Nonparametric Dynamic Network Modeling

Ayan Acharya
Dept. of ECE, UT Austin
aacharya@utexas.edu

Avijit Saha
Dept. of CSE, IIT Madras
avijit@cse.iitm.ac.in

Mingyuan Zhou
Dept. of IROM, UT Austin
mzhou@utexas.edu

Dean Teffer
Applied Research Laboratories, UT Austin
dean.teffer@arlut.utexas.edu

Joydeep Ghosh
Dept. of ECE, UT Austin
ghosh@ece.utexas.edu

ABSTRACT

Certain relational datasets, such as those representing social networks, evolve over time, motivating the development of time-series models where the characteristics of the underlying groups of entities can adapt with time. This paper proposes the **D**ynamic **G**amma **P**rocess **P**oisson **F**actorization for **N**etwork (D-NGPPF) modeling framework, wherein binary network entries are modeled using a truncated Poisson distribution and the ideal number of network groups is discovered from the data itself. Crucially, a Gamma-markov chain enables the characteristics of these groups to smoothly evolve over time. Exploiting the properties of the Negative Binomial distribution and a novel data augmentation technique, closed form Gibbs sampling updates are derived that yield superior empirical results for both synthetic and real world datasets.

Keywords

Dynamic Network modeling, Poisson factorization, Gamma Process

1. INTRODUCTION

Many complex social and biological interactions can be naturally represented as graphs. Often these graphs evolve over time. For example, an individual in a social network can get acquainted with a new person, an author can collaborate with a new author to write a research paper and proteins can change their interactions to form new compounds. Consequently, a variety of statistical and graph-theoretic approaches have been proposed for modeling both static and dynamic networks [2; 10; 11; 21; 23; 30; 31; 37; 38].

Of particular interest in this work are scalable techniques that can identify groups or communities and track their evolution. Existing non-parametric Bayesian approaches for this task promise to solve the model selection problem of identifying an appropriate number of groups, but are computationally intensive, and often do not match the characteristics of real datasets. All such models assume that the data comes from a latent space that has *either* discrete sets of configurations [9; 23; 30] *or* is modeled using Gaussian distribution [10; 15; 37]. Approaches that employ discrete latent states do not have closed-form inference updates, mostly due to the presence of probit or logit links. On the other hand, Gaussian assumption is often overly restrictive for modeling binary matrices. Since the inference techniques for linear dynamical systems are well-developed, one usually is tempted to connect a binary observation to a latent Gaussian random variable using the probit or logit links. Such approaches, however, involve heavy computation and lack intuitive interpretation of the latent states.

This work attempts to address such inadequacies by introducing an efficient and effective model for binary matrices that evolve over time. Its contributions include:

- A novel non-parametric Gamma Process dynamic network model that predicts the number of latent network communities from the data itself.
- A technique for allowing the weights of these latent communities to vary smoothly over time using a Gamma-Markov chain, the inference of which is solved using an augmentation trick associated with the Negative Binomial distribution together with a forward-backward sampling algorithm, each step of which has closed-form updates.
- Empirical results indicating clear superiority of the proposed dynamic network model as compared to existing baselines for dynamic and static network modeling.

The rest of the paper is organized as follows. Pertinent background and related works are outlined in Section 2. A detailed description of the Gamma Process network modeling is provided in Section 3 which is then followed by a description of the Dynamic Gamma Process network model in Section 4. Empirical results for both synthetic and real-world data are reported in Section 5. Finally, the conclusions and future works are listed in Section 6.

2. BACKGROUND AND RELATED WORK

2.1 Negative Binomial Distribution

Lemma 2.1. *Let $x_k \sim Pois(\zeta_k)\ \forall k$, $X = \sum_{k=1}^{K} x_k$, $\zeta = \sum_{k=1}^{K} \zeta_k$. If $(y_1, \cdots, y_K) \sim mult(X; \zeta_1/\zeta, \cdots, \zeta_K/\zeta)$, then the following holds:*

$$P(x_1, \cdots, x_K) = P(y_1, \cdots, y_K; X). \tag{1}$$

The negative binomial (NB) distribution $m \sim \text{NB}(r, p)$, with probability mass function (PMF) $\Pr(M = m) = \frac{\Gamma(m+r)}{m!\Gamma(r)} p^m (1-p)^r$ for $m \in \mathbb{Z}$, can be augmented into a gamma-Poisson construction as $m \sim \text{Pois}(\lambda)$, $\lambda \sim \text{Gamma}(r, p/(1-p))$, where the gamma distribution is parameterized by its shape r and scale $p/(1-p)$. It can also be augmented under a compound Poisson representation as $m = \sum_{t=1}^{l} u_t, u_t \overset{iid}{\sim} \text{Log}(p), l \sim \text{Pois}(-r\ln(1-p))$, where $u \sim \text{Log}(p)$ is the logarithmic distribution [18].

WOODSTOCK '97 El Paso, Texas USA

Lemma 2.2 ([39]). *If $m \sim \mathrm{NB}(r,p)$ is represented under its compound Poisson representation, then the conditional posterior of l given m and r has PMF:*

$$\Pr(l=j|m,r) = \tfrac{\Gamma(r)}{\Gamma(m+r)}|s(m,j)|r^j,\ j=0,1,\cdots,m, \quad (2)$$

where $|s(m,j)|$ are unsigned Stirling numbers of the first kind. We denote this conditional posterior as $l \sim \mathrm{CRT}(m,r)$, a Chinese restaurant table (CRT) count random variable, which can be generated via $l=\sum_{n=1}^{m} z_n, z_n \sim \mathrm{Bernoulli}(r/(n-1+r))$.

Lemma 2.3. *If $\lambda \sim$ Gamma$(r,1/c), x_i \sim$ Poisson$(m_i\lambda)$, then $x=\sum_i x_i \sim$ NB(r,p), where $p=\frac{\sum_i m_i}{c+\sum_i m_i}$.*

Lemma 2.4. *If $\lambda \sim$ Gamma$(r,1/c), x_i \sim$ Poisson$(m_i\lambda)$, then $\lambda|\{x_i\} \sim$ Gamma$(r+\sum_i x_i, 1/(c+\sum_i m_i))$.*

Lemma 2.5. *If $x_i \sim \mathrm{Pois}(m_i r_2)$, $r_2 \sim \mathrm{Gamma}(r_1, 1/d)$, $r_1 \sim \mathrm{Gamma}(a, 1/b)$, , then $(r_1|-) \sim \mathrm{Gamma}(a+\ell, 1/(b-log(1-p)))$ where $(\ell|x,r_1) \sim \mathrm{CRT}(\sum_i x_i, r_1), p=\sum_i m_i/(d+\sum_i m_i)$. The proof and illustration can be found in Section 3.3 of [1].*

