Mass Preserving Exponential Random Graph Model

Lin Yuan
yuanl@purdue.edu
School of ECE
Purdue University

Sergey Kirshner
skirshne@purdue.edu
Dept. of Statistics
Purdue University

Robert Givan
givan@purdue.edu
School of ECE
Purdue University

Abstract

Exponential random graph models (ERGMs) are commonly used for modeling network data. However, they often suffer from degeneracy manifested in having the learned model place very little mass on or near the observed network(s). We propose a *mass preserving* ERGM, which, in addition to matching the mean statistics for the features used with ERGMs, also ensures the resulting model places at least a predetermined amount of mass on graphs similar to the observed graphs. The resulting model is thus directly resistant to the degeneracy of assigning near zero probability to the observations. This claim is further confirmed by our experiments on several social networks from small to moderate in size (16 to 1000+ nodes).

1 Modeling Graphs with Mass Preserving ERGMs

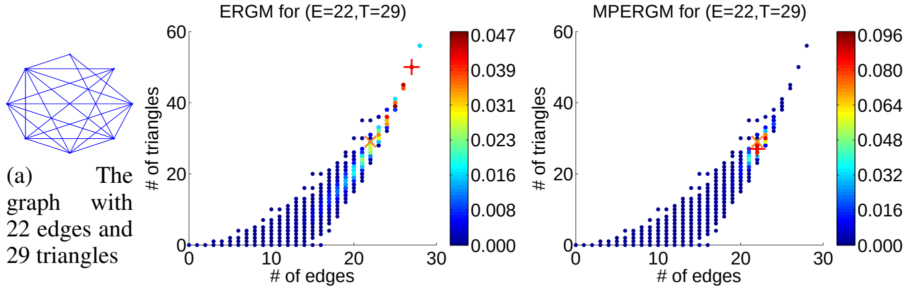
An ERGM (or $p^\star$ model) is an exponential family model over a space $\mathcal{G}_n$ of n-node graphs[1] which uses graph statistics as its features (sufficient statistics). The model is very popular in the social network literature [ERGMs, e.g., 1–3] due to the inherited virtues of the exponential family and the ability to match the graph statistics. The features of the model are typically motivated by the properties of the networks that are of interest to domain scientists (e.g., sociologists); some examples of such features are the number of edges, $t_e(G) = \sum\sum_{1\le i<j\le n} e_{ij}$, and triangles $t_\triangle(G) = \sum\sum\sum_{1\le i<j<k\le n} e_{ij}e_{ik}e_{jk}$, where for $G \in \mathcal{G}_n$, $e_{ij} = 1$ if there is an edge between nodes i and j, and 0 otherwise. The probability mass function for a graph $G \in \mathcal{G}_n$ is defined as

$$P(G|\boldsymbol{\lambda}) = \frac{1}{Z(\boldsymbol{\lambda})} \exp\langle \boldsymbol{\lambda}, \boldsymbol{t}(G)\rangle, \qquad Z(\boldsymbol{\lambda}) = \sum_{G\in\mathcal{G}_n} \exp\langle \boldsymbol{\lambda}, \boldsymbol{t}(G)\rangle. \tag{1}$$

A standard setup for the problem is to estimate a set of parameters $\boldsymbol{\lambda}$ for observed networks $G_1, \ldots, G_m \in \mathcal{G}_n$ which is commonly done by maximizing the likelihood of the data (MLE).

However, the combination of having few training instances (a common setting is $m = 1$, i.e., only *one* training example, which we denote as $G^\star$) and a very large sample space ($|\mathcal{G}_n| = 2^{\binom{n}{2}}$) leads to issues with parameter estimation often referred to as degeneracy of the estimated model. The first type of degeneracy has been traced to the proximity of the feature vector $\boldsymbol{t}(G^\star)$ to the relative boundary of the convex hull of $\mathcal{H} = \{\boldsymbol{t}(G) : G \in \mathcal{G}_n\}$, the set of all possible feature values for graphs under consideration [2, 4]. When $\boldsymbol{t}(G^\star) \in \operatorname{rint}(\operatorname{conv}(\mathcal{H}))$, MLE exists and is unique. However, the estimated model may place little probability mass in the vicinity of the only observation $G^\star$ (c.f. Figure 1), with a large portion of the mass placed on unrealistic graphs (e.g., empty or complete graphs). There have been several approaches to fixing the latter (second type of) degeneracy which can be summarized in two categories: 1) modifying the geometry [5–7], and 2) limiting exploration in the canonical parameter space [8, 9]. Our proposed approach is also aimed at alleviating this latter type of degeneracy.

