Dynamic Probabilistic Models for Latent Feature Propagation in Social Networks

Creighton Heaukulani CKH28@CAM.AC.UK
Zoubin Ghahramani ZOUBIN@ENG.CAM.AC.UK
University of Cambridge, Dept. of Engineering, Trumpington St., Cambridge, CB2 1PZ, UK

Abstract

Current Bayesian models for dynamic social network data have focused on modelling the influence of evolving unobserved structure on observed social interactions. However, an understanding of how observed social relationships from the past affect future unobserved structure in the network has been neglected. In this paper, we introduce a new probabilistic model for capturing this phenomenon, which we call *latent feature propagation*, in social networks. We demonstrate our model's capability for inferring such latent structure in varying types of social network datasets, and experimental studies show this structure achieves higher predictive performance on link prediction and forecasting tasks.

1. Introduction

Social networks have received a large amount of attention in recent literature due both to the popularity of sites such as Facebook and Twitter and to the abundance of data collected by these services about users and their social relationships. As with other forms of relational data, one often wants to be able to predict the behaviour or social interactions between two entities. Probabilistic approaches to analysing relational data have focused on latent variable representations of the objects in a network (Nowicki & Snijders, 2001; Kemp et al., 2006; Airoldi et al., 2008; Miller et al., 2009; Palla et al., 2012). For social networks, this is an intuitive approach, since the latent variables can represent the unobserved hobbies or interests of individuals in a network, which interact to explain observed social behaviour. In addition to predicting unobserved relationships, one often wants to be able to forecast behaviour in the network at some future time. To approach this problem, probabilistic models have been extended to accommodate dynamic network data with demonstrated success (Fu et al., 2009; Xing et al., 2010; Foulds et al., 2011).

Current probabilistic models for dynamic relational data lack the ability to directly use information from previous network observations to model future latent structure. Instead, they rely only on latent representations which evolve independently from the observations. This is inadequate for social networks, since our social relationships certainly influence both our personal interests and our future social interactions. We call this phenomenon *latent feature propagation* and to capture such behaviour, we introduce a new approach for modelling dynamic social network data. Our model uses observed social relationships in the network to model distributions over the latent structure in the network at the next time point. We motivate the particular form of our model with intuition from social network theory, and perform inference using Markov chain Monte Carlo (MCMC). We will demonstrate our model's ability to infer latent feature propagation in real social networks and provide experimental results which indicate such structure provides higher predictive power on link prediction and forecasting tasks.

The paper is organised as follows: section 2 provides background material on latent feature models for dynamic network data. Section 3 presents our generative model and the motivation for its parameterisation. We discuss related works in section 4, and in section 5 we present the MCMC inference procedure. In section 6, we provide the experimental results and present an example visualising latent feature propagation.

2. Dynamic Network Models

A network of N *actors* is represented by an $N \times N$ binary *link adjacency matrix* $\boldsymbol{Y}$, where $y_{ij} = 1$ if there is a "link" between person i and j and $y_{ij} = 0$ if there

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

is no link. In social networks, a link can be interpreted as friendship or correspondence, though, the applications for such a model can be far more general in many branches of the biological and physical sciences. In this work, we do not consider self-links, and edges are undirected (i.e., $\boldsymbol{Y}$ is a symmetric matrix and the diagonal elements are meaningless). We associate each actor i with a binary latent feature vector $\boldsymbol{h}_i$ of length K, with $h_{ik} = 1$ indicating actor i possesses feature k and $h_{ik} = 0$ indicating he does not. These *latent feature models* can be viewed as assigning the actors to multiple, overlapping latent clusters (Airoldi et al., 2008; Miller et al., 2009). Entries in $\boldsymbol{Y}$ are then conditionally independent Bernoulli random variables, given the latent feature assignments. In our social network application, we interpret these latent features as the hobbies or interests of person i. For example, feature k could mean "plays tennis" and $h_{ik} = 1$ means person i plays tennis. We will refer to the set of all feature vectors as the $N \times K$ binary matrix $\boldsymbol{H}$ where $\boldsymbol{h}_i$ is the i-th row of the matrix.

With dynamic network data, we observe a sequence of networks $\boldsymbol{Y}^{(1)}, \boldsymbol{Y}^{(2)}, \ldots, \boldsymbol{Y}^{(T)}$, each an observation of the edges in the network at time slices $t = 1, \ldots, T$. We assume that the corresponding sequence of latent features $\boldsymbol{H}^{(1)}, \boldsymbol{H}^{(2)}, \ldots, \boldsymbol{H}^{(T)}$ comprise a latent Markov chain. In this framework, the latent features evolve through time according to some Markov dynamics:

$$h_{ik}^{(t+1)} \Big| h_{ik}^{(t)} \sim \boldsymbol{Q}\left(h_{ik}^{(t)}, h_{ik}^{(t+1)}\right), \tag{1}$$

where $\boldsymbol{Q}(r, s)$ is a Markov transition probability of moving from feature state r to s, which can be a fixed parameter, feature specific, or otherwise arbitrary, as long as it defines a Markov transition matrix. In the context of *hidden Markov models* (HMM), this transition probability does not depend on any observations. While the state space of possible latent feature configurations may seem large, models of the form (1) actually factor the states into a matrix of state variables and are known as *factorial hidden Markov models*, within which tractable inference can be performed (Ghahramani & Jordan, 1997). With an HMM, static snapshots of the network are generated independently from all other observations, given the latent structure $\boldsymbol{H}^{(t)}$ at time t:

