Graph Gamma Process Linear Dynamical Systems

Rahi Kalantari
Department of Electrical & Computer Engineering
The University of Texas at Austin
rkalantari@utexas.edu

Mingyuan Zhou
McCombs School of Business
The University of Texas at Austin
mzhou@utexas.edu

Abstract

We introduce graph gamma process (GGP) linear dynamical systems to model real-valued multivariate time series. GGP generates S latent states that are shared by K different communities, each of which is characterized by its own pattern of activation probabilities imposed on a $S \times S$ directed sparse graph, and allow both S and K to grow without bound. For temporal pattern discovery, the latent representation under the model is used to decompose the time series into a parsimonious set of multivariate sub-sequences generated by formed communities. In each sub-sequence, different data dimensions often share similar temporal patterns but may exhibit distinct magnitudes, and hence allowing the superposition of all sub-sequences to exhibit diverse behaviors at different data dimensions. On both synthetic and real-world time series, the proposed nonparametric Bayesian dynamic models, which are initialized at random, consistently exhibit good predictive performance in comparison to a variety of baseline models, revealing interpretable latent state transition patterns and decomposing the time series into distinctly behaved sub-sequences.

1 INTRODUCTION

Linear dynamical systems (LDSs) have been widely used to model real-valued time series (Kalman, 1960; West and Harrison, 1997; Ghahramani and Roweis, 1999; Ljung, 1999), with diverse applications such as financial time series analysis (Carvalho and Lopes, 2007) and movement trajectory modeling (Gao et al., 2016; Zhang et al., 2017). They have become standard tools in optimal filtering, smoothing, and control (Imani and Braga-Neto, 2018; Hardt et al., 2018; Koyama, 2018). An LDS consists of two main blocks, including an observation model, which assumes that the observations are translated from their latent states via a linear system with added Gaussian noise, and a transition block, which is represented by a Markov chain that linearly transforms a latent state from time $t-1$ to time t with added Gaussian noise. The transition block plays an important role in capturing the underlying dynamics of the data. An LDS, which has limited representation power due to its linear assumption, allows one to examine the temporal trajectory of each latent dimension to understand the role played by the corresponding latent factor. While it is often considered to be simple to interpret, its interpretability often quickly deteriorates as its latent state dimension increases.

To enhance the representation power of LDSs, in particular, to model non-linear behaviors of the time series and improve their interpretability, one may consider switching LDSs (Fox et al., 2009; Linderman et al., 2017; Nassar et al., 2018), which learn how to divide the time series into separate temporal segments and fit them by switching between different LDSs. Important parameters include the number of different LDSs, their latent state dimensions, and the transition mechanism from one LDS to another. While nonparametric Bayesian techniques have been applied to switching LDSs to learn the number of LDSs that is needed, the latent state dimensions often stay as tuning parameters to be set (Fox et al., 2009; Nassar et al., 2018). Moreover, switching LDSs do not allow different LDSs to share latent states, making it difficult to capture smooth transitions between different temporal patterns, and false positives/negatives and delays in detecting the switching points will also compromise their performance. In addition, existing optimal smoothing and filtering techniques developed for LDSs, such as Kalman filtering (Kalman, 1960), require non-trivial modifications before being able to be applied to switching LDSs

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

(Murphy, 1998).

Moving beyond switching LDSs where different LDSs neither share their latent states nor overlap in time, we propose the graph gamma process (GGP) LDS that encourages forming multiple LDSs that can share their latent states and co-occur at the same time. GGP-LDS uses a flexible combination of multiple LDSs to fit the observation at any given time point, allowing smooth transitions between different dynamical patterns across time. A notable feature of GGP-LDS is that existing optimal filtering and smoothing techniques developed for a canonical LDS can be readily applied to GGP-LDS without any modification. Therefore, GGP-LDS can serve as a plug-in replacement of the LDS in an existing system.

The introduced nonparametric Bayesian construction in GGP-LDS will support S latent states that are shared by K different types of LDSs, each of which is characterized by its own pattern of activation probabilities imposed on a $S \times S$ sparse state-transition matrix, and allow both S and K to grow without bound. This unique construction is realized by modeling the sparsity structure of the $S \times S$ state-transition matrix as the adjacency matrix of a directed random graph, which is resulted from the logical OR operation over K latent binary adjacency matrices, each of which is drawn according to the interaction strengths between the states (nodes) of a type of LDS (node community). While a latent state is associated with all communities, the association strengths can clearly differ. Note that the sparsity pattern of the state-transition matrix is determined by the logical OR of these community-specific binary adjacency matrices. Therefore, to facilitate interpretation and visualization, one can hard assign a state to a community whose binary adjacency matrix best explains how this state is being influenced by the states of the previous time, or to a community that best explains how this state is influencing the states of the next time.

GGP-LDS allows approximating complex nonlinear dynamics by activating a certain combination of communities to model a particular type of linear dynamics at any given time, and using smooth transitions between overlapping communities to model smooth transitions between distinct linear dynamics. The characteristics of each community can be visualized by reconstructing the observations using the inferred latent representation and a community-specific reweighted latent state-transition matrix, where the weights are determined by the activation strength of that community relative to the combined activation strength of all communities.

It is noteworthy to mention that while the LDS (Kalman, 1963) has been chosen as the transition and observation model of GGP-LDS, the proposed GGP can potentially be applied to many other nonlinear systems that have a latent state transition module (Johnson et al., 2016).

2 NONPARAMETRIC BAYESIAN MODELING

For LDSs, let us denote $\boldsymbol{y}_t \in \mathbb{R}^V$ and $\boldsymbol{x}_t \in \mathbb{R}^S$ as the observed data and latent state vectors, respectively, at time $t \in \{1, \ldots, T\}$, $\mathbf{D} = (\boldsymbol{d}_1, \ldots, \boldsymbol{d}_S) \in \mathbb{R}^{V \times S}$ as the observation factor loading matrix, and both $\boldsymbol{\Phi} \in \mathbb{R}^{V \times V}$ and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_S)$ as precision (inverse covariance) matrices. Inspired by Kalantari et al. (2018), we first modify the usual LDS hierarchical model by utilizing a spike-and-slab construction (Mitchell and Beauchamp, 1988; Ishwaran and Rao, 2005; Zhou et al., 2009), which imposes binary mask $\mathbf{Z} \in \{0, 1\}^{S \times S}$ element-wise on the real-valued latent state transition matrix $\mathbf{W} \in \mathbb{R}^{S \times S}$ as

$$\boldsymbol{y}_t \sim \mathcal{N}(\mathbf{D}\boldsymbol{x}_t, \boldsymbol{\Phi}^{-1}),\ \boldsymbol{x}_t \sim \mathcal{N}\left((\mathbf{W} \odot \mathbf{Z})\, \boldsymbol{x}_{t-1}, \boldsymbol{\Lambda}^{-1}\right).$$

The $K \times K$ latent state-transition matrix $\mathbf{W} \odot \mathbf{Z}$, in particular, the sparsity structure of $\mathbf{Z}$, plays an important role in determining the model's dynamical behaviors. First, the nonzero locations in $\mathbf{Z}$ determine the temporal dependencies between the latent states (Kalantari et al., 2018). For example, if z_{ij}, the (i, j)th element of $\mathbf{Z}$, is zero, then at time t, x_{ti} will be independent of $x_{(t-1)j}$, and x_{tj} will not influence $x_{(t+1)i}$. Thus in what follows, we consider that there is a directed link (edge) from states (nodes) j to i if $z_{ij} = 1$.

