A Dynamic Relational Infinite Feature Model for Longitudinal Social Networks

James Foulds[†] **Christopher DuBois**[‡] **Arthur U. Asuncion**[†] **Carter T. Butts**[*] **Padhraic Smyth**[†]

[†]Department of Computer Science
University of California, Irvine
{*jfoulds,asuncion, smyth*}
@ics.uci.edu

[‡]Department of Statistics
University of California, Irvine
duboisc@ics.uci.edu

[*]Department of Sociology and Institute for Mathematical Behavioral Sciences
University of California, Irvine
buttsc@uci.edu

Abstract

Real-world relational data sets, such as social networks, often involve measurements over time. We propose a Bayesian nonparametric latent feature model for such data, where the latent features for each actor in the network evolve according to a Markov process, extending recent work on similar models for static networks. We show how the number of features and their trajectories for each actor can be inferred simultaneously and demonstrate the utility of this model on prediction tasks using synthetic and real-world data.

1 Introduction

Statistical modeling of social networks and other relational data has a long history, dating back at least as far as the 1930s. In the statistical framework, a static network on N actors is typically represented by an $N \times N$ binary sociomatrix $\mathbf{Y}$, where relations between actors i and j are represented by binary random variables y_{ij} taking value 1 if a relationship exists and 0 otherwise. The sociomatrix can be interpreted as the adjacency matrix of a graph, with each actor being represented by a node. A useful feature of the statistical framework is that it readily allows for a variety of extensions such as handling missing data and incorporating additional information such as weighted edges, time-varying edges, or covariates for actors and edges.

Exponential-family random graph models, or ERGMs, are the canonical approach for parametrizing statistical network models—but such models can be difficult to work with both from a computational and statistical estimation viewpoint [Handcock et al., 2003]. An alternative approach is to use latent vectors $\mathbf{z}_i$ as "coordinates" to represent the characteristics of each network actor i. The presence or absence of edges y_{ij} are modeled as being conditionally independent given the latent vectors $\mathbf{z}_i$ and $\mathbf{z}_j$ and given the parameters of the model. Edge probabilities in these models can often be cast in the following form,

$$P(y_{ij} = 1 | \ldots) = f(\alpha_0 + \alpha^T \mathbf{x}_{i,j} + g(\mathbf{z}_i, \mathbf{z}_j)),$$

where f is a link function (such as logistic); α_0 is a parameter controlling network density; $\mathbf{x}_{i,j}$ is a vector of observed covariates (if known) with weight vector α; and $g(\mathbf{z}_i, \mathbf{z}_j)$ is a function that models the interaction of latent variables $\mathbf{z}_i$ and $\mathbf{z}_j$.

We are often interested in modeling latent structure, for example, when there are no observed covariates $\mathbf{x}_{i,j}$ or to complement such covariates. As discussed by Hoff [2008] there are a number of options for modeling the interaction term $g(\mathbf{z}_i, \mathbf{z}_j)$, such as:

- additive sender and receiver effects with $g(\mathbf{z}_i, \mathbf{z}_j) = \mathbf{z}_i + \mathbf{z}_j$;
- latent class models where $\mathbf{z}_i$ is a vector indicating if individual i belongs to one of K clusters [Nowicki and Snijders, 2001, Kemp et al., 2006], or allowing individuals to have probabilities of membership in multiple groups as in the mixed-membership blockmodel [Airoldi et al., 2008];
- distance models, e.g., where $\mathbf{z}_i \in \mathcal{R}^K$ and $g(\mathbf{z}_i, \mathbf{z}_j)$ is negative Euclidean distance [Hoff et al., 2002];
- multiplicative models, such as eigendecompositions of $\mathbf{Y}_{ij}$ [Hoff, 2007]; relational topic models with multinomial probability $\mathbf{z}_i$'s [Chang and Blei, 2009]; and infinite feature models with binary feature vector $\mathbf{z}_i$'s [Miller et al., 2009].

Appearing in Proceedings of the 14^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2011, Fort Lauderdale, FL, USA. Volume 15 of JMLR: W&CP 15.

Given the increasing availability of social network data sets with a temporal component (email, online social networks, instant messaging, etc.) there is considerable motivation to develop latent representations for network data over time. Rather than a single observed network $\mathbf{Y}$ we have a sequence of observed networks $\mathbf{Y}^{(t)}$ indexed by time $t = 1, \ldots, T$, often referred to as *longitudinal* network data. In this paper, we extend the infinite latent feature model of Miller et al. [2009] by introducing temporal dependence in the latent $\mathbf{z}_i$'s via a hidden Markov process. Consider first the static model. Suppose individuals are characterized by latent features that represent their job-type (e.g., dentist, graduate student, professor) and their leisure interests (e.g., mountain biking, salsa dancing), all represented by binary variables. The probability of an edge between two individuals is modeled as a function of the interactions of the latent features that are turned "on" for each of the individuals. For example, graduate students that salsa dance might have a much higher probability of having a link to professors that mountain bike, rather than to dentists that salsa dance. We extend this model to allow each individual's latent features to change over time. Temporal dependence at the feature level allows an individual's features $\mathbf{z}_i^{(t)}$ to change over time t as that individual's interests, group memberships, and behavior evolve. In turn the relational patterns in the networks $\mathbf{Y}^{(t)}$ will change over time as a function of the $\mathbf{z}_i^{(t)}$'s.

The remainder of the paper begins with a brief discussion of related work in Section 2. Sections 3 and 4 discuss the generative model and inference algorithms respectively. In Section 5 we evaluate the model (relative to baselines) on prediction tasks for both simulated and real-world network data sets. Section 6 contains discussion and conclusions.

