Discrete Temporal Models of Social Networks

Steve Hanneke and Eric Xing

Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213 USA
{shanneke,epxing}@cs.cmu.edu

Abstract. We propose a family of statistical models for social network evolution over time, which represents an extension of Exponential Random Graph Models (ERGMs). Many of the methods for ERGMs are readily adapted for these models, including MCMC maximum likelihood estimation algorithms. We discuss models of this type and give examples, as well as a demonstration of their use for hypothesis testing and classification.

1 Introduction

The field of social network analysis is concerned with populations of *actors*, interconnected by a set of *relations* (e.g., friendship, communication, etc.). These relationships can be concisely described by directed graphs, with one vertex for each actor and an edge for each relation between a pair of actors. This network representation of a population can provide insight into organizational structures, social behavior patterns, emergence of global structure from local dynamics, and a variety of other social phenomena.

There has been increasing demand for flexible statistical models of social networks, for the purposes of scientific exploration and as a basis for practical analysis and data mining tools. The subject of modeling a static social network has been investigated in some depth. In particular, there is a rich (and growing) body of literature on the *Exponential Random Graph Models* (ERGM) [1–4]. Specifically, if N is some representation of a social network, and $\mathcal{N}$ is the set of all possible networks in this representation, then the probability distribution function for any ERGM can be written in the following general form.

$$\mathcal{P}(N) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left\{\boldsymbol{\theta}'\mathbf{u}(N)\right\}.$$

Here, $\boldsymbol{\theta} \in \mathbb{R}^k$, and $\mathbf{u} : \mathcal{N} \to \mathbb{R}^k$. $Z(\boldsymbol{\theta})$ is a normalization constant, which is typically intractable to compute. The $\mathbf{u}$ function represents the sufficient statistics for the model, and, in a graphical modeling interpretation, can be regarded as a vector of clique potentials. The representation for N can vary widely, possibly including multiple relation types, valued or binary relations, symmetric or asymmetric relations, and actor and relation attributes. The most widely studied models of this form are for single-relation social networks, in which case N is generally taken to be the weight matrix A for the network (sometimes referred to as a *sociomatrix*), where A_{ij} is the strength of directed relation between the i^{th} actor and j^{th} actor.

Often one is interested in modeling the evolution of a network over multiple sequential observations. For example, one may wish to model the evolution of coauthorship networks in a specific community from year to year, trends in the evolution of the World Wide Web, or a process by which simple local relationship dynamics give rise to global structure. In the following sections, we propose a model family that is capable of modeling network evolution, while maintaining the flexibility of ERGMs. Furthermore, these models build upon ERGMs, so that existing methods developed for ERGMs over the past two decades are readily adapted to apply to the temporal models as well.

2 Discrete Temporal Models

We begin with the simplest case of the proposed models, before turning to the fully general derivation. One way to simplify a statistical model for social networks is to make a Markov assumption on the network from one time step to the next. Specifically, if A^t is the weight matrix representation of a single-relation social network at time t, then we can assume A^t is independent of $A^1, \ldots, A^{t-2}$ given A^{t-1}. Put another way, a sequence of network observations $A^1, \ldots, A^t$ has the property that

$$\mathcal{P}(A^2, A^3, \ldots, A^t | A^1) = \mathcal{P}(A^t|A^{t-1})\mathcal{P}(A^{t-1}|A^{t-2}) \cdots \mathcal{P}(A^2|A^1).$$

With this assumption in mind, we can now set about deciding what the form of the conditional PDF $\mathcal{P}(A^t|A^{t-1})$ should be. Given our Markov assumption, one natural way to generalize ERGMs for evolving networks is to assume $A^t|A^{t-1}$ admits an ERGM representation. That is, we can specify a function $\boldsymbol{\Psi} : \mathbb{R}_{n \times n} \times \mathbb{R}_{n \times n} \to \mathbb{R}^k$ and parameter vector $\boldsymbol{\theta} \in \mathbb{R}^k$, such that the conditional PDF has the following form.