Second, viewing $\mathbf{Z}$ as the adjacency matrix of a directed random graph and the LDS states as the graph nodes, we may introduce inductive bias to encourage its nodes to be formed into overlapping communities, reflected by overlapping dense blocks along the diagonal of the adjacency matrix after appropriately rearranging the orders of the nodes. We may then view each community as an LDS, which forms its own state-transition matrix, using a submatrix of $\mathbf{W} \odot \mathbf{Z}$, to model the transitions between the corresponding subset of states. This construction allows approximating complex nonlinear dynamics by activating different communities at different levels to model a particular type of linear dynamics at any given time, and using smooth transitions between overlapping communities to model the smooth transitions between distinct linear dynamics.

To induce the structure of overlapping communities into $\mathbf{Z}$, the adjacency matrix of a directed random graph, and allow both the number of communities and number of nodes (dimension of $\mathbf{Z}$) to grow without bound, we propose the graph gamma process (GGP). A

draw from the GGP consists of countably infinite latent communities, each of which is associated with a positive weight indicating the overall activation strength of the community. These communities all share the same set of countably infinite nodes (states) but place different weights on how strongly a node is associated with a community. We describe the detail in what follows.

2.1 Graph Gamma Process

Denote $Z(i,:)$ and $Z(:,i)$ as row i and column i of $\mathbf{Z}$, respectively. Since

$$\mathbb{E}[\boldsymbol{x}_t \,|\, \boldsymbol{x}_{t-1}, \mathbf{W}, \mathbf{Z}] = (\mathbf{W} \odot \mathbf{Z})\boldsymbol{x}_{t-1}$$

we have:

$$\begin{aligned}\mathbb{E}[x_{ti} \,|\, \boldsymbol{x}_{t-1}, \mathbf{W}, \mathbf{Z}] &= (W(i,:) \odot Z(i,:))\boldsymbol{x}_{t-1}, \text{ and} \\ \mathbb{E}[\boldsymbol{x}_{t+1} \,|\, \boldsymbol{x}_t, \mathbf{W}, \mathbf{Z}] &= (W(:,i) \odot Z(:,i))x_{ti} \\ &\quad + \sum_{j\neq i}^{\infty}(W(:,j) \odot Z(:,j))x_{tj} \qquad (1)\end{aligned}$$

which means x_{ti} will be dependent on $\boldsymbol{x}_{t-1}$ if $Z(i,:)$ contains non-zero elements, and it will influence $\boldsymbol{x}_{t+1}$ if $Z(:,i)$ contains non-zero elements. To construct a nonparametric Bayesian model that removes the need to tune the hidden state dimension, our first goal is to allow $\mathbf{Z}$ to have an unbounded number of rows and columns, which means that the model can support countably infinite state-specific factors $\boldsymbol{d}_i$, with which the mean of $\boldsymbol{y}_t$ given $\boldsymbol{x}_t$ is factorized as $\mathbb{E}[\boldsymbol{y}_t \,|\, \boldsymbol{x}_t, \mathbf{D}] = \mathbf{D}\boldsymbol{x}_t = \sum_{t=1}^{\infty} \boldsymbol{d}_i x_{ti}$.

2.1.1 Forming Unbounded Number of States

To achieve this goal, with $c_\rho > 0$ and $G_{0,\rho}$ defined as a finite and continuous base measure over a complete and separable metric space Ω, we first introduce a gamma process $G_\rho \sim \Gamma\mathrm{P}(c_\rho, G_{0,\rho})$ on the product space $\mathbb{R}^+ \times \Omega$, where $\mathbb{R}^+ := \{x : x > 0\}$, such that for each subset $A \subset \Omega$, we have $G_\rho(A) \sim \mathrm{Gamma}(G_{0,\rho}(A), 1/c_\rho)$. The Lévy measure of this gamma process can be expressed as $\nu(d\rho d\boldsymbol{d}) = \rho^{-1}e^{-c_\rho\rho}d\rho G_{0,\rho}(d\boldsymbol{d})$. A draw from this gamma process can be expressed as $G_\rho = \sum_{i=1}^{\infty}\rho_i d\boldsymbol{d}_i$, consisting of countably infinite atoms (factors) $\boldsymbol{d}_i$ with weights ρ_i. We view $\boldsymbol{d}_i$ as the factor loading vector for latent state i, and will make ρ_i determine the number of nonzero elements in $Z(i,:)$ and, consequently, how strongly x_{ti}, the activation of state i at time t, is influenced by $\boldsymbol{x}_{t-1}$ of the previous time. As the number of ρ_i that are larger than an arbitrarily small constant ϵ follows a Poisson distribution with a finite mean as $\gamma_{0,\rho}\int_\epsilon^\infty \rho^{-1}e^{-c_\rho\rho}d\rho$, where $\gamma_{0,\rho} := G_{0,\rho}(\Omega)$ is the mass parameter, this can be used to express the idea that only a finite number of elements in $\{x_{ti}\}_{i=1,\infty}$ at time t will be dependent on $\boldsymbol{x}_{t-1}$ of the previous time.

We further mark each ρ_i with a degenerate gamma random variable, changing the Lévy measure of the gamma process to that of a marked gamma process (Kingman, 1993) as $\nu(d\rho\, d\boldsymbol{d}\, d\tau) = \rho^{-1}e^{-c_\rho\rho}d\rho G_{0,\rho}(d\boldsymbol{d})\gamma_{0,\tau}\tau^{-1}e^{-c_\tau\tau}d\tau$; we express a draw from this marked gamma process as $G_{\rho,\tau} = \sum_{i=1}^{\infty}(\rho_i,\tau_i)\delta_{\boldsymbol{d}_i}$. We will make τ_i determine the random number of nonzero elements in $Z(:,i)$ and, consequently, how strongly x_{ti}, the factor score of state i at time t, will influence $\boldsymbol{x}_{t+1}$ of the next time point. As the number of τ_i that are larger than an arbitrarily small constant ϵ follows a Poisson distribution with a finite mean as $\gamma_{0,\tau}\int_\epsilon^\infty \tau^{-1}e^{-c_\tau\tau}d\tau$, this can be used to express the idea that only a finite number elements in $\{x_{ti}\}_{i=1,\infty}$ at time t will influence $\boldsymbol{x}_{t+1}$.