2 Background and Related Work

The model proposed in this paper builds upon the Indian buffet process (IBP) [Griffiths and Ghahramani, 2006], a probability distribution on (equivalence classes of) sparse binary matrices with a finite number of rows but an unbounded number of columns. The IBP is named after a metaphorical process that gives rise to the probability distribution, where N customers enter an Indian Buffet restaurant and sample some subset of an infinitely long sequence of dishes. The first customer samples the first Poisson(α) dishes, and the kth customer then samples the previously sampled dishes proportionately to their popularity, and samples Poisson(α/k) new dishes. The matrix of dishes sampled by customers is a draw from the IBP distribution.

A typical application of the IBP is to use it as a prior on a matrix that specifies the presence or absence of latent features which explain some observed data. The motivation of such an infinite latent feature model in this context is that the number of features can be automatically adjusted during inference, and hence does not need to be specified ahead of time. Meeds et al. [2007] introduced a probabilistic matrix decomposition method for row and column-exchangeable binary matrices using a generative model with IBP priors. This model was subsequently adapted for modeling static social networks by Miller et al. [2009].

The primary contribution of this paper is to build on this work to develop a nonparametric Bayesian generative model for longitudinal social network data. The model leverages ideas from the recently introduced infinite factorial HMM [Van Gael et al., 2009], an approach that modifies the IBP into a factorial HMM with an unbounded number of hidden chains. Modeling temporal changes in latent variables, for actors in a network, has been also proposed by Sarkar and Moore [2005], Sarkar et al. [2007] and Fu et al. [2009]— a major difference in our approach in that we model an actor's evolution by Markov switching rather than via the Gaussian linear motion models used in these papers. Our approach explicitly models the dynamics of the actors' latent representations, unlike the model of Fu et al. [2009], making it more suitable for forecasting. Other statistical models for dynamic network data have been also proposed but typically deal only with the observed graphs $\mathbf{Y}^{(t)}$ (e.g. Snijders [2006], Butts [2008]) and do not use latent representations.

3 Generative Process for the Dynamic Relational Infinite Feature Model

We introduce a dynamic relational infinite feature model (abbreviated as DRIFT) which extends the nonparametric latent feature relational model (LFRM) of Miller et al. [2009] to handle longitudinal network data. In the LFRM model, each actor is described by a vector of binary latent features, of unbounded dimension. These features (along with other covariates, if desired) determine the probability of a link between two actors. Although the features are not a priori associated with any specific semantics, the intuition is that these features can correspond to an actor's interests, club memberships, location, social cliques and other real-world features related to an actor. Latent features can be understood as clusters or class memberships that are allowed to overlap, in contrast to the mutually exclusive classes of traditional blockmodels [Fienberg and Wasserman, 1981] from the social network literature. Unlike LFRM, our proposed model allows the feature memberships to evolve over time—

LFRM can be viewed as a special case of DRIFT with only one time step.

We start with a finite version of the model with K latent features. The final model is defined to be the limit of this model as K approaches infinity. Let there be N actors, and T discrete time steps. At time t, we observe $\mathbf{Y}^{(t)}$, an $N \times N$ binary sociomatrix representing relationships between the actors at that time. We will typically assume that $\mathbf{Y}^{(t)}$ is constrained to be symmetric. At each time step t there is an $N \times K$ binary matrix of latent features $\mathbf{Z}^{(t)}$, where $z_{ik}^{(t)} = 1$ if actor i has feature k at that time step. The $K \times K$ matrix $\mathbf{W}$ is a real-valued matrix of weights, where entry $w_{kk'}$ influences the probability of an edge between actors i and j if i has feature k turned on and j has feature k' turned on. The edges between actors at time t are assumed to be conditionally independent given $\mathbf{Z}^{(t)}$ and $\mathbf{W}$. The probability of each edge is:

$$Pr(y_{ij}^{(t)} = 1) = \sigma(\mathbf{z}_i^{(t)} \mathbf{W} \mathbf{z}_j^{(t)\intercal}) \ , \tag{1}$$

where $\mathbf{z}_i^{(t)}$ is the ith row of $\mathbf{Z}^{(t)}$, and $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the logistic function. There are assumed to be null states $z_{ik}^{(0)} = 0$, which means that each feature is effectively "off" before the process begins. Each feature k for each actor i has independent Markov dynamics, wherein if its current state is zero, the next value is distributed Bernoulli with a_k, otherwise it is distributed Bernoulli with the persistence parameter b_k for that feature. In other words, the transition matrix for actor i's kth feature is $\mathbf{Q}^{(ik)} = \left(\begin{smallmatrix} 1-a_k & a_k \\ 1-b_k & b_k \end{smallmatrix}\right)$. These Markov dynamics resemble the infinite factorial hidden Markov model [Van Gael et al., 2009]. Note that $\mathbf{W}$ is not time-varying, unlike $\mathbf{Z}$. This means that the features themselves do not evolve over time; rather, the network dynamics are determined by the changing presence and absence of the features for each actor.

