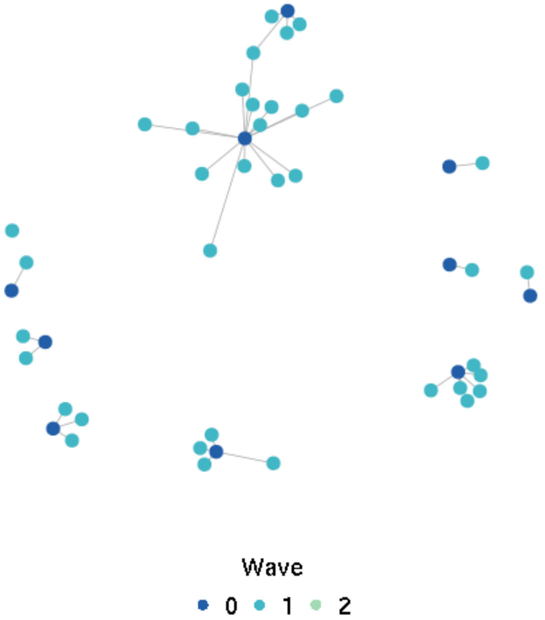
Snowball sampling

- Start with N_0 seed nodes (wave 0).
- Follow all 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...
- We can use conditional estimation procedures (Pattison *et al.* 2013) on each snowball sample.

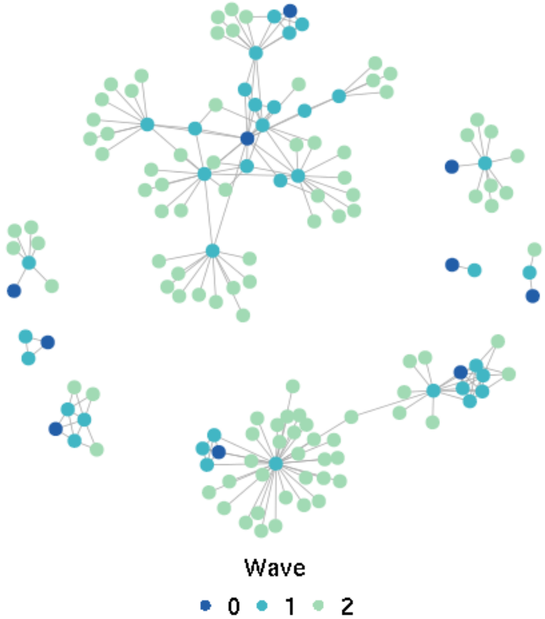
Snowball sampling example, wave 0



Snowball sampling example, wave 1



Snowball sampling example, wave 2



Estimating confidence intervals with bootstrap method

For both the WLS and the median point estimators:

- 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.

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	ρ	ρ_B
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 $\rho = -2\rho_B$).

Snowball estimation — methods

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

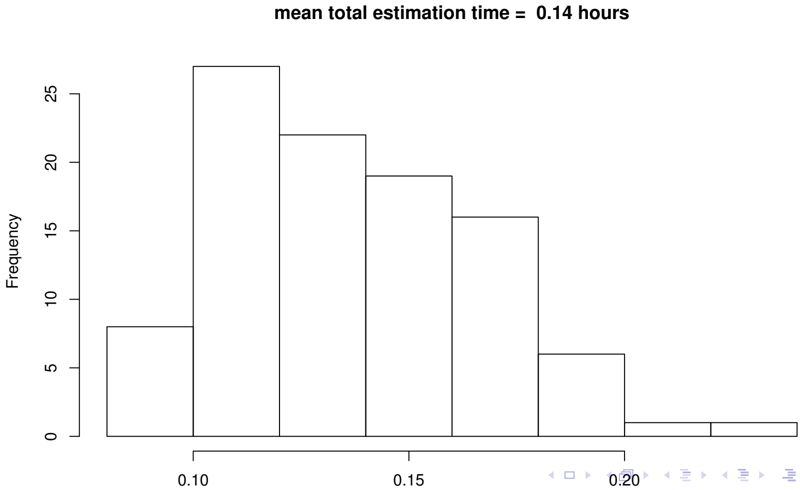
- 2 waves
- 10 seeds
- 20 seed sets, i.e. snowball samples (so one per parallel process)

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.