$$\begin{aligned} y_{ij}^{(t)} | \boldsymbol{h}_i^{(t)}, \boldsymbol{h}_j^{(t)} &\sim \text{Bernoulli}\left(\pi_{ij}^{(t)}\right) \\ \pi_{ij}^{(t)} &= \sigma\left(\boldsymbol{h}_i^{(t)\,T} \boldsymbol{V} \boldsymbol{h}_j^{(t)} + s\right) \qquad (2) \\ \nu_{kk'} &\sim \mathcal{N}(0, \sigma_\nu^2). \end{aligned}$$

where $\nu_{kk'}$, $k, k' = 1, \ldots, K$ are the elements of the $K \times K$ feature-interaction weight matrix $\boldsymbol{V}$, the function $\sigma(\cdot)$ is the logistic sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$, and s is a link-bias parameter representing an underlying global probability of a link. Different variants of this model can also be considered, for example a non-negative and diagonal feature-interaction weight matrix $\boldsymbol{V}$ corresponds to allowing links to only be affected by the possession of common features, and such an interaction can only increase the probability of a link.

3. A Latent Feature Propagation Model

In the context of social networks, HMMs assume that social interactions such as friendships are determined by latent hobbies or interests, and that the evolution of these interests over time do not depend on past observations of social interactions. Consider, however, the following two examples:

- If my friends enjoy playing tennis, I am likely to start playing recreational tennis.
- If a friend gets me to start playing tennis, I will likely befriend other tennis players.

In these examples, a person's interests are influenced by his current friends and, as he adopts their hobbies, his future friendships are influenced by his new interests. Viewed in this manner, we wish to capture the information propagating between the network observations and the latent structure over time.

In order to encode the latent feature propagation assumption, we model the Markov transition probability for a latent state as dependent on a weighted sum of neighbour features. In particular, if we set the initial or "null" states of the features off, i.e., $h_{ik}^{(0)} = 0$, $i = 1, \ldots, N$, $k = 1, \ldots, K$, then the latent features dynamically evolve according to

$$h_{ik}^{(t+1)} \Big| \mu_{ik}^{(t+1)} \sim \text{Bernoulli}\left[\sigma\left(c_k\left[\mu_{ik}^{(t+1)} - b_k\right]\right)\right] \tag{3}$$

$$\mu_{ik}^{(t+1)} = (1-\lambda_i) h_{ik}^{(t)} + \lambda_i \frac{h_{ik}^{(t)} + \sum_{j \in \varepsilon(i,t)} w_j h_{jk}^{(t)}}{1 + \sum_{j' \in \varepsilon(i,t)} w_{j'}} \tag{4}$$

where $\varepsilon(i, t)$ is the set of actors which are linked to actor i at time t and we interpret the parameters as:

1. $\lambda_i \in [0, 1]$: actor i's susceptibility to the influence of friends and $1 - \lambda_i$ is a corresponding measure of actor i's social independence,
2. $w_i \in \mathbb{R}_+$: the weight of influence of actor i,
3. $c_k \in \mathbb{R}_+$: a scale parameter for the persistence of feature k, and
4. $b_k \in \mathbb{R}_+$: a bias parameter for feature k.

We will discuss the interpretation of these *social parameters* in section 3.1, though, without strong prior knowledge, we give them the following broad priors $s \sim \mathcal{N}(-1, 4)$, $b_k \sim \text{Gamma}(1, 1)$, $w_i \sim \text{Gamma}(1, 1)$, $\lambda_i \sim \text{Beta}(2, 2)$, and $c_k \sim \text{Gamma}(1, 1)$. The generative process for our model is then completed with the likelihood function and prior distribution over the feature-interaction weights in $\boldsymbol{V}$ given by (2).

We can see from (3) that the distribution of $h_{ik}^{(t+1)}$ depends on the local topology of the observed network around actor i at the previous time step. A graphical model representing our latent feature propagation assumption is shown in figure 1, where the dependence on the parameters of the model are excluded for simplicity. Note in the graphical model that if we drop the edges from the observed networks to future latent feature states, we recover the HMM given by (1). Finally, although we focus in this work on social networks, we conjecture that feature propagation may also be useful for structure in networks studied in other domains.

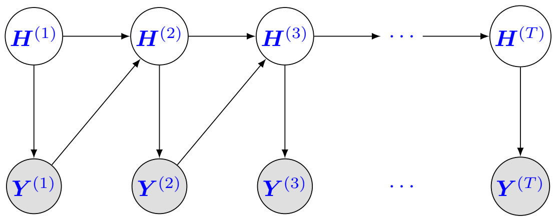


Figure 1. Graphical representation of the latent feature propagation model. Network observations (shaded grey) influence future latent features, thereby propagating information between the observed and latent structures throughout the network over time.

3.1. Social parameters

We briefly discuss the motivation for the particular form of our model; specifically, we focus on the *social parameters* in the feature transition probability, given by (3). This expression is well-defined for all configurations of the features $\boldsymbol{H}^{(1:T)}$. The state of feature $h_{ik}^{(t)}$ at time t stochastically depends on a weighted combination of the current state of the feature and a contribution from the local topology of the network around actor i at time $t-1$, given by

$$\frac{h_{ik}^{(t-1)} + \sum_{j \in \varepsilon(i,t-1)} w_j h_{jk}^{(t-1)}}{1 + \sum_{j' \in \varepsilon(i,t-1)} w_{j'}}. \qquad (5)$$

This contribution is determined by actor i's friends and is a normalized sum of their social influence weights $w_j \in \mathbb{R}_+$. Clearly, as w_j increases, the more influential actor j is to its connections. We can also see that if friend j has feature k turned off at the previous time point $h_{jk}^{(t)} = 0$, then the feature transition probability is decreased since w_j (a positive number) is present in the denominator of the ratio (5).