The a_k's have prior probability $\text{Beta}(\frac{\alpha}{K}, 1)$, which is the same prior as for the features in the IBP. Importantly, this choice of prior allows for the number of introduced (i.e. "activated") features to have finite expectation when $K \to \infty$, with the expected number of "active" features being controlled by hyper-parameter α. The b_k's are drawn from a beta distribution, and the $w_{kk'}$'s are drawn from a Gaussian with mean zero. More formally, the complete generative model is

$$\begin{aligned} a_k &\sim \text{Beta}(\frac{\alpha}{K}, 1) \\ b_k &\sim \text{Beta}(\gamma, \delta) \\ z_{ik}^{(0)} &= 0 \\ z_{ik}^{(t)} &\sim \text{Bernoulli}(a_k^{1-z_{ik}^{(t-1)}} b_k^{z_{ik}^{(t-1)}}) \\ w_{kk'} &\sim \text{Normal}(0, \sigma_w) \\ y_{ij}^{(t)} &\sim \text{Bernoulli}(\sigma(\mathbf{z}_i^{(t)} \mathbf{W} \mathbf{z}_j^{(t)\intercal})) \ . \end{aligned}$$

Our proposed framework is illustrated with a graphical model in Figure 1. The model is a factorial hidden Markov model with a hidden chain for each actor-feature pair, and with the observed variables being the networks ($\mathbf{Y}$'s). It is also possible to include additional covariates as used in the social network literature (see e.g. Hoff [2008]), inside the logistic function for Equation 1. In our experiments we only use an additional intercept term α_0 that determines the prior probability of an edge when no features are present. Note that this does not increase the generality of the model, as the same effect could be achieved by introducing an additional feature shared by all actors.

3.1 Taking the Infinite Limit

The full model is defined to be the limit of the above model as the number of features approaches infinity. Let c_k^{00}, c_k^{01}, c_k^{10}, c_k^{11} be the total number of transitions from $0 \to 0$, $0 \to 1$, $1 \to 0$, $1 \to 1$ over all actors, respectively, for feature k. In the finite case with K features, we can write the prior probability of $\mathbf{Z} = \mathbf{z}$, for $\mathbf{z} = (\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(T)})$ in the following way:

$$Pr(\mathbf{Z} = \mathbf{z} | a, b) = \prod_{k=1}^{K} a_k^{c_k^{01}} (1 - a_k)^{c_k^{00}} b_k^{c_k^{11}} (1 - b_k)^{c_k^{10}} \ . \tag{2}$$

Before taking the infinite limit, we integrate out the transition probabilities with respect to their priors,

$$Pr(\mathbf{Z} = \mathbf{z} | \alpha, \gamma, \delta) = \prod_{k=1}^{K} \frac{\frac{\alpha}{K} \Gamma(\frac{\alpha}{K} + c_k^{01}) \Gamma(1 + c_k^{00}) \Gamma(\gamma + \delta) \Gamma(\delta + c_k^{10}) \Gamma(\gamma + c_k^{11})}{\Gamma(\frac{\alpha}{K} + c_k^{00} + c_k^{01} + 1) \Gamma(\gamma) \Gamma(\delta) \Gamma(\gamma + \delta + c_k^{10} + c_k^{11})} \tag{3}$$

where $\Gamma(x)$ is the gamma function. Similar to the construction of the IBP and the iFHMM, we compute the infinite limit for the probability distribution on *equivalence classes* of the binary matrices, rather than on the matrices directly. Consider the representation $\bar{\mathbf{z}}$ of $\mathbf{z}$, an $NT \times K$ matrix where the chains of feature values for each actor are concatenated to form a single matrix, according to some fixed ordering of the actors. The equivalence classes are on the left-ordered form (lof) of $\bar{\mathbf{z}}$. Define the history of a column k to be the binary number that it encodes when its entries are interpreted to be binary digits. The lof of a matrix $\mathbf{M}$ is a copy of $\mathbf{M}$ with the columns permuted so that their histories are sorted in decreasing order. Note that the model is column-exchangeable so transforming $\bar{\mathbf{z}}$ to lof does not affect its probability. We denote $[\mathbf{z}]$ to be the

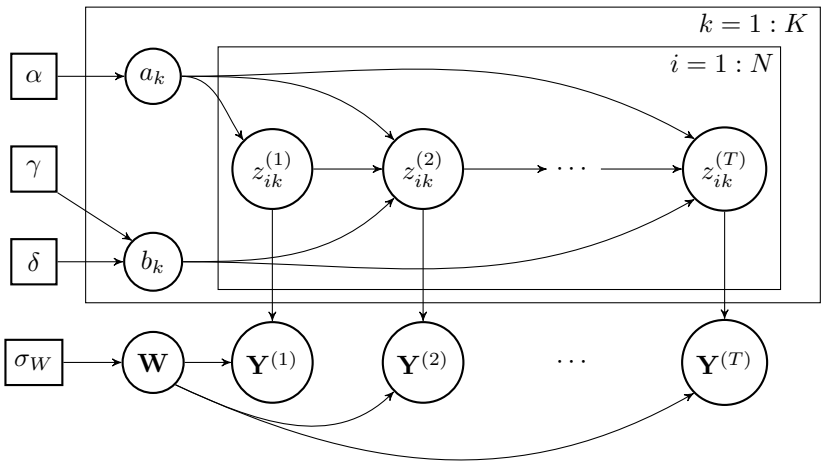


Figure 1: Graphical model for the finite version of DRIFT. The full model is defined to be the limit of this model as $K \to \infty$.

