Benchmarking Equilibrium Expectation as an Estimation Method for Exponential Random Graph Models

Youhan Cheery

July 15, 2021

Contents

1 Abstract

Exponential Random Graph Models (ERGMs) are a family of network models that are useful in modelling complex network structures, such as corporate connections and social relations. ERGMs benefit from many of the properties that make their namesake, the exponential family, such a powerful device in the statistical toolkit.

Given the nature and complexity of modelling network interactions both in terms of statistical methodology and computational execution - many methods have been devised to estimate parameters of a network model accurately and within computational constraints. This challenge is due to the normalising constant that is core to the definition of an ERGM, often making parameter estimation intractable and difficult to compute. As such, initial methods approximated the likelihood in order to estimate parameters, an approach known as Maximum Pseudo Likelihood Estimation (MPLE) Strauss and Ikeda (1990), while more current approaches leverage improvements in computation power and aim to reach the MLE via the Markov Chain Monte Carlo (MCMC). Benchmarking these MCMC methods will be the crux of this paper.

Within the class of MCMC estimation methods, a recent approach, Equilibrium Expectation (EE), proposes an efficient means to compute the parameters of large networks, a typically challenging computational problem Byshkin et al. (2018). By using Markov Chain properties at equilibrium it is able to make infrequent, albeit complex updates to the parameter values to reduce computation effort required. It follows on from methods popularised by Robbins and Monro Robbins and Monro (1951) and Stochastic Approximation Snijders (2002) which have been staples in the estimation of ERGM parameters for a number of years. This paper will look to benchmark the Equilibrium Expectation algorithm against these two methods by comparing properties across some of the more challenging network datasets covered in Hummel et al. (2012).

The next section introduces ERGMs and the challenge at hand, followed by a literature review spanning Robbins and Monro, to Stochastic Approximation by Snijder, followed on by a summary of Equilibrium Expectation. This is then followed by the methodology and results obtained in the analysis.

2 Introduction

- Mention something about ERGMs and how they're used
- This includes the math side - and about network statistics
- Talk a little about the intractable denominator when it comes to ERGM estimation
- Computational challenges in ERGMs

Exponential family models find a wide variety of use: from social networks, to spatial statistics, to even image analysis. Exponential Random Graph Models (ERGMs) take the form:

$$\pi(x, \boldsymbol{\theta}) = \frac{1}{k(\boldsymbol{\theta})} \exp\left(\sum_A \theta_A z_A(x)\right) \tag{1}$$

Where $z_A(x)$ are the statistics that define the network, θ_A are the parameters of the network, and $k(\theta)$ is the normalising constant.

Due to this normalising constant often being intractable, parameter calculation approaches have needed to find a way around this summation of network statistics over all possible networks. Strauss and Ikeda in 1991 confronted this challenge by approximating the maximum likelihood form with an approximated form, or a 'pseudo maximum likelihood'. Using traditional maximum likelihood calculations Strauss and Ikeda then estimate the underlying parameter values. This method had drawbacks of its own: as it's namesake implies, the estimate was calculated for an approximation of the maximum likelihood equation of the network, and not the true likelihood itself. This meant that the possibility of erroneously selecting parameters that defined the observed network came with additional uncertainty.

With the advent of computation storage and power, Markov Chain approaches became viable ways of reaching the solution to optimisation problems iteratively. These methods used sampling techniques to gradually direct the model to the maximum likelihood estimate at the global minima. Beginning with Roberts and Munro in 1951 with the Stochastic Approximation method that sought to iteratively reach the parameters by the Newton-Raphson method. The relevance of Robberts and Munro's methods were extended to ERGMs by Sneijders in 2002, and onwards to Equilibrium Expectation in 2018 by Stivala et. al.