$$\mathcal{P}(A^t|A^{t-1}, \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta}, A^{t-1})} \exp\left\{\boldsymbol{\theta}' \boldsymbol{\Psi}(A^t, A^{t-1})\right\} \tag{1}$$

2.1 An Example

To illustrate the expressivity of this framework, we present the following simple example model. For simplicity, assume the weight matrix is binary (i.e., an adjacency matrix). Define the following statistics, which represent *density, stability, reciprocity,* and *transitivity*, respectively.

$$\Psi_D(A^t, A^{t-1}) = \frac{1}{(n-1)} \sum_{ij} A^t_{ij}$$

$$\Psi_S(A^t, A^{t-1}) = \frac{1}{(n-1)} \sum_{ij} \left[A^t_{ij} A^{t-1}_{ij} + (1-A^t_{ij})(1-A^{t-1}_{ij})\right]$$

$$\Psi_R(A^t, A^{t-1}) = n \left[\sum_{ij} A^t_{ji} A^{t-1}_{ij}\right] \Big/ \left[\sum_{ij} A^{t-1}_{ij}\right]$$

$$\Psi_T(A^t, A^{t-1}) = n \left[\sum_{ijk} A^t_{ik} A^{t-1}_{ij} A^{t-1}_{jk}\right] \Big/ \left[\sum_{ijk} A^{t-1}_{ij} A^{t-1}_{jk}\right]$$

The statistics are each scaled to a constant range (in this case $[0, n]$) to enhance interpretability of the model parameters. The conditional probability mass function (1) is governed by four parameters: θ_D controls the *density*, or the number of ties in the network as a whole; θ_S controls the *stability*, or the tendency of a link that does (or does not) exist at time $t-1$ to continue existing (or not existing) at time t; θ_R controls the *reciprocity*, or the tendency of a link from i to j to result in a link from j to i at the next time step; and θ_T controls the *transitivity*, or the tendency of a tie from i to j and from j to k to result in a tie from i to k at the next time step. The transition probability for this temporal network model can then be written as follows.

$$\mathcal{P}(A^t|A^{t-1}, \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta}, A^{t-1})} \exp\left\{ \sum_{j\in\{D,S,R,T\}} \theta_j \Psi_j(A^t, A^{t-1}) \right\}$$

2.2 General Models

We can generalize the form of (1) by replacing $A^1, A^2, \ldots, A^T$ with general networks $N^1, N^2, \ldots, N^T \in \mathcal{N}$, which may include multiple relations, actor attributes, etc. Furthermore, we generalize the Markov assumption to allow any K-order dependencies, so that the previous discussion was for $K = 1$. In this case, the function $\boldsymbol{\Psi}$ is also generalized by $\boldsymbol{\Psi} : \mathcal{N}^{K+1} \rightarrow \mathbb{R}^k$. The fully general model can therefore be written as

$$\mathcal{P}(N^{K+1}, N^{K+2}, \ldots, N^T|N^1, \ldots, N^K, \boldsymbol{\theta}) = \prod_{t=K+1}^{T} \mathcal{P}(N^t|N^{t-K}, \ldots, N^{t-1}, \boldsymbol{\theta}),$$

where

$$\mathcal{P}(N^t|N^{t-K}, ..., N^{t-1}, \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta}, N^{t-K}, ..., N^{t-1})} \exp\{\boldsymbol{\theta}' \boldsymbol{\Psi}(N^t, N^{t-1}, ..., N^{t-K})\}\,.$$

Note that specifying the joint distribution requires one to specify a distribution over the first K networks. This can generally be accomplished fairly naturally using an ERGM for N^1 and exponential family conditional distributions for $N^i|N^1 \ldots N^{i-1}$ for $i \in \{2, \ldots, K\}$. For simplicity of presentation, we avoid these details in subsequent sections by assuming the distribution over these initial K networks is functionally independent of the parameter $\boldsymbol{\theta}$.