set of $\mathbf{Z}$s that have the same lof $\bar{\mathbf{Z}}$ as $\mathbf{z}$. Let K_h be the number of columns in $\bar{\mathbf{z}}$ whose history has decimal value h. Then the number of elements of $[\mathbf{z}]$ equals $\frac{K!}{\prod_{h=0} 2^{NT-1} K_h!}$, yielding the following:

$$\begin{aligned} Pr([\mathbf{Z}] = [\mathbf{z}]) &= \sum_{\hat{\mathbf{z}} \in [\mathbf{z}]} Pr(\mathbf{Z} = \hat{\mathbf{z}} | \alpha, \gamma, \delta) \\ &= \frac{K!}{\prod_{h=0} 2^{NT-1} K_h!} Pr(\mathbf{Z} = \mathbf{z} | \alpha, \gamma, \delta). \end{aligned} \tag{4}$$

The limit of $Pr([\mathbf{Z}])$ as $K \to \infty$ can be derived similarly to the iFHMM model [Van Gael et al., 2009]. Let K^+ be the number of features that have at least one non-zero entry for at least one actor. Then we obtain

$$\begin{aligned} \lim_{K \to \infty} Pr([\mathbf{Z}] = [\mathbf{z}]) = \frac{\alpha^{K^+}}{\prod_{h=0}^{2^{NT}-1} K_h!} \exp(-\alpha H_{NT}) \\ \prod_{k=1}^{K^+} \frac{(c_k^{01}-1)! c_k^{00}! \Gamma(\gamma+\delta) \Gamma(\delta + c_k^{10}) \Gamma(\gamma + c_k^{11})}{(c_k^{00}+c_k^{01})! \Gamma(\gamma) \Gamma(\delta) \Gamma(\gamma + \delta + c_k^{10} + c_k^{11})}, \end{aligned} \tag{5}$$

where $H_i = \sum_{k=1}^{i} \frac{1}{k}$ is the ith harmonic number. It is also possible to derive Equation 5 as a stochastic process with a culinary metaphor similar to the IBP, but we omit this description for space. A restaurant metaphor equivalent to $Pr(\mathbf{Z})$ with one actor is provided in Van Gael et al. [2009].

For inference, we will make use of the stick-breaking construction of the IBP portion of DRIFT [Teh et al., 2007]. Since the distribution on the a_k's is identical to the feature probabilities in the IBP model, the stick breaking properties of these variables carry over to our model. Specifically, if we order the features so that they are strictly decreasing in a_k, we can write them in stick-breaking form as $v_k \sim \text{Beta}(\alpha, 1), a_k = v_k a_{k-1} = \prod_{l=1}^{k} v_l$.

4 MCMC Inference Algorithm

We now describe how to perform posterior inference for DRIFT using a Markov chain Monte Carlo algorithm. The algorithm performs blocked Gibbs sampling updates on subsets of the variables in turn. We adapt a slice sampling procedure for the IBP that allows for correct sampling despite the existence of a potentially infinite number of features, and also mixes better relative to naive Gibbs sampling [Teh et al., 2007]. The technique is to introduce an auxiliary "slice" variable s to adaptively truncate the represented portion of $\mathbf{Z}$ while still performing correct inference on the infinite model. The slice variable is distributed according to

$$s | \mathbf{Z}, a \sim \text{Unif}(0, \min_{k: \exists t,i, \mathbf{Z}_{ik}^{(t)}=1} a_k) \ . \tag{6}$$

We first sample the slice variable s according to Equation 6. We condition on s for the remainder of the MCMC iteration, which forces the features for which $a_k < s$ to be inactive, allowing us to discard them from the represented portion of $\mathbf{Z}$. We now extend the representation so that we have a and b parameters for all features k such that $a_k \geq s$. Here we are using the semi-ordered stick-breaking construction of the IBP feature probabilities [Teh et al., 2007], so we view the active features as being unordered, while the inactive features are in decreasing order of their a_k's. Consider the matrix whose columns each correspond to an inactive feature and consist of the concatenation of each

actor's $\mathbf{Z}$ values at each time for that feature. Since each entry in each column is distributed Bernoulli(a_k), we can view this as the inactive portion of an IBP with $M = NT$ rows. So we can follow Teh et al. [2007] to sample the a_k's for each of these features:

$$Pr(a_k|a_{k-1}, \mathbf{Z}^{:}_{:,>k} = 0) \propto \exp(\alpha \sum_{i=1}^{M} \frac{1}{i}(1-a_k)^i) \\ a_k^{\alpha-1}(1-a_k)^M \mathbb{I}(0 \leq a_k \leq a_{k-1}) \ , \quad (7)$$

where $\mathbf{Z}^{:}_{:,>k}$ is the entries of $\mathbf{Z}$ for all timesteps and all actors, with feature index greater than k. We do this for each introduced feature k, until we find an a_k such that $a_k < s$. The $\mathbf{Z}$s for these features are initially set to $\mathbf{Z}^{(t)}_{ik} = 0$, and the other parameters ($\mathbf{W}$, b_k) for these are sampled from their priors, e.g. $Pr(b_k|\gamma, \delta) \sim \text{Beta}(\gamma, \delta)$.