In this work, we implement the equilibrium expectation algorithm (Byshkin et al. (2018)) in the R package `ergm` (Hunter et al. (2008)) and benchmark the implementation to ERGM parameter estimation and compare it to the existing Monte Carlo Maximum Likelihood approaches of Robbins-Monro (Robbins and Monro (1951)) and Stochastic Approximation (Snijders (2002)). The implementation of the equilibrium expectation algorithm is important in this case, given that the methods used for comparison are themselves being run from the ERGM R package. Finally, we compare estimates using each method, using the *E. coli* transcriptional regulation network, and *Kapferers* sociational dataset as illustrated by Hummel et al. (2012).

We consider the computational tractibility of the MCMC-based approaches by considering three different starting points: starting from all parameters being set to zero, from the MPLE, and from a mostly zero starting point sans having values at the edges of the network pre-calculated with a run of `ergm` at default settings.

3 Literature Review

In order to benchmark the Equilibrium Expectation estimation method we use the Robbins-Monro algorithm as well as the later work by extension of this process known as Stochastic Approximation. The aforementioned estimation approaches are detailed in this section, focusing largely on the philosophy of each algorithm reaching meaningful estimates.

We specifically note that the these techniques sought the "actual" MLE, though it is worth beginning our discussion with the Maximum Pseudo Likelihood Estimation (MPLE) method (Strauss and Ikeda (1990)) as we use this method as an initialisation point for some of our benchmarking.

3.1 Maximum Pseudo-Likelihood Estimation

Due to the normalising constant referred to in Section 2 being a summation over parameters of all possible graphs, estimating the true parameters of the ERGM is often an intractable problem. Strauss and Ikeda proposed an approach where calculating the maximum likelihood function wasn't done on the direct likelihood but rather on a 'pseudo' likelihood which approximated the true underlying model.

For a dyad independent network model, that is, a model where every dyad (or pair of ties) is independent from the next - maximum likelihood can be calculated. While this assumption simplifies the mathematics of estimation, it limits the practicality of the network model greatly. For dyad dependent networks, Strauss and Ikeda proposed the pseudo-likelihood as the product of the probabilities of the y_{ij}, with each of the probabilities conditional on the rest of the data Strauss and Ikeda (1990).

When not conditioning on data, the ERGM takes the log linear form

$$Pr(G) = \frac{1}{Z(\theta)} e^{\theta' x(G)} \tag{2}$$

where θ is a vector of parameters and x(G) is a vector of graph statistics (on the observed graph).

Conditioning on the rest of the graph produces a model form that now does not depend on

θ

$$\begin{aligned} Pr(y_{ij} = 1|C) = \frac{Pr(G^{-})}{Pr(G^{-}) + Pr(G^{+})} \\ logit(Pr(y_{ij} = 1|C) = \theta' x(G^{+}) - x(G^{-}) \\ logit(Pr(y_{ij} = 1|C) = \theta' \delta x_{ij} \end{aligned} \tag{3}$$

3.2 Markov Chain Methods For Parameter Estimation

3.3 Monte Carlo Maximum Likelihood Estimation (MCMLE)

The Monte Carlo Maximuim Likelihood Estimate was first introduced by Geyer and Thompson (1992), and subsequently extended to curved ERGMs by Hunter and Handcock (2006). MCMLE is a method developed as a consequence to the fact that the Laplace transformations (or normalising constants as is often used in this paper) for exponential family models with dependent data cannot be exactly calculated, and approximations are difficult to find. As such, the general idea comes down to translating from the intractable integral, to a probability distribution such that Monte Carlo methods become applicable, as shown with the method of moments equation below:

$$M_\theta(t) = \int exp < t(x), \tau > dP_\theta(x) = \frac{c(\theta + t)}{c(\theta)} \tag{4}$$

Where $t(x)$ are canonical statistics, and P_θ denoting the measure having density f with respect to μ.