Lemma 2.6. *If $r_i \sim$ Gamma$(a_i, 1/b)\ \forall i \in \{1,2,\cdots,K\}, b \sim$ Gamma$(c,1/d)$, then $b|\{r_i\} \sim$ Gamma$(\sum_{i=1}^{K} a_i + c, 1/(\sum_{i=1}^{K} r_i + d))$.*

2.2 Gamma Process

Following [36], for any $\nu^+ \geq 0$ and any probability distribution $\pi(dpd\omega)$ on the product space $\mathbb{R}\times\Omega$, let $K^+ \sim \mathrm{Pois}(\nu^+)$ and $(p_k,\omega_k) \overset{iid}{\sim} \pi(dpd\omega)$ for $k=1,\cdots,K^+$. Defining $\mathbf{1}_A(\omega_k)$ as being one if $\omega_k \in A$ and zero otherwise, the random measure $\mathcal{L}(A) \equiv \sum_{k=1}^{K^+} \mathbf{1}_A(\omega_k)p_k$ assigns independent infinitely divisible random variables $\mathcal{L}(A_i)$ to disjoint Borel sets $A_i \subset \Omega$, with characteristic functions:

$$\mathbb{E}\left[e^{it\mathcal{L}(A)}\right] = \exp\left\{\int\int_{\mathbb{R}\times A}(e^{itp}-1)\nu(dpd\omega)\right\}, \quad (3)$$

where $\nu(dpd\omega) \equiv \nu^+\pi(dpd\omega)$. A random signed measure $\mathcal{L}$ satisfying the above characteristic function is called a Lévy random measure. More generally, if the Lévy measure $\nu(dpd\omega)$ satisfies $\int\int_{\mathbb{R}\times S}\min\{1,|p|\}\nu(dpd\omega) < \infty$ for each compact $S \subset \Omega$, the Lévy random measure $\mathcal{L}$ is well defined, even if the Poisson intensity ν^+ is infinite. A nonnegative Lévy random measure $\mathcal{L}$ satisfying the integration condition is called a completely random measure [24; 25] which was introduced to machine learning in [19; 33].

The Gamma Process [8; 36] $G \sim \Gamma\mathrm{P}(c,H)$ is a completely random measure defined on the product space $\mathbb{R}_+\times\Omega$, with concentration parameter c and a finite and continuous base measure H over a complete separable metric space Ω, such that $G(A_i) \sim \mathrm{Gamma}(H(A_i),1/c)$ are independent gamma random variables for disjoint partition $\{A_i\}_i$ of Ω. The Lévy measure of the Gamma Process can be expressed as $\nu(drd\omega) = r^{-1}e^{-cr}drH(d\omega)$. Since the Poisson intensity $\nu^+ = \nu(\mathbb{R}_+\times\Omega) = \infty$ and the value of $\int_{\mathbb{R}_+\times\Omega} r\nu(drd\omega)$ is finite, a draw from the Gamma Process consists of countably infinite atoms, which can be expressed as follows:

$$G=\sum_{k=1}^{\infty} r_k\delta_{\omega_k},\ (r_k,\omega_k) \overset{iid}{\sim} \pi(drd\omega),\ \pi(drd\omega)\nu^+ \equiv \nu(drd\omega). \quad (4)$$

A gamma process based model has an inherent shrinkage mechanism, as in the prior the number of atoms with weights greater than $\tau \in \mathbb{R}_+$ follows a Poisson distribution with parameter

$$H(\Omega)\int_{\tau}^{\infty} r^{-1}\exp(-cr)dr,$$

the value of which decreases as τ increases.

2.3 Poisson Factor Analysis

A large number of discrete latent variable models for count matrix factorization can be united under Poisson factor analysis (PFA) [42], which factorizes a count matrix $\mathbf{Y} \in \mathbb{Z}^{D\times V}$ under the Poisson likelihood as $\mathbf{Y} \sim \mathrm{Pois}(\mathbf{\Phi\Theta})$, where $\mathbf{\Phi} \in \mathbb{R}_+^{D\times K}$ is the factor loading matrix or dictionary, $\mathbf{\Theta} \in \mathbb{R}_+^{K\times V}$ is the factor score matrix. A wide variety of algorithms, although constructed with different motivations and for distinct problems, can all be viewed as PFA with different prior distributions imposed on $\mathbf{\Phi}$ and $\mathbf{\Theta}$. For example, non-negative matrix factorization [7; 26], with the objective to minimize the Kullback-Leibler divergence between $\mathbf{N}$ and its factorization $\mathbf{\Phi\Theta}$, is essentially PFA solved with maximum likelihood estimation. LDA [4] is equivalent to PFA, in terms of both block Gibbs sampling and variational inference, if Dirichlet distribution priors are imposed on both $\boldsymbol{\phi}_k \in \mathbb{R}_+^D$, the columns of $\mathbf{\Phi}$, and $\boldsymbol{\theta}_k \in \mathbb{R}_+^V$, the columns of $\mathbf{\Theta}$. The gamma-Poisson model [6; 34] is PFA with gamma priors on $\mathbf{\Phi}$ and $\mathbf{\Theta}$. A family of negative binomial (NB) processes, such as the beta-NB [5; 42] and gamma-NB processes [39; 41], impose different gamma priors on $\{\theta_{vk}\}$, the marginalization of which leads to differently parameterized NB distributions to explain the latent counts. Both the beta-NB and gamma-NB process PFAs are nonparametric Bayesian models that allow K to grow without limits [14].

2.4 Static and Dynamic Network Modeling

We mention select, most relevant approaches from a substantial literature on this topic. Among static latent variable based models, the Infinite Relational Model (IRM [21]) allows for multiple types of relations between entities in a network and an infinite number of clusters, but restricts these entities to belong to only one cluster. The Mixed Membership Stochastic Blockmodel (MMSB [2]) assumes that each node in the network can exhibit a mixture of communities. Though the MMSB has been applied successfully to discover complex network structure in a variety of applications, the computational complexity of the underlying inference mechanism is in the order of N^2, which limits its use to small networks. Computation complexity is also a problem with many other existing latent variable network models, such as the latent feature relational model [27] and its max margin version [43], and the infinite latent attribute model [29]. Regardless, such models are adept at identifying high-level clusters and perform particularly well for link prediction in small, dense, static networks. The Assortative Mixed-Membership Stochastic Blockmodel (a-MMSB [11]) bypasses the quadratic complexity of the MMSB by making certain assumptions about the network structure that might not be true in general, such as assuming the probability of linking distinct communities is small, sub-sampling the network, and employing stochastic variational inference that uses *only* a noisy estimate of the gradients. The hierarchical Dirichlet process relational model [22] allows mixed membership with an unbounded number of latent communities; however, it is built on the a-MMSB whose assumptions could be restrictive.