The weighting in the combination of $h_{ik}^{(t-1)}$ and (5) is determined by the susceptibility parameter $\lambda_i \in [0, 1]$ for actor i (shown by (4)). It is clear that as λ_i increases (and thus the social independence measure $1 - \lambda_i$ decreases), actor i will tend to adopt the features of his friends. If we consider the case when actor i has no friends at the previous time step, then the weighted combination collapses down to the state $h_{ik}^{(t)}$.

Finally, the combination of features is scaled and shifted by the feature-specific persistency parameter $c_k \in \mathbb{R}_+$ and $b_k \in \mathbb{R}_+$, respectively. We can see that as c_k increases, the part of the expression for the feature probability in (3) inside the sigmoid gets pushed further into the extremes of the logistic function, i.e., it becomes less likely for feature k to change from it's current state. Thus, large values could be appropriate for a feature such as “lives in London” which is unlikely to change over time, and low values appropriate for features such as “likes rock music”.

4. Related work

A classical approach to analysing both static and dynamic networks is with a family of models called *exponential random graph models*, all of which can be represented via a particular canonical parameterisation (Hanneke et al., 2010). These methods, however, suffer from both inconsistencies in their modelling framework and difficulties with computation (Handcock et al., 2003). Latent variable representations of dynamic networks have been presented in Xing et al. (2010), Fu et al. (2009), and Sarkar & Moore (2005), where the evolution of the network through time is determined by the underlying latent variables evolving according to a linear Gaussian model. The work in Westveld & Hoff (2011), and a particular study of social networks in Hoff et al. (2002), also use latent representations and treat network evolution as a regression problem, where parameters in the model represent expectations and covariances of connectivity patterns in the network. In contrast to these approaches, our model instead uses the framework of HMMs, where the latent variables stochastically depend on their state at the previous time step. Highly related to our model in this sense is the *dynamic relational infinite feature model* (DRIFT) from Foulds et al. (2011), which is the dynamic extension of the *latent feature relational model* (LFRM) from Miller et al. (2009). Like our model, DRIFT uses Markov switching dynamics to model the evolution of multiple latent Markov chains through time, however, it is unable to capture the latent feature propagation associated with social networks.

Previous work on modelling dynamic network structure as dependent on the observed structure at previous time points are usually referred to as *autoregressive* models. This literature is far too vast to attempt to summarise, however, these models typically deal with only the observations $\boldsymbol{Y}^{(t)}$ and do not use latent variable representations or Bayesian modelling approaches, for example, see Snijders (2006) and their application to social networks in Snijders (2001). In the social network literature, there is a great deal of work which studies the latent feature propagation phenomenon, often termed *social influence*, *selection*, and *trust propagation*, for instance see Crandall et al.. Some of these works do use latent variable approaches, but lack a probabilistic framework. To the best of our knowledge, our method is the first Bayesian model for dynamic network data which is able to model latent structure which directly depends on observed data from the network.

5. MCMC Inference

We develop a Markov chain Monte Carlo procedure to approximate samples from the posterior distributions of the latent variables in our model. Let $\boldsymbol{H}_{-ik}^{(t)}$ be the current states of all latent features in the model at time t, excluding $h_{ik}^{(t)}$. Let $\boldsymbol{\Omega}$ be the current state of all parameters and variables in the model which are constant across time, and let $\boldsymbol{Y}^{(1:t)}$ denote the sequence $(\boldsymbol{Y}^{(1)}, \ldots, \boldsymbol{Y}^{(t)})$ and similarly for $\boldsymbol{H}^{(1:t)}$.

5.1. Sample latent features $\boldsymbol{H}^{(1:T)}$

We use the forward-backward recursion algorithm from Scott (2002) to sample each latent feature chain $h_{ik}^{(1:T)}$ one at a time given the current state of all other variables and parameters in the model. The algorithm first defines a deterministic forward pass which runs down the chain starting at time one and, at each time point t, collects information from the data and parameters up to time t in a dynamic programming cache. A stochastic backwards pass is then defined which starts at time T and samples each $h_{ik}^{(t)}$ in backwards order $T, T-1, \ldots, 1$ conditioned on all of the data $\boldsymbol{Y}^{(1:T)}$, using the information collected during the forward pass.

In particular, the forward pass creates the dynamic programming cache $\boldsymbol{P}_2, \ldots, \boldsymbol{P}_T$, where $\boldsymbol{P}_t$ is the joint distribution of $(h_{ik}^{(t-1)}, h_{ik}^{(t)})$ given all variables and data up to time t. Thus, $\boldsymbol{P}_t = (p_{trs})$ is a 2×2 matrix with elements

$$\begin{aligned} p_{trs} &= P\left(h_{ik}^{(t-1)} = r, h_{ik}^{(t)} = s \middle| \boldsymbol{Y}^{(1:t)}, \boldsymbol{H}_{-ik}^{(1:t)}, \boldsymbol{\Omega}\right) \\ &\propto \pi_{ik}^{(t-1)}(r)\, Q_{ik}^{(t-1,t)}(r,s)\, P\left(\boldsymbol{Y}^{(t)} \middle| h_{ik}^{(t)} = s, \boldsymbol{H}_{-ik}^{(t)}, \boldsymbol{\Omega}\right) \end{aligned}$$

where the quantity

$$\pi_{ik}^{(t)}(s) = P\left(h_{ik}^{(t)} = s \middle| \boldsymbol{H}_{-ik}^{(1:t)}, \boldsymbol{Y}^{(1:t)}, \boldsymbol{\Omega}\right) = \sum_r p_{trs}$$