The above refers to the “Monte Carlo” side of the eponymous section. Standard Gibbs or Metropolis sampling methods are used to generate an ergodic Markov chain X_1, X_2, ... having equilibrium distribution P_θ. This leads to the observation that

$$d_n(\theta') = \frac{1}{n}\Sigma_{i=1}^{n} exp {<}\, t(x_i), \theta' - \theta > \to d(\theta) = \frac{c(\theta')}{c(\theta)} \tag{5}$$

almost surely by the ergodic theorem. For a given observation x the log likelihood can be written as

$$I_x(\theta) = log(f_\theta(x)) + log(c(\theta)) =< t(x), \theta > -log(d(\theta)) \tag{6}$$

which has a Monte Carlo approximation given by

$$I_{n,x}(\theta) =< t(x), \theta > -log(d_n(\theta)) \tag{7}$$

Then, for any fixed θ,

$$I_{n,x}(\theta) \to I_x(\theta) \tag{8}$$

almost surely by expression 5. Thus, under the concavity of $I_{n,x}(\theta)$ and $I_x(\theta)$ (a consequence of the exponential nature of the objective function in this case, though not a not necessary condition for the expression above to be true) we have that

$$\hat{\theta}_n \to \hat{\theta}\text{almost surely} \tag{9}$$

Using simulations from one distribution P_θ we are able to approximate $\hat{\theta}$ regardless of the intractability of the normalising constant. The algorithm is iterated numerous times, and we note that there is a dependence on a reasonable starting point (i.e. something within the realm of reality).

- **Monte Carlo MLE methods**
  - Uses importance sampling integration to find $\frac{\kappa(\theta')}{\kappa(\theta)}$

  - Given a sample from the ERGM at point θ^t we then update θ^{t+1}
  - There are a number of benefits in using the MCMLE approach - namely that it uses the entire distribution of y^{θ^t} instead of just the first moment, incorporates nonlinear effects on θ to determine the next guess, and automatically determines the step length = less steps to convergence.
  - That said, it is even more susceptible to a bad θ^0 guess.
  - Can also fail for non-curved ERGMs when the convex hull of the simulated statistic does not contain $g(y^{obs})$ (i think this just means the observed graph itself?)
  - Hummel et. al (2012) proposed two ways to address the bad first guess problem. Firstly, assuming $g(Y)$ is log-normal, then $exp[\eta(\theta') - n(\theta)^T g(Y)]$ is lognormal and we can approximate the expectation such that the maximiser depends only on the first two moments of y^{θ^t} which has a closed form solution for non-curved ERGMs. The second fix is a partial stepping technique that shifts the observed statistic to the centroid of the simulated statistic, reducing the step but preserving its direction. This approach *survives* poor starting values but does not make make immune to them.

- There are two approaches to finding good values for y^{θ^0}. One is the maximum pseudo/composite likelihood estimation (MPLE/MCLE) and the second is contrastive divergence.

3.3.1 Robbins-Monro/ Stochastic Approximation

Stochastic Approximation methods were among the first approaches to estimating the ERGM parameters by finding the actual MLE. First introduced by Robbins and Monro in 1951, the stochastic approximation method aims to iteratively find the roots of an optimisation function as represented by an expected value. It does so by iteratively solving equations of the form

$$E(Z_\theta) = 0 \tag{10}$$

Where Z_θ are the observed network statistics under the unknown θ.

For a root proposed root $\hat{\theta}$ the Robbins and Monro algorithms states that $\lim_{\hat{\theta_n} \to \inf} = \theta$, the true root. This is done by iterating through

$$\theta_{n+1} = \theta_n - \alpha_n(N(\theta) - \alpha) \tag{11}$$

In the context of ERGMs, this is ...