2.1.2 Forming Unbounded Number of Overlapping State Communities

Given $G_{\rho,\tau} = \sum_{i=1}^{\infty}(\rho_i,\tau_i)\delta_{\boldsymbol{d}_i}$, we further need to build a stochastic process to form unbounded number of communities among states $\{x_{(t+1)i}\}_{i=1:\infty}$ and $\{x_{ti}\}_{i=1:\infty}$ where each of these communities will help to form one of concurrent LDSs. To facilitate that objective, we further define a gamma process $G_o \sim \Gamma\mathrm{P}(c_o, G_{\rho,\tau})$, with Lévy measure $\nu(dr d\boldsymbol{\theta} d\boldsymbol{\psi}) = r^{-1}e^{-cr}dr G_o(d\boldsymbol{\theta}d\boldsymbol{\psi})$, a draw from which is expressed as $G_o = \sum_{\kappa=1}^{\infty} r_\kappa \delta_{\{\boldsymbol{\theta}_\kappa,\boldsymbol{\psi}_\kappa\}}$. In this random draw, $r_\kappa \in \mathbb{R}_+$, reflecting the activation strength of community κ, is the weight of the κth atom $\{\boldsymbol{\theta}_\kappa, \boldsymbol{\psi}_\kappa\}$, where $\boldsymbol{\theta}_\kappa = (\theta_{1\kappa},\ldots,\theta_{\infty\kappa})^T$, $\boldsymbol{\psi}_\kappa = (\psi_{1\kappa},\ldots,\psi_{\infty\kappa})^T$, and $\theta_{i\kappa}$ and $\psi_{i\kappa}$, representing how strongly that node i is associated with community κ, are defined on ρ_i and τ_i, the weights of the atoms of the gamma process $G_{\rho,\tau}$, using $\theta_{i\kappa} \sim \mathrm{Gamma}(\rho_i, 1/e)$, $\psi_{i\kappa} \sim \mathrm{Gamma}(\tau_i, 1/f)$. We refer to the hierarchical stochastic process constructed in this way as the GGP. We denote the mass parameter of the GGP as $\gamma_0 := \int G_o(d\boldsymbol{\theta}d\boldsymbol{\psi})$. Inherited from the property of a gamma process, the GGP has an inherent shrinkage mechanism that its number of atoms (node communities) with weights greater than $\epsilon > 0$ is a finite random number drawn from $\mathrm{Pois}(\gamma_0\int_\epsilon^\infty r^{-1}e^{-cr}dr)$.

Given a random draw from the GGP as $G_o = \sum_{\kappa=1}^{\infty} r_\kappa\delta_{\{\boldsymbol{\theta}_\kappa,\boldsymbol{\psi}_\kappa\}}$, we will let r_κ determine the overall activation strength of community κ, $\theta_{i\kappa}$ how strongly state i in community κ is influenced by the states of the previous time in the same community, and $\psi_{j\kappa}$ how strongly state j in community κ influences the states of the next time in the same community. To express this idea, for community κ parameterized by $\{r_\kappa, \boldsymbol{\theta}_\kappa, \boldsymbol{\psi}_\kappa\}$, we generate a community-specific sparse adjacency matrix, whose (i,j)th element is drawn as

$$z_{ij\kappa} \sim \mathrm{Bernoulli}(1 - e^{-r_\kappa\theta_{i\kappa}\psi_{j\kappa}}). \qquad (2)$$

Thus from nodes j to i, community κ defines its own in-

teraction probability, expressed as $p_{ij\kappa} = 1-e^{-r_\kappa \theta_{i\kappa}\psi_{j\kappa}}$, and draws a binary edge $z_{ij\kappa}$ based on $p_{ij\kappa}$. While there are countably infinite nodes, in community κ, the total number of edges is a finite random number and hence the number of nodes with nonzero degrees is also finite.

Lemma 1. *The number of edges in community κ, expressed as $\sum_{i=1}^{\infty}\sum_{j=1}^{\infty} z_{ij\kappa}$, is finite.*

As in (2), whether $z_{ij\kappa} = 1$ or 0 is related to both the overall strength of community κ and how strongly nodes i and j are affiliated with community κ. Lemma 1, whose proof is deferred to the Appendix, suggests that we can extract a finite submatrix $\mathcal{Z}_\kappa := \{z_{ij\kappa}\}_{i,j\in\mathcal{S}_\kappa}$, where $\mathcal{S}_\kappa := \{i : \sum_j z_{ij\kappa} + \sum_j z_{ji\kappa} > 0\}$ is the set of nodes with non-zero degrees in community κ. We consider $\mathcal{S}_\kappa$ as the nodes activated by community κ and $\mathcal{Z}_\kappa$ as its nonempty graph adjacency matrix. Thus under the proposed GGP construction, different communities could overlap in the nodes belonging to their respective nonempty graph adjacency matrices, which means it is possible that $\mathcal{S}_\kappa \cap \mathcal{S}_{\kappa'} \neq \emptyset$ for $\kappa \neq \kappa'$. If $\mathcal{S}_\kappa \cap \mathcal{S}_{\kappa'} = \emptyset$, then we consider communities κ and κ' as two non-overlapping communities.

Our previous analysis shows whether $z_{ij} = 1$ determines not only whether state i at a given time will be dependent of the states of the previous time, but also whether state j at a given time will influence the states of the next time. To express the idea that whether $z_{ij} = 1$ is collectively decided by all countably infinite communities, whose nonempty adjacency matrices could overlap in their selections of nodes, we take the OR operation over all elements in $\{z_{ij\kappa}\}_\kappa$ to define the adjacency matrix of the full model as $z_{ij} = \vee_{\kappa=1}^{\infty} z_{ij\kappa}$, which means $z_{ij} = 1$ if at least one $z_{ij\kappa} = 1$, indicating community κ places a directed edge from nodes j to i, and $z_{ij} = 0$ otherwise. In a matrix format, we have $\mathbf{Z} = \vee_{\kappa=1}^{\infty}\mathbf{Z}^{(\kappa)}$, where $\mathbf{Z}^{(\kappa)}$ represents the graph adjacency matrix of community κ, whose (i,j)th element is $z_{ij\kappa}$.

We note that marginalizing out $\{z_{ij\kappa}\}_\kappa$, we can directly draw the graph adjacency matrix defined by $z_{ij} = \vee_{\kappa=1}^{\infty} z_{ij\kappa}$ as $\mathbf{Z} \sim \text{Bernoulli}(1 - e^{-\sum_{\kappa=1}^{\infty} r_\kappa \boldsymbol{\theta}_k \boldsymbol{\psi}_k^T})$, which can also be equivalently generated under the Bernoulli-Poisson link (Zhou, 2015) as $\mathbf{Z} = \delta(\mathbf{M} \geq 1)$, $\mathbf{M} = \sum_{\kappa=1}^{\infty}\mathbf{M}_\kappa$, $\mathbf{M}_\kappa \sim \text{Pois}(r_\kappa \boldsymbol{\theta}_\kappa \boldsymbol{\psi}_\kappa^T)$, where $\delta(\cdot)$ returns one if the condition is true and zero otherwise. While the graph defined by $\mathbf{Z}$ has countably infinite nodes, the total number of edges is finite and hence the number of nodes with nonzero degrees is also finite; the proof of the following Lemma is deferred to the Appendix.

Lemma 2. *The number of edges in $\mathbf{Z}$, expressed as $\sum_{i=1}^{\infty}\sum_{j=1}^{\infty} z_{ij}$, is finite.*

In summary, the GGP uses a gamma process to support countably infinite node communities in the prior, and another marked gamma process to support countably infinite number of nodes (states) shared by these communities. The adjacency matrix of the GGP generated random graph can be either viewed as taking the OR operation over all community-specific binary adjacency matrices, or viewed as thresholding a latent count matrix that aggregates the activation strengths across all communities for each node pair. Under this model construction, with the inherent shrinkage mechanisms of the gamma processes, only a finite number of communities will contain edges between the nodes, and the nonempty communities overlap with each other on their selections of nonzero-degree nodes, the total number of which across all communities is finite.