Having adaptively chosen the number of features to consider, we can now sample the feature values. The $\mathbf{Z}$s are sampled one $\mathbf{Z}_{ik}$ chain at a time via the forward-backward algorithm [Scott, 2002]. In the forward pass, we create the dynamic programming cache, which consists of the 2×2 matrices $\mathbf{P}_2 \dots \mathbf{P}_T$, where $\mathbf{P}_t = (p_{trs})$. Letting θ_{ik} be all other parameters and hidden variables not in $\mathbf{Z}_{ik}$, we have the following standard recursive computation,

$$p_{trs} = Pr(\mathbf{Z}^{(t-1)}_{ik} = r, \mathbf{Z}^{(t)}_{ik} = s|\mathbf{Y}^{(1)} \dots \mathbf{Y}^{(t)}, \theta_{ik}) \\ \propto \pi_{t-1}(r|\theta)\mathbf{Q}^{(ik)}(r,s)Pr(\mathbf{Y}^{(t)}|\mathbf{Z}^{(t)}_{ik} = s, \theta_{ik}), \\ \text{where } \ \pi_t(s|\theta) = Pr(\mathbf{Z}^{(t)}_{ik} = s|\mathbf{Y}^{(1)} \dots \mathbf{Y}^{(t)}, \theta_{ik}) \\ = \sum_r p_{trs} \quad (8)$$

In the backward pass, we sample the states in backwards order via $\mathbf{Z}^{(T)}_{ik} \sim \pi_T(.|\theta_{ik})$, and $Pr(\mathbf{Z}^{(t)}_{ik} = s) \propto p_{t+1,r,\mathbf{Z}^{(t+1)}_{ik}}$. We drop all inactive columns, as they are relegated to the non-represented portion of $\mathbf{Z}$. Next, we sample α, for which we assume a Gamma(α_a, α_b) hyper-prior, where α_a is the shape parameter and α_b is the inverse scale parameter. After integrating out the a_k's, $Pr(\mathbf{Z}|\alpha) \propto \alpha^{K^+} e^{-\alpha H_{NT}}$ from Equation 5. By Bayes' rule, $Pr(\alpha|\mathbf{Z}) \propto \alpha^{K^+ + \alpha_a - 1} e^{-\alpha(H_{NT} + \alpha_b)}$ is a Gamma($K^+ + \alpha_a, H_{NT} + \alpha_b$).

Next, we sample the a's and b's for non-empty columns. Starting with the finite model, using Bayes' rule and taking the limit as $K \to \infty$, we find that $a_k \sim \text{Beta}(c_k^{01}, c_k^{00} + 1)$. It is straightforward to show that $b_k \sim \text{Beta}(c_k^{11} + \gamma, c_k^{10} + \delta)$.

We next sample $\mathbf{W}$, which proceeds similarly to Miller et al. [2009]. Since it is non-conjugate, we use Metropolis-Hastings updates on each of the entries in $\mathbf{W}$. For each entry $w_{kk'}$, we propose $w^*_{kk'} \sim$ Normal($w_{kk'}, \sigma_w$). When calculating the acceptance ratio, since the proposal distribution is symmetric, the transition probabilities cancel, leaving the standard acceptance probability

$$Pr(\text{accept } w^*_{kk'}) = \min\{\frac{Pr(\mathbf{Y}|w^*_{kk'}, \dots)Pr(w^*_{kk'})}{Pr(\mathbf{Y}|w_{kk'}, \dots)Pr(w_{kk'})}, 1\} \ . \quad (9)$$

The intercept term α_0 is also sampled using Metropolis-Hastings updates with a Normal proposal centered on the current location.

5 Experimental Analysis

We analyze the performance of DRIFT on synthetic and real-world longitudinal networks. The evaluation tasks considered are predicting the network at time t given networks up to time $t-1$, and prediction of missing edges. For the forecasting task, we estimate the posterior predictive distribution for DRIFT,

$$Pr(\mathbf{Y}^t|\mathbf{Y}^{t-1}) = \sum_{\mathbf{Z}^t} \sum_{\mathbf{Z}^{1:(t-1)}} Pr(\mathbf{Y}^t|\mathbf{Z}^t)Pr(\mathbf{Z}^t|\mathbf{Z}^{t-1}) \\ Pr(\mathbf{Z}^{1:(t-1)}|\mathbf{Y}^{1:(t-1)}), \quad (10)$$

in Monte Carlo fashion by obtaining samples for $\mathbf{Z}^{1:(t-1)}$ from the posterior, using the MCMC procedure outlined in the previous section. For each sample, we then repeatedly draw $\mathbf{Z}^t$ by incrementing the Markov chains one step from $\mathbf{Z}^{(t-1)}$, using the learned transition matrix. Averaging the likelihoods of these samples gives a Monte Carlo estimate of the predictive distribution. This procedure also works in principle for predicting more than one timestep into the future.

An alternative task is to predict the presence or absence of edges between pairs of actors when this information is missing. Assuming that edge data are missing completely at random, we can extend the MCMC sampler to perform Gibbs updates on missing edges by sampling the value of each pair independently using Equation 1. To make predictions on the missing entries, we estimate the posterior mean of the predictive density of each pair by averaging the edge probabilities of Equation 1 over the MCMC samples. This was found to be more stable than estimating the edge probabilities from the sample counts of the pairs.

In our experiments, we compare DRIFT to its static counterpart, LFRM. Several variations of LFRM were considered. LFRM (all) treats the networks at each timestep as i.i.d samples. For forecasting, LFRM (last) only uses the network at the last time step $t-1$ to predict timestep t, while for missing data prediction LFRM (current) trains a LFRM model on the training entries for each timestep. The inference algorithm for LFRM is the algorithm for DRIFT with one time

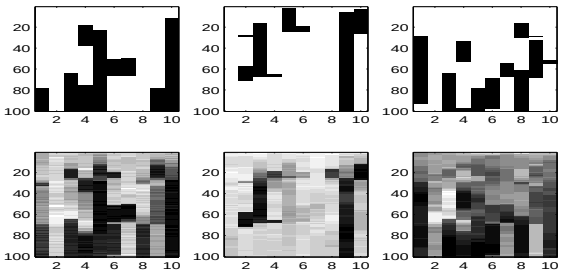


Figure 2: Ground truth (top) versus $\mathbf{Z}$'s learned by DRIFT (bottom) on synthetic data. Each image represents one feature, with rows corresponding to timesteps and columns corresponding to actors.