A Markov chain is a sequence of random variables such that the value taken by the random variable only depends upon the value taken by the previous variable. We can hence consider a network in the form of an adjacency matrix in which each entry is a random variable. By switching the values of these variables to 0 or to 1(adding or removing a link from the network) one can generate a sequence of graphs such that each graph only depends upon the previous graph. This would be a Markov chain. The hypothesis is then that if the value at step t is drawn from the correct distribution than so will the value at step t+1. Unlike regular Monte-Carlo methods, the observations that are sampled are close to each-other since they vary by a single link. However, one would need a method for selecting which variable should change state in order to get closer to the MLE, this is done using the Metropolis-Hastings algorithm or the Gibbs sampler

Snijders introduced the Stochastic Approximation as an extension of the original Robbins-Monro algorithm in his paper Snijders (2001). The algorithm in his subsequent paper (Snijders (2002)) takes the ideas proposed for this *stochastic actor oriented model* and uses a simplification due to the scaling matrix used in the first phase (itself a matrix of derivatives) can be estimated using the covariance matrix of the generated statistics rather than by a finite quotient difference.

Due to the normalising constant in ERGMs $k\boldsymbol{\theta}$ typically being unknown, the means of estimating the probability distribution remains an intractable problem. And while stochastic approximation can be used to estimate the parameters, the rise of computing power has meant Markov Chain Monte Carlo MCMC simulation is typically used to address the value of $k\boldsymbol{\theta}$ (and the MLE as a whole).

Using MCMC estimation typically via the Metropolis-Hastings algorithm estimates network parameters by using the Markov process that asymptotically reaches a unique stationary distribution. A new state, x' is proposed with some probability given by $q(x \rightarrow x'$.

Snijder's algorithm consists of three main phases, with each phase adopting the Metropolis-Hastings sampling approach to draw from the model. Before diving into each of the phases, the developer has control on two parameters at this stage: the burn-in, and the gain factor. The burn in is a MCMC-specific hyperparameter and needs to be set for any MCMC-based method, while the gain factor controls the size of the steps the algorithm takes in reaching the MLE. A large gain factor means reaching the general vicinity of the MLE quicker, albeit with less 'precise' movements.

Phase 1: Initialisation

The goal of Stochastic Approximation's first phase is to determine the scaling matrix D_0 and the initial values of the parameters being estimated using a small number of steps. The scaling matrix is used to scale the updates of the different elements of the parameter vector. At the end of the first phase, there is an optional initiation of the Newton-Raphson process (used in the subsequent step), $\hat{\theta}^{(N_1)} = \theta^{(1)} - \alpha_1 D^{-1}(\bar{u} - u_0)$ where the values D and $\bar{u}$ are respectively defined as

Phase 2: Optimisation

The second phase is the main phase of Stochastic Approximation, with a number of subphases each containing several iterations used to calculate $Y(n)$ according to the current parameter value $\hat{\theta}^{(n)}$. After of these minor iterations, the θ is updated according to the equation

$$\hat{\theta}^{(n+1)} = \hat{\theta}^{(n)} - \alpha_n D_0^{-1} Z(n) \tag{12}$$

Where α_n is a constant consistent through each subphase, and $Z_k(n) = P(n)u(1 - Y(n)) + (1 - P(n))u(Y(n)) - u_0$. At the end of each of these subphases, the algorithm estimates a new value for $\theta^{(n)}$, with the final estimate, $\hat{\theta}$, being the average of the $\theta^{(n)}$ estimated at the final subphase.

Note that with each subphase, the value of α_n is halved, so that the steps become incrementally smaller with more iterations (i.e. the algorithm makes smaller, more 'careful' steps as it reaches the final $\hat{\theta}$).

Phase 3: Conversion check and measurement calculation

As the previous phase calculated the final value for $\hat{\theta}$, the final phase of Stochastic Approximation seeks to estimate the covariance matrix of the estimator and $\Sigma(\theta)$.

3.4 Equilibrium Expectation

The focus of this paper, the equilibrium expectation algorithm by Byshkin et al. (2018) proposes a fast algorithm for exponential random graph model parameters using maximum likelihood estimation which in turn affords an increase in the scale of the network being estimated. The fast and scalable nature of the equilibrium expectation algorithm are a consequence of the

4 Method

Before considering analysis methodology, for an equivalent comparison of Equilibrium Expectation to other estimation methods, we must first implement Stivala's algorithm within the Statnet `ergm` package by Hunter et al. (2008).