can be computed once $\boldsymbol{P}_t$ is known, setting up the next step in the recursion. Proportionality is reconciled with $\sum_r \sum_s p_{trs} = 1$. Here, $Q_{ik}^{(t-1,t)}(r,s)$ is the Markov transition probability from state $h_{ik}^{(t-1)} = r$ to $h_{ik}^{(t)} = s$. It is expressed as the Bernoulli density

$$\begin{aligned} Q_{ik}^{(t-1,t)}(r,s) &= \left[\rho_{ik}^{(t)}\right]^{s} \left[1 - \rho_{ik}^{(t)}\right]^{(1-s)} \\ \rho_{ik}^{(t)} &= \sigma(c_k[\mu_{ik}^{(t)} - b_k]) \end{aligned}$$

where the expression for $\mu_{ik}^{(t)}$ is given by (4), evaluated at $h_{ik}^{(t-1)} = r$ and the current states of all other variables in the model. Finally, the partial likelihood term $P(\boldsymbol{Y}^{(t)} | h_{ik}^{(t)} = s, \boldsymbol{H}_{-ik}^{(t)}, \boldsymbol{\Omega})$ is computed only for the observations $\boldsymbol{Y}^{(t)}$ at time t, evaluated at $h_{ik}^{(t)} = s$ and the current states of all other variables.

For the stochastic backwards pass, we first sample $h_{ik}^{(T)} \sim \pi_{ik}^{(T)}(\cdot)$, then sample each remaining state in the chain $h_{ik}^{(1:T-1)}$ in backwards order (i.e., $t = T-1, T-2, \ldots, 1$) via

$$\begin{aligned} P(h_{ik}^{(T-t)} = r \mid h_{ik}^{(T-t+1)}, \boldsymbol{Y}^{(1:T-t+1)}, \boldsymbol{\Omega}) \\ \propto p_{T-t+1, r, h_{ik}^{(T-t+1)}}. \end{aligned}$$

Scott (2002) shows that this procedure avoids the label-switching problem commonly encountered with mixture models. Sampling one chain with this algorithm computes as $O(2^{KT})$, while computing the likelihood in (2) for a single time point costs $O(K^2 N^2)$. The latter computation can be straightforwardly reduced to $O(K^2 L)$, where L is the number of observed links in the network, using the likelihood model in Mørup et al. (2011). Such a model variant allows scalability to large realistically-sized datasets, however, it suffers from reduced expressibility, allowing only positive interaction weights (given by the matrix $\boldsymbol{V}$ in (2)).

5.2. Sample feature interaction weights and social parameters

We use slice sampling (Neal, 2003) to learn the feature-interaction weights $\boldsymbol{V}$ and the social parameters (with prior distributions specified in section 3) in turn given the current values of all other variables in the model.

6. Experiments

We evaluate our method on two tasks: 1) the prediction of held-out links, and 2) forecasting a future, unseen network. We also provide an example which visualises latent feature propagation in a network.

6.1. Datasets and baseline methods

We experiment on a synthetic dataset generated from our model and two social network datasets. Comparison is made against two baseline methods.

Synthetic data: simulated network of 50 actors over 100 time steps with 10 features and parameters randomly drawn from their prior distributions. We train the data on the correct number of features ($K = 10$).

NIPS co-authorship: a subset of the widely studied NIPS co-authorship dataset.[1] The full dataset consists of co-authorship information among researchers on publications in the NIPS conference over the years from 1987 to 2003 ($T = 17$ years). We take the 110 researchers which are most connected across *all* time steps and use $K = 15$ latent features for training.

INFOCOM '06: proximity interactions between 78 students at the INFOCOM 2006 conference (Scott et al., 2009), recorded using wireless detector remotes given to student attendees over a period of about 93 continuous hours. We agglomerated the recordings into one hour-long time slices, symmetricised the link matrices (remote sightings aren't necessarily reciprocated) by keeping only reciprocated sightings. We also removed slices with less than 80 links (corresponding to late night and early morning hours), resulting in 50 time steps. We use $K = 10$ latent features for training.

Comparison to baseline methods: we compare the performance of our model (LFP) against two baseline methods. The first is the finite (parametric) version of the DRIFT model from Foulds et al. (2011), which is presented therein before taking the infinite limit of their model. We also implement a finite version of the LFRM model from Miller et al. (2009), which is a model for static networks. This is equivalent to using the finite version of the *Indian buffet process* presented in Griffiths & Ghahramani (2011) as a prior distribution over the feature matrix. Essentially, each feature assignment has a beta-Bernoulli distribution with the Bernoulli parameter integrated out. By re-examining figure 1, we can see that (finite) DRIFT corresponds to dropping the edges from observations to latent features, and (finite) LFRM corresponds to additionally dropping the edges between latent features.