3 Estimation

The estimation task for models of the above form is to use the sequence of observed networks, $N^1, N^2, \ldots, N^T$, to find an estimator $\hat{\boldsymbol{\theta}}$ that is close to the actual parameter values $\boldsymbol{\theta}$ in some sensible metric. As with ERGMs, the intractability of the normalizing constant Z often makes explicit solution of maximum likelihood estimation difficult. However, general techniques for MCMC sampling to enable approximate maximum likelihood estimation for ERGMs have been studied in some depth and have proven successful for a variety of models [3]. By a slight modification of these algorithms, we can apply the same general techniques as follows.

Let

$$\mathcal{L}(\boldsymbol{\theta}; N^1, N^2, \ldots, N^T) = \log \mathcal{P}(N^{K+1}, N^{K+2}, \ldots, N^T | N^1, \ldots, N^K, \boldsymbol{\theta}), \tag{2}$$

$$\mathbf{M}(t, \boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}} \left[\mathit{\Psi}(\underline{\mathbf{N}}^t, N^{t-1}, \ldots, N^{t-K}) | N^{t-1}, \ldots, N^{t-K} \right],$$

$$\mathbf{C}(t, \boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}} \left[\mathit{\Psi}(\underline{\mathbf{N}}^t, N^{t-1}, \ldots, N^{t-K}) \mathit{\Psi}(\underline{\mathbf{N}}^t, N^{t-1}, \ldots, N^{t-K})' | N^{t-1}, \ldots, N^{t-K} \right].$$

where expectations are taken over the random variable $\underline{\mathbf{N}}^t$, the network at time t. Note that

$$\nabla \mathcal{L}(\boldsymbol{\theta}; N^1, \ldots, N^T) = \sum_{t=K+1}^{T} \left(\mathit{\Psi}(N^t, N^{t-1}, \ldots, N^{t-K}) - M(t, \boldsymbol{\theta}) \right)$$

and

$$\nabla^2 \mathcal{L}(\boldsymbol{\theta}; N^1, \ldots, N^T) = \sum_{t=K+1}^{T} \left(M(t, \boldsymbol{\theta}) M(t, \boldsymbol{\theta})' - C(t, \boldsymbol{\theta}) \right).$$

The expectations can be approximated by Gibbs sampling from the conditional distributions [3], so that we can perform an unconstrained optimization procedure akin to Newton's method: approximate the expectations, update parameter values in the direction that increases (2), repeat until convergence. A related algorithm is described by [5] for general exponential families, and variations are given by [3] that are tailored for ERG models. The following is a simple version of such an estimation algorithm.

```
Randomly initialize $\boldsymbol{\theta}^{(1)}$
For $i = 1$ up until convergence
    For $t = K+1, K+2, \ldots, T$
        Sample $\hat{N}^{t,1}_{(i)}, \ldots, \hat{N}^{t,B}_{(i)} \sim \mathcal{P}(\underline{\mathbf{N}}^t | N^{t-K}, \ldots, N^{t-1}, \boldsymbol{\theta}^{(i)})$
        $\hat{\mu}^t_{(i)} = \frac{1}{B} \sum_{b=1}^{B} \boldsymbol{\Psi}(\hat{N}^{t,b}_{(i)}, N^{t-1}, \ldots, N^{t-K})$
        $\hat{C}^t_{(i)} = \frac{1}{B} \sum_{b=1}^{B} \boldsymbol{\Psi}(\hat{N}^{t,b}_{(i)}, N^{t-1}, \ldots, N^{t-K}) \boldsymbol{\Psi}(\hat{N}^{t,b}_{(i)}, N^{t-1}, \ldots, N^{t-K})'$
    $\hat{H}_{(i)} = \sum_{t=K+1}^{T} [\hat{\mu}^t_{(i)} \hat{\mu}^{t\prime}_{(i)} - \hat{C}^t_{(i)}]$
    $\boldsymbol{\theta}^{(i+1)} \leftarrow \boldsymbol{\theta}^{(i)} - \hat{H}^{-1}_{(i)} \sum_{t=K+1}^{T} \left[ \boldsymbol{\Psi}(N^t, N^{t-1}, \ldots, N^{t-K}) - \hat{\mu}^t_{(i)} \right]$
```

