Deep Poisson gamma dynamical systems

Dandan Guo, Bo Chen∗, Hao Zhang
National Laboratory of Radar Signal Processing
Collaborative Innovation Center of Information Sensing and Understanding
Xidian University, Xi’an, China
gdd_xidian@126.com, bchen@mail.xidian.edu.cn, zhanghao_xidian@163.com

Mingyuan Zhou
McCombs School of Business
The University of Texas at Austin
Austin, TX 78712, USA
mingyuan.zhou@mccombs.utexas.edu

Abstract

We develop deep Poisson-gamma dynamical systems (DPGDS) to model sequentially observed multivariate count data, improving previously proposed models by not only mining deep hierarchical latent structure from the data, but also capturing both first-order and long-range temporal dependencies. Using sophisticated but simple-to-implement data augmentation techniques, we derived closed-form Gibbs sampling update equations by first backward and upward propagating auxiliary latent counts, and then forward and downward sampling latent variables. Moreover, we develop stochastic gradient MCMC inference that is scalable to very long multivariate count time series. Experiments on both synthetic and a variety of real-world data demonstrate that the proposed model not only has excellent predictive performance, but also provides highly interpretable multilayer latent structure to represent hierarchical and temporal information propagation.

1 Introduction

The need to model time-varying count vectors \(x_1, ..., x_T\) appears in a wide variety of settings, such as text analysis, international relation study, social interaction understanding, and natural language processing [1–9]. To model these count data, it is important to not only consider the sparsity of high-dimensional data and robustness to over-dispersed temporal patterns, but also capture complex dependencies both within and across time steps. In order to move beyond linear dynamical systems (LDS) [10] and its nonlinear generalization [11] that often make the Gaussian assumption [12], the gamma process dynamic Poisson factor analysis (GP-DPFA) [5] factorizes the observed time-varying count vectors under the Poisson likelihood as \(x_t \sim \text{Poisson}(\Phi \theta_t)\), and transmit temporal information smoothly by evolving the factor scores with a gamma Markov chain as \(\theta_t \sim \text{Gamma}(\theta_{t-1}, \beta)\), which has highly desired strong non-linearity. To further capture cross-factor temporal dependence, a transition matrix \(\Pi\) is further used in Poisson–gamma dynamical system (PGDS) [7] as \(\theta_t \sim \text{Gamma}(\Pi \theta_{t-1}, \beta)\). However, these shallow models may still have shortcomings in capturing long-range temporal dependencies [8]. For example, if given \(\theta_t\), then \(\theta_{t+1}\) no longer depends on \(\theta_{t-k}\) for all \(k \geq 1\).

Deep probabilistic models are widely used to capture the relationships between latent variables across multiple stochastic layers [4, 8, 13, 16]. For example, deep dynamic Poisson factor analysis (DDPFA)

∗Corresponding author

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.
In this paper, we present deep Poisson gamma dynamical systems (DPGDS), a deep probabilistic dynamical model that takes the advantage of the hierarchical structure to efficiently incorporate both between-layer and temporal dependencies, while providing rich interpretation. Moving beyond DTSBN using binary hidden units, we build a deep dynamic directed network with gamma distributed nonnegative real hidden units, inferring a multilayer contextual representation of multivariate time-varying count vectors. Consequently, DPGDS can handle highly overdispersed counts, capturing the correlations between the visible/hidden features across layers and over times using the gamma belief network [15]. Combing the deep and temporal structures shown in Fig. 1(a) DPGDS breaks the assumption that given \( \theta_l \), \( \theta_{l+1} \) no longer depends on \( \theta_{l-k} \) for \( k \geq 1 \), suggesting that it may better capture long-range temporal dependencies. As a result, the model can allow more specific information, which are also more likely to exhibit fast temporal changing, to transmit through lower layers, while allowing more general information, which are also more likely to slowly evolve over time, to transmit through higher layers. For example, as shown in Fig. 1(b) that is learned from GDELT2003 with DPGDS, when analyzing these international events, the factors at lower layers are more specific to discover the relationships between the different countries, whereas those at higher layers are more general to reflect the conflicts between the different areas consisting of several related countries, or the ones occurring simultaneously, and the latent representation \( \theta_l \) at a lower layer varies more intensely than that at a higher layer.

Distinct from DDPPFA [8] that adopts a two-stage inference, the latent variables of DPGDS can be jointly trained with both a Backward-Upward–Forward-Downward (BUFD) Gibbs sampler and a sophisticated stochastic gradient MCMC (SGMCMC) algorithm that is scalable to very long multivariate time series [17][21]. Furthermore, the factors learned at each layer can refine the understanding and analysis of sequentially observed multivariate count data, which, to the best of our knowledge, may be very challenging for existing methods. Finally, based on a diverse range of real-world data sets, we show that DPGDS exhibits excellent predictive performance, inferring interpretable latent structure with well captured long-range temporal dependencies.