6.2. Prediction of missing links

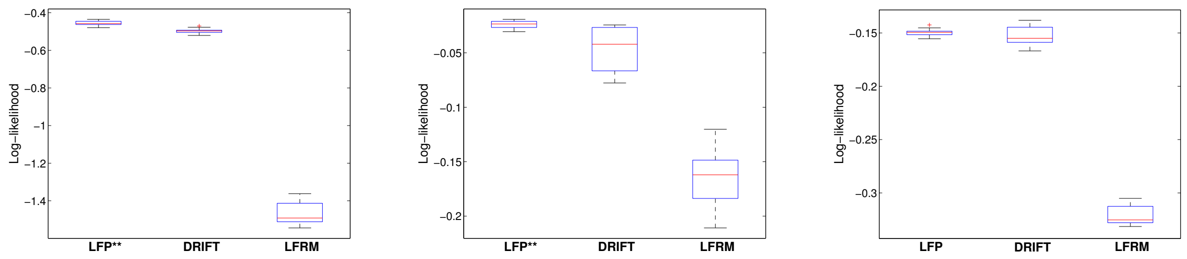
We first evaluate our model and the baseline methods on the task of predicting missing edges in a network, interpreted as observing some social relations (or lack thereof) in a network and having to predict a set of unobserved interactions. At each time point, we hold out a different 20% of the interactions (either links or non-links) as a test set chosen uniformly at random. We run the MCMC inference procedure; for the synthetic and INFOCOM datasets, we use 1,000 burn-in iterations for LFP and finite DRIFT, and 800 iterations for finite LFRM. For the NIPS dataset, we used 800 burn-in iterations for LFP and finite DRIFT and 600 iterations for finite LFRM. We found all of these burn-in times to be sufficient. Since LFRM is a static model, we train it on each time step individually.

After the Markov chains have burned-in, we collected 300 samples of the latent features and model parameters from their approximated steady state distribution. We then estimate the posterior mean of the link probability for each interaction in the test set by averaging (2) over the collected samples. These link probabilities are then used to evaluate the log-likelihood of each model computed on the test set. We also compute the area under the curve (AUC) of the receiver operating characteristic metric. We perform 10 repeats of this procedure, each time holding out a different random 20% of the data as a test set. Box-plots of the results on the 10 repeats are shown in figures 2 and 3 for the log-likelihoods and AUC scores, respectively. In each plot, we designate with ($\star\star$) any statistically significant results compared to the second best performing method, based on a T-test at a 0.05 significance level.

On the synthetic dataset, LFP achieves a higher log-likelihood under the test data than both baseline methods, with a statistically significant improvement over the next best result (from finite DRIFT) based on the T-test (p-value of 1.7×10^{-09}). Without a capability to model the dynamics of the network, finite LFRM is drastically outperformed by both LFP and finite DRIFT and is noticeably less robust (higher variance over the repeats). LFP also achieves the highest AUC score, with a statistically significant improvement (p-value of 9×10^{-09}) over finite LFRM, the second best performing method. Interestingly, finite LFRM significantly outperforms finite DRIFT on this dataset, perhaps indicating that modelling latent feature dynamics based only on their popularity is a bad model when feature propagation is truly present in a network.

On the NIPS dataset, LFP again outperforms both baselines on the log-likelihood and AUC metrics, the result being statistically significant in both cases (p-

[1] Obtained from http://ai.stanford.edu/∼gal/data.html

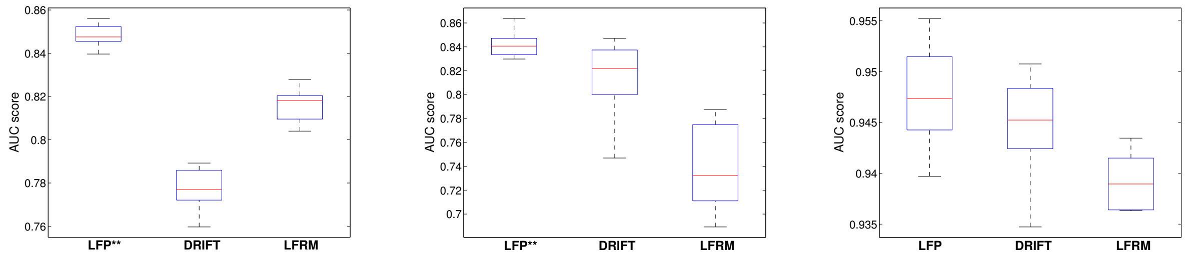


(a) Simulated data

(b) NIPS dataset

(c) INFOCOM dataset

Figure 2. Log-likelihood of the test set. Box-plots are over 10 repeats, each holding out a different 20% of the data. All results are averaged over 300 samples drawn from the steady state distribution following a burn-in period as described in the text. Statistically significant results are indicated by a (⋆⋆) based on a T-test at a 0.05 significance level.



(a) Simulated data

(b) NIPS dataset

(c) INFOCOM dataset

Figure 3. AUC scores for classifying the test set in the same experiment as in figure 2. Statistically significant results are indicated by (⋆⋆) based on a T-test at a 0.05 significance level.

values of 0.0091 and 0.0098, respectively). Upon examining the result plots, LFP appears to also provide the most robust estimates. On the INFOCOM dataset, the median (and mean) of both the log-likelihood and AUC score are highest for LFP, however, the results are not statistically significant based on a T-test (p-values of 0.1806 and 0.2055, respectively). Despite this, we will see in the forecasting task some evidence that LFP is still a preferable model on this dataset.