The choice of B can affect the convergence of this algorithm. Generally, larger B values will give more accurate updates, and thus fewer iterations needed until convergence. However, in the early stages of the algorithm, precise updates might not be necessary if the likelihood function is sufficiently smooth, so that a B that grows larger only when more precision is needed may be appropriate. If computational resources are limited, it is possible (though less certain) that the algorithm might still converge even for small B values (see [6] for an alternative approach to sampling-based MLE, which seems to remain effective for small B values).

To examine the convergence rate empirically, we display in Figure 1 the convergence of this algorithm on data generated from the example model given in Section 2.1.

The simulated data is generated by sampling from the example model with randomly generated $\boldsymbol{\theta}$, and the loss is plotted in terms of Euclidean distance of the estimator from the true parameters. To generate the initial N^1 network, we sample from the pmf $\frac{1}{Z(\boldsymbol{\theta})}\exp\{\boldsymbol{\theta}'\boldsymbol{\Psi}(N^1, N^1)\}$. The number of actors n is 100. The parameters are initialized uniformly in the range $[0, 10)$, except for θ_D, which is initialized to $-5\theta_S - 5\theta_R - 5\theta_T$. This tends to generate networks with reasonable densities. The results in Figure 1 represent averages over 10 random initial configurations of the parameters and data. In the estimation algorithm used, $B = 100$, but increases to 1000 when the Euclidean distance between parameter estimates from the previous two iterations is less than 1. Convergence is defined as the Euclidean distance between $\boldsymbol{\theta}^{(i+1)}$ and $\boldsymbol{\theta}^{(i)}$ being within 0.1. Since this particular model is simple enough for exact calculation of the likelihood and derivatives thereof (see below), we also compare against Newton's method with exact updates (rather than sampling-based). We can use this to determine how much of the loss is due to the approximations being performed, and how much of it is intrinsic to the estimation problem. The parameters returned by the sampling-based approximation are usually almost identical to the MLE obtained by Newton's method, and this behavior manifests itself in Figure 1 by the losses being visually indistinguishable.

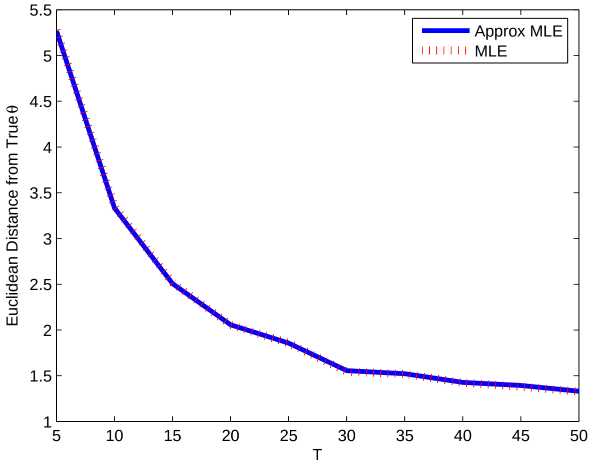

Fig. 1. Convergence of estimation algorithm on simulated data, measured in Euclidean distance of the estimated values from the true parameter values. The approximate MLE from the sampling-based algorithm is almost identical to the MLE obtained by direct optimization.

4 Hypothesis Testing

As an example of how models of this type might be used in practice, we present a simple hypothesis testing application. Here we see the generality of this framework pay off, as we can use models of this type to represent a broad range of scientific hypotheses. The general approach to hypothesis testing in this framework is first to write down potential functions representing transitions one expects to be of some significance in a given population, next to write down potential functions representing the usual "background" processes (to serve as a null hypothesis), and third to plug these potentials into the model, calculate a test statistic, and compute a p-value.