There has been quite a bit of research with non-Bayesian [12; 31] as well as Bayesian approaches [15; 17; 30; 37] to study dynamic networks. The Bayesian approaches differ among themselves due to the assumptions in structures of the latent space they make. For example, Euclidean space models [16; 30] place nodes in a low dimensional Euclidean space and the network evolution is then modeled as a regression problem of future latent node location. On the other hand, certain models [10; 15; 17] assume that the latent variables stochastically depend on the state at the previous time step.

Some other models use multi-memberships [9; 13; 23] wherein a node's membership to one group does not limit its membership to other groups. Compared to these approaches, D-NGPPF models the latent factors using Gamma distribution and the shape parameter of the distribution of the latent factor at time t is modeled by the latent factor at time $(t-1)$. The network entries are generated from a Truncated Poisson distribution whose rate is given by the underlying latent variables, some of which evolve over time and will be described in more details later.

3. GAMMA PROCESS POISSON FACTORIZATION FOR NETWORKS (N-GPPF)

Let there be a network of N users encoded as an $N \times N$ binary matrix $\boldsymbol{B}$. To model the latent factors in a network, a Gamma process $G \sim \Gamma\mathrm{P}(c, G_0)$ is maintained, a draw from which is expressed as:

$$G = \sum_{k=1}^{\infty} r_k \delta_{\boldsymbol{\phi}_k}, \tag{5}$$

where $\boldsymbol{\phi}_k \in \Omega$ is an atom drawn from an N-dimensional base distribution as $\boldsymbol{\phi}_k \sim \prod_{n=1}^{N} \text{Gamma}(e_0, 1/c_n)$ and $r_k = G(\boldsymbol{\phi}_k)$ is the associated weight. Also, $\gamma_0 = G_0(\Omega)$ is defined as the mass parameter corresponding to the base measure G_0. The $(n,m)^{\text{th}}$ entry in the matrix $\boldsymbol{B}$ is assumed to be derived from a latent count as:

$$b_{nm} = \mathbb{I}_{\{x_{nm} \geq 1\}},\ x_{nm} \sim \text{Pois}\left(\lambda_{nm}\right), \lambda_{nm} = \sum_{k} \lambda_{nmk}, \tag{6}$$

where $\lambda_{nmk} = r_k \phi_{nk} \phi_{mk}$. This is called as the Poisson-Bernoulli (PoBe) link in [1; 38]. The distribution of b_{nm} given λ_{nm} is named as the Poisson-Bernoulli distribution, with the PMF:

$$f(b_{nm}|\lambda_{nm}) = e^{-\lambda_{nm}(1-b_{nm})}(1 - e^{-\lambda_{nm}})^{b_{nm}}.$$

One may consider λ_{nmk} as the strength of mutual latent community membership between nodes n and m in the network for latent community k, and λ_{nm} as the interaction strength aggregating all possible community membership. For example, consider professional and recreational interactions between people n, m, and m' who all work together. Person n has about the same level of professional interactions with both persons m and m'. Yet if we add the condition that person n and m' go fishing together during the weekend, n and m' will have membership in the "fishing together" latent community while n and m will not. The strength of interactions between any two persons could be considered as the aggregation of a possibly infinite kinds of latent community memberships. Using Lemma 2.1, one may augment the above representation as:

$$x_{nm} = \sum_{k} x_{nmk},\ x_{nmk} \sim \text{Pois}\left(\lambda_{nmk}\right). \tag{7}$$

Thus each interaction pattern contributes a count and the total latent count aggregates the countably infinite interaction patters.

Unlike the usual approach that links the binary observations to latent Gaussian random variables with a logistic or probit function, the above approach links the binary observations to Poisson random variables. Thus, this approach transforms the problem of modeling binary network interaction into a count modeling problem, providing several potential advantages. First, it is more interpretable because r_k and $\boldsymbol{\phi}_k$ are non-negative and the aggregation of different interaction patterns increases the probability of establishing a link between two nodes. Second, the computational benefit is significant since the computational complexity is approximately linear in the number of non-zeros S in the observed binary adjacency matrix $\boldsymbol{B}$. This benefit is especially pertinent in many real-word datasets where S is significantly smaller than N^2. To complete the generative process, we put Gamma priors over c and c_n as:

$$c \sim \text{Gamma}(c_0, 1/d_0), c_n \sim \text{Gamma}(f_0, 1/g_0). \tag{8}$$

3.1 Gibbs Sampling for N-GPPF

Though N-GPPF supports countably infinite number of latent communities for network modeling, in practice it is impossible to instantiate all of them. Instead of marginalizing out the underlying stochastic process [3; 28] or using slice sampling [35] for non-parametric modeling, for simplicity, a finite approximation of the infinite model is considered by truncating the number of graph communities K. Such an approximation approaches the original infinite model as K approaches infinity. With such finite approximation, the generative process of N-GPPF is further summarized in Table 1.

Sampling of x_{nmk} : x_{nm}'s are sampled only corresponding to the following entries:

$$(n,m) : n = \{1, \cdots, (N-1)\}, m = \{(n+1), \cdots, N\}.$$

For the above entries the sampling goes as follows:

$$x_{nm}| \sim b_{nm}\text{Poisson}_{+}\left(\sum_{k=1}^{K} r_k \phi_{nk} \phi_{mk}\right), \tag{9}$$

where $\text{Poisson}_{+}(.)$ is the truncated Poisson distribution, the sampling from which is detailed in [38]. Since, one can augment $x_{nm} \sim \text{Pois}\left(\sum_{k=1}^{K} \lambda_{nmk}\right)$ as $x_{nm} = \sum_{k=1}^{K} x_{nmk}$, where $x_{nmk} \sim \text{Pois}\left(\lambda_{nmk}\right)$, equivalently, one obtains the following according to Lemma 2.1:

$$(x_{nmk})_{k=1}^{K}| \sim \text{mult}\left((r_k \phi_{nk} \phi_{mk})_{k=1}^{K} / \sum_{k=1}^{K} r_k \phi_{nk} \phi_{mk}; x_{nm}\right). \tag{10}$$