[1] In this paper, we consider undirected graphs. An extension to directed graphs is straightforward.



(a) The graph with 22 edges and 29 triangles

(b) Probability mass for ERGM

(c) Probability mass for MPERGM

Figure 1: Illustration of degenerate ERGM and non-degenerate MPERGM for $\mathcal{G}_8$. The models are trained based on the observation $t_e(G^\star) = 22, t_\triangle(G^\star) = 29$. The orange × is the observed statistics, and the red + is the mode of the learned model. The color bar on the right from red to blue represents the probability mass changing from high to low.

1.1 Mass Preserving ERGMs

We propose to modify ERGMs by introducing additional *mass-preservation* constraints, ensuring that a portion of probability mass is concentrated around the observations. Recall that an exponential family model can be viewed as a solution to maximizing the entropy of a distribution while matching the moments for the features:

$$P(G) = \arg\max_{P'} H_{P'}[G] \quad \text{subject to} \quad E_{P'}[\boldsymbol{t}(G)] = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{t}(G_i). \tag{2}$$

The components of the parameter vector $\boldsymbol{\lambda}$ correspond to the constraints for individual features in (2). One possible approach to guarantee that graphs generated by the model are similar to the observed is to add constraints to concentrate the mass on the graphs with feature values close to those of the observed. We accomplish this by adding the constraints on the mass in the neighborhood of each observation. An intuitive way to introduce such constraint is using the indicator functions $\mathcal{I}_{G_i}(G) = 1$ if and only if $\|\boldsymbol{t}(G) - \boldsymbol{t}(G_i)\|_2 < \frac{h}{2}$ for a predefined bandwidth $h > 0$. Then $E_P[\mathcal{I}_{G_i}]$ is the mass inside the ball in the space of feature values centered at $\boldsymbol{t}(G_i)$ according to P. One can replace the indicator function $\mathcal{I}_{G_i}$ with a kernel function $t_a^i(G) = K_{\mathbf{H}}(\boldsymbol{t}(G); \boldsymbol{t}(G_i))$ (with bandwidth parameter $\mathbf{H}$) to obtain a weighted mass indicator function. Adding constraints

$$min_i \leq E_{P'}\left[t_a^i(G)\right] \leq max_i \tag{3}$$

to (2) ensures that weighted mass over graphs with features similar to G_i is at least min_i and at most max_i. The solution to (2) with additional constraints in (3) (provided it exists) is also within the exponential family [10],

$$P(G) = \frac{1}{Z(\boldsymbol{\lambda}, \boldsymbol{\lambda}_a)} \exp\left[\langle \boldsymbol{\lambda}, \boldsymbol{t}(G)\rangle + \langle \boldsymbol{\lambda}_a, \boldsymbol{t}_a(G)\rangle\right] \tag{4}$$

with $(\boldsymbol{\lambda}, \boldsymbol{\lambda}_a)$ found by maximizing the concave objective function

$$l(\boldsymbol{\lambda}, \boldsymbol{\lambda}_a) = \frac{1}{m}\sum_{i=1}^{m} \langle (\boldsymbol{\lambda}, \boldsymbol{\lambda}_a), (\boldsymbol{t}(G_i), \boldsymbol{t}_a(G_i))\rangle - \ln Z(\boldsymbol{\lambda}, \boldsymbol{\lambda}_a) - \sum_{i=1}^{m} \beta_i |\lambda_i^a|$$

where each β_i is determined based on min_i and max_i. Details can be found in [11]. We refer to the model in Equation 4 as *mass preserving* ERGM (MPERGM).

There are several challenges with parameter estimation, most encountered before in ERGM fitting [e.g., 6]: in particular, the gradient cannot be computed in closed form except for graphs of small size ($\mathcal{G}_n$ for $n \leq 11$). We therefore apply the MCMC-MLE approach of Hunter and Handcock [12], computing $E_P[\boldsymbol{t}(G)]$ as a sampled average $\frac{1}{S}\sum_{j=1}^{S}\left(\boldsymbol{t}\left(G^j\right)\right)$ where $G^{1:S} \overset{i.i.d}{\sim} f(G|\boldsymbol{\lambda}, \boldsymbol{\lambda}_a)$. There are, however, two complications with this approach. One, graph sampling from ERGMs is performed using Gibbs sampling and is computationally expensive. Therefore, graphs $G^{1:S}$ are re-sampled only once in several iterations, and reused for other iterations with weights equal to the posterior probabilities. Two, the resulting distribution over graphs can be multi-modal, and according to Jin and Liang [9], Hunter and Handcock [12], the sampler can get stuck around the closest mode leading to an incorrect estimate of the gradient. Thus instead of performing line search, we use the direction of the gradient with a predefined step-size.