2 Deep Poisson gamma dynamic systems

Shown in Fig. 1(a) is the graphical representation of a three-hidden-layer DPGDS. Let us denote \( \theta \sim \text{Gam}(a, c) \) as a gamma random variable with mean \( a/c \) and variance \( a/c^2 \). Given a set of \( V \)-dimensional sequentially observed multivariate count vectors \( x_1, ..., x_T \), represented as a \( V \times T \) matrix \( X \), the generative process of a \( L \)-hidden-layer DPGDS, from top to bottom, is expressed as

\[
\begin{align*}
\theta^{(L)}_t &\sim \text{Gam} \left( \tau_0 \Pi^{(L)} \theta^{(L)}_{t-1} , \tau_0 \right), \ldots, \\
\theta^{(2)}_t &\sim \text{Gam} \left( \tau_0 \left( \Phi^{(2)} \theta^{(2)}_{t-1} + \Pi^{(1)} \theta^{(1)}_{t-1} \right) , \tau_0 \right), \\
\theta^{(1)}_t &\sim \text{Gam} \left( \tau_0 \left( \Phi^{(1)} \theta^{(1)}_{t-1} \right) , \tau_0 \right), \quad x^{(1)}_t \sim \text{Pois} \left( \delta^{(1)}_t \Phi^{(1)} \theta^{(1)}_{t-1} \right),
\end{align*}
\]

where \( \Phi^{(l)} \in \mathbb{R}^{K_{l-1} \times K_l} \) is the factor loading matrix at layer \( l \), \( \theta^{(l)}_t \in \mathbb{R}^{K_l} \) the hidden units of layer \( l \) at time \( t \), and \( \Pi^{(l)} \in \mathbb{R}^{K_{l-1} \times K_l} \) a transition matrix of layer \( l \) that captures cross-factor temporal dependencies. We denote \( \delta^{(1)}_t \in \mathbb{R}_+ \) as a scaling factor, reflecting the scale of the counts at time \( t \); one may also set \( \delta^{(1)}_t = \sigma^{(1)} \) for \( t = 1, ..., T \). We denote \( \tau_0 \in \mathbb{R}_+ \) as a scaling hyperparameter that controls the temporal variation of the hidden units. The multilayer time-varying hidden units \( \theta^{(l)}_t \) are well suited for downstream analysis, as will be shown below.

DPGDS factorizes the count observation \( x^{(1)}_t \) into the product of \( \delta^{(1)}_t \), \( \Phi^{(1)} \), and \( \theta^{(1)}_t \) under the Poisson likelihood. It further factorizes the shape parameters of the gamma distributed \( \theta^{(l)}_t \) of layer \( l \) at time \( t \) into the sum of \( \Phi^{(l+1)} \theta^{(l+1)}_{t-1} \), capturing the dependence between different layers, and \( \Pi^{(l)} \theta^{(l)}_{t-1} \), capturing the temporal dependence at the same layer. At the top layer, \( \theta^{(L)}_t \) is only
dependent on $\mathbf{\Pi}^{(L)}\mathbf{\theta}^{(L)}_{t-1}$, and at $t = 1$, $\mathbf{\theta}_1^{(1)} \sim \text{Gam} \left( \tau_0 \Phi^{(1)} \mathbf{\theta}_1^{(L+1)}, \tau_0 \right)$ for $l = 1, \ldots, L - 1$ and $\mathbf{\theta}_1^{(L)} \sim \text{Gam} \left( \tau_0 \nu^{(L)}_{1k}, \tau_0 \right)$. To complete the hierarchical model, we introduce $K_l$ factor weights $\nu^{(l)} = (\nu^{(l)}_1, \ldots, \nu^{(l)}_{K_l})$ in layer $l$ to model the strength of each factor, and for $l = 1, \ldots, L$, we let

$$
\pi_k^{(l)} \sim \text{Dir}(\nu^{(l)}_1, \ldots, \nu^{(l)}_{K_l}),
$$

Note that $\pi_k^{(l)}$ is the $k$th column of $\mathbf{\Pi}^{(l)}$ and $\pi_{k_2}^{(l)}$ can be interpreted as the probability of transiting from topic $k_2$ of the previous time to topic $k_1$ of the current time at layer $l$.

Finally, we place Dirichlet priors on the factor loadings and draw other parameters from a noninformative gamma prior: $\mathbf{\phi}_k^{(l)} = (\phi_{1k}^{(l)}, \ldots, \phi_{K_l-1k}^{(l)}) \sim \text{Dir}(\nu^{(l)}_1, \ldots, \eta^{(l)})$, and $\delta^{(1)}_k, \xi^{(1)}_k, \beta^{(1)}_k \sim \text{Gam}(\epsilon_0, \epsilon_0)$.

Note that imposing Dirichlet distributions on the columns of $\mathbf{\Pi}^{(l)}$ and $\mathbf{\Phi}^{(l)}$ not only makes the latent representation more identifiable and interpretable, but also facilitates inference, as will be shown in the next section. Clearly when $L = 1$, DPGDS reduces to PGDS [7]. In real-world applications, a binary observation can be linked to a latent count using the Bernoulli-Poisson link as $b = 1(n \geq 1), n \sim \text{Pois}(\lambda)$ [22]. Nonnegative-real-valued matrix can also be linked to a latent count matrix via a Poisson randomized gamma distribution as $x \sim \text{Gam}(n, c), n \sim \text{Pois}(\lambda)$ [23].