Sampling of ϕ_{nk}**,** r_k **,** c_n **and** c : Sampling of these parameters follow from Lemma 2.4 and are given as follows:

$$\phi_{nk}| \sim \text{Gamma}\left(e_0 + \sum_{m=1}^{(n-1)} x_{mnk} + \sum_{m=(n+1)}^{N} x_{nmk}, 1/\left(c_n + r_k \sum_{\substack{m=1 \\ m \neq n}}^{N} \phi_{mk}\right)\right), \tag{11}$$

$$r_k| \sim \text{Gamma}\left(\gamma_k + \sum_{n=1,n<m}^{(N-1),N} x_{nmk}, 1/\left(c + \sum_{n=1,n<m}^{(N-1),N} \phi_{nk}\phi_{mk}\right)\right), \tag{12}$$

$$c_n| \sim \text{Gamma}\left(f_0 + Ke_0, 1/\left(g_0 + \sum_{k=1}^{K} \phi_{nk}\right)\right), \tag{13}$$

$$c| \sim \text{Gamma}\left(c_0 + \sum_{k=1}^{K} \gamma_k, 1/\left(d_0 + \sum_{k=1}^{K} r_k\right)\right). \tag{14}$$

Sampling of γ_k : Using Lemma 2.1, one can show that:

$$x_{..k} \sim \text{Pois}(r_k s_k), \tag{15}$$

$$x_{..k} = \sum_{n=1}^{(N-1)} \sum_{m=(n+1)}^{N} x_{nmk}, s_k = \sum_{n=1}^{(N-1)} \sum_{m=(n+1)}^{N} \phi_{nk}\phi_{mk}.$$

$$b_{nm} = I_{\{x_{nm}\geq 1\}}, x_{nm} \sim \text{Pois}\left(\sum_k r_k \phi_{nk}\phi_{mk}\right),$$
$$r_k \sim \text{Gamma}(\gamma_k, 1/c), \phi_{nk} \sim \text{Gamma}(e_0, 1/c_n),$$
$$c_n \sim \text{Gamma}(f_0, 1/g_0),$$
$$\gamma_k \sim \text{Gamma}(a_0, 1/b_0), c \sim \text{Gamma}(c_0, 1/d_0).$$

Table 1: Generative Process of N-GPPF

Since $r_k \sim \text{Gam}(\gamma_k, 1/c)$ and one can augment $\ell_k \sim \text{CRT}(x_{..k}, \gamma_k)$, following Lemma 2.5 one can sample

$$\gamma_k| \sim \text{Gamma}\left(a_0 + \ell_k, 1/(b_0 - \log(1-p_k))\right), \quad (16)$$

where $p_k = s_k/(c+s_k)$.

3.2 Gibbs Sampling for N-GPPF with Missing Entries

Variables whose update get affected in presence of missing entries $\mathcal{M}$ are ϕ_{nk}'s and r_k's. Sampling of these parameters follow from Lemma 2.4 and are given as follows:

$$\phi_{nk}| \sim \text{Gamma}\left(e_0 + \sum_{\substack{m=1\\(m,n)\notin\mathcal{M}}}^{(n-1)} x_{mnk} \right. \quad (17)$$
$$\left. + \sum_{\substack{m=(n+1)\\(n,m)\notin\mathcal{M}}}^{N} x_{nmk}, 1/\left(c_n + r_k \sum_{\substack{m=1\\m\neq n;(n,m)\notin\mathcal{M}}}^{N} \phi_{mk}\right)\right),$$

$$r_k| \sim \text{Gamma}\left(\gamma_k + \sum_{\substack{n=1,n<m\\(n,m)\notin\mathcal{M}}}^{(N-1),N} x_{nmk}, 1/\left(c + \sum_{\substack{n=1,n<m\\(n,m)\notin\mathcal{M}}}^{(N-1),N} \phi_{nk}\phi_{mk}\right)\right). \quad (18)$$

4. DYNAMIC GAMMA PROCESS POISSON FACTORIZATION FOR NETWORKS (D-NGPPF)

Consider a tensor $\mathbb{B} \in \mathbb{Z}^{N\times N\times T}$, whose T columns are sequentially observed $N \times N$-dimensional binary matrices, and are indexed by $\{\boldsymbol{B}_t\}_{t=1}^T$. Further, consider a gamma process $G \sim \Gamma\text{P}(c, G_0)$, a draw from which is expressed as $G = \sum_{k=1}^{\infty} r_{0k}\delta_{\boldsymbol{\phi}_k}$, where $\boldsymbol{\phi}_k \in \Omega$ is an atom drawn from an N-dimensional base distribution $\boldsymbol{\phi}_k \sim \prod_{n=1}^N \text{Gamma}(e_0, 1/c_n)$ and $r_{0k} = G(\boldsymbol{\phi}_k)$ is the associated weight. We mark each atom $\boldsymbol{\phi}_k$ with an r_{1k} and generate a gamma Markov chain by letting:

$$r_{tk}|r_{(t-1)k} \sim \text{Gam}(r_{(t-1)k}, 1/c),\ t = \{1, \ldots, T\}.$$

The $(n,m)^{\text{th}}$ entry at time t is assumed to be generated as follows:

$$b_{tnm} = \mathbb{I}_{x_{tnm}\geq 1}, x_{tnm} \sim \text{Pois}\left(\sum_k r_{tk}\phi_{nk}\phi_{mk}\right).$$

Similar to NGPPF, to complete the generative process, we put Gamma priors over c and c_n as:

$$c \sim \text{Gamma}(c_0, 1/d_0), c_n \sim \text{Gamma}(f_0, 1/g_0). \quad (19)$$

In the formulation above of the dynamic network model, we assume that the weights of the latent factors evolve over time using a Gamma markov chain. At the t^{th} time instance, the proximity (or assignment) of the n^{th} entity of the network to the k^{th} latent factor is given by $r_{tk}\phi_{nk}$ and hence the evolution of r_{tk} alone can capture the changes in characteristics of the n^{th} network entity. In many applications, one may also evolve ϕ_{nk} over time, but we leave that as an interesting future work.

$$b_{tnm} = \mathbb{I}_{\{x_{tnm}\geq 1\}}, x_{tnm} \sim \text{Poisson}\left(\sum_{k=1}^K r_{tk}\phi_{nk}\phi_{mk}\right),$$
$$r_{tk} \sim \text{Gam}(r_{(t-1)k}, 1/c), \phi_{nk} \sim \text{Gam}(e_0, 1/c_n),$$
$$c_n \sim \text{Gam}(f_0, 1/g_0), r_{0k} \sim \text{Gam}(\gamma_k, 1/c),$$
$$\gamma_k \sim \text{Gam}(a_0, 1/b_0), c \sim \text{Gam}(c_0, 1/d_0).$$

Table 2: Generative Process of D-NGPPF

4.1 Gibbs Sampling for D-NGPPF

Similar to the implementation for N-GPPF, a finite approximation of the infinite model is considered by truncating the number of factors to K which approaches the original infinite model as $K \to \infty$. The generative process with the approximation is detailed in Table 2.