The data involved in this example come from the United States 108^{th} Senate, having $n = 100$ actors. Every time a proposal is made in the Senate, be it a bill, amendment, resolution, etc., a single Senator serves as the proposal's *sponsor* and there may possibly be several *cosponsors*. Given records of all proposals voted on in the full Senate, we create a sliding window of 100 consecutive proposals. For a particular placement of the window, we define a binary directed relation existing between two Senators if and only if one of them is a sponsor and the other a cosponsor for the same proposal within that window (where the direction is toward the sponsor). The data is then taken as evenly spaced snapshots of this sliding window, A^1 being the adjacency matrix for the first 100 proposals, A^2 for proposal 31 through 130, and so on shifting the window by 30 proposals each time. In total, there are 14 observed networks in this series, corresponding to the first 490 proposals addressed in the 108^{th} Senate.

In this study, we propose to test the hypothesis that intraparty reciprocity is inherently stronger than interparty reciprocity. To formalize this, we use a model similar to the example given previously. The main difference is the addition of party membership indicator variables. Let $P_{ij} = 1$ if the i^{th} and j^{th} actors are in the same political party, and 0 otherwise, and let $\bar{P}_{ij} = 1 - P_{ij}$. Define the following potential functions, representing *stability*, *intraparty density*, *interparty density*,[1] *overall reciprocity*, *intraparty reciprocity*, and *interparty reciprocity*.

$$
\begin{aligned}
\varPsi_S(A^t, A^{t-1}) &= \frac{1}{(n-1)} \sum_{ij} \left[A^t_{ij} A^{t-1}_{ij} + (1 - A^t_{ij})(1 - A^{t-1}_{ij}) \right] \\
\varPsi_{WD}(A^t, A^{t-1}) &= \frac{1}{(n-1)} \sum_{ij} A^t_{ij} P_{ij} \\
\varPsi_{BD}(A^t, A^{t-1}) &= \frac{1}{(n-1)} \sum_{ij} A^t_{ij} \bar{P}_{ij} \\
\varPsi_R(A^t, A^{t-1}) &= n \left[\sum_{ij} A^t_{ji} A^{t-1}_{ij} \right] \Bigg/ \left[\sum_{ij} A^{t-1}_{ij} \right] \\
\varPsi_{WR}(A^t, A^{t-1}) &= n \left[\sum_{ij} A^t_{ji} A^{t-1}_{ij} P_{ij} \right] \Bigg/ \left[\sum_{ij} A^{t-1}_{ij} P_{ij} \right] \\
\varPsi_{BR}(A^t, A^{t-1}) &= n \left[\sum_{ij} A^t_{ji} A^{t-1}_{ij} \bar{P}_{ij} \right] \Bigg/ \left[\sum_{ij} A^{t-1}_{ij} \bar{P}_{ij} \right]
\end{aligned}
$$

[1] We split density to intra- and inter-party terms so as to factor out the effects on reciprocity of having higher intraparty density.

The null hypothesis supposes that the reciprocity observed in this data is the result of an overall tendency toward reciprocity amongst the Senators, regardless of party. The alternative hypothesis supposes that there is a stronger tendency toward reciprocity among Senators within the same party than among Senators from different parties. Formally, the transition probability for the null hypothesis can be written as

$$\mathcal{P}_0(A^t|A^{t-1},\boldsymbol{\theta}^{(0)}) = \frac{1}{Z_0(\boldsymbol{\theta}^{(0)},A^{t-1})}\exp\left\{\sum_{j\in\{S,WD,BD,R\}}\theta_j^{(0)}\Psi_j(A^t,A^{t-1})\right\},$$

while the transition probability for the alternative hypothesis can be written as

$$\mathcal{P}_1(A^t|A^{t-1},\boldsymbol{\theta}^{(1)}) = \frac{1}{Z_1(\boldsymbol{\theta}^{(1)},A^{t-1})}\exp\left\{\sum_{j\in\{S,WD,BD,WR,BR\}}\theta_j^{(1)}\Psi_j(A^t,A^{t-1})\right\}.$$