The methods we will be benchmarking against are mentioned in Section 2 and detailed in Section 3.

To begin, we use the models used by Hummel et al. (2012) to compare the effectiveness of Equilibrium Expectation against MCMLE and Stochastic-Approximation as implemented in `ergm` under different combinations of hyperparameters, such as starting point, while controlling for other hyperparameters such as burn-in. The datasets we use to compare estimation algorithms are the E. Coli dataset for protein location sites, as well as Kapferer's Zambian tailor shop. For more detail on the datasets refer to Appendix A and B respectively.

We consider four models:

```
ecoli2 ~ edges + degree(2:5) + gwdegree(0.25, fixed = TRUE)
ecoli2 ~ edges + degree(2:5) + gwdegree(0.25, fixed = TRUE) + nodemix(”self”,
kapferer ~ edges + gwesp(0.25, fixed = TRUE) + gwdsp(0.25, fixed = TRUE)
kapferer ~ edges + gwdegree(0.25, fixed = TRUE) + gwesp(0.25, fixed = TRUE) +
```

¡reason for why we had each of these equations¿

For each of the network, we then considered three different starting point configurations across each of the three aforementioned estimation methods:

1. Starting from a parameter vector of all zeros, i.e. $\boldsymbol{\theta_0} = \mathbf{0}$.
2. Starting from the maximum pseudo-likelihood estimate.
3. Starting from a parameter vector of mostly zeros, except for the edges of the network which were calculated beforehand with a run of the network at the edges only.

1. Estimate parameters on the same dataset using EE, RM, and MCMCMLE
2. Modify the starting points of the algorithm
3. Test a missing data implementation

What constitutes a more effective ”algorithm”?

5 Results

6 Notes from papers

6.1 Using contrastive divergence to seed Monte Carlo MLE for exponential-family random graph models (Krivitsky, 2015)

- This paper aims to address the starting point problem of MCMC methods using something known as contrastive divergence which uses a series of abridged MCMC runs instead of running one to stationarity.
- Intractable normalizing constants make life in the ERGM space challenging hence the need for MCMC methods. Sampling techniques need a starting point. Performance and feasibility depend on this value.
- Combined with importance sampling MC MLE to find the initial values
- What are curved models? What is a binary ERGM? (i.e. h(theta) proportional 1)
- The body of techniques currently used to find the MLE can be split into two broad categories: stochastic approximation (SA) and MCMLE.
- **Stochastic approximation methods**
  - The first methods were these types
  - Given a guess θ^t, these techniques simulate a sample from ERGM(θ^t) and update
  - We require an initial guess

  - A bad initial guess may result in issues of near-degeneracy, concentrating on the edge of the convex hull of the set of attainable statistics.
  - If all possible values for the parameter θ are not in the realm of Real (q dim) numbers, then MCMC sampling for θ^0 will diverge in the first place
  - Poor choice of θ^0 may induce a very dense network (after simulation), thus resulting in a large impact on computation
  - SA methods are inefficient as each new θ^t requires a burn-in and a sample to estimate $U(\theta^t)$ - **what does this mean? why do we need a new MCMC for each θ^t?** Note here that U is the score function used to find the MLE.
  - SA will fail if the entirety of the sample lies on the edge of the convex hull.