Sampling of x_{tnm} : x_{tnm}'s are sampled only corresponding to the following entries:

$$(t,n,m) : t = \{1, \cdots, T\}, n = \{1, \cdots, (N-1)\}, m = \{(n+1), \cdots, N\}.$$

For the above entries, the sampling goes as follows:

$$x_{tnm} \sim b_{tnm}\text{Pois}_+\left(\sum_{k=1}^K r_{tk}\phi_{nk}\phi_{mk}\right). \quad (20)$$

Since, one can augment $x_{tnm} \sim \text{Pois}\left(\sum_{k=1}^K r_{tk}\phi_{nk}\phi_{mk}\right)$ as $x_{tnm} = \sum_{k=1}^K x_{tnmk}$, where $x_{tnmk} \sim \text{Pois}\left(r_{tk}\phi_{nk}\phi_{mk}\right)$, equivalently, one obtains the following according to Lemma 2.1:

$$(x_{tnmk})_{k=1}^K| \sim \text{mult}\left((r_{tk}\phi_{nk}\phi_{mk})_{k=1}^K / \sum_{k=1}^K r_{tk}\phi_{nk}\phi_{mk}; x_{tnm}\right). \quad (21)$$

Sampling of r_{tk} : The data augmentation and marginalization techniques specific to the NB distribution [1; 39; 40] are utilized to sample r_{tk}. Despite the challenge present in inferring the gamma shape parameters, closed-form Gibbs sampling update equations can be derived for all the r_{tk}'s. For $t = T$, one can sample

$$r_{Tk}| \sim \text{Gam}\left(r_{(T-1)k} + x_{T..k}, 1/(c+s_k)\right), \quad (22)$$

$$x_{T..k} = \sum_{n=1}^{(N-1)}\sum_{m=(n+1)}^{N} x_{Tnmk},\ s_k = \sum_{n=1}^{(N-1)}\sum_{m=(n+1)}^{N}\phi_{nk}\phi_{mk}.$$

For $t = (T-1)$, one needs to augment $\ell_{Tk} \sim \text{CRT}(x_{T..k}, r_{(T-1)k})$, after which, using Lemma 2.5 one obtains the following:

$$r_{(T-1)k}| \sim \text{Gam}\left(r_{(T-2)k} + x_{(T-1)..k} + \ell_{Tk}, \right. \quad (23)$$
$$\left. 1/(c + s_k - \log(1-p_{Tk}))\right),$$
$$x_{(T-1)..k} = \sum_{n=1}^{(N-1)}\sum_{m=(n+1)}^{N} x_{(T-1)nmk},\ p_{Tk} = \frac{s_k}{(c+s_k)}.$$

For $1 \leq t \leq (T-2)$, the augmentation and sampling trick is very similar. One needs to augment $\ell_{(t+1)k} \sim \text{CRT}(x_{(t+1)..k} + \ell_{(t+2)k}, r_{tk})$ and then sample, according to Lemma 2.5

$$r_{tk}| \sim \text{Gam}\left(r_{(t-1)k} + x_{t..k} + \ell_{(t+1)k}, \right. \quad (24)$$
$$\left. 1/(c + s_k - \log(1-p_{(t+1)k}))\right),$$

$$x_{t..k} = \sum_{n=1}^{(N-1)} \sum_{m=(n+1)}^{N} x_{tnmk},\ p_{(t+1)k} = \frac{s_k - \log(1-p_{(t+2)k})}{(c + s_k - \log(1-p_{(t+2)k}))}.$$

For $t = 0$, augment $\ell_{1k} \sim \text{CRT}(x_{1..k} + \ell_{2k}, r_{0k})$. Then sample

$$r_{0k}| \sim \text{Gam}(\gamma_k + \ell_{1k}, 1/(c - \log(1-p_{1k}))), \tag{25}$$

$$x_{1..k} = \sum_{n=1}^{(N-1)} \sum_{m=(n+1)}^{N} x_{1nmk},\ p_{1k} = \frac{s_k - \log(1-p_{2k})}{(c + s_k - \log(1-p_{2k}))}.$$

Sampling of γ_k : Augment $\ell_{0k} \sim \text{CRT}(\ell_{1k}, \gamma_k)$. Then sample

$$\gamma_k| \sim \text{Gam}\left(a_0 + \ell_{0k}, 1/\left(b_0 - \log(1-p_{0k})\right)\right), \tag{26}$$
$$p_{0k} = \frac{\log(1-p_{1k})}{(\log(1-p_{1k}) - c)}.$$

Sampling of ϕ_{nk}**,** c_n **and** c : Sampling of these parameters follow from Lemma 2.4 and are given as follows:

$$\phi_{nk}| \sim \text{Gam}\left(d_0 + \sum_{t=1}^{T}\left(\sum_{m=1}^{(n-1)} x_{tmnk} + \sum_{m=(n+1)}^{N} x_{tnmk}\right), 1/\left(c_n + \sum_{t=1}^{T}\sum_{m=1,m\neq n}^{N} r_{tk}\phi_{mk}\right)\right). \tag{27}$$

$$c_n| \sim \text{Gam}\left(f_0 + Ke_0, 1/\left(g_0 + \sum_{k=1}^{K}\phi_{nk}\right)\right). \tag{28}$$

$$c| \sim \text{Gam}\left(\sum_{k=1}^{K}\left(\gamma_k + \sum_{t=0}^{(T-1)} r_{tk}\right) + c_0, 1/\left(\sum_{k=1}^{K}\sum_{t=0}^{T} r_{tk} + d_0\right)\right). \tag{29}$$

4.2 Gibbs Sampling for D-NGPPF with Missing Entries

Variables whose update get affected in presence of missing values are r_{tk}'s and ϕ_{nk}'s. Rest of the update equations are same as in D-NGPPF without any missing value. Below, the updates are enlisted where $\mathcal{M}_t$ denotes the set of missing entries in the network at the t^{th} time instance.

Sampling of r_{tk} : For $t = T$,

$$r_{Tk}| \sim \text{Gam}\left(r_{(T-1)k} + x_{T..k}, 1/(c + s_{Tk})\right), \tag{30}$$

$$x_{T..k} = \sum_{\substack{n=1,m=(n+1)\\(n,m)\notin\mathcal{M}_T}}^{(N-1),N} x_{Tnmk},\ s_{Tk} = \sum_{\substack{n=1,m=(n+1)\\(n,m)\notin\mathcal{M}_T}}^{(N-1),N} \phi_{nk}\phi_{mk}.$$