For our test statistic, we use the likelihood ratio. To compute this, we compute the maximum likelihood estimators for each of these models, and take the ratio of the likelihoods. For the null hypothesis, the MLE is

$$(\hat{\theta}_S^{(0)} = 336.2, \hat{\theta}_{WD}^{(0)} = -58.0, \hat{\theta}_{BD}^{(0)} = -95.0, \hat{\theta}_R^{(0)} = 4.7)$$

with likelihood value of $e^{-9094.46}$. For the alternative hypothesis, the MLE is

$$(\hat{\theta}_S^{(1)} = 336.0, \hat{\theta}_{WD}^{(1)} = -58.8, \hat{\theta}_{BD}^{(1)} = -94.3, \hat{\theta}_{WR}^{(1)} = 4.2, \hat{\theta}_{BR}^{(1)} = 0.03)$$

with likelihood value of $e^{-9088.96}$. The likelihood ratio statistic (null likelihood over alternative likelihood) is therefore about 0.0041. Because the null hypothesis is composite, determining the p-value of this result is a bit more tricky, since we must determine the probability of observing a likelihood ratio at least this extreme under the null hypothesis for the parameter values $\boldsymbol{\theta}^{(0)}$ that *maximize* this probability. That is,

$$\text{p-value} = \sup_{\boldsymbol{\theta}^{(0)}}\mathcal{P}_0\left\{\frac{\sup_{\hat{\boldsymbol{\theta}}^{(0)}}\mathcal{P}_0(A^1,\ldots,A^{14}|\hat{\boldsymbol{\theta}}^{(0)})}{\sup_{\hat{\boldsymbol{\theta}}^{(1)}}\mathcal{P}_1(A^1,\ldots,A^{14}|\hat{\boldsymbol{\theta}}^{(1)})} \leq 0.0041\middle|\boldsymbol{\theta}^{(0)}\right\}.$$

In general this seems not to be tractable to analytic solution, so we employ a genetic algorithm to perform the unconstrained optimization, and approximate the probability for each parameter vector by sampling. That is, for each parameter vector $\boldsymbol{\theta}^{(0)}$ (for the null hypothesis) in the GA's population on each iteration, we sample a large set of sequences from the joint distribution. For each sequence, we compute the MLE under the null hypothesis and the MLE under the alternative hypothesis, and then calculate the likelihood ratio and compare it to the observed ratio. We calculate the empirical frequency with which the likelihood ratio is at most 0.0041 in the set of sampled sequences for each vector $\boldsymbol{\theta}^{(0)}$, and use this as the objective function value in the genetic algorithm. Mutations consist of adding Gaussian noise (with variance decreasing on each iteration), and recombination is performed as usual. Full details of the algorithm are omitted for brevity (see [7] for an introduction to GAs). The resulting approximate p-value we obtain by this optimization procedure is 0.024.

This model is nice in that we can compute the likelihoods and derivatives thereof analytically. In fact, it is representative of an interesting subfamily of models, in which the distributions of edges at time t are independent of each other given the network at time $t-1$. In models of this form, we can compute likelihoods and perform Newton-Raphson optimization directly, without the need of sampling-based approximations. However, in general this might not be the case. For situations in which one cannot tractably compute the likelihoods, an alternative possibility is to use bounds on the likelihoods. Specifically, one can obtain an upper bound on the likelihood ratio statistic by dividing an upper bound on the null likelihood by a lower bound on the alternative likelihood. When computing the p-value, one can use a lower bound on the ratio by dividing a lower bound on the null likelihood by an upper bound on the alternative likelihood. See [8,9] for examples of how such bounds on the likelihood can be tractably attained, even for intractable models.