- **Composite likelihood methods**
  - Before simulation methods became popularised we had pseudolikelihood estimators (Strauss + Ikeda, 1974). Here the likelihood was approximated and $y_{i,j}$ is an indicator for the presence of ties. This lead to a score fuction that is a nonlinear logistic regression.
  - MPLE can be quite different from the MLE and it's mainly used to initialise them (what is meant by ”them”?)
  - MPLE/MCLE result in a multinomial model to enumerate the set of possible pairs of toggles - thus adding more burden to the modeller.
- **Contrastive divergence**
  - Hinton (2002) suggested we don't run the MCMC simulation to convergence, but instead make parallel updates starting at the observed data and calculating the gradient based on that.
- **Questions for Pavel**
  - Emphasis in this research on methods that do well for θ^0? ”Poor choice may result in a dense network” = bad simulation? Are some methods more sensitive to initial values e.g. importance sampling vs. Robins Munro
  - Often in the CD paper we refer to curved vs. non-curved ERGMs - what is the difference and is this something that I should be separately learning about? When you have an exponential family model you have exponential form - you can transform theta before taking the dot product. Now you have model parameter and the mapping mapped to a vector it needs to be the same length of the statistics. size of theta ¡ g = strict definition of curved family. From the point of view of computation, whether the theta is shorter or not/transformation. Not curved ones are special cases of curved ones. Helpful for the algorithm to accommodate some transformations of theta
  - In what instances can a model only be done via MCMC simulation and not the MPLE/MCLE methods?
  - Confusing result on page 7 regarding contrastive divergence as discovered by Asuncion et al. (2010).
  - What is CD_∞ Confirm my understanding: CD_1 is the MPLE equivalent (it finds the MLE under the pseudo method), while CD_∞ is the actual MLE?
  - The three approaches to finding $theta^0$
- **Papers off the back of this I think I should read**
  - Hinton (2002)
  - Wang et. al. (2014)
  - Hummel (2011) perhaps?
  - Hummel et. al. (2012) (partial stepping techniques)

How well do different algorithms behave if say you start with a vector of 0s for theta, or try different configurations for the starting values

6.2 Fast Maximum Likelihood Estimation via Equilibrium Expectation for Large Network Data

- This paper proposes a fast algorithm for MLE which allows for larger network estimation. It leverages properties of the Markov chain at equilibrium.
- Existing approaches for MLE such as MCMCMLE, MoM, and Bayesian estimation use iterative algorithms that successively update θ until the expectation of the network under the statistics is equivalent to the statistic of the observed network. To do this, MCMC draws a large number of simulated networks for various values of θ, $x_s(\theta)$. The simulated network x_s is a network drawn from probability distribution $\pi(x, \theta)$.
- Each time a new simulated network is drawn, we need to satisfy the convergence criteria $\Sigma_{x,x'}\pi(x, \theta)P(x \to x', \theta)(z_A(x') - z_A(x))$. This is very expensive computationally.
- Consider the equation $E_{\pi(\theta)}(\Delta_{Z_A}(x, \theta)) = 0$. This suggests that if network x is drawn from probability distribution $\pi(x, \theta)$ then the expected value of $\Delta_{Z_A}(x, \theta)$ is 0. Only valid when at the limiting distribution.
- $\Delta_{Z_A}(x, \theta)$ is found using Monte Carlo integration. Assuming that are s_i Monte Carlo sample of networks that are i.i.d from $\pi(x, \theta)$ then the expectation can be calculated. This is obviously not a realistic assumption which will later be removed.
- I.e. $E_{\pi(\theta)}(\Delta_{Z_A}(x, \theta)) = \frac{1}{n}\Sigma_i\Delta_{Z_A}(x_{s_i}, \theta)$
- With enough sample of networks we can efficiently compute the LHS and then solve with respect to θ.
- No burn-in time using MC integration so it's much more efficient.
- Ergodicity of systems suggests that if the network x_s is very large then the true θ^* may be estimated from $f_A(x_{obs}, \theta)$, thus dropping the need to sum. This leads to the result $\Delta Z_A(x_s, \theta^{EE}) = 0$.
- If the network x_s is very large then $\theta^{EE} = \theta^*$
- **However in reality we only have ONE x_{obs}**
- In reality - this looks like Contrastive Divergence (CD) as applied to ERGM parameter estimation (instead of for finding θ^0)
- So what's done instead is that if a solution to the method of moments exists then we can draw an x_s from $\pi(x, \theta)$ such that we satisfy $z_A(x_s) = z_A(x_obs)$. The network x_s can be drawn from an MCMC simulation. θ is updated iteratively until we have a solution for $Z_A(x_s, \theta^{EE}) = 0$. It employs Metropolis-Hastings
- Since we are not drawing many simulated networks from various θ the algorithm is considerably faster.
- To start the EE algorithm, we use CD1 estimate as the starting point. EE is more sensitive to K_A then it is to θ_0. K_A is subtracted from the old θ to produce the next θ. This value is used to make sure that θ_A increases when dz_A is negative and decrease when dz_A is positive. We also require that the exponential family expectation is a monotonically increasing function in θ
- **Questions for Pavel**
  - In reality we only have one x_{obs} - but why can't we simulate more networks based on a starting θ and use those?