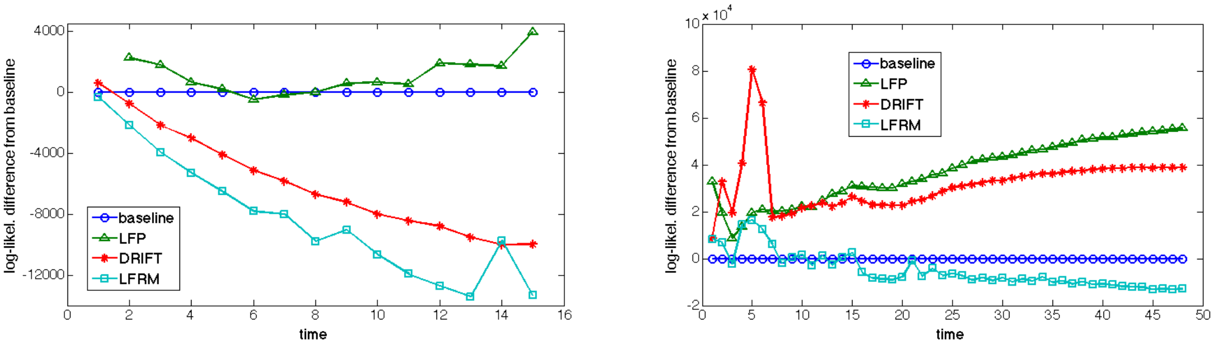
6.3. Forecasting

Our next task is to forecast the interactions in a previously unseen network. Following Foulds et al. (2011), for each time $t = 1, \dots, T$, we train a model on the data from time 1 to $t-1$ and estimate the predictive distribution of the unseen network $\boldsymbol{Y}^{(t)}$ using

$$P(\boldsymbol{Y}^{(t)}|\boldsymbol{Y}^{(1:t-1)}) = \sum_{\boldsymbol{H}^{(t)}} \sum_{\boldsymbol{H}^{(1:t-1)}} P(\boldsymbol{Y}^{(t)}|\boldsymbol{H}^{(t)}) \times P(\boldsymbol{H}^{(t)}|\boldsymbol{H}^{(t-1)}, \boldsymbol{Y}^{(t-1)}) P(\boldsymbol{H}^{(1:t-1)}|\boldsymbol{Y}^{(1:t-1)}).$$

In order to approximate this distribution, we obtain multiple samples of the features $\boldsymbol{H}^{(1:t-1)}$ following the MCMC inference procedure. Using each of these samples, we draw repeated samples of $\boldsymbol{H}^{(t)}$ using the learnt feature transition probabilities and the data $\boldsymbol{Y}^{(t)}$ from time t. In particular, for each sample obtained for $\boldsymbol{H}^{(1:t-1)}$, we must draw multiple samples of $\boldsymbol{H}^{(t)}$ to accurately approximate its posterior distribution in order to marginalise it out. In our experiments, we generate 10 such samples for each of the 300 samples collected for $\boldsymbol{H}^{(1:t-1)}$. For LFRM, we predict a network $\boldsymbol{Y}^{(t)}$ at time t by training on only $\boldsymbol{Y}^{(t-1)}$ and using the predictive distribution for $\boldsymbol{Y}^{(t-1)}$ as a model for $\boldsymbol{Y}^{(t)}$.

We perform this experiment on the NIPS and INFOCOM datasets. We used the same number of latent chains for each dataset as in the prediction task, and burn-in the incremental time steps as follows: for $t = 1$ we used 300 burn-in iterations. We then use the burned-in states of the variables and parameters in the model to initialise training on the next time point $t = 2$ (after adding the data $\boldsymbol{Y}^{(2)}$), and so on. Upon analysing the burn-in periods for all time increments, we found this amount of training time to be more than sufficient for both datasets on all models. For accu-



(a) NIPS dataset, $K = 15$ features.

(b) INFOCOM dataset, $K = 10$ features.

Figure 4. Forecasting a future unseen network. Differences from a naive baseline of the log-likelihoods of $\boldsymbol{Y}^{(t)}$ after training on $\boldsymbol{Y}^{(1:t-1)}$, performed sequentially for each $t = 1, \ldots, T$. No data was held out of the training sets.

rate comparison, we also implement the same naive baseline method used for comparison in Foulds et al. (2011), where at each time t, the posterior predictive probability for a link in the network is proportional to the previous number of occurrences of that link in the training data in $t = 1, \ldots, t-1$. This is equivalent to an independent Dirichlet-multinomial distribution on each $y_{ij}^{(t)}$, $i, j = 1, \ldots, N$, and a symmetric Dirichlet prior distribution is used with parameter $t/5$ (note this increases with the amount of training data).

In figure 4, we show the difference of the test log-likelihoods from the baseline for each predicted time point t. First, we note that the NIPS dataset is very sparse so it is not surprising that the baseline method performs relatively well, since it will more often than not predict a non-link. LFP is the only method which is able to consistently achieve higher likelihoods at each time point than the naive baseline method, with finite DRIFT and finite LFRM consistently under performing. On the INFOCOM dataset, we see that LFP and finite DRIFT both perform consistently well, with LFP clearly providing the best results. Near the beginning of the dataset, there are a few irregular time points in which both finite LFRM and finite DRIFT perform unusually well, perhaps attributing to the lack of a statistically significant result for LFP in the prediction tasks on the INFOCOM dataset.