In practice, the problem of formulating an appropriate model to encode one's hypothesis is ill-posed. One general approach which seems intuitively appealing is to write down the types of motifs or patterns one expects to find in the data, and then specify various other patterns which one believes those motifs could likely transition to (or would likely *not* transition to) under the alternative hypothesis. For example, perhaps one believes that densely connected regions of the network will tend to become more dense and clique-like over time, so that one might want to write down a potential representing the transition of, say, k-cliques to more densely connected structures.

5 Classification

One can additionally consider using these temporal models for classification. Specifically, consider a transductive learning problem in which each actor has a static class label, but the learning algorithm is only allowed to observe the labels of some random subset of the population. The question is then how to use the known label information, in conjunction with observations of the network evolving over time, to accurately infer the labels of the remaining actors whose labels are unknown.

As an example of this type of application, consider the alternative hypothesis model from the previous section (model 1), in which each Senator has a class label (party affiliation). We can slightly modify the model so that the party labels are no longer constant, but random variables drawn independently from a known multinomial distribution. Assume we know the party affiliations of a randomly chosen 50 Senators. This leaves 50 Senators with unknown affiliations. If we knew the parameters $\boldsymbol{\theta}$, we could predict these 50 labels by sampling from the posterior distribution and taking the mode for each label. However, since *both* the parameters *and* the 50 labels are unknown, this is not possible. Instead, we can perform Expectation Maximization to *jointly* infer the maximum likelihood estimator $\hat{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}$ *and* the posterior mode given $\hat{\boldsymbol{\theta}}$.

Specifically, let us assume the two class labels are *Democrat* and *Republican*, and we model these labels as independent Bernoulli(0.5) random variables. The distribution on the network sequence given that all 100 labels are fully observed is the same as given in the previous section. Since one can compute likelihoods in this model, sampling from the posterior distribution of labels given the network sequence is straightforward using Gibbs sampling. We can therefore employ a combination of MCEM and Generalized EM algorithms (call it MCGEM) [10] with this model to infer the party labels as follows. In each iteration of the algorithm, we sample from the posterior distribution of the unknown class labels under the current parameter estimates given the observed networks and known labels, approximate the expectation of the gradient and Hessian of the log likelihood using the samples, and then perform a single Newton-Raphson update using these approximations.

We run this algorithm on the 108^{th} Senate data from the previous section. We randomly select 50 Senators whose labels are observable, and take the remaining Senators as having unknown labels. As mentioned above, we assume all Senators are either Democrat or Republican; Senator Jeffords, the only independent Senator, is considered a Democrat in this model. We run the MCGEM algorithm described above to infer the maximum likelihood estimator $\hat{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}$, and then sample from the posterior distribution over the 50 unknown labels under that maximum likelihood distribution, and take the sample mode for each label to make a prediction.

The predictions of this algorithm are correct on 70% of the 50 Senators with unknown labels. Additionally, it is interesting to note that the parameter values the algorithm outputs $(\hat{\theta}_S = 336.0, \hat{\theta}_{WD} = -59.7, \hat{\theta}_{BD} = -96.0, \hat{\theta}_{WR} = 3.8, \hat{\theta}_{BR} = 0.28)$ are very close (Euclidean distance 2.0) to the maximum likelihood estimator obtained in the previous section (where all class labels were known). Compare the above accuracy score with a baseline predictor that always predicts Democrat, which would get 52% correct for this train/test split, indicating that this statistical model of network evolution provides at least a somewhat reasonable learning bias. However, there is clearly room for improvement in the model specification, and it is not clear whether modeling the evolution of the graph is actually of any benefit for this particular example. For example, after collapsing this sequence of networks into a single weighted graph with edge weights equal to the sum of edge weights over all graphs in the sequence, running Thorsten Joachims' Spectral Graph Transducer algorithm [11] gives a 90% prediction accuracy on the Senators with unknown labels. These results are summarized in Table 1. Further investigation is needed into what types of problems can benefit from explicitly modeling the network evolution, and what types of models are most appropriate for basing a learning bias on.