2.2 Hierarchical Model and Inference

To facilitate implementation, we truncate the GGP by setting K as an upper-bound of the number of communities, and S as an upper-bound of the number of states (nodes). We set $\gamma_{0,\rho} = \gamma_{0,\tau} = \gamma_0$. We make the scales of $\theta_{i\kappa}$ and $\psi_{j\kappa}$ change with κ to increase model flexibility. Letting $e_\kappa, f_\kappa \sim \text{Gamma}\,(\alpha_0, 1/\beta_0)$ and $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{m}_0, \mathbf{H}_0)$, the hierarchical model of the truncated GGP-LDS is expressed as

$$
\begin{aligned}
&\boldsymbol{y}_t \sim \mathcal{N}(\mathbf{D}\boldsymbol{x}_t,\, \boldsymbol{\Phi}^{-1}), \quad \boldsymbol{\Phi} \sim \text{Wishart}(\mathbf{V}, V+2),\\
&\boldsymbol{x}_t \sim \mathcal{N}\left[(\mathbf{W}\odot\mathbf{Z})\boldsymbol{x}_{t-1},\, \text{diag}(\lambda_1,\ldots,\lambda_S)^{-1}\right],\\
&\boldsymbol{d}_s \sim \mathcal{N}\left(\mathbf{0},\, \mathbf{I}_V/\sqrt{V}\right), \quad \lambda_s \sim \text{Gamma}\,(a, 1/b),\\
&w_{ij} \sim \mathcal{N}(0, \varphi_{ij}^{-1}), \quad \varphi_{ij} \sim \text{Gamma}\,(\alpha_0, 1/\beta_0),\\
&z_{ij} = \vee_{\kappa=1}^{K} z_{ij\kappa}, \quad z_{ij\kappa} = \delta(m_{ij\kappa} \geq 1),\\
&m_{ij\kappa} \sim \text{Pois}\,(r_\kappa\theta_{i\kappa}\psi_{j\kappa}), \; r_\kappa \sim \text{Gamma}(\gamma_0/K, 1/c),\\
&\theta_{i\kappa} \sim \text{Gamma}\,(\rho_i, 1/e_\kappa), \; \rho_i \sim \text{Gamma}\,(\gamma_0/S, 1/c_\rho),\\
&\psi_{j\kappa} \sim \text{Gamma}\,(\tau_j, 1/f_\kappa), \; \tau_j \sim \text{Gamma}\,(\gamma_0/S, 1/c_\tau).
\end{aligned}
$$

The graphical model is depicted in the Appendix A. As in Lemma 2, the total number of nonzero elements in $\mathbf{Z}$ has a finite expectation. Thus if the GGP truncation levels K and S are set large enough, it is expected for some state i that $\sum_j z_{ij} = 0$, which means its corresponding row in $\mathbf{Z}$ has no nonzero elements, and/or $\sum_j z_{ji} = 0$, which means its corresponding column in $\mathbf{Z}$ has no nonzero elements. If node i has zero degree that $\sum_j z_{ij} = \sum_j z_{ji} = 0$, then x_{ti} will neither depend on $\boldsymbol{x}_{t-1}$ nor influence $\boldsymbol{x}_{t+1}$, which means $\{x_{ti}\}_t$, the factor scores of state i, capture only the non-dynamic noise component of the data. Moreover, the proposed model will penalize the total energy captured by zero-degree node (state) i, expressed as $\sum_{t=1}^{T} x_{ti}^2$ if it is a zero-degree node (see Appendix B for more details).

We perform Bayesian inference via Gibbs sampling. Exploiting a variety of data augmentation and marginalization techniques developed for discrete data (Zhou

and Carin, 2013; Zhou, 2015) , we provide closed-form Gibbs sampling updated equations for all model parameters, as described in detail in Appendix E. Unless specified otherwise, we consider 6000 Gibbs sampling iterations, treat the first 3000 samples as burnin, and collect one sample per 60 iterations afterwards, resulting in a collection of 50 posterior MCMC samples that are used to predict the means and estimate the uncertainty of future observations. We provide a review of related work in Appendix D, where we compare our proposed models with a variety of dynamical systems, including switching LDSs and autoregressive, nonparametric, and deep neural network based models, and we clarify our distinct contributions.

Another set of time-series models are nonparametric Bayesian switching LDSs (Fox et al., 2009; Linderman et al., 2017), in which every temporal segment of the time series is fitted by one LDS. These models are focused on finding a mixture of LDSs, which are used to fit different time series segments, and a switching mechanism between different LDSs is learned to model the transitions between segments. Switching LDSs, however, may not provide satisfactory predictive performance on test data, as false switching, missed switching, and delayed switching could all compromise their predictions. Chiuso and Pillonetto (2010) design another type of nonparametric Bayesian models that identify sparse linear systems. Unlike the proposed GGP-LDS, it assumes no latent state transitions and models each observation as a linear combination of previous observations and some external input.

3 RELATED WORK

Modeling $\mathbf{Z}$ as the adjacency matrix of an infinite latent sparse graph is inspired by SGLDS (Kalantari et al., 2018), but the overlapping latent community structure added on top of that is unique to GGP-LDS. The Bernoulli-Poisson link used by Zhou (2015) and Caron and Fox (2015) to construct observed graphs are used by both SGLDS and GGP-LDS to construct latent spare graphs. SGLDS tries to model a single LDS with infinite number of states, while GGP-LDS tries to form infinite smaller overlapping LDSs in an infinite dimensional transition model, with smooth transition between them to support non-linear dynamics. Consequently, interpretation provided by SGLDS is not directly comparable to that of GGP-LDS, as SGLDS only provides one LDS while the overlapping latent community structures of GGP-LDS is the foundation to visualize and interpret the model's latent representation, such as decomposing a time series into community-specific subsequences, as will be shown below.

There are models that use the hierarchical Dirichlet process (Teh et al., 2006) priors over the states in hidden Markov models (Johnson and Willsky, 2013; Fox et al., 2009; Valera et al., 2015; Hayden et al., 2020). There are also models that perform clustering on the time series use a Pitman-Yor process based mixture prior on non-linear state-space models (Nieto-Barajas et al., 2014), and Dirichlet process mixtures (Caron et al., 2008) for modeling noise distributions. These models are not fully nonparametric as they typically have some parametric assumptions as part of the model such as having a fixed number of hidden states or imposing explicit specifications of the underlying temporal dynamics, such as seasonality and trends.

Chiuso and Pillonetto (2010) design a nonparametric Bayesian model to identify sparse linear systems. It assumes no latent transitions and believes each observation is a linear combination of previous observations plus some external input. Saad and Mansinghka (2018) introduce a recurrent Chinese restaurant process based mixture to capture temporal dependencies and a hierarchical prior to discover groups of time series whose underlying dynamics are modeled jointly. This model is able to cluster the observations to a set of trajectories with similar behaviors, although it is prone to creating unnecessary clusters as if the same pattern repeats with different magnitudes in two different segments of the observation, these two segments are likely to be assigned to two different clusters. This may result in many unnecessary clusters for high dimensional and/or lengthy data.

Another widely used type of time series models are autoregressive models (Harrison et al., 2003; Davis et al., 2016; Saad and Mansinghka, 2018). There also exist several other parametric models, such as Barber et al. (2011), that provide additional tools to model time series. Most of these parametric models require searching over a large set of possible parameter settings or model configurations to achieve satisfactory performance. More comprehensive literature review has been provided in the Appendix D.

4 EXPERIMENTAL RESULTS

In this section, we will demonstrate the interpretablity of GGP-LDS and its predictive performance on several different datasets. Details on how we create overlapping community based model interpretation and visualization are provided in Appendix C.1. Due to the inherent shrinkage mechanisms of the GGP, we find that the proposed nonparametric Bayesian model is not sensitive to the choice of the truncation levels S and K as long as they are set large enough. For all the datasets in this section, we truncate them at $K = 16$ and $S = 30$, which are found to be large enough to

accommodate all nonempty node communities, with interpretable latent representation and good predictive performance. Our Gibbs sampling based inference is not sensitive to initialization, allowing us to randomly initialize the model parameters. In this paper, we set $\gamma_0 = \alpha_0 = \beta_0 = c = c_\rho = c_\tau = 1$ for all experiments. We set $a = 1$ and $b = 0.1$ for all experiments (except for all visualizations, we set $b = 1$ to encourage sparser latent state-transition matrices), encouraging λ_s^{-1} to be small and hence encouraging the latent state representation vector to be constituted more by the autoregressive components and less by the white noise, generated by $\mathcal{N}\left[\mathbf{0}, \text{diag}(\lambda_1, \ldots, \lambda_S)^{-1}\right]$.