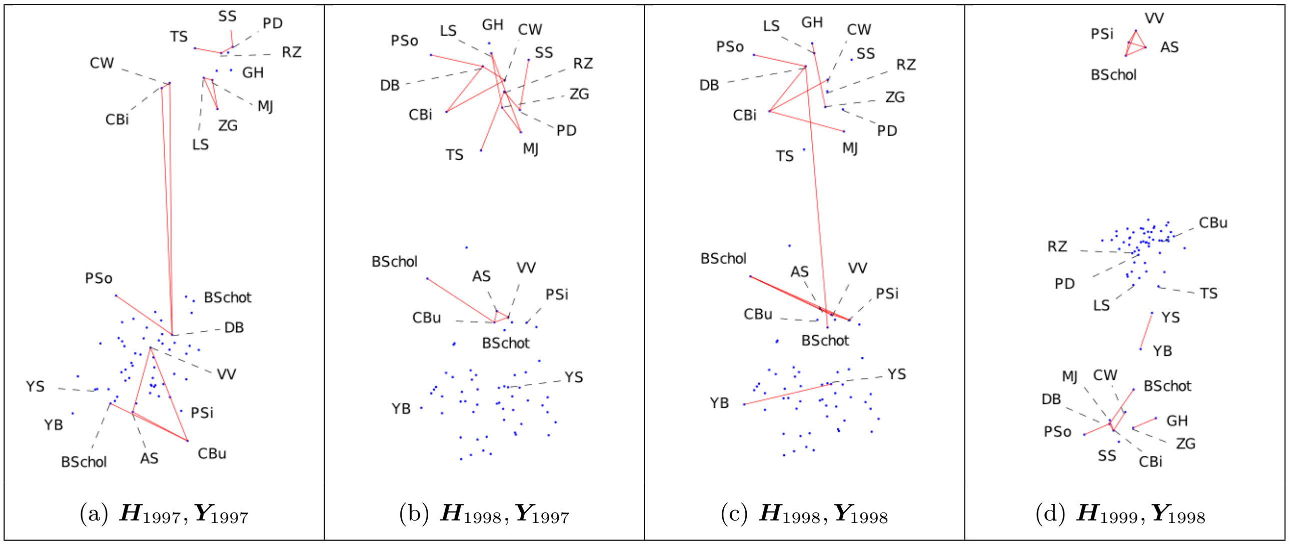
We make one final note that the two social network datasets are very different in nature. As already mentioned during the forecasting task, the NIPS dataset is very sparse across the time steps and, conversely, the INFOCOM dataset is very dense. The superior performance of LFP in our experiments on both of these datasets evidences the appropriateness of the latent feature propagation assumption for widely varying types of social network data.

6.4. Feature propagation

We demonstrate our model's ability to capture the phenomenon of latent feature propagation throughout a network over time. We run the MCMC inference procedure described in section 5 to learn a latent feature representation of a small ($N = 70$) subset of the NIPS dataset with no data held out. We learn a set of $K = 15$ features resulting in an $N \times K$ matrix of Bernoulli probabilities at each time t. These probabilities define the distribution over the latent feature matrix $\boldsymbol{H}^{(t+1)}$ at time $t + 1$, the expressions of which are given by (3) (the transition probability for $h_{ik}^{(t+1)}$ is the Bernoulli parameter). We perform a singular value decomposition of these feature probability matrices in order to reduce the representation of each author's feature vector to a location in two-dimensional space, using the two most significant components.

In figure 5, we display the evolution of the authors' latent feature representations from 1997 to 1999. Each blue dot represents an author, 20 of which are labelled in order to track their movement across time (see the caption of figure 5 for the author names). Here, closeness in the 2D-reduced space represents similarity in the original 15-dimensional feature space. We display the co-authorship observations (as red lines) in a particular sequence to demonstrate that if two researchers co-author a paper in one year, their latent feature representations become closer in the following year.

In figure 5(a), we begin with the authors' latent feature representations in 1997, along with the observed co-authorships in 1997. In figure 5(b), we advance the latent feature representations by one year to 1998, while still displaying the same co-authorships from 1997 so that the reader can easily compare the movement of linked authors. We can see that the feature representa-



(a) $\boldsymbol{H}_{1997}, \boldsymbol{Y}_{1997}$ (b) $\boldsymbol{H}_{1998}, \boldsymbol{Y}_{1997}$ (c) $\boldsymbol{H}_{1998}, \boldsymbol{Y}_{1998}$ (d) $\boldsymbol{H}_{1999}, \boldsymbol{Y}_{1998}$

Figure 5. Visualising feature propagation in a subset of the NIPS dataset from 1997 to 1999 using the LFP model with $K = 15$ features. A sequence of learnt feature matrices are plotted (as blue dots) onto a 2D-reduced feature space and overlaid with the co-authorship observations (as red lines) in a particular sequence to show that if two researchers co-author a paper, their latent feature representations become closer in the following year (see the text for details). Several authors are tracked: Barber_D (DB); Bengio_Y (YB); Bishop_C (CBi); Burges_C (CBu); Dayan_P (PD); Ghahramani_Z (ZG); Hinton_G (GH); Jordan_M (MJ); Saul_L (LS); Scholkopf_B (BSchol); Schottky_B (BSchot); Sejnowski_T (TS); Simard_P (Psi); Singer_Y (YSi); Singh_S (SS); Smola_A (AS); Sollich_P (PSo); Vapnik_V (VV); Williams_C (CW); Zemel_R (RZ).

tions of one inter-connected group of authors have clustered towards the top of the figure. Another smaller set of co-authors have grouped in the middle of the figure. This convergence of authors in latent feature space following a co-authorship in the previous year is an example of latent feature propagation. Such configurations improve the explanatory power of the model, as evidenced by the experimental results on the prediction and forecasting tasks.

We can see the same pattern in the following year. Continuing to figure 5(c), we again show the feature representation from 1998 and we update the co-authorships to 1998 (one year following those in figure 5(b)). Many of the co-authorships in 1997 have occurred again in 1998 and some authors that are far apart in feature space in 1997 have collaborated on a paper in 1998. With latent feature propagation, we expect these authors to move closer in latent feature space in the following year, and indeed this is the case. In figure 5(d), we advance the feature representations to 1999 while still displaying the co-authorships from the previous year. We can see that the feature space has reorganised itself using latent feature propagation to form two inter-connected groups of authors which have been distinctly separated from the large number of authors with no co-authorship information.