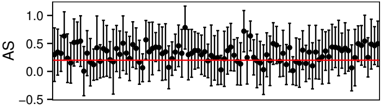
Simulated networks — elapsed time

5000 node network, no attributes. 83.5 hours for parameter estimation on full network with PNet.



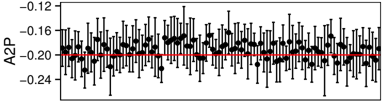
Results: 5000 node balanced 50/50 network

no Edge parameter



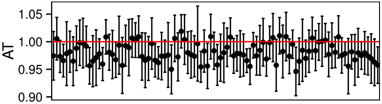
bias = 0.1198, RMSE = 0.1951

% In CI = 96, FNR% = 66



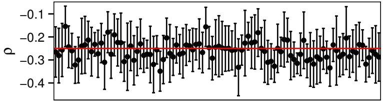
bias = 0.007509, RMSE = 0.01493

% In CI = 96, FNR% = 0



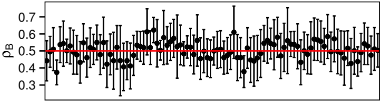
bias = −0.01895, RMSE = 0.02522

% In CI = 83, FNR% = 0



bias = −0.01076, RMSE = 0.03981

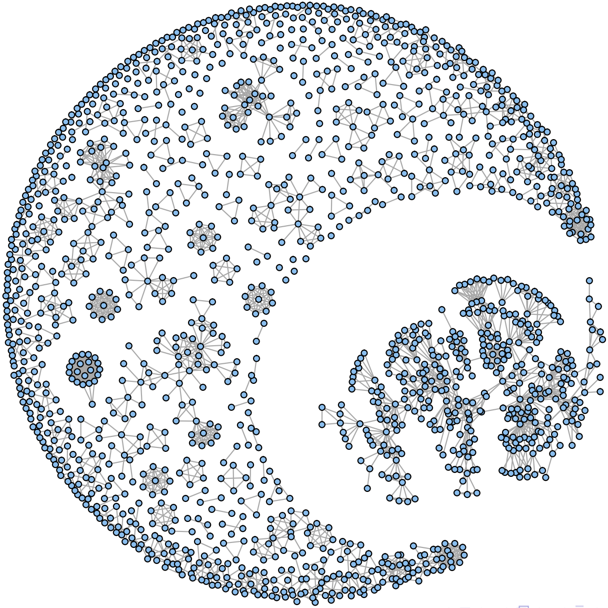
% In CI = 94, FNR% = 0



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	convergence statistic	
A2P	-0.0216	0.0085	0.0493	*
AT	3.8091	0.0602	-0.0401	*
Edge	-7.3165	0.1335	-0.0321	*
AS	-0.7046	0.0635	-0.0784	*

Standard PNet (full network), 9.5 days.

Effect	N_s	Estimate	C.I.		
			lower	upper	
A2P	20	-0.0451	-0.1281	0.0380	
AT	20	3.7423	3.1692	4.3155	*
Edge	20	-7.9978	-8.7875	-7.2081	*
AS	20	-0.4655	-0.6893	-0.2417	*

Snowball PNet: 2 waves, 10 seeds, 20 processors, 5 hours elapsed (15 hours total CPU time).

Results for condmatcollab2005 network, N = 40 421

Effect	N_s	Estimate	C.I.		
			lower	upper	
A2P	63	-0.0004	-0.0044	0.0037	
AT	63	4.3729	3.6009	5.1448	*
Edge	63	-9.2179	-10.5882	-7.8477	*
AS	63	-0.6983	-1.1029	-0.2936	*

100 snowball samples (100 tasks), 3 seeds, 2 waves. 102 hours.

Effect	N_s	Estimate	C.I.		
			lower	upper	
A2P	57	0.0001	-0.0045	0.0046	
AT	57	4.2161	3.4736	4.9586	*
Edge	57	-8.8726	-10.2970	-7.4482	*
AS	57	-0.8375	-1.2084	-0.4666	*

Results after 7 hours.