6.3 Markov Chain Monte Carlo Estimation of Exponential Random Graph Models

- This paper is about the simulation and MCMC estimation of exponential random graph models.

- A major problem with ERGMs is that for certain parameter values they can have bimodal (or multimodal) distributions for the sufficient statistics such as the number of ties

- The modality is reflected in the outcome space being divided into two or more regions whereby the usual MCMC approaches have long sojourn times, with a negligible probability of moving from one region to the next. This leads to slow convergence

- MCMC algorithms must be able to make transitions from a given graph to a very different graph. It is proposed to include transitions to the graph complement as updating steps to improve the speed of convergence to the target distribution

- Recall that a random graph, according to Strauss and Frank (1986) is a Markov graph if there are finite numbers of nodes and if edges between disjoint pairs are independent conditional on the rest of the graph

- Frank and Strauss proposed a simulation based approach to approximate the MLE of any one of the three θ_k (talking here about the standard p* triad model), given that the other two are fixed at 0. They also proposed a logistic regression method to estimate the full θ (pseudo MLE).

- While the pseudo MLE approach is nice computationally and intuitively appealing, the properties of the resulting estimator for ERGMs are unknown.

- In addition, the pseudo MLE is not a function of the entire sufficient statistic $u(Y)$ which implies that it is not an admissible estimator for squared error loss function (lehmann, 1983)

- Many of the older MCMC estimation methods relied on MC simulations of the Markov graph at current parameter values. However, simulation algorithms for the ERGM methods can suffer from severe convergence problems not really outlined previously

- For many models (including triad), there is a large region in which the demarcation between the subset of parameters θ leads to graphs with relatively low expected densities, and the subset of parameters θ that lead to graphs with high expected densities; is quite sharp. Paramerters in or near the demarcation zone can have bimodal density.

- With more nodes, the demarcation becomes more marked. (demarcation = fixing the boundary)

- One MCMC process: assuming you are at $\theta^(n)$, a MC simulation of the Markov graph is made which is used to estimate the moments of the distribution, which are then used to make an expansion approximating $\mu(\theta)$ for θ in the neighbourhood of $\theta^(n)$. This is then used to solve the moment equation to provide an estimate for $\theta^(n+1)$

- This paper uses a version of the Robbins-Monro algorithm, which is itself considered by some a MC variant of the Newton-Raphson algorithm.

- Robbins-Monro is a stochastic iterative algorithm intended to solve equations of the form $E\{Z_\theta\} = 0$. In the ERGM case, consider $Z_\theta = u(Y) - u_0$ where $u_0 = u(y)$ is the observed value of the sufficient statistic.

- The iteration step in the Robbins-Monro algorithm with step-size a_n is $\hat{\theta}^{(n+1)} = \hat{\theta}^{(n)} - a_n D_n^{-1} Z(n)$ where $Z(n)$ are random variables such that the conditional distribution of $Z(n)$ given $Z(1)$,...,$Z(n-1)$ is the distribution of $Z_(\theta)$ obtained for $\theta = \hat{\theta}^(n)$

- The stochastic process $\hat{\theta}^(n)$ generated by this MC simulation is a Markov chain, because for any n_0 the sequence $\{\hat{\theta}^(n)\}$ for $n > n_0$ depends on the past value via the last value, $\hat{\theta}^(n_0)$

- **Questions for Pavel**

  - There is a lot of mention of the triad model in these older papers. With simulation based methods, how often are they impervious to model specification?
  - Is simulating an ERGM a different problem to estimating an ERGM?