step. For both DRIFT and LFRM, all variables were initialized by sampling them from their priors. We also consider a baseline method which has a posterior predictive probability for each edge proportional to the number of times that edge has appeared in the training data (i.e. a multinomial), using a symmetric Dirichlet prior with concentration parameter set to the number of timesteps divided by 5 (so it increases with the amount of training data). We also consider a simpler method ("naive") whose posterior predictive probability for all edges is proportional to the mean density of the network over the observed time steps. In the experiments, hyperparameters were set to $\alpha_a = 3$, $\alpha_b = 1$, $\gamma = 3$, $\delta = 1$, and $\sigma_W = .1$. For the missing data prediction tasks, twenty percent of the entries of each dataset were randomly chosen as a test set, and the algorithms were trained on the remaining entries.

5.1 Synthetic Data

We first evaluate DRIFT on synthetic data to demonstrate its capabilities. Ten synthetic datasets were each generated from a DRIFT model with 10 actors and 100 timesteps, using a $\mathbf{W}$ matrix with 3 features chosen such that the features were identifiable, and a different $\mathbf{Z}$ sampled from its prior for each dataset.

Given this data, our MCMC sampler draws 20 samples from the posterior distribution, with each sample generated from an independent chain with 100 burn in iterations. Figure 2 shows the $\mathbf{Z}$s from one scenario, averaged over the 20 samples (with the number of features constrained to be 3, and with the features aligned so as to visualize the similarity with the true $\mathbf{Z}$). This figure suggests that the $\mathbf{Z}$s can be correctly recovered in this case, noting as in Miller et al. [2009] that the $\mathbf{Z}$s and $\mathbf{W}$s are not in general identifiable.

Table 1 shows the average AUC and log-likelihood scores for forecasting an additional network at

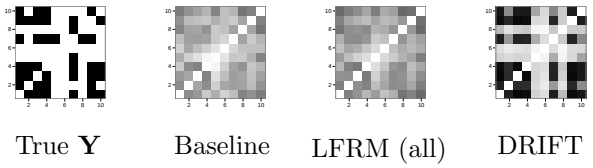


Figure 3: Held out $\mathbf{Y}$, and posterior predictive distributions for each method, on synthetic data.

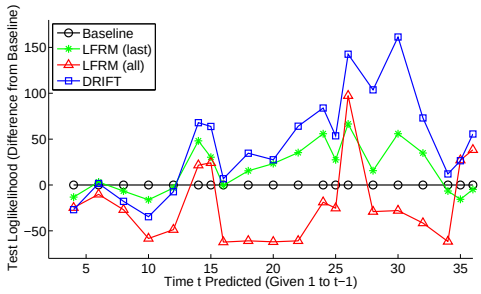


Figure 4: Test log-likelihood difference from baseline on Enron dataset at each time t.

timestep 101, and for predicting missing edges (the number of features was not constrained in these experiments). DRIFT outperforms the other methods in both log-likelihood and AUC on both tasks. Figure 3 illustrates this with the held-out $\mathbf{Y}$ and the posterior predictive distributions for one forecasting task.

5.2 Enron Email Data

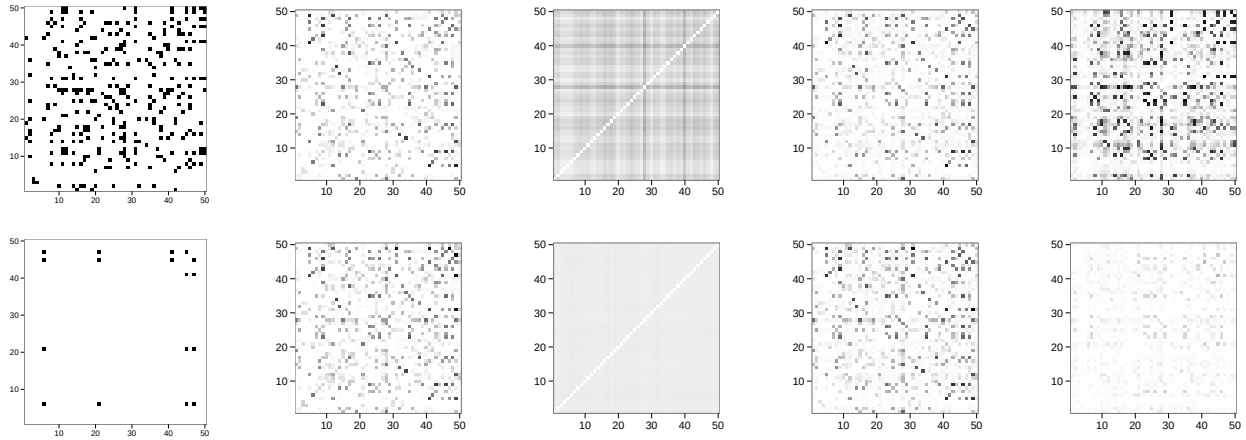
We also evaluate our approach on the widely-studied Enron email corpus [Klimt and Yang, 2004]. The Enron data contains 34182 emails among 151 individuals over 3 years. We aggregated the data into monthly snapshots, creating a binary sociomatrix for each snapshot indicating the presence or absence of an email between each pair of actors during that month. In these experiments, we take the subset involving interactions among the 50 individuals with the most emails.