Conclusions and future work

- We have shown how to make inferences from ERGM parameters for large (over 40 000 nodes) networks.
- Previously, this was only possible for a few thousands nodes at most.
- Future (ongoing current) work:
  - Directed networks (the subject of my upcoming talk at INSNA Sunbelt XXXV, June 23–28, Brighton UK).
  - Handle networks with hubs ("power law" degree distribution).
  - Bias correction.
  - Bipartite networks.
  - Goodness-of-fit procedure.

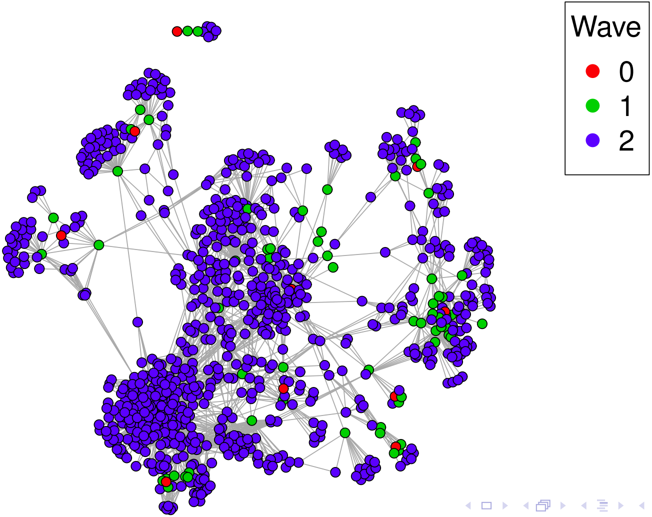
Acknowledgments

- Co-authors: Peng Wang, Johan Koskinen, Garry Robins, David Rolls, Alessandro Lomi
- 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.
- And the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1053575, with Prof. Noshir Contractor and Dr Yun Huang at Northwestern U.

Hidden bonus slides

Snowball sampling example from Nexus condmatcollab2005

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



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.

Also, we can use the median as an estimator with few assumptions.

Simulated networks

N	Attributes	Mean	Mean	Mean	Mean
		#comp.	degree	density	clust. coef.
5000	None	1.00	8.76	0.00175	0.02451
5000	50/50	1.00	9.54	0.00191	0.02661
5000	70/30	1.00	9.99	0.00200	0.02762
5000	50/50 balanced	1.01	8.51	0.00170	0.02428
10000	None	1.00	10.04	0.00100	0.01553

Type I error rate: 5000 node balanced 50/50 network

N	Attributes	Effect	Bias	RMSE	Type I error rate (%)			Std. dev.	Mean
					Estim.	95% C.I.		estimate	samples
						lower	upper		converged
5000	None	AT	-0.0087	0.0683	3	1	8	0.0681	18.94
5000	None	AS	0.2483	0.4185	6	3	12	0.3386	10.16
5000	70/30	AT	-0.0027	0.0477	3	1	8	0.0479	19.96
5000	70/30	AS	0.2394	0.2925	5	2	11	0.1689	16.08
5000	70/30	ρ	-0.0419	0.0753	6	3	12	0.0629	18.17
5000	70/30	ρ_B	-0.0099	0.0646	4	2	10	0.0641	18.17
5000	50/50 balanced	AT	-0.0091	0.0706	3	1	8	0.0703	19.16
5000	50/50 balanced	AS	0.1648	0.4174	7	3	14	0.3854	10.48
5000	50/50 balanced	ρ	-0.0245	0.0500	7	3	14	0.0438	16.96
5000	50/50 balanced	ρ_B	-0.0022	0.0639	2	1	7	0.0642	15.17
5000	50/50	AT	-0.0128	0.0480	0	0	4	0.0465	19.87
5000	50/50	AS	0.2369	0.2949	5	2	11	0.1764	14.00
5000	50/50	ρ	-0.0245	0.0500	6	3	12	0.0438	16.96
5000	50/50	ρ_B	0.0103	0.0526	4	2	10	0.0519	17.25