 Hierarchical structure: To interpret the hierarchical structure of $\mathbf{\Pi}$, we notice that $\mathbb{E} \left[ \mathbf{x}_t^{(l)} | \mathbf{\theta}_t^{(l)}, \mathbf{\Phi}^{(p)} \right] = \prod_{p=1}^{\Pi} \Phi^{(p)}$ if the temporal structure is ignored. Thus it is straightforward to interpret $\phi_k^{(l)}$ by projecting them to the bottom data layer as $\prod_{l=1}^{L} \Phi^{(l)}$, which are often quite specific at the bottom layer and become increasingly more general when moving upwards, as will be shown below in Fig. 3(a).

 Long-range temporal dependencies: Using the law of total expectations on $\mathbf{\Pi}$, for a three-hidden-layer DPGDS shown in Fig. 1(a), we have

$$
\mathbb{E}[x_t^{(l)} | \mathbf{\theta}_{t-1}^{(l)}, \mathbf{\theta}_{t-2}^{(l)}, \mathbf{\theta}_{t-3}^{(3)}] = \Phi^{(1)} \mathbf{\Pi}^{(1)} \mathbf{\theta}_{t-1}^{(1)} + \Phi^{(1)} \Phi^{(2)} \mathbf{\Pi}^{(2)} \mathbf{\theta}_{t-2}^{(2)} + \Phi^{(1)} \Phi^{(2)} \Phi^{(3)} \mathbf{\Pi}^{(3)} \mathbf{\Pi}^{(3)} \mathbf{\theta}_{t-3}^{(3)},
$$

which suggests that $\mathbf{\Pi}^{(l)}$ play the role of transiting the latent representation across time and, different from most existing dynamic models, DPGDS can capture and transmit long-range temporal information (often general and change slowly over time) through its higher hidden layers.

### 3 Scalable MCMC inference

In this paper, in each iteration, across layers and times, we first exploit a variety of data augmentation techniques for count data to “backward” and “upward” propagate auxiliary latent counts, with which
In order to marginalize out \( \theta \), latent counts that are smaller or equal as \( x \), we exploit three useful properties for count data, denoted as \( \text{P1, P2, and P3} \) [7][24], respectively, as presented in the Appendix. Let us denote \( x \sim \text{NB}(r, p) \) as the negative binomial distribution with probability mass function \( P(x = k) = \frac{\Gamma(k+r)}{k! \Gamma(r)} p^k (1 - p)^r \), where \( k \in \{0, 1, \ldots\} \). First, we can augment each count \( x^{(1)}_{1t} \) in (1) into the summation of \( K_1 \) latent counts that are smaller or equal as \( x^{(1)} = \sum_{k=1}^{K_1} A^{(1)}_{kT} \), \( A^{(1)}_{kT} \sim \text{Pois}(\phi^{(1)}_{ik} \theta_{k1,T}) \), with \( A^{(1)}_{kT} = \sum_{v=1}^{V} A^{(1)}_{vk} \). Since \( \sum_{v=1}^{V} \phi^{(1)}_{vk} = 1 \) by construction, we also have \( A^{(1)}_{kT} \sim \text{Pois}(\delta^{(1)}_{k1} \theta^{(1)}_{k1}) \), as shown in Fig. 2(b). We start with \( \theta^{(1)}_{k1} \) at the last time point \( T \), as none of the other time-step factors depend on it in their priors. Via \( \text{P2} \), as shown in Fig. 2(b), we can marginalize out \( \theta^{(1)}_{k1} \) to obtain

\[
A^{(1)}_{kT} \sim \text{NB} \left[ \tau_0 \left( \sum_{k_2=1}^{K_2} \phi^{(2)}_{kk2} \phi^{(2)}_{k2T} + \sum_{k_1=1}^{K_1} \pi^{(1)}_{kk1} \theta^{(1)}_{k1,T-1} \right), g(\zeta^{(1)}_T) \right],
\]

where \( \zeta^{(1)}_T = \ln(1 + \frac{\delta^{(1)}_{k1}}{\tau_0}) \) and \( g(\zeta) = 1 - \exp(-\zeta) \).

In order to marginalize out \( \theta^{(1)}_{k1-1} \), as shown in Fig. 2(c), we introduce an auxiliary variable following the Chinese restaurant table (CRT) distribution [24] as

\[
x^{(2)}_{kT} \sim \text{CRT} \left[ A^{(1)}_{kT}, \tau_0 \left( \sum_{k_2=1}^{K_2} \phi^{(2)}_{kk2} \phi^{(2)}_{k2T} + \sum_{k_1=1}^{K_1} \pi^{(1)}_{kk1} \theta^{(1)}_{k1,T-1} \right) \right].
\]