For each month t, we train LFRM (all), LFRM (last), and DRIFT on all previous months 1 to $t-1$. In the MCMC sampler, we use 3 chains and a burnin length of 100, which we found to be sufficient. To compute predictions for month t for DRIFT, we draw 10 samples from each chain, and for each of these samples, we draw 10 different instantiations of $\mathbf{Z}^t$ by advancing the Markov chains one step. For LFRM, we simply use the sampled $\mathbf{Z}$'s from the posterior for prediction.

Table 1 shows the test log-likelihoods and AUC scores, averaged over the months from $t = 3$ to $t = 37$. Here, we see that DRIFT achieves a higher test log-likelihood and AUC than the LFRM models, the baseline and the "naive" method. Figure 4 shows the test log-likelihood

Table 1: Experimental Results

Synthetic Dataset	**Naive**	**Baseline**	**LFRM (last/current)**	**LFRM (all)**	**DRIFT**
Forecast LL	-31.6	-32.6	-28.4	-31.6	**−11.6**
Missing Data LL	-575	-490	-533	-478	**−219**
Forecast AUC	N/A	0.608	0.779	0.596	**0.939**
Missing Data AUC	N/A	0.689	0.675	0.691	**0.925**
Enron Dataset	**Naive**	**Baseline**	**LFRM (last/current)**	**LFRM (all)**	**DRIFT**
Forecast LL	-141	-108	-119	-98.3	**−83.5**
Missing Data LL	-1610	-1020	-1410	-981	**−639**
Forecast AUC	N/A	0.874	0.777	0.891	**0.910**
Missing Data AUC	N/A	0.921	0.803	0.933	**0.979**



(a) True $\mathbf{Y}$ (b) Baseline (c) LFRM (last) (d) LFRM (all) (e) DRIFT

Figure 5: Held out $\mathbf{Y}$ at time $t = 30$ (top row) and $t = 36$ (bottom row) for Enron, and posterior predictive distributions for each of the methods.

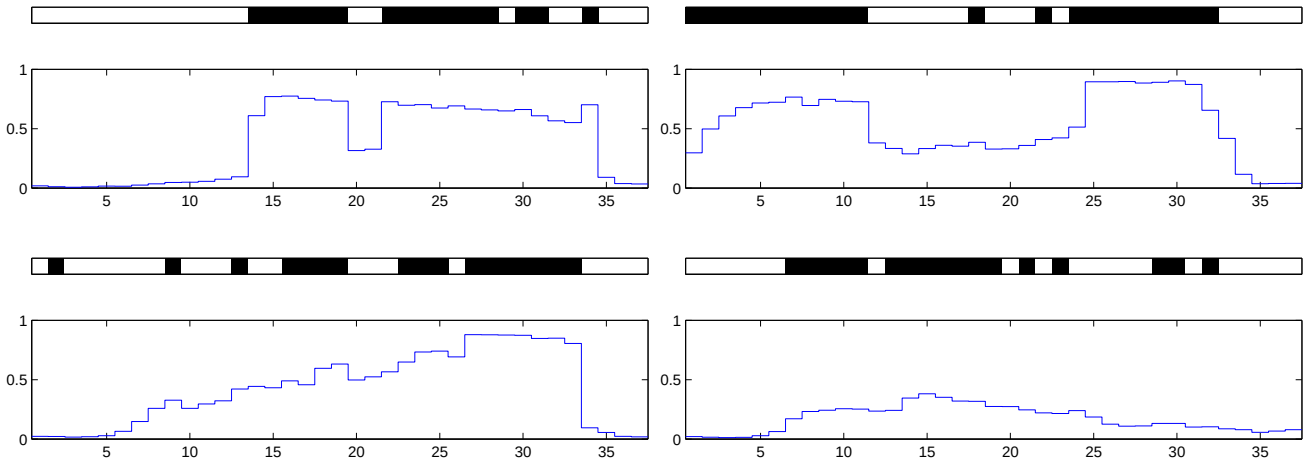


Figure 6: Estimated edge probabilities vs timestep for four pairs of actors from the Enron dataset. Above each plot the presence and absence of edges is shown, with black meaning that an edge is present.

k	**Baseline**	**LFRM** (current)	**LFRM** (all)	**DRIFT**
10	**10**	5	**10**	**10**
20	19	6	19	**20**
50	36	12	36	**48**
100	60	22	62	**90**
500	192	78	197	**301**
1000	285	142	290	**361**

Table 2: Number of true positives for the k missing entries predicted most likely to be an edge on Enron.

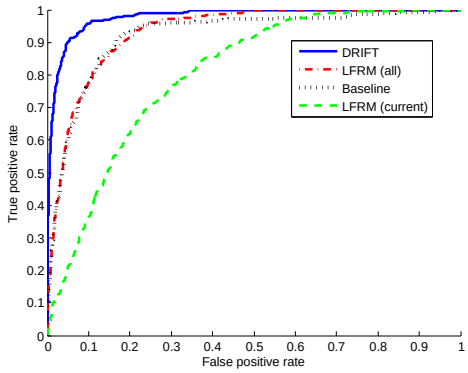


Figure 7: ROC curves for Enron missing data.

for each time step t predicted (given 1 to $t-1$). This plot suggests that all of the probabilistic models have difficulty beating the simple baseline early on (for $t < 12$). However, when t is larger, DRIFT performs better than the baseline and the other methods. For the last time step, LFRM (last) also does well relative to the other methods, since the network has become sparse at both that time step and the previous time step.