For $t = (T-1)$, augment $\ell_{Tk} \sim \text{CRT}(x_{T..k}, r_{(T-1)k})$ and then sample

$$r_{(T-1)k}| \sim \text{Gam}\left(r_{(T-2)k} + x_{(T-1)..k} + \ell_{Tk}, 1/(c + s_{(T-1)k} - \log(1-p_{Tk}))\right), \tag{31}$$

$$x_{(T-1)..k} = \sum_{\substack{n=1,m=(n+1)\\(n,m)\notin\mathcal{M}_{(T-1)}}}^{(N-1),N} x_{(T-1)nmk},$$

$$s_{(T-1)k} = \sum_{\substack{n=1,m=(n+1)\\(n,m)\notin\mathcal{M}_{(T-1)}}}^{(N-1),N} \phi_{nk}\phi_{mk},\ p_{Tk} = \frac{s_{Tk}}{(c + s_{Tk})}.$$

For $1 \leq t \leq (T-2)$, augment $\ell_{(t+1)k} \sim \text{CRT}(x_{(t+1)..k} + \ell_{(t+2)k}, r_{tk})$ and then sample

$$r_{tk}| \sim \text{Gam}\left(r_{(t-1)k} + x_{t..k} + \ell_{(t+1)k}, 1/(c + s_{tk} - \log(1-p_{(t+1)k}))\right), \tag{32}$$

$$x_{t..k} = \sum_{\substack{n=1,m=(n+1)\\(n,m)\notin\mathcal{M}_t}}^{(N-1),N} x_{tnmk},$$

$$s_{tk} = \sum_{\substack{n=1,m=(n+1)\\(n,m)\notin\mathcal{M}_t}}^{(N-1),N} \phi_{nk}\phi_{mk},\ p_{(t+1)k} = \frac{s_{(t+1)k} - \log(1-p_{(t+2)k})}{(c + s_{(t+1)k} - \log(1-p_{(t+2)k}))}.$$

Sampling of ϕ_{nk} :

$$\phi_{nk}| \sim \text{Gam}\left(d_0 + \sum_{t=1}^{T}\left(\sum_{\substack{m=1\\(m,n)\notin\mathcal{M}_t}}^{(n-1)} x_{tmnk} + \sum_{\substack{m=(n+1)\\(n,m)\notin\mathcal{M}_t}}^{N} x_{tnmk}\right), 1/\left(c_n + \sum_{t=1}^{T}\sum_{\substack{m=1,m\neq n\\(n,m)\notin\mathcal{M}_t}}^{N} r_{tk}\phi_{mk}\right)\right). \tag{33}$$

5. EXPERIMENTS

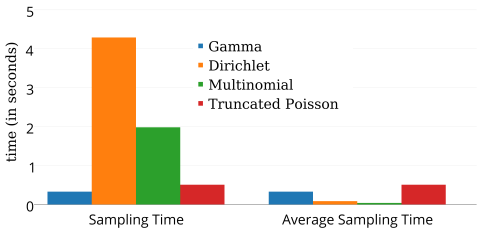


Figure 1: Time to generate a million of samples

In this section, experimental results are reported for a synthetic data and three real world datasets. Before presenting the results, the computation complexity of sampling from different samplers are illustrated. In Fig. 1, the computation times for generating one million samples from Gamma, Dirichlet (of dimension 50), multinomial (of dimension 50) and truncated Poisson distributions are shown. The experiments are carried out with the samplers available from GNU Scientific Library (GSL) on an Intel 2127U machine with 2 GB of RAM and 1.90 GHz of processor base frequency. To highlight the average complexity of sampling from Dirichlet and multinomial distributions, we further display another plot where the computation time is divided by 50 for these samplers only. One can see that to draw one million samples, our implementation of the sampler for truncated Poisson distribution takes the longest, though the difference from the Gamma sampler in GSL is not significant. For all the experiments with synthetic and real world data, the Gibbs sampler is run with 2000 burn-in and 2000 collection iterations, and $K = 50$ is maintained.

5.1 Synthetic Data

We generate a set of synthetic networks of size 60×60 with three different groups that evolve over six different time stamps. These datasets are displayed in column (a) in both Fig. 2 and 3. In practice, this may represent a group of users in a social network