As shown in Fig. 2(d) we re-express the joint distribution over \( A^{(1)}_{kT} \) and \( x^{(2)}_{kT} \) according to \( \text{P3} \) as

\[
A^{(1)}_{kT} \sim \text{SumLog}(x^{(2)}_{kT}, g(\zeta^{(1)}_T)), \quad x^{(2)}_{kT} \sim \text{Pois} \left[ r^{(1)}_T \tau_0 \left( \sum_{k_2=1}^{K_2} \phi^{(2)}_{kk2} \phi^{(2)}_{k2T} + \sum_{k_1=1}^{K_1} \pi^{(1)}_{kk1} \theta^{(1)}_{k1,T-1} \right) \right].
\]
We describe the BUFD Gibbs sampling algorithm for DPGDS in Algorithm 1 and provide more detail. Although the proposed BUFD Gibbs sampling algorithm for DPGDS has closed-form update equations, it requires processing all time-varying vectors at each iteration and hence has limited scalability. To allow for scalable inference, we apply the topic-layer-adaptive stochastic gradient MCMC inference.

where the sum-logarithmic (SumLog) distribution is defined as in Zhou and Carin [24]. Via the deep structure, the latent counts \( x_{kT}^{(l)} \) at layer two is divided into two parts: one from time \( T - 1 \) at layer one, while the other from time \( T \) at layer two. Furthermore, \( \zeta_t^{(1)} \) is the scaling factor at layer two, which is propagated from the one at layer one \( \delta_t^{(1)} \). Repeating the process all the way back to \( t = 1 \), and from \( l = 1 \) up to \( l = L \), we are able to marginalize out all gamma latent variables \( \{ \Theta_t^{(t:L)} \} \) and provide closed-form conditional posteriors for all of them.

### 3.2 Backward-upward–forward-downward Gibbs sampling

**Sampling auxiliary counts:** This step is about the “backward” and “upward” pass. Let us denote \( Z_{kT}^{(l)} = \sum_{k=1}^{K_l} Z_{kT}^{(l)} \), \( Z_{k,T+1}^{(l)} = 0 \), and \( x_{kT}^{(1)} = x_{kT}^{(1)} \). Working backward for \( t = T, \ldots, 2 \) and upward for \( l = 1, \ldots, L \), we draw

\[
\begin{align*}
(A_{kT}^{(l)}, \ldots, A_{k,K(l)}^{(l)}) & \sim \text{Multi} \left( x_{kT}^{(l)}; \frac{\phi_{k1}^{(l)} \theta_{k1}^{(1)}}{\sum_{k=1}^{K_l} \phi_{k1}^{(l)} \theta_{k1}^{(1)}}, \ldots, \frac{\phi_{kK_l}^{(l)} \theta_{kK_l}^{(1)}}{\sum_{k=1}^{K_l} \phi_{kK_l}^{(l)} \theta_{kK_l}^{(1)}} \right), \\
x_{kT}^{(l+1)} & \sim \text{CRT} \left[ A_{kT}^{(l)} + Z_{k,T+1}^{(l)} \tau_0 \left( \frac{\sum_{k=1}^{K_l+1} \phi_{kkT}^{(l+1)} \theta_{kk1}^{(l+1)}}{\sum_{k=1}^{K_l} \phi_{kk1}^{(l)} \theta_{kk1}^{(1)}}, ..., \frac{\sum_{k=1}^{K_l} \phi_{kK_l}^{(l)} \theta_{kK_l}^{(1)}}{\sum_{k=1}^{K_l} \phi_{kK_l}^{(l)} \theta_{kK_l}^{(1)}} \right) \right].
\end{align*}
\]

Note that via the deep structure, the latent counts \( x_{kT}^{(l+1)} \) will be influenced by the effects from both of time \( t - 1 \) at layer \( l \) and time \( t \) at layer \( l + 1 \). With \( p_1 := \sum_{k=1}^{K_l} \phi_{kk1}^{(l)} \theta_{kk1}^{(1)} \) and \( p_2 := \sum_{k=1}^{K_l} \phi_{kk1}^{(l+1)} \theta_{kk1}^{(1)} \), we can sample the latent counts at layer \( l \) and \( l + 1 \) by

\[
(x_{kT}^{(l+1)}, x_{kT}^{(l+1)}; p_1 + p_2) \sim \text{Multi} \left( x_{kT}^{(l+1)}, p_1/(p_1 + p_2), p_2/(p_1 + p_2) \right),
\]

and then draw

\[
(Z_{k1T}, \ldots, Z_{k,K(l)}^{(l)}) \sim \text{Multi} \left( x_{kT}^{(l+1)}; \frac{\pi_{k1}^{(l+1)} \theta_{k1}^{(1)}}{\sum_{k=1}^{K_l+1} \pi_{k1}^{(l)} \theta_{k1}^{(1)}}, ..., \frac{\pi_{kK_l}^{(l)} \theta_{kK_l}^{(1)}}{\sum_{k=1}^{K_l} \pi_{kK_l}^{(l)} \theta_{kK_l}^{(1)}} \right).
\]