7 Meeting 24/03

- **Meeting preparation**
  - Papers read in the last fortnight: New specifications for ERGM models (Snijders et. al.), inference for curved exponential family random models (Hunter and Handcock), Modelling Social Networks from Sampled Data (Handcock and Gile)
  - A lot of material in the themes of simulation, inference and missing data. Need to start boiling down all of this raw material into a coherent thesis that I can start writing.
  - High level I am doing a bench marking of the equilibrium expectation algorithm - can we incorporate missing data effectiveness into the tests? Are there other algorithms we want to benchmark?
  - Any meta analysis papers that are worth looking into to gain some kind of inspiration for benchmarking various inference algorithms? What are the types of things people look for including speed, scalability, robustness, etc
  - Different types of error: are we only looking at MCMC error and MLE error? In the case of curved ERGMs are there any other errors we are interested in testing? What about finding the expectation and variance in curved ERGMs?
  - On the topic of curved ERGM inference - I understand the traditional Thompson and Geyer approach to MCMC. Generate initial guess, approximate the difference between current and initial by a MC integration, then approximate r and continue. But now are we talking iterative methods like Newton-Raphson?
  - Is there any relation with this work to simulation of ERGMs?
- **Meeting notes**
  - MCMC and Robbins Monro are the two basis methods for fitting these models, MCMCMLE, while Robbins Monro draws small samples and downweights them... These are competitors to equilibrium expectation
  - MCMC errors are more obvious, versus Robbins Monro not really having a good error approximator. The same goes for equilibrium expectation.
  - Okobayashi and Geyer
  - Missing data you can still calculate the likelihood - two normalising constants on top of each other. Each paramter updates requires you to run MCMC twice.
  - For Robbins Monro - there's a question of how do you run that... two MCs running at once? one for constrained and one for unconstrained etc... implementing this is a stretch goal
  - Reach goal for equilibrium expectation handling data
  - Go to ergm.R - check the mainfit ¡- switch(...). This is where the equilibrium expectation alogirthm will go
  - Method to prefix in control.ergm.R which contains the parameters
  - EE is infrequent complex updates so it should probably be written in C
  - Go to SEXP_DISPATCH_MCMCPhase12 as an example for the implementation of the algorithm
  - Taking the outputs of MCMC and using them
  - Check out MCMC.c.template.do_not_include_directly.h

  - Testing a hybrid method - starting with Robbins Monro and then going into MCMCMLE

Appendices

A E. Coli dataset

B Kapferer's Zambian tailoring shop dataset

References

Maksym Byshkin, Alex Stivala, Antonietta Mira, Garry Robins, and Alessandro Lomi. Fast maximum likelihood estimation via equilibrium expectation for large network data. *Springer Nature*, 2018.

Charles J. Geyer and Elizabeth A. Thompson. Constrained monte carlo maximum likelihood for dependent data (with discussion). *Journal of the Royal Statistical Society Series B*, 1992.

Ruth M. Hummel, David R. Hunter, and Mark S. Handcock. Improving simulation-based algorithms for fitting ergms. *Journal of Computation and Graphical Statistics*, 2012.

David R. Hunter and Mark S. Handcock. Inference in curved exponential family models for networks. *Journal of Computation and Graphical Statistics*, 2006.

David R. Hunter, Mark S. Handcock, Carter T. Butts, Steven M. Goodreau, and Martina Morris. ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of Statistical Software*, 2008.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 1951.

Tom A.B. Snijders. The statistical evaluation of social network dynamics. *Sociological Methodology*, 2001.

Tom A.B. Snijders. Markov chain monte carlo estimation of exponential random graph models. *Journal of Social Structure*, 2002.

David Strauss and Michael Ikeda. Pseudolikelihood estimation for social networks. *Journal of the American Statistical Association*, 1990.