We compare the predictive performance of GGP-LDS with several representative time series models, whose description and parameter setup for each dataset are described in Appendix F. For each dataset, we consider tuning important parameters for each competing algorithm. Notably for GGP-LDS, when evaluating predictive performance, we simply use a same set of non-informative hyperparameters across all datasets and initialize all learnable parameters at random. Additional experiments on a synthetic dataset (the FitzHugh-Nagumo model) and a real dataset (closing stock price of 12 companies) will also be provided in Appendix F.

4.1 Lorenz Attractor

To demonstrate the performance of GGP-LDS on a dataset that has an underlying nonlinear dynamical pattern, we consider the Lorenz Attractor. We show how GGP-LDS finds an interpretable approximation to the generated time series with nonlinear dynamics. The Lorenz system is a classical nonlinear differential equation with three independent variables, defined as $\frac{dx_1}{dt} = \alpha(x_2 - x_1)$, $\frac{dx_2}{dt} = x_1(\beta - x_3) - x_2$, $\frac{dx_3}{dt} = x_1 x_2 - \gamma x_3$. There exist approximate solutions for this differential equation (Hernandez et al., 2018; Linderman et al., 2017; Nassar et al., 2018). A linear approximation will be very useful as we can leverage for this non-linear system many canonical algorithms developed for filtering and smoothing on linear systems. To show how our model approximates the latent states, we generate numerical solutions of the Lorenz system with a randomly generated initial state, $\alpha = 1$, $\beta = 2$, $\gamma = 1$, and $T = 578$ time points. The original generated variables under the Lorenz system have three dimensions (x_1, x_2, and x_3). We treat them as latent variables and use a randomly generated 10×3 matrix to map them to a 10-dimensional observation space. We use this $10 \times T$ observed data with added white Gaussian noise to train both GGP-LDS and a variety of baseline models.

Fig. 1 illustrates a single posterior sample of GGP-LDS, focusing on the inferred graph adjacency matrix, and the underlying activation probabilities of the edges of the graph adjacency matrix. More specifically, in the top row, we show on the left the graph adjacency matrix $\mathbf{Z}$, whose rows and columns have been separately reordered following the description in Section C.1, and on the right the underlying edge activation probabilities. In the bottom row, we show $\mathbf{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K) \in \mathbb{R}_+^{S \times K}$, $\mathbf{R} = \text{diag}(r_1, \ldots, r_K) \in \mathbb{R}_+^{K \times K}$, and $\mathbf{\Psi}^T = (\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_K)^T \in \mathbb{R}_+^{K \times S}$, where $\theta_{i\kappa}$ shows the affiliation strength of $x_{(t+1)i}$ to the κ^{th} community (κ^{th} LDS) and $\psi_{\kappa,j}$ shows the association strength of x_{tj} to the κ^{th} LDS.

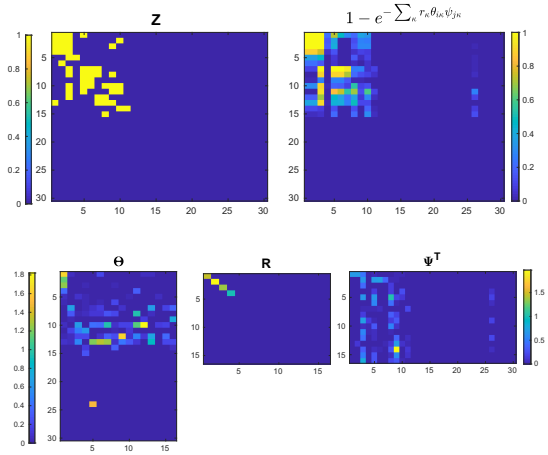

Figure 1: Visualization of a GGP-LDS inferred posteriori sample on a Lorenz Attractor synthesized time series. Top Left: $\mathbf{Z}$ from this posterior sample, where the rows and columns are separately reordered with the method described in Section C.1; Top Right: The inferred activation probability of $\mathbf{Z}$; Bottom Left: $\mathbf{\Theta}$, where $\theta_{i\kappa}$ shows how strongly state i is influenced by the states of the previous time due to its association with community κ; Bottom Middle: $\mathbf{R}$, whose diagonal elements show the activation strength of different communities; Bottom Right: $\mathbf{\Psi}^T$, where $\psi_{\kappa,j}$ shows how strongly state j influences the states of the next time due to its association with community κ.

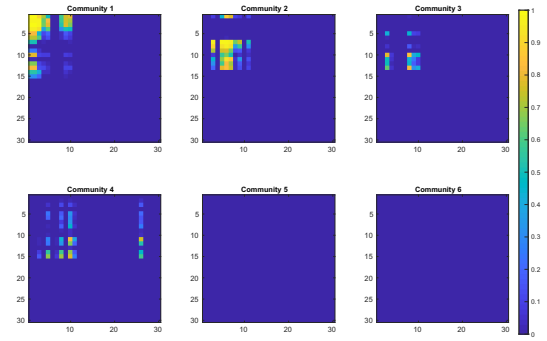

Figure 2: Relative activation strength $\mathbf{A}_\kappa$, as defined in (3), of the top six communities; note that 4 active communities formed over 15 active latent states are inferred by GGP-LDS while the truncation levels of the GGP are set as $K = 16$ and $S = 30$.

It can be observed how the shrinkage property of the gamma process $G_{\rho,\tau}$ has been effective in sparsifying the rows of $\mathbf{\Theta}$ and columns of $\mathbf{\Psi}^T$, with unnecessary

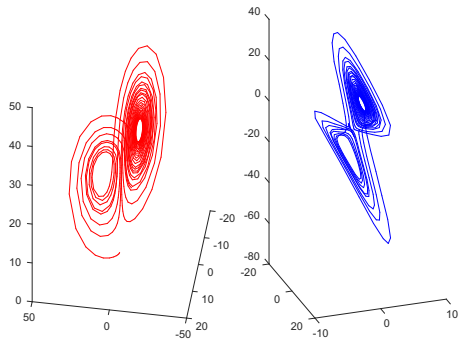

Figure 3: The red trajectory shown on the left is synthesized by a Lorenz Attractor and used as the 3D latent state sequence to generate $\boldsymbol{y}_{1:T}$, a 10D time series observation. Training GGP-LDS on $\boldsymbol{y}_{1:T}$, the blue trajectory shown on the right is a 3D visualization of the inferred latent dynamics based on $(\hat{\boldsymbol{x}}^{(1)}_{1:T}, \hat{\boldsymbol{x}}^{(2)}_{1:T}, \hat{\boldsymbol{x}}^{(3)}_{1:T})$, the sub-sequences of the three strongest communities decomposed from the reconstructed time series by GGP-LDS.

elements being shrunk towards zero. In addition, it can be seen that each active row of $\boldsymbol{\Theta}$, or active column of $\boldsymbol{\Psi}^T$ can potentially be a member of several different communities. The shrinkage property of the GGP G_o drives many elements of r_k towards zero and hence helps the model to pick which types of LDSs to be utilized. This is equivalent to say that the model infers which of these associations should be amplified or suppressed in expressing the underlying dynamics of the data. Moreover, for the $\boldsymbol{\Theta}$ matrix, it has 7 members (rows) associated with community one, which implies there are 7 corresponding states at time $t+1$ that will be influenced by $\boldsymbol{x}_t$ of the previous time due to their associations with community one, and $\boldsymbol{\Psi}^T$ shows that it has 4 members (columns) associated with community one, which implies that there are 4 corresponding states at time t that will influence $\boldsymbol{x}_{t+1}$ of the next time due to their associations with community one. Thus the transition matrix of the first member of overlapping LDSs will be the 7×4 block shown on the top left corner, as shown in both $\mathbf{Z}$ and its corresponding probability matrix in Fig. 1.