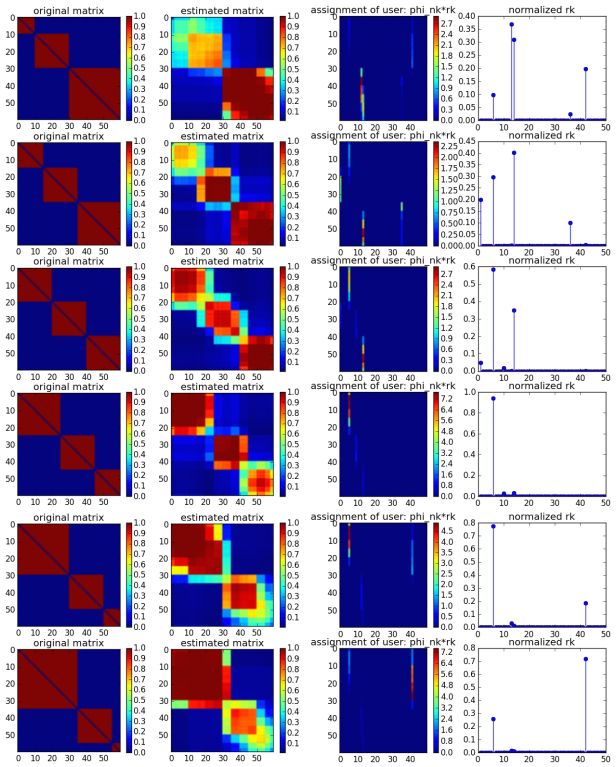


Figure 2: Results from D-NGPPF

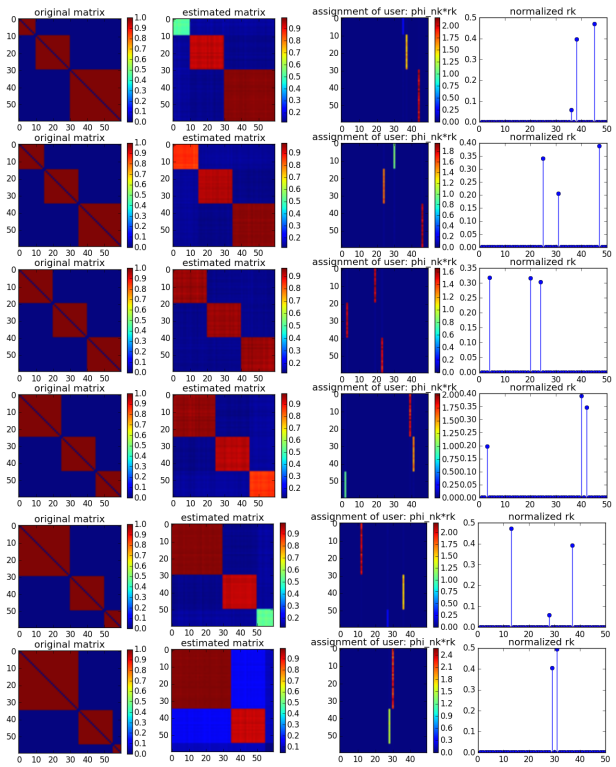


Figure 3: Results from N-GPPF

whose friend circles change over time. The links in these graphs are presented by brown and the non-links are illustrated by deep blue. The performance of D-NGPPF is displayed in columns (b), (c), and (d) of Fig. 2. Column (b) in Fig. 2 shows the groups discovered by D-NGPPF in the graph over different time-stamps. Note that the discovery of groups at any time instance is influenced by the groups present in other time instances. In column (c) of Fig. 2, the proximity of the users to the latent groups are displayed. The x-axis in each of these plots imply different latent groups and the y-axis represents the proximity of the n^{th} user to the k^{th} latent group at the t^{th} time instance, which is calculated as $r_{tk}\phi_{nk}$. In our experiments, 50 different latent groups are maintained ($K = 50$), but the model assigns the users to only a few of the latent groups, a desired outcome. This observation is also reinforced by the plots in column (c) of Fig. 2. These plots denote the normalized weights of the different latent groups ($r_{tk}/\sum_{k=1}^{K} r_{tk}$) at different time instances. In each time instance, only a few latent groups have positive weight. Expectedly, as displayed in columns (c) and (d) of Fig. 2, the latent factors that are dominant over different time instances vary smoothly with time. In Fig. 3, results are displayed for a baseline model that uses only N-GPPF for modeling the networks isolatedly at each different time slice. One can see that N-GPPF reconstructs the groups perfectly at each time instance as the groups are very clear-cut. However, different sets of latent groups dominate in modeling the networks at different time slices, as revealed in plots of columns (c) and (d) of Fig. 3. Unlike this toy example, most real world networks are sparse and groups are less distinct at any given time. The performance of a static network model is expected to be poorer in such settings, as it cannot link the solutions across time. This is explained more clearly alongside the results reported in the next subsection.

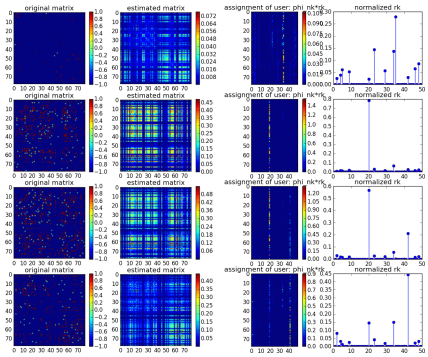
5.2 Real World Data

NIPS Authorship Network Data: The NIPS co-authorship network connects two people if they appear on the same publication in the NIPS conference in a given year. Network spans $T = 17$ years (1987 to 2003). Following [13], only a subset of 110 authors, who are most connected over all the time periods, are considered. For evaluating the predictive performance, 25% of the links and equal number of non-links are held out from each of the 17 time instances. The rest of the data is used as training. DSBM [37], N-GPPF and MMSB [2] are considered as the baselines in the prediction problem. For both N-GPPF and MMSB, the networks for the different time instances are modeled isolatedly. We use the implementation from the authors of DBSM for the corresponding set of experiments. Since both DBSM and MMSB are parametric methods, we use $K = 10$ for all the experiments which, as the literature reports, is found to produce best results for these set of models with these datasets. The objective is to infer the labels of the held out links and non-links. The quality of prediction is measured by AUC and the results are displayed in Table 3.

Dataset	D-NGPPF	DSBM	N-GPPF	MMSB
NIPS	$\mathbf{0.797} \pm 0.016$	0.780 ± 0.010	0.766 ± 0.012	0.740 ± 0.009
DBLP	$\mathbf{0.836} \pm 0.013$	0.810 ± 0.013	0.756 ± 0.020	0.749 ± 0.014
Infocom	$\mathbf{0.907} \pm 0.008$	0.901 ± 0.006	0.856 ± 0.011	0.831 ± 0.006

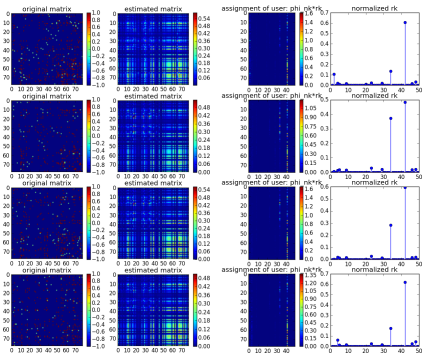
Table 3: AUC Results on Real World Data

DBLP Data: The DBLP co-authorship network is obtained from 21 Computer Science conferences from 2000 to 2009 ($T = 10$) [32]. Only top 209 people are considered in this datasets by taking 7-core of the aggregated network for the entire time. For each different time slice, 10% of the links and equal number of non-links are held out. The results are displayed in Table 3.



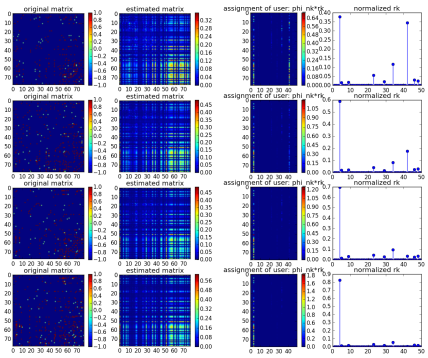
(a) (b) (c) (d)

Figure 4: Infocom: Hour 5^{th} to 8^{th}



(a) (b) (c) (d)

Figure 5: Infocom: Hour 9^{th} to 12^{th}



(a) (b) (c) (d)

Figure 6: Infocom: Hour 13^{th} to 16^{th}

Infocom Data: The Infocom dataset represents the physical proximity interactions between 78 students at the 2006 Infocom conference, recorded by wireless detector remotes given to each attendee [20]. As in [13], the recordings are agglomerated into one hour-long time slices and only the reciprocated sightings are maintained. Also, the slices with less than 80 links (corresponding to late night and early morning hours), are removed, resulting in only 50 time slices. For each different time slice, 10% of the links and equal number of non-links are held out. The results are displayed in Table 3. One can see that D-NGPPF outperforms DSBM, a strong baseline for dynamic network modeling, and two other baselines for static network modeling.

To illustrate the effectiveness of D-NGPPF further in real world data, some findings are presented in Fig. 4 to Fig. 6 for the Infocom dataset. One can see the smooth transition of the dominant factors over time. Fig. 4, 5 and 6 present the results corresponding to the datasets at times $T = 4$ to $T = 8$, $T = 9$ to $T = 12$ and $T = 13$ to $T = 16$ respectively. Column (a) in each of these figures present the original network with some of the entities held out (indicated by green). Column (b) represents the cluster structures discovered by D-NGPPF, while column (c) and (d) signify the assignment of the users in the latent space and the weights of the latent factors respectively. Note that, for each time slice, very few links are available (indicated by deep brown) and hence the performance of N-GPPF for prediction of held-out links is poorer, as illustrated in Table 3.