2 Experimental Evaluation

We evaluate the fit of the estimated models by comparing local statistics of the observed graph to that of the samples generated from the estimated distribution.[2]

Table 1: Social network data sets [11]. `g8`: The 8-node graph as in Figure 1(a); `Do`: The dolphins data set; `Kp`: The Kapferer data set; `Fl`: The Florentine Business data set; `Fa`: The Faux.Mesa.High data set; `Ja`: The Jazz data set; `Ad`: The AddHealth data set; `Fb`: The Facebook data set; `Em`: The Email data set.

	g8	Do	Kp	Fl	Fa	Ja	Ad	Fb	Em
$\|V\|$	8	62	39	16	206	198	803	1024	1133
$t_e(G^\star)$	22	159	158	15	203	2742	1985	1012	5451
$t_\triangle(G^\star)$	29	95	201	5	62	17899	649	116	5343
No. unique sampled graph	99	100	100	100	100	100	100	96	100
No. unique features	33	33	30	27	14	70	72	74	70
No. max graph-edit distance	22	306	263	39	341	1435	999	385	1275

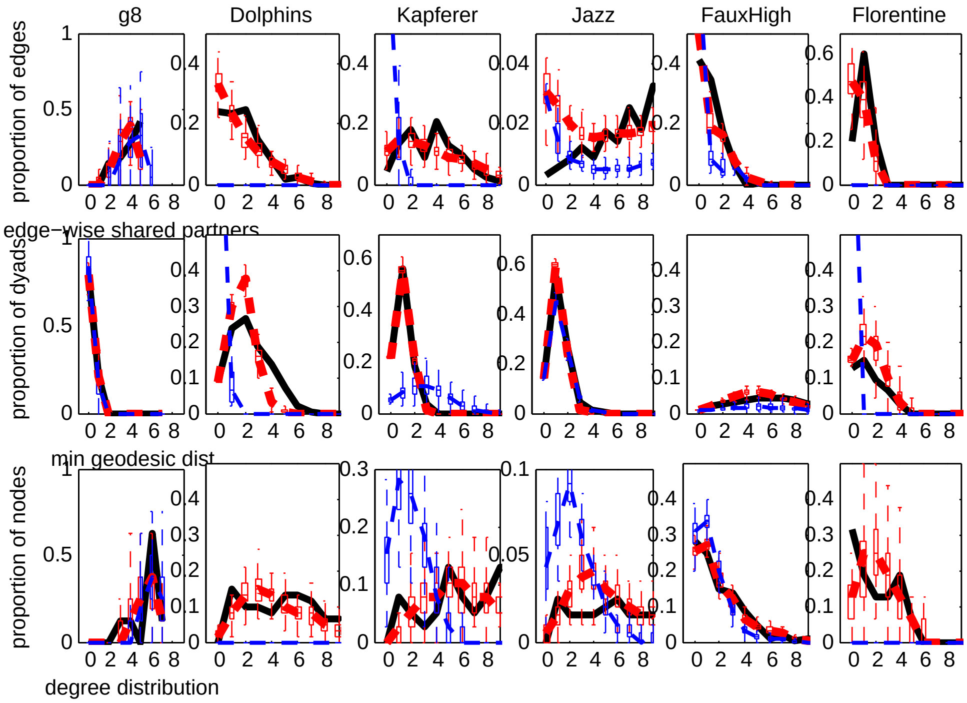


Figure 2: Goodness of fit for small graphs. Gaussian kernel functions are used for MPERGM. ERGM is shown in blue dashed lines, and MPERGM is shown in red dashed lines. Black lines are the statistics for $G^\star$, being closer to black line means better fit.

We make use of three sets of local statistics commonly used as goodness-of-fit measures for ERGMs [6]: the *degree distribution* (the proportion of nodes with exactly k neighbors), *edgewise shared*

[2] See [6] for a discussion on the evaluation of fit for social networks.

partner distribution (the proportion of edges joining nodes with exactly k neighbors in common), and the *minimum geodesic distance* (the proportion of connected node-pairs which have a minimum distance of k).