Out of $K = 16$ (truncation level) possible communities, we show in Fig. 2 the top six formed communities, extracted from the inferred transition matrix for Lorenz Attractor, in six different subplots; we display each of these six communities using its relative strength defined in (3). It is shown in Fig. 2 that our nonparametric Bayesian model finds four communities in total to model the underlying pattern of the data. The number of linear solutions that our model has discovered is similar to that of Nassar et al. (2018), in which a tree based stick breaking process has been used as the prior. Moreover, it can be observed from Figs. 1 and 2 that these 4 active communities are formed over 15 active states. Note for GGP-LDS, we have truncated its number of communities at $K = 16$ and that of states at $S = 30$. The results in Fig. 2 demonstrate the ability of GGP-LDS in inferring a parsimonious set of active communities and states to model the time series.

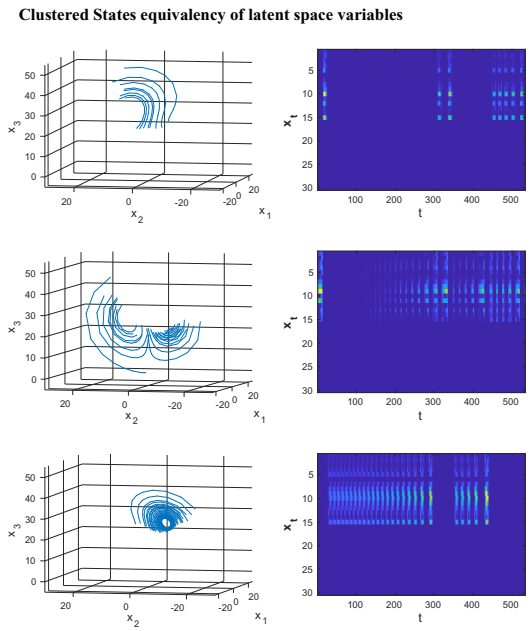

Figure 4: The bottom row visualizes the inferred latent states $\boldsymbol{x}_{1:T}$ of GGP-LDS, which are assigned into three non-overlapping clusters via the K-means algorithm, and the top row visualizes the corresponding segments of the Lorenz Attractor synthesized 3D time series.

Fig. 8 in Appendix shows how each community can reconstruct the observed data. Each row corresponds to a data dimension of the observed time series. The first three columns show how the three strongest communities contribute to data reconstruction, while the last column shows the superpositions of the first three columns and compares them against the observed time series. It can be seen from Fig. 8 that the different dimensions of each community specific sub-sequence share similar temporal patterns, but may exhibit clearly different magnitudes.

In Fig. 3, the red trajectory in the left plot represents the Lorenz Attractor synthesized 3D time series that is used as the latent state representation to generate the observed 10D time series, and the blue trajectory in the right plot illustrates a 3D representation of the latent dynamics of GGP-LDS trained on this 10D time series. More specifically, the blue trajectory is the visualization of the inferred community-specific latent sub-sequences $(\hat{\boldsymbol{x}}^{(1)}_{1:T}, \hat{\boldsymbol{x}}^{(2)}_{1:T}, \hat{\boldsymbol{x}}^{(3)}_{1:T})$, where $\hat{\boldsymbol{x}}^{(\kappa)}_{1:T}$, defined as in (4), is the latent sub-sequence extracted according to the relative strength of the κ^{th} strongest community to the aggregation of all communities, as illustrated in Figs. 2 and 8 and described in detail in section C.1. It can be seen from Fig. 3 that the latent dynamics (*e.g.*,

moving between two spirals) of GGP-LDS, visualized in 3D based on its inferred sub-sequences of its three strongest communities, are closely synchronized with the underlying dynamics of the Lorenz Attract synthesized 3D time series (a video showing how the red and blue trajectories move synchronously with each other is provided in the supplement). This shows that our model infers a close linear approximation to the underlying nonlinear dynamics.

We provide another visualization of the latent dynamics inferred by GGP-LDS in Fig. 4. Instead of decomposing the time series into sub-sequences, we now cluster it in time according to the inferred latent states $\boldsymbol{x}_t$. In the bottom row of Fig. 4, the $\boldsymbol{x}_t$'s are partitioned into three non-overlapping clusters with the K-means algorithm, which means each $\boldsymbol{x}_t$ is assigned to one of the three clusters. In the top row of Fig. 4, the same cluster assignment is applied to segment the Lorenz Attractor time series into three sequences that do not overlap in time. It is clear that the segmentation points based on the $\boldsymbol{x}_t$'s inferred by GGP-LDS well align with the switching points between different linear dynamics, demonstrating the ability of GGP-LDS to seamlessly transit between different temporal patterns, each of which is modeled by adjusting the activation strengths of different latent state communities that can co-occur at the same time.

In addition to these qualitative analyses, we quantitatively compare GGP-LDS and a variety of baseline algorithms on their predictive performance on the same 10D time series $\boldsymbol{y}_t$, generated by adding Gaussian noise to $\mathbf{D}\boldsymbol{x}_t$, where $\boldsymbol{x}_{1:T}$ is a Lorenz Attractor synthesized 3D time series. As $t = 445$ is one of the switching time points at which $\boldsymbol{x}_t$ moves from one spiral to another, we choose $\boldsymbol{y}_{1:445}$ for training. This set up can measure how well an algorithm detects and responds to changes in the underlying dynamics. The predictive performance of each algorithm is measured by mean absolute error defined in 23 over a horizon of 10 time points. The results are presented in Table 1. As shown in Table 1, most of the competing algorithms are not making good predictions following the switching point, likely because they expect that the trajectory will keep following the same spiral pattern before the switching point. In reality, the trajectory quickly switches to the other spiral pattern for a few steps before coming back to the same spiral pattern observed before $t = 445$. To further illustrate this point, we pick three different dimensions of the 10D time series $\boldsymbol{y}_t$, and show in Fig. 9 the prediction of four different algorithms, including SGLDS, TrLDS, TCRCP, and the proposed GGP-LDS, on these three dimensions over a horizon of 10 time points. It is evident from Fig. 9 that at the switching point, SGLDS, TrLDS, and TCRCP all fail to detect

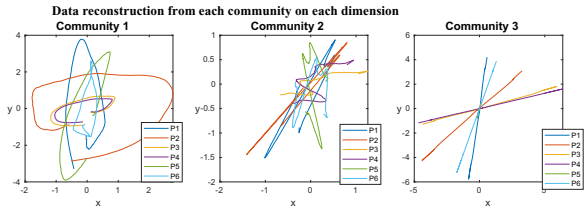

Figure 5: GGP-LDS applied on Pedestrians' trajectories data in 2D. Reconstruction of all 6 pedestrians' trajectories using three strongest communities

the transition from one spiral to another. More specifically, SGLDS closely follows the pattern of the same spiral, TrSLDS is experiencing delays in switching to the correct LDS that better fits the second spiral, and TRCRP creates wrong patterns.