7. Future Work

In this work, we only consider a fixed, finite number of features, however, Bayesian non-parametrics employ methods which can automatically learn the complexity of a network model (Kemp et al., 2006; Miller et al., 2009; Palla et al., 2012) (see also Roy & Teh (2009) and Lloyd et al. (2012) for interesting alternatives). Non-parametric extensions of the HMM allowing a potentially infinite number of latent states are based on the *infinite HMM* from Beal et al. (2002) (see also Teh et al. (2006)) and the *infinite factorial HMM* from Van Gael et al. (2009). Applications to network data were done by Xing et al. (2010) and Foulds et al. (2011) (see also Xu et al. (2006)). These methods require the distribution over the data to be invariant to permutations of the transitions in the state space, a property called *Markov exchangeability* which has yet to be reconciled in our model. However, we envision that some form of this symmetry may exist on the joint space $\{\boldsymbol{H}^{(t)}, \boldsymbol{Y}^{(t)}\}_{t\geq 1}$, which could likely be exploited to take the infinite limit $K \to \infty$ of our model.

Acknowledgments

We thank Konstantina Palla, Dave Knowles and Emil Hewage for helpful discussions. This work was supported by EPSRC Grant EP/I036575/1.

References

Airoldi, E.M., Blei, D.M., Fienberg, S.E., and Xing, E.P. Mixed-membership stochastic blockmodels. *JMLR*, 9:1981–2014, 2008.

Beal, M.J., Ghahramani, Z., and Rasmussen, C. The infinite hidden Markov model. In *Proc. NIPS*, 2002.

Crandall, D., Cosley, D., Huttenlocher, D., Kleinberg, J., and Suri, S. Feedback effects between similarity and social influence. In *Proc. SIGKDD*.

Foulds, J., DuBois, C., Asuncion, A.U., Butts, C.T., and Smyth, P. A dynamic relational infinite feature model for longitudinal social networks. In *Proc. AISTATS*, April 2011.

Fu, W., Song, L., and Xing, E.P. Dynamic mixed membership stochastic blockmodel for evolving networks. In *Proc. ICML*, 2009.

Ghahramani, Z. and Jordan, M.I. Factorial hidden Markov models. *Machine Learning*, 29:245–273, 1997.

Griffiths, T. and Ghahramani, Z. The Indian buffet process: An introduction and review. *JMLR*, 12: 1185–1224, 2011.

Handcock, M. S., Robins, G., Snijders, T., Moody, J., and Besag, J. Assessing degeneracy in statistical models of social networks. *J. Am. Statist. Assoc.*, 76:33–50, 2003.

Hanneke, S., Fu, W., and Xing, E.P. Discrete temporal models of social networks. *Electronic Journal of Statistics*, 4:585–605, 2010.

Hoff, P., Raftery, A., and Handcock, M. Latent space approaches to social network analysis. *J. Am. Statist. Assoc.*, 97(460):1090–1098, 2002.

Kemp, Charles, Tenenbaum, Joshua B., Griffiths, Thomas L., Yamada, Takeshi, and Ueda, Naonori. Learning systems of concepts with an infinite relational model. In *Proc. AAAI*, 2006.

Lloyd, J.R., Orbanz, P., Ghahramani, Z., and Roy, D.M. Random function priors for exchangeable arrays with applications to graphs and relational data. In *Proc. NIPS*, 2012.

Miller, K.T., Griffiths, T.L., and Jordan, M.I. Nonparametric latent feature models for link prediction. In *Proc. NIPS*, 2009.

Mørup, M., Schmidt, M.N., and Hansen, L.K. Infinite multiple membership relational modeling for complex networks. In *Proc. MLSP*. IEEE, 2011.

Neal, R.M. Slice sampling. *Annals of Statistics*, 31(3): 705–741, 2003.

Nowicki, K. and Snijders, T. Estimation and prediction for stochastic blockstructures. *J. Am. Statist. Assoc.*, 96(455):1077–1087, 2001.

Palla, K., Knowles, D., and Ghahramani, Z. An infinite latent attribute model for network data. In *Proc. ICML*, 2012.

Roy, D.M. and Teh, Y.W. The Mondrian process. In *Proc. NIPS*, 2009.

Sarkar, P. and Moore, A. Dynamic social network analysis using latent space models. *SIGKDD Explor. Newsl.*, 7(2):31–40, December 2005.

Scott, J., Gass, R., Crowcroft, J., Hui, P., Diot, C., and Chaintreau, A. CRAWDAD data set cambridge/haggle (v. 2009-05-29). Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle, May 2009.

Scott, S. Bayesian methods for hidden Markov models: Recursive computing in the 21st century. *J. Am. Statist. Assoc.*, 97(457):337–351, 2002.

Snijders, T. The statistical evaluation of social network dynamics. *Sociological Methodology*, 31(1): 361–395, 2001.

Snijders, T. Statistical methods for network dynamics. In *Proc. Scientific Meeting*, pp. 281–296. Italian Statistical Society, 2006.

Teh, Y.W., Jordan, M.I., Beal, M.J., and Blei, D.M. Hierarchical dirichlet processes. *J. Am. Statist. Assoc.*, 101(476):1566–1581, 2006.

Van Gael, J., Teh, Y.W., and Ghahramani, Z. The infinite factorial hidden Markov model. In *Proc. NIPS*, 2009.

Westveld, A. and Hoff, P. A mixed effects model for longitudinal relational and network data, with applications to international trade and conflict. *Ann. Appl. Stat.*, 5(2A):843–872, 2011.

Xing, E., Fu, W., and Song, L. A state-space mixed-membership blockmodel for dynamic network tomography. *Ann. Appl. Stat.*, 4(2):535–566, 2010.

Xu, Z., Tresp, V., Yu, K., and Kriegel, H.P. Learning infinite hidden relational models. In *Proc. UAI*, 2006.