Method	Accuracy
Baseline	52%
Temporal Model	70%
SGT	90%

Table 1. Summary of classification results.

6 Open Problems and Future Work

If we think of this type of model as describing a process giving rise to the networks one observes in reality, then one can think of a single network observation as a snapshot of this Markov chain at that time point. Traditionally one would model a network at a single time point using an ERGM. It therefore seems natural to investigate the formal relation between these Markov chain models and ERGMs. Specifically, any Markov chain of the form described here has a stationary distribution which can be characterized by an ERGM. Can one give a general analytic derivation of this stationary ERGM for any Markov chain of the form described here? To our knowledge, this remains an open problem. One can also ask the reverse question of whether, given any ERGM, one can describe an interesting set of Markov chains having it as a stationary distribution. Answering this would not only be of theoretical interest, but would potentially also lead to practical techniques for sampling from an ERGM distribution by formulating a more tractable Markov chain giving rise to it. Indeed, one can ask these same questions about general Markov chains (not just networks) having transition probabilities in an exponential family, the stationary distributions of which can be described by exponential families.

Moving forward, we hope to move beyond these ERG-inspired models toward models that incorporate latent variables, which may also evolve over time with the network. For example, it may often be the case that the phenomena represented in data can most easily be described by imagining the existence of unobserved groups or factions, which form, dissolve, merge and split as time progresses. The flexibility of the ERG models and the above temporal extensions allows a social scientist to “plug in” his or her knowledge into the formulation of the model, while still providing general-purpose estimation algorithms to find the right trade-offs between competing and complementary factors in the model. We would like to retain this flexibility in formulating a general family of models that include evolving latent variables in the representation, so that the researcher can “plug in” his or her hypotheses about latent group dynamics, evolution of unobservable actor attributes, or a range of other possible phenomena into the model representation. At the same time, we would like to preserve the ability to provide a “black box” inference algorithm to determine the parameter and variable values of interest to the researcher, as can be done with ERG models and their temporal extensions.

Acknowledgments

We thank Stephen Fienberg and all participants in the statistical network modeling discussion group at Carnegie Mellon for helpful comments and feedback. We also thank the anonymous reviewers for insightful suggestions.

References

[1] Anderson, C.J., Wasserman, S., Crouch, B.: A p* primer: logit models for social networks. Social Networks **21** (1999) 37–66

[2] Robins, G.L., Pattison, P.E.: Interdependencies and social processes: Generalized dependence structures. Cambridge University Press (2004)

[3] Snijders, T.A.B.: Markov Chain Monte Carlo estimation of exponential random graph models. Journal of Social Structure **3**(2) (2002)

[4] Frank, O., Strauss, D.: Markov graphs. Journal of the American Statistical Association **81** (1986)

[5] Geyer, C., Thompson, E.: Constrained Monte Carlo maximum likelihood for dependent data. Journal of the Royal Statistical Society, B **54**(3) (1992) 657–699

[6] Carreira-Perpignán, M., Hinton, G.E.: On contrastive divergence learning. In: Artificial Intelligence and Statistics. (2005)

[7] Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press (1996)

[8] Opper, M., Saad, D., eds.: Advanced Mean Field Methods: Theory and Practice. MIT Press (2001)

[9] Wainwright, M., Jaakkola, T., Willsky, A.: A new class of upper bounds on the log partition function. IEEE Trans. on Information Theory **51**(7) (2005)

[10] McLachlan, G., Krishnan, T.: The EM algorithm and Extensions. Wiley (1997)

[11] Joachims, T.: Transductive learning via spectral graph partitioning. In: Proceedings of the International Conference on Machine Learning. (2003)