4.2 Pedestrians' Trajectories

We analyze a dataset that records by camera the 3D motions of pedestrians and their interactions, downloaded from `https://motchallenge.net/` and available in the provided code repository. For visualization, we use only 2 dimensional data for each pedestrian (x, y). We select six pedestrians over 120 time points to train our model. The next 10 time points are used to measure the predictive performance of various algorithms.

Table 2 compares the predictive performance of various algorithms on this dataset. In most of the 10 forecast horizons our model has outperformed the other competing models. Fig. 11 in Appendix provides the interpretation of the latent factors for this dataset, analogous to Fig. 1 used to provide interpretation of the latent structure inferred from Lorenz Attractor. Fig. 5 and Fig. 12(b) in Appendix, analogous to Fig. 8 for Lorenz Attractor, represents the reconstruction of all 6 pedestrians' trajectories using the three strongest communities. A total of four communities is inferred by GGP-LDS to model the underlying pattern of the data as shown by Fig. 12(a) in Appendix. Fig. 5 shows how each community can decompose the observed data in 2 dimensions (x, y) into a community-specific subsequence. The last subplot in Fig. 12(b) superposes the first three sub-sequences and compares it against the true trajectory (shown in dashed lines).

It is interesting to see how each community can create one type of motion (*e.g.*, straight movement, circular trajectory, and spiral movement). It is evident that regardless of the property of motion, such as "turn direction," "radius of circular motion," or "direction of straight," the trajectories of the same nature have appeared in a same community-specific sub-sequence. It can be seen in the figure that some of the communities have a very small contribution for some of

Table 1: Lorenz Attractor predictive performance. The best result and the results that are not considered as statistically different are highlighted in bold.

	Mean absolute error for 10 forecast horizons									
Algorithm	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$	$t=9$	$t=10$
LDS	$11.12_{(1.10)}$	$13.76_{(2.47)}$	$18.54_{(2.78)}$	$22.34_{(2.90)}$	$18.81_{(3.18)}$	$13.12_{(3.30)}$	$9.68_{(2.87)}$	$8.54_{(2.37)}$	$7.21_{(2.51)}$	$7.54_{(2.48)}$
rLDSg	$8.12_{(0.90)}$	$12.76_{(1.31)}$	$17.33_{(1.98)}$	$20.52_{(1.61)}$	$15.62_{(2.05)}$	$10.54_{(3.21)}$	$8.62_{(3.28)}$	$9.42_{(3.65)}$	$7.46_{(3.16)}$	$6.28_{(3.49)}$
rLDSr	$12.46_{(1.72)}$	$19.22_{(3.72)}$	$21.51_{(4.22)}$	$25.32_{(3.67)}$	$18.21_{(3.11)}$	$11.51_{(4.16)}$	$13.21_{(4.21)}$	$10.21_{(4.11)}$	$8.57_{(3.95)}$	$6.91_{(3.18)}$
SGLDS	$8.84_{(1.32)}$	$10.43_{(1.66)}$	$14.51_{(2.43)}$	$15.32_{(3.61)}$	$16.31_{(3.24)}$	$15.32_{(3.83)}$	$12.21_{(3.94)}$	$9.13_{(4.24)}$	$9.57_{(3.95)}$	$7.91_{(3.68)}$
TrSLDS	$5.21_{(0.62)}$	$5.76_{(0.98)}$	$6.23_{(1.31)}$	$7.45_{(1.42)}$	$\mathbf{5.31_{(1.19)}}$	$\mathbf{5.12_{(1.53)}}$	$\mathbf{4.21_{(1.24)}}$	$\mathbf{2.31_{(1.18)}}$	$\mathbf{2.57_{(1.45)}}$	$\mathbf{5.78_{(1.11)}}$
Multi-output GP	$11.52_{(1.58)}$	$15.35_{(1.73)}$	$16.21_{(2.21)}$	$19.08_{(2.36)}$	$17.21_{(3.79)}$	$12.37_{(3.77)}$	$8.38_{(3.98)}$	$8.21_{(2.98)}$	$6.31_{(3.21)}$	$7.21_{(3.98)}$
FB Prophet	$5.57_{(1.31)}$	$11.82_{(1.45)}$	$13.42_{(1.98)}$	$15.21_{(2.04)}$	$16.26_{(1.76)}$	$9.41_{(1.86)}$	$8.78_{(2.01)}$	$7.66_{(1.91)}$	$6.54_{(2.14)}$	$6.72_{(2.23)}$
DeepAR	$9.42_{(0.26)}$	$10.21_{(0.31)}$	$16.22_{(0.54)}$	$16.42_{(1.28)}$	$15.24_{(1.21)}$	$11.21_{(1.61)}$	$13.25_{(2.08)}$	$12.83_{(2.83)}$	$14.21_{(3.01)}$	$16.25_{(3.21)}$
TRCRP	$5.66_{(0.86)}$	$7.91_{(1.01)}$	$11.23_{(1.35)}$	$15.37_{(2.31)}$	$16.21_{(2.42)}$	$9.68_{(2.68)}$	$8.21_{(2.98)}$	$6.85_{(2.71)}$	$5.63_{(2.38)}$	$7.35_{(2.81)}$
GGP-LDS (10 steps)	$\mathbf{2.12_{(0.84)}}$	$\mathbf{3.76_{(1.87)}}$	$\mathbf{4.77_{(2.68)}}$	$\mathbf{5.04_{(3.16)}}$	$\mathbf{4.83_{(3.32)}}$	$\mathbf{4.50_{(3.27)}}$	$\mathbf{4.15_{(3.34)}}$	$4.14_{(3.51)}$	$4.60_{(3.66)}$	$\mathbf{5.24_{(3.86)}}$
GGP-LDS (1 step)	$2.10_{(0.52)}$	$0.37_{(0.23)}$	$0.32_{(0.17)}$	$0.41_{(0.18)}$	$0.40_{(0.24)}$	$0.64_{(0.27)}$	$0.84_{(0.28)}$	$0.81_{(0.25)}$	$0.66_{(0.23)}$	$0.57_{(0.23)}$

the pedestrians' trajectory reconstruction since those pedestrians did not use that specific pattern in their recorded walking.

4.3 Stock Price

This dataset contains 12 companies' relative closing price ($P_t - P_{t-1}$) over the course of three years. These 12 companies have been selected from four different industries, and the stock closing prices of the ones in the same industry share similar temporal behaviors. Table 3 compares the predictive performance of various algorithms on this dataset. In most of the 10 forecast horizons our model has outperformed the other competing models. Analogous to Fig. 1 on Lorenz attractor data set, Fig. 6 is visualization of a posterior sample of GGP-LDS applied to this data-set. Fig. 13(a) shows the formed communities. Eight non-zero communities has been formed with the first four communities having at least one non-overlapping member, while the next four communities do not have members that are exclusive to them. Having four major communities, Fig. 13(b) shows how each of these major communities can contribute to reconstruct the observed data. Rows of Fig. 13(b) correspond to 6 stocks out of 12. The first four columns of the figure describe how the four strongest detected communities contributed to reconstruct the data in each dimension. There are two noticeable observations in Fig. 13(b). First, it can be seen that each community represents similar behavior for all 6 selected stocks in the figure, and these behaviors are distinct from one community to another. Second, it is evident that if a behavior represented by a community does not play a significant role in reconstruction of the data in a specific dimension, that community contribution will be trivial. As an example, community 3's role in reconstructing the second stock is trivial, while playing a much more significant role to reconstruct the observation of the fourth stock.