For the missing data prediction task, thirty MCMC samples were drawn for LFRM and DRIFT by taking only the last sample from each of thirty chains, with three hundred burn in iterations. AUC and log-likelihood results are given in Table 1. Under both metrics, DRIFT achieves the best performance of the models considered. Receiver operating characteristic curves are shown in Figure 7. Table 2 shows the number of true positives for the k most likely edges of the missing entries predicted by each method, for several values of k. As some pairs of actors almost always have an edge between them in each timestep, the baseline method is very competitive for small k, but DRIFT becomes the clear winner as k increases.

We now look in more detail at the ability of DRIFT to model the dynamic aspects of the network. Figure 5 shows the predictive distributions for each of the methods, at times $t = 30$ and $t = 36$. At time $t = 30$, the network is dense, while at $t = 36$, the network has become sparse. While LFRM (all) and the baseline method have trouble predicting a sparse network at $t = 36$, DRIFT is able to scale back and predict a sparser structure, since it takes into account the temporal sequence of the networks and it has learned that the network has started to sparsify before time $t = 36$. Figure 6 shows the edge probabilities over time for four pairs of actors. The pairs shown were hand picked "interesting" cases from the fifty most frequent pairs, although the performance on these pairs is fairly typical (with the exception of the bottom right plot). The bottom right plot shows a rare case where the model has arguably underfit, consistently predicting low edge probabilities for all timesteps.

We note that for some networks there may be relatively little difference in the predictive performance of DRIFT, LFRM, and the baseline method. For example, if a network is changing very slowly, it can be modeled well by LFRM (all), which treats graphs at each timestep as i.i.d. samples. However, DRIFT should perform well in situations like the Enron data where the network is systematically changing over time.

6 Conclusions

We have introduced a nonparametric Bayesian model for longitudinal social network data that models actors with latent features whose memberships change over time. We have also detailed an MCMC inference procedure that makes use of the IBP stick-breaking construction to adaptively select the number of features, as well as a forward-backward algorithm to sample the features for each actor at each time slice. Empirical results suggest that the proposed dynamic model can outperform static and baseline methods on both synthetic and real-world network data.

There are various interesting avenues for future work. Like the LFRM, the features of DRIFT are not directly interpretable due to the non-identifiability of $\mathbf{Z}$ and $\mathbf{W}$. We intend to address this in future work by exploring constraints on $\mathbf{W}$ and extending the model to take advantage of additional observed covariate information such as text. We also envision that one can generate similar models that handle continuous-time dynamic data and more complex temporal dynamics.

Acknowledgments This work was supported in part by an NDSEG Graduate Fellowship (CDB), an NSF Fellowship (AA), and by ONR/MURI under grant number N00014-08-1-1015 (CB, JF, PS). PS was also supported by a Google Research Award.

References

E.M. Airoldi, D.M. Blei, S.E. Feinberg, and E.P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.

C.T. Butts. A relational event framework for social action. *Sociological Methodology*, 38(1):155–200, 2008.

J. Chang and D.M. Blei. Relational topic models for document networks. *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2009.

Steven E. Fienberg and Stanley Wasserman. Categorical data analysis of single sociometric relations. *Sociological Methodology*, 12:156–192, 1981.

Wenjie Fu, Le Song, and Eric P. Xing. Dynamic mixed membership blockmodel for evolving networks. *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, pages 1–8, 2009.

T. Griffiths and Z Ghahramani. Infinite latent feature models and the Indian buffet process. *Advances in Neural Information Processing Systems*, 18:475–482, 2006.

Mark Handcock, Garry Robins, Tom Snijders, and Julian Besag. Assessing degeneracy in statistical models of social networks. *Journal of the American Statistical Association*, 76:33–50, 2003.

Peter Hoff. Modeling homophily and stochastic equivalence in symmetric relational data. In *Advances in Neural Information Processing Systems 20*, 2007.

Peter Hoff. Multiplicative latent factor models for description andprediction of social networks. *Computational and Mathematical Organization Theory*, 15(4):261–272, October 2008.

Peter Hoff, Adrian E Raftery, and Mark S Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

C. Kemp, J.B. Tenenbaum, T.L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, 2006.

B. Klimt and Y. Yang. Introducing the Enron corpus. In *First Conference on Email and Anti-Spam (CEAS)*, 2004.

Edward Meeds, Zoubin Ghahramani, Radford Neal, and Sam Roweis. Modeling dyadic data with binary latent factors. In *Advances in neural information processing systems*, 2007.

K.T. Miller, T.L. Griffiths, and M.I. Jordan. Nonparametric latent feature models for link prediction. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

Krzysztof Nowicki and Tom A B Snijders. Estimation and prediction of stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.

P. Sarkar and A.W. Moore. Dynamic social network analysis using latent space models. *SIGKDD Explorations: Special Edition on Link Mining*, 7(2):31–40, 2005.

P. Sarkar, S.M. Siddiqi, and G.J. Gordon. A latent space approach to dynamic embedding of co-occurrence data. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2007.

S.L. Scott. Bayesian hidden Markov models : Recursive computing in the 21st century. *Journal of the American Statistical Association*, 97(457):337– 351, 2002.

T.A.B. Snijders. Statistical methods for network dynamics. In *Proceedings of the XLIII Scientific Meeting, Italian Statistical Society*, pages 281–296, 2006.

Y.W. Teh, D. Görür, and Z. Ghahramani. Stick-breaking construction for the Indian buffet process. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2007.

J. Van Gael, Y.W. Teh, and Z. Ghahramani. The infinite factorial hidden Markov model. In *Advances in Neural Information Processing Systems*, volume 21, pages 1697 – 1704, 2009.