6. CONCLUSION AND FUTURE WORK

This paper introduces the Dynamic Gamma Process Poisson Factorization framework for analyzing a network that evolves over time. Efficient inference technique has been developed for modeling the temporal evolution of the latent components of the network using a gamma Markov chain. Superior empirical performance on both synthetic and real world datasets makes the approach a promising candidate for modeling other count time-series data; for example, time-evolving rating matrices and tensors that appear quite frequently in text mining, recommendation systems and analysis of electronic health records.

References

[1] A. Acharya, J. Ghosh, and M. Zhou. Nonparametric Bayesian Factor Analysis for Dynamic Count Matrices. In *Proc. of AISTATS*, pages 1–9. 2015.

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

[3] D. Blackwell and J. MacQueen. Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, 1973.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *JMLR*, 3:993–1022, 2003.

[5] T. Broderick, L. Mackey, J. Paisley, and M. I. Jordan. Combinatorial clustering and the beta negative binomial process. *arXiv:1111.1802v5*, 2013.

[6] J. Canny. Gap: a factor model for discrete data. In *SIGIR*, 2004.

[7] A. T. Cemgil. Bayesian inference for nonnegative matrix factorisation models. *Intell. Neuroscience*, 2009.

[8] T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *Ann. Statist.*, 1973.

[9] J. R. Foulds, C. Dubois, A. U. Asuncion, C. T. Butts, and P. Smyth. A dynamic relational infinite feature model for longitudinal social networks. In *Proc. of AISTATS*, volume 15, pages 287–295, 2011.

[10] W. Fu, L. Song, and E. P. Xing. Dynamic mixed membership blockmodel for evolving networks. In *Proc. of ICML*, pages 329–336, 2009.

[11] P. Gopalan, D. M. Mimno, S. Gerrish, M. J. Freedman, and D. M. Blei. Scalable inference of overlapping communities. In *Proc. of NIPS*, pages 2258–2266, 2012.

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

[13] C. Heaukulani and Z. Ghahramani. Dynamic probabilistic models for latent feature propagation in social networks. In *Proc. of ICML*, pages 275–283, 2013.

[14] N. L. Hjort. Nonparametric Bayes estimators based on beta processes in models for life history data. *Ann. Statist.*, 1990.

[15] Q. Ho, L. Song, and E. Xing. Evolving cluster mixed-membership blockmodel for time-varying networks. In *Proc. of AISTATS*. 2011.

[16] P. Hoff, A. Raftery, and M. Handcock. Latent space approaches to social network analysis. *JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION*, 97:1090–1098, 2001.

[17] K. Ishiguro, T. Iwata, N. Ueda, and J. B. Tenenbaum. Dynamic infinite relational model for time-varying relational data analysis. In *Proc. of NIPS*, pages 919–927. 2010.

[18] N. L. Johnson, A. W. Kemp, and S. Kotz. *Univariate Discrete Distributions*. John Wiley & Sons, 2005.

[19] M. I. Jordan. Hierarchical models, nested models, and completely random measures. In *Frontiers of Statistical Decision Making and Bayesian Analysis: In Honor of James O. Berger*, pages 207–217. Springer, 2010.

[20] J.Scott, R.Gass, J.Crowcroft, P.Hui, C.Diot, and A.Chaintreau. CRAWDAD data set dartmouth/campus (v. 2009-05-29). Downloaded from http://crawdad.org/dartmouth/campus/, May 2009.

[21] C. Kemp, J. Tenenbaum, T. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *Proc. of AAAI*, pages 381–388, 2006.

[22] D. I. Kim, P. Gopalan, D. M. Blei, and E. B. Sudderth. Efficient online inference for bayesian nonparametric relational models. In *Proc. of NIPS*, pages 962–970, 2013.

[23] M. Kim and J. Leskovec. Nonparametric multi-group membership model for dynamic networks. In *Proc. of NIPS*, pages 1385–1393. 2013.

[24] J. Kingman. *Poisson Processes*. Oxford University Press, 1993.

[25] J. F. C. Kingman. Completely random measures. *Pacific Journal of Mathematics*, 21(1):59–78, 1967.

[26] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2001.

[27] K. T. Miller, T. L. Griffiths, and M. I. Jordan. Nonparametric latent feature models for link prediction. In *Proc. of NIPS*, pages 1276–1284, 2009.

[28] R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of computational and graphical statistics*, 2000.

[29] K. Palla, Z. Ghahramani, and D. A. Knowles. An infinite latent attribute model for network data. In *Proc. of ICML*, pages 1607–1614, 2012.

[30] P. Sarkar and A. Moore. Dynamic social network analysis using latent space models. *SIGKDD Explor. Newsl.*, 7:31–40, dec 2005.

[31] T. Snijders, G. Bunt, and C. Steglich. Introduction to stochastic actor-based models for network dynamics. *Social Networks*, 32(1):44–60, 2010.

[32] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: Extraction and mining of academic social networks. In *Proc. of KDD*, pages 990–998, 2008.

[33] R. Thibaux and M. Jordan. Hierarchical beta processes and the indian buffet process. In *Proc. of AISTATS*, 2007.

[34] M. K. Titsias. The infinite gamma-Poisson feature model. In *Proc. of NIPS*, 2008.

[35] S. G. Walker. Sampling the Dirichlet mixture model with slices. *Communications in Statistics Simulation and Computation*, 2007.

[36] R. L. Wolpert, M. A. Clyde, and C. Tu. Stochastic expansions using continuous dictionaries: Lévy Adaptive Regression Kernels. *Annals of Statistics*, 2011.

[37] K. Xu and A. Hero. Dynamic stochastic blockmodels for time-evolving social networks. *J. Sel. Topics Signal Processing*, 8(4):552–562, 2014.

[38] M. Zhou. Infinite edge partition models for overlapping community detection and link prediction. In *Proc. of AISTATS*, pages 1135–1143. 2015.

[39] M. Zhou and L. Carin. Augment-and-conquer negative binomial processes. In *Proc. of NIPS*, 2012.

[40] M. Zhou and L. Carin. Augment-and-conquer negative binomial processes. In *NIPS*, 2012.

[41] M. Zhou and L. Carin. Negative binomial process count and mixture modeling. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2015.

[42] M. Zhou, L. Hannah, D. Dunson, and L. Carin. Beta-negative binomial process and Poisson factor analysis. In *Proc. of AISTATS*, pages 1462–1471, 2012.

[43] J. Zhu. Max-margin nonparametric latent feature models for link prediction. In *Proc. of ICML*, 2012.