**Sampling hidden units \( \theta_{kT}^{(l)} \) and calculating \( \zeta_t^{(l)} \):** Given the augmented latent count variables, working forward for \( t = 1, \ldots, T \) and downward for \( l = L, \ldots, 1 \), we can sample

\[
\theta_{kT}^{(l)} \sim \text{Gamma} \left[ A_{kT}^{(l)} + Z_{k,T+1}^{(l)} + \tau_0 \left( \frac{\sum_{k=1}^{K_l} \phi_{kk1}^{(l+1)} \theta_{kk1}^{(1)}}{\sum_{k=1}^{K_l} \pi_{k1}^{(l)} \theta_{k1}^{(1)}}, ..., \frac{\sum_{k=1}^{K_l} \phi_{kK_l}^{(l+1)} \theta_{kK_l}^{(1)}}{\sum_{k=1}^{K_l} \pi_{kK_l}^{(l)} \theta_{kK_l}^{(1)}}, \tau_0 (1 + \zeta_t^{(l)} + \zeta_t^{(l-1)}) \right) \right],
\]

where \( \zeta_t^{(l)} = \delta_t^{(l)} / \tau_0 \) and \( \zeta_t^{(l)} = \ln \left( 1 + \zeta_t^{(l-1)} + \zeta_t^{(l-1)} \right) \). Note if \( \delta_t^{(l)} = \delta_t^{(l)} \) for \( t = 1, \ldots, T \), then we may let \( \zeta_t^{(l)} = -W_{t-1}(-\exp(-1 - \zeta_t^{(l-1)})) - 1 - \zeta_t^{(l-1)} \), where the function \( W_{t-1} \) is the lower real part of the Lambert \( W \) function [25]. From [22], we can find that the conditional posterior of \( \theta_t^{(l)} \) is parameterized by not only both \( \Phi_t^{(l-1)} \theta_t^{(l-1)} \) and \( \Pi_t^{(l-1)} \theta_t^{(l-1)} \), which represent the information from layer \( l - 1 \) (downward) and time \( t - 1 \) (forward), respectively, but also both \( A_t^{(l-1)} \) and \( Z_t^{(l-1)} \), which record the message from layer \( l - 1 \) (upward) in [8] and time \( t + 1 \) (backward) in [11], respectively. We describe the BUFD Gibbs sampling algorithm for DPGDS in Algorithm 1 and provide more details in the Appendix.

### 3.3 Stochastic gradient MCMC inference

Although the proposed BUFD Gibbs sampling algorithm for DPGDS has closed-form update equations, it requires processing all time-varying vectors at each iteration and hence has limited scalability. To allow for scalable inference, we apply the topic-layer-adaptive stochastic gradient
Riemannian (TLASGR) MCMC algorithm described in Cong et al. [27] and Zhang et al. [29], which can be used to sample simplex-constrained global parameters [28] in a mini-batch based manner. It improves its sampling efficiency via the use of the Fisher information matrix (FIM) [29], with adaptive step-sizes for the latent factors and transition matrices of different layers. More specifically, for $\pi_k^{(l)}$, column $k$ of the transition matrix $\Pi^{(l)}$ of layer $l$, its sampling can be efficiently realized as

$$
\left( \pi_k^{(l)} \right)_{n+1} = \left( \pi_k^{(l)} \right)_n + \frac{\varepsilon_n}{M_k^{(l)}} \left( \rho z_{k,.}^{(l)} + \eta_k^{(l)} - \left( \rho z_{k,.}^{(l)} + \eta_k^{(l)} \right) \left( \pi_k^{(l)} \right)_n \right) + \mathcal{N} \left( 0, \frac{2\varepsilon_n}{M_k^{(l)}} \left[ \text{diag} \left( \pi_k^{(l)} \right)_n - \left( \pi_k^{(l)} \right)_n \left( \pi_k^{(l)} \right)^T \right] \right),
$$

where $M_k^{(l)}$ is calculated using the estimated FIM, both $z_{k,.}^{(l)}$ and $\eta_k^{(l)}$ come from the augmented latent counts $Z^{(l)}$, $\cdot$ denotes a simplex constraint, and $\eta_k^{(l)}$ denotes the prior of $\pi_k^{(l)}$. The update of $\Phi^{(l)}$ is the same with Cong et al. [27], and all the other global parameters are sampled using SGNHT [20]. We provide the details of the SGMC MCMC for DPGDS in Algorithm 2 in the Appendix.

4 Experiments

In this section, we present experimental results on a synthetic dataset and five real-world datasets. For a fair comparison, we consider PGDS [7], GP-DPFA [5], DTSBN [4], and GPDM [11] that can be considered as a dynamic generalization of the Gaussian process latent variable model of Lawrence [30], using the code provided by the authors. Note that as shown Schein et al. [7] and Gan et al. [4], PGDS and DTSBN are state-of-the-art count time series modeling algorithms that outperform a wide variety of previously proposed ones, such as LDS [12] and DRFM [31]. The hyperparameter settings of PGDS, GP-DPFA, GPDM, TSBN, and DTSBN are the same as their original settings [4,5,7,11]. For DPGDS, we set $\tau_0 = 1, \gamma_0 = 100, \eta_0 = 0.1$ and $\epsilon_0 = 0.1$. We use $K^{(1)}, K^{(2)}, K^{(3)} = [200, 100, 50]$ for both DPGDS and DTSBN and $K^{(3)} = 200$ for PGDS, GP-DPFA, GPDM, and TSBN. For PGDS, GP-DPFA, GPDM, and DPGDS, we run 2000 Gibbs sampling as burn-in and collect 3000 samples for evaluation. We also use SGMC MCMC to infer DPGDS, with 5000 collection samples after 5000 burn-in steps, and use 100000 SGMC MCMC iterations for both TSBN and DTSBN to evaluate their performance.