We consider the number of edges and triangles as sufficient statistics, $\boldsymbol{t}(G) = (t_e(G), t_\triangle(G))$. First, we consider the toy domain of graphs with 8 nodes, $\mathcal{G}_8$. We enumerate all possible $K = 12346$ non-isomorphic graphs and resulting feature tuples, and compute probability mass entries $\pi_1, \ldots, \pi_K$. We trained our MPERGM with a Gaussian kernel function with $h = 8$ and $\beta = 0.2$. Figure 1 shows that the MPERGM puts larger probability mass around $G^\star$.

We also estimated MPERGMs for several social network data sets, ranging in the number of nodes from 16 to 1024, and with varying density of edges. Since $|\mathcal{G}_n|$ is too large to enumerate, the graphs are drawn using a Gibbs sampler, and the parameters for the MPERGMs (and ERGMs, using the R package `ergm` [6]) are estimated using MCMC-MLE. For each estimated model, the statistics in Figure 2 (see [11] for large networks) were generated from 100 sampled graphs obtained by running Gibbs with 1000 iterations for burn-in and 100 iterations between samples. We initialized our Markov chain with the example graph; however, we lack documentation on what initial state is used by the R package `ergm` procedure. We used a set of hand-tuned step-sizes and h values for different data sets, and re-scaled the edge and triangle features by factors of $\frac{1}{t_e(G^\star)}$ and $\frac{1}{t_\triangle(G^\star)}$. Empirically, we find $h \approx 8$ and a predefined step-size 10 works well for small graphs. In Figure 2, ERGM is degenerate for the `Florentine` and `Dolphins` dataset, because most sampled graphs have 0-degree nodes (third row), while MPERGM is able to generate samples scoring a similar set of graph statistics to $G^\star$. In order to investigate the variance of the learned MPERGM, we count the number of samples that are different in structure (not counting isomorphism) or different in features (number of edges and triangles), while recording the maximum *graph-edit distance*[3] to the initial state over all sampled graphs. The results in Table 1 suggests that our sampler explores $\mathcal{G}_n$ with a considerable range.

References

[1] S. Wasserman and P. Pattison. Logit models and logistic regression for social networks: An introduction to Markov graphs and p* model. *Psychometrii*, 61(3), September 1996.

[2] M. S. Handcock. Assessing degeneracy in statistical models of social networks. Technical Report 39, Center for Statistics and the Social Sciences, University of Washington, 2003.

[3] G. Robins, T. Snijders, P. Wang, M. Handcock, and P. Pattison. Recent developments in exponential random graph (p*) models for social networks. *Social Networks*, 29(2):192 – 215, 2007.

[4] A. Rinaldo, S. E. Fienberg, and Y. Zhou. On the geometry of discrete exponential families with application to exponential random graph models. *Electronic Journal of Statistics*, 3:446–484, 2009.

[5] M. S. Handcock, D. R. Hunter, C. T. Butts, S. M. Goodreau, and M. Morris. `statnet`: Software tools for the representation, visualization, analysis and simulation of network data. *Journal of Statistical Software*, 24(3), 2008.

[6] D. R. Hunter, M. S. Handcock, C. T. Butts, S. M. Goodreau, and M. Morris. `ergm`: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of Statistical Software*, 24(3), 2008.

[7] D. Lunga and S. Kirshner. Generating similar graphs from spherical features. In *Ninth Workshop on Mining and Learning with Graphs (MLG'11)*, San Diego, CA, August 2011.

[8] A. Caimo and N. Friel. Bayesian inference for exponential random graph models. *ArXiv e-prints*, July 2010.

[9] I. H. Jin and F. Liang. Fitting social network models using varying truncation stochastic approximation MCMC algorithm. *Journal of Computational and Graphical Statistics*, In Press, 2012.

[10] M. Dudik, S. Phillips, and R. Schapire. Maximum entropy density estimation with generalized regularization and an application to species distribution modeling. *Journal of Machine Learning Research*, 8: 1217–1260, Jun 2007.

[11] L. Yuan, S. Kirshner, and R. Givan. Estimating Densities with Non-Parametric Exponential Families. *ArXiv e-prints*, June 2012. URL `http://arxiv.org/abs/1206.5036`.

[12] D. R. Hunter and M. Handcock. Inference in curved exponential family models for networks. *ASA, Journal of Computational and Graphical Statistics*, 15(2), 2006.

[3] minimum number of unique edge-flips to change one graph to another