4.1 Synthetic dataset

Following the literature [1,4], we consider sequences of different lengths, including $T = 10, 50, 100, 200, 300, 400, 500$ and 600, and generate 50 synthetic bouncing ball videos for training, and 30 ones for testing. Each video frame is a binary-valued image with size $30 \times 30$, describing the location of three balls within the image. Both TSBN and DTSBN model it with the Bernoulli likelihood, while both PGDS and DPGDS use the Bernoulli-Poisson link [22].

As shown in Fig. 3(b), the average prediction errors of all algorithms decrease as the training sequence length increases. A higher-order TSBN, TSBN-4, performs much better than the first-order TSBN.
Another advantage of DPGDS is that its inferred deep latent structure often has meaningful interpre-
tion, as shown in Tab. 1. We also use the Mean Precision (MP) and Mean Recall (MR) over all the years appearing in the training set to evaluate different models. As another criterion, the Predictive Precision (PP) shows the predictive precision for the final year, for which all the observations are held out. Similar as previous methods, for each corpus, the entire data of the last year is held out, and for the documents in the previous years we randomly partition the words of each document into 80%/20% in each trial, and we conduct five random trials to report the sample mean and standard deviation. Note that to apply GPDM, we have used Anscombe transform.

### 4.2 Real-world datasets

Besides the binary-valued synthetic bouncing ball dataset, we quantitatively and qualitatively evaluate all algorithms on the following real-world datasets used in Schein et al. [7]. The State-of-the-Union (SOTU) dataset consists of the text of the annual SOTU speech transcripts from 1790 to 2014. The Global Database of Events, Language, and Tone (GDELT) and Integrated Crisis Early Warning System (ICEWS) are both datasets for international relations extracted from news corpora. Note that ICEWS consists of undirected pairs, while GDELT consists of directed pairs of countries. The NIPS corpus contains the text of every NIPS conference paper from 1987 to 2003. The DBLP corpus is a database of computer science research papers. Each of these datasets is summarized as a $V \times T$ count matrix, as shown in Tab. 1. Unless specified otherwise, we choose the top 1000 most frequently used terms to form the vocabulary, which means we set $V = 1000$ for all real-data experiments.

#### 4.2.1 Quantitative comparison

For a fair and comprehensive comparison, we calculate the precision and recall at top-$M$ [4, 5, 31, 32], which are calculated by the fraction of the top-$M$ words that match the true ranking of the words and appear in the top-$M$ ranking, respectively, with $M = 50$. We also use the Mean Precision (MP) and Mean Recall (MR) over all the years appearing in the training set to evaluate different models. As another criterion, the Predictive Precision (PP) shows the predictive precision for the final year, for which all the observations are held out. Similar as previous methods, for each corpus, the entire data of the last year is held out, and for the documents in the previous years we randomly partition the words of each document into 80%/20% in each trial, and we conduct five random trials to report the sample mean and standard deviation. Note that to apply GPDM, we have used Anscombe transform.
to preprocess the count data to mitigate the mismatch between the data and model assumption. The results on all five datasets are summarized in Tab. [1] which clearly show that the proposed DPGDS has achieved the best performance on most of the evaluation criteria, and again a deep model often improves its performance by increasing its number of layers. To add more empirical study on scalability, we have also tested the efficiency of our model on a GDELT data (from 2001 to 2005, temporal granularity of 24 hrs, with a total of 1825 time points), which is not too large so that we can still run DPGDS-Gibbs and GPDM. As shown in Fig. 4, we present how various algorithms progress over time, evaluated with MP. It takes about 1000s for DTSBN and DPGDS-SGMC to converge, 3.5 hrs for DPGDS-Gibbs, 5 hrs for GPDM. Clearly, our DPGDS-SGMC is scalable and clearly outperforms both DTSBN and GPDM. We also present in Appendix C the results of DPGDS-SGMC on a very long time series, on which it becomes too expensive to run a batch learning algorithm.

4.2.2 Exploratory data analysis

Compared to previously proposed dynamic systems, the proposed DPGDS, whose inferred latent structure is simple to visualize, provides much richer interpretation. More specifically, we may not only exhibit the content of each factor (topic), but also explore both the hierarchical relationships between them at different layers, and the temporal relationships between them at the same layer. Based on the results inferred on ICEWS 2001-2003 via a three hidden layer DPGDS, with the size of 200-100-50, we show in Fig. 5 how some example topics are hierarchically and temporally related to each other, and how their corresponding latent representations evolve over time.

In Fig. 5(a) we select two large-weighted topics at the top hidden layer and move down the network to include any lower-layer topics that are connected to them with sufficiently large weights. For each topic, we list all its terms whose values are larger than 1% of the largest element of the topic. It is interesting to note that topic 2 at layer three is connected to three topics at layer two, which are characterized mainly by the interactions of Israel (ISR)-Palestinian Territory (PSE), Iraq (IRQ)-USA-Iran (IRN), and North Korea (PRK)-South Korea (KOR)-USA-China (CHN)-Japan (JPN), respectively. The activation strength of one of these three interactions, known to be dominant in general during 2001-2003, can be contributed not only by a large activation of topic 2 at layer three, but also by a large activation of some other topic of the same layer (layer two) at the previous time. For example, topic 41 of layer two on “ISR-PSE, IND-PAK, RUS-UKR, GEO-RUS, AFG-PAK, SYR-USA, MNE-SRB” could be associated with the activation of topic 46 of layer two on “IND-PAK, RUS-TUR, ISR-PSE, BLR-RUS” at the previous time; and topic 99 of layer two on “PRK-KOR, JPN-USA, CHN-USA, CHN-KOR, CHN-JPN, USA-RUS” could be associated with the activation of topic 63 of layer two on “IRN-USA, CHN-USA, AUS-CHN, CHN-KOR” at the previous time.

Another instructive observation is that topic 140 of layer one on “IRQ-USA, IRQ-GBR, IRN-IRQ, IRQ-KWT, AUS-IRQ” is related not only in hierarchy to topic 34 of the higher layer on “IRQ-USA, IRQ-GBR, GBR-USA, IRQ-KWT, IRN-IRQ, SYR-USA,” but also in time to topic 166 of the same layer on “ESP-USA, ESP-GBR, FRA-GBR, POR-USA,” which are interactions between the member states of the North Atlantic Treaty Organization (NATO). Based on the transitions from topic 13 on “PRK-KOR” to both topic 140 on “IRQ-USA” and 77 on “ISR-PSE,” we can find that the ongoing Iraq war and Israeli–Palestinian relations regain attention after the six-party talks [7].

To get an insight of the benefits attributed to the deep structure, how the latent representations of several representative topics evolve over days are shown in Fig. 5(b). It is clear that relative to these temporal factor trajectories at the bottom layer, which are specific for the bilateral interactions between two countries, these from higher layers vary more smoothly, whose corresponding high-layer topics capture the multilateral interactions between multiple closely related countries. Similar phenomena have also been demonstrated in Fig. 1(b) on GDELT2003. Moreover, we find that a spike of the temporal trajectory of topic 166 (NATO) appears right before a one of topic 140 (Iraq war), matching the above description in Fig. 5(a). Also, topic 14 of layer three and its descendants, including topic 23 of layer two and topic 48 at layer one are mainly about a breakthrough between RUS and Azerbaijan (AZE), coinciding with Putin’s visit in January 2001. Additional example results for the topics and their hierarchical and temporal relationships, inferred by DPGDS on different datasets, are provided in the Appendix.

Figure 4: MP as a function of time for GDELT.
We propose deep Poisson gamma dynamical systems (DPGDS) that take the advantage of a probabilistic deep hierarchical structure to efficiently capture both across-layer and temporal dependencies. The inferred latent structure provides rich interpretation for both hierarchical and temporal information propagation. For Bayesian inference, we develop both a Backward-Upward–Forward-Downward Gibbs sampler and a stochastic gradient MCMC (SGMCMC) that is scalable to long multivariate count/binary time series. Experimental results on a variety of datasets show that DPGDS not only exhibits excellent predictive performance, but also provides highly interpretable latent structure.
Acknowledgements

D. Guo, B. Chen, and H. Zhang acknowledge the support of the Program for Young Thousand Talent by Chinese Central Government, the 111 Project (No. B18039), NSFC (61771361), NSFC for Distinguished Young Scholars (61525105) and the Innovation Fund of Xidian University. M. Zhou acknowledges the support of Award IIS-1812699 from the U.S. National Science Foundation.

References

[1] I. Sutskever and G. E. Hinton, “Learning multilevel distributed representations for high-dimensional sequences,” in AISTATS, 2007.
[2] C. Wang, D. Blei, and D. Heckerman, “Continuous time dynamic topic models,” in UAI, 2008, pp. 579–586.
[3] M. Hermans and B. Schrauwen, “Training and analysing deep recurrent neural networks,” in NIPS, 2013, pp. 190–198.
[4] Z. Gan, C. Li, R. Henao, D. E. Carlson, and L. Carin, “Deep temporal sigmoid belief networks for sequence modeling,” in NIPS, 2015, pp. 2467–2475.
[5] A. Acharya, J. Ghosh, and M. Zhou, “Nonparametric Bayesian factor analysis for dynamic count matrices,” in AISTATS, 2015.
[6] L. Charlin, R. Ranganath, J. Mcinerney, and D. M. Blei, “Dynamic Poisson factorization,” in ACM, 2015, pp. 155–162.
[7] A. Schein, M. Zhou, and H. Wallach, “Poisson–gamma dynamical systems,” in NIPS, 2016.
[8] C. Y. Gong and W. Huang, “Deep dynamic Poisson factorization model,” in NIPS, 2017.
[9] H. R. Rabiee, H. R. Rabiee, H. R. Rabiee, H. R. Rabiee, H. R. Rabiee, and H. R. Rabiee, “Recurrent Poisson factorization for temporal recommendation,” in KDD, 2017, pp. 847–855.
[10] Z. Ghahramani and S. T. Roweis, “Learning nonlinear dynamical systems using an EM algorithm,” in NIPS, 1999, pp. 431–437.
[11] J. M. Wang, A. Hertzmann, and D. M. Blei, “Gaussian process dynamical models,” in NIPS, 2006.
[12] R. E. Kalman, “Mathematical description of linear dynamical systems,” Journal of The Society for Industrial and Applied Mathematics, Series A: Control, vol. 1, no. 2, pp. 152–192, 1963.
[13] R. M. Neal, “Connectionist learning of belief networks,” Artificial Intelligence, vol. 56, no. 1, pp. 71–113, 1992.
[14] R. Ranganath, L. Tang, L. Charlin, and D. M. Blei, “Deep exponential families,” in AISTATS, 2014, pp. 762–771.
[15] M. Zhou, Y. Cong, and B. Chen, “The Poisson gamma belief network,” in NIPS, 2015, pp. 3043–3051.
[16] R. Henao, Z. Gan, J. T. Lu, and L. Carin, “Deep Poisson factor modeling,” in NIPS, 2015, pp. 2800–2808.
[17] Y. A. Ma, T. Chen, and E. B. Fox, “A complete recipe for stochastic gradient MCMC,” in NIPS, 2015, pp. 2917–2925.
[18] M. Welling and Y. W. Teh, “Bayesian learning via stochastic gradient Langevin dynamics,” in ICML, 2011, pp. 681–688.
[19] S. Patterson and Y. W. Teh, “Stochastic gradient Riemannian Langevin dynamics on the probability simplex,” in NIPS, 2013, pp. 3102–3110.
[20] N. Ding, Y. Fang, R. Babbush, C. Chen, R. D. Skeel, and H. Neven, “Bayesian sampling using stochastic gradient thermostats,” in NIPS, 2014, pp. 3203–3211.
[21] C. Li, C. Chen, D. Carlson, and L. Carin, “Preconditioned stochastic gradient Langevin dynamics for deep neural networks,” in AAAI, 2016, pp. 1788–1794.
[22] M. Zhou, “Infinite edge partition models for overlapping community detection and link prediction,” in AISTATS, 2015, pp. 1135–1143.
[23] M. Zhou, Y. Cong, and B. Chen, “Augmentable gamma belief networks,” *Journal of Machine Learning Research*, vol. 17, no. 163, pp. 1–44, 2016.

[24] M. Zhou and L. Carin, “Negative binomial process count and mixture modeling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 307–320, 2015.

[25] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, “On the LambertW function,” *Advances in Computational Mathematics*, vol. 5, no. 1, pp. 329–359, 1996.

[26] H. Zhang, B. Chen, D. Guo, and M. Zhou, “WHAI: Weibull hybrid autoencoding inference for deep topic modeling,” in *ICLR*, 2018.

[27] Y. Cong, B. Chen, H. Liu, and M. Zhou, “Deep latent Dirichlet allocation with topic-layer-adaptive stochastic gradient Riemannian MCMC,” in *ICML*, 2017.

[28] Y. Cong, B. Chen, and M. Zhou, “Fast simulation of hyperplane-truncated multivariate normal distributions,” *Bayesian Anal.*, vol. 12, no. 4, pp. 1017–1037, 2017.

[29] M. A. Girolami and B. Calderhead, “Riemann manifold Langevin and Hamiltonian Monte Carlo methods,” *Journal of The Royal Statistical Society Series B-statistical Methodology*, vol. 73, no. 2, pp. 123–214, 2011.

[30] N. D. Lawrence, “Probabilistic non-linear principal component analysis with gaussian process latent variable models,” *Journal of Machine Learning Research*, vol. 6, pp. 1783–1816, 2005.

[31] S. Han, L. Du, E. Salazar, and L. Carin, “Dynamic rank factor model for text streams,” in *NIPS*, 2014, pp. 2663–2671.

[32] P. Gopalan, F. J. R. Ruiz, R. Ranganath, and D. M. Blei, “Bayesian nonparametric Poisson factorization for recommendation systems,” in *AISTATS*, 2014.

[33] F. J. Anscombe, “The transformation of Poisson, binomial and negative-binomial data,” *Biometrika*, vol. 35, no. 3/4, pp. 246–254, 1948.

[34] D. B. Dunson and A. H. Herring, “Bayesian latent variable models for mixed discrete outcomes,” *Biostatistics*, vol. 6, no. 1, pp. 11–25, 2005.

[35] M. Zhou, L. Hannah, D. B. Dunson, and L. Carin, “Beta-negative binomial process and Poisson factor analysis,” in *AISTATS*, 2012, pp. 1462–1471.

[36] M. Zhou, “Nonparametric Bayesian negative binomial factor analysis,” *Bayesian Anal.*, vol. 13, no. 4, pp. 1061–1089, 2018.