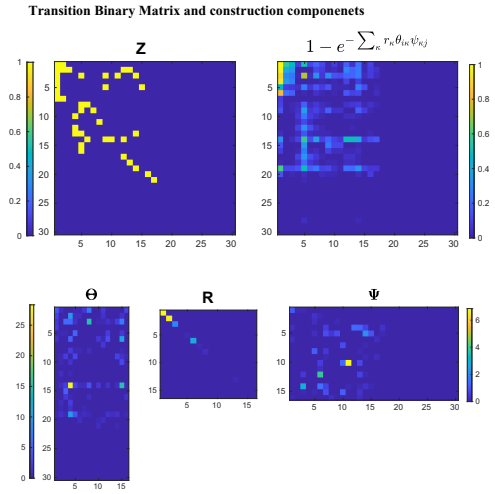

Figure 6: GGP-LDS posteriori sample visualization for Stock price dataset

5 CONCLUSION

We introduce the graph gamma process (GGP) to form an infinite dimensional transition model with a finite random number of nonzero-degree nodes and a finite random number of nonzero-edge communities over these nodes. The GGP is used to promote sparsity on the state-transition matrix of a linear dynamical system (LDS), and encourage forming overlapping communities among the nonzero-degree nodes of the graph. The model is designed such that each node community models a behavior described with an LDS. Instead of assigning one behavior to a temporal segment of an observation trajectory, it allows any observation point to be a combination of different simple trajectories, each modeled by one of the discovered communities and a smooth transition process forming one trajectory to another. This way, we can break the sophisticated behavior in a trajectory to a combination of simple behaviors which are modeled by linear systems, which helps model the nonlinearities of the data by smooth transitioning between these linear systems.

Acknowledgements

The authors acknowledge the support of Grants IIS-1812699 and ECCS-1952193 from the U.S. National Science Foundation, and the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper. URL: http://www.tacc.utexas.edu

References

M. Alvarez and N. D. Lawrence. Sparse convolved gaussian processes for multi-output regression. In *Advances in neural information processing systems*, pages 57–64, 2009.

D. Barber, A. T. Cemgil, and S. Chiappa. *Bayesian time series models*. Cambridge University Press, 2011.

F. Caron and E. B. Fox. Sparse graphs using exchangeable random measures. *arXiv:1401.1137v3*, 2015.

F. Caron, M. Davy, A. Doucet, E. Duflos, and P. Vanheeghe. Bayesian inference for linear dynamic models with dirichlet process mixtures. *IEEE Transactions on Signal Processing*, 56(1):71–84, 2008.

C. M. Carvalho and H. F. Lopes. Simulation-based sequential analysis of Markov switching stochastic volatility models. *Computational Statistics & Data Analysis*, 51(9):4526–4542, 2007.

A. Charles, M. S. Asif, J. Romberg, and C. Rozell. Sparsity penalties in dynamical system estimation. In *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*, pages 1–6. IEEE, 2011.

A. Chiuso and G. Pillonetto. Learning sparse dynamic linear systems using stable spline kernels and exponential hyperpriors. In *Advances in Neural Information Processing Systems*, pages 397–405, 2010.

R. A. Davis, P. Zang, and T. Zheng. Sparse vector autoregressive modeling. *Journal of Computational and Graphical Statistics*, 25(4):1077–1096, 2016.

V. Flunkert, D. Salinas, and J. Gasthaus. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *arXiv preprint arXiv:1704.04110*, 2017.

E. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. Nonparametric bayesian learning of switching linear dynamical systems. In *NIPS*, pages 457–464. 2009.

E. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. Bayesian nonparametric inference of switching dynamic linear models. *IEEE Transactions on Signal Processing*, 59(4):1569–1585, 2011.

Y. Gao, E. W. Archer, L. Paninski, and J. P. Cunningham. Linear dynamical neural population models through nonlinear embeddings. In *Advances in neural information processing systems*, pages 163–171, 2016.

Z. Ghahramani and G. E. Hinton. Parameter estimation for linear dynamical systems. Technical report, 1996.

Z. Ghahramani and S. T. Roweis. Learning nonlinear dynamical systems using an EM algorithm. In *NIPS*, pages 431–437, 1999.

M. Hardt, T. Ma, and B. Recht. Gradient descent learns linear dynamical systems. *The Journal of Machine Learning Research*, 19(1):1025–1068, 2018.

L. Harrison, W. D. Penny, and K. Friston. Multivariate autoregressive modeling of fmri time series. *Neuroimage*, 19(4):1477–1491, 2003.

D. S. Hayden, J. Pacheco, and J. W. Fisher. Nonparametric object and parts modeling with lie group dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7426–7435, 2020.

D. Hernandez, A. K. Moretti, Z. Wei, S. Saxena, J. Cunningham, and L. Paninski. A novel variational family for hidden nonlinear markov models. *arXiv preprint arXiv:1811.02459*, 2018.

M. Imani and U. M. Braga-Neto. Particle filters for partially-observed boolean dynamical systems. *Automatica*, 87:238–250, 2018.

H. Ishwaran and J. S. Rao. Spike and slab variable selection: frequentist and Bayesian strategies. *Annals of statistics*, pages 730–773, 2005.

M. Johnson, D. K. Duvenaud, A. Wiltschko, R. P. Adams, and S. R. Datta. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in neural information processing systems*, pages 2946–2954, 2016.

M. J. Johnson and A. S. Willsky. Bayesian nonparametric hidden semi-markov models. *Journal of Machine Learning Research*, 14(Feb):673–701, 2013.

R. Kalantari, J. Ghosh, and M. Zhou. Nonparametric bayesian sparse graph linear dynamical systems. *arXiv preprint arXiv:1802.07434*, 2018.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

R. E. Kalman. Mathematical description of linear dynamical systems. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, 1(2): 152–192, 1963.

C. E. Kennedy and J. P. Turley. Time series analysis as input for clinical predictive modeling: Modeling cardiac arrest in a pediatric icu. *Theoretical Biology and Medical Modelling*, 8(1):40, 2011.

J. F. C. Kingman. *Poisson Processes*. Oxford University Press, 1993.

S. Koyama. Projection smoothing for continuous and continuous-discrete stochastic dynamic systems. *Signal Processing*, 144:333–340, 2018.

G. Lai, W.-C. Chang, Y. Yang, and H. Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 95–104. ACM, 2018.

J. Li, C. Hu, D. Xu, J. Xiao, and H. Wang. Application of time-series autoregressive integrated moving average model in predicting the epidemic situation of newcastle disease. In *World Automation Congress (WAC), 2010*, pages 141–144. IEEE, 2010.

S. Linderman, M. Johnson, A. Miller, R. Adams, D. Blei, and L. Paninski. Bayesian learning and inference in recurrent switching linear dynamical systems. In *AISTATS*, volume 54, pages 914–922, Fort Lauderdale, FL, USA, 20–22 Apr 2017.

Z. Liu and M. Hauskrecht. A regularized linear dynamical system framework for multivariate time series analysis. In *AAAI*, pages 1798–1804, 2015.

L. Ljung. *System Identification: Theory for the User, 2nd edition*. Prentice Hall, 1999.

T. J. Mitchell and J. J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.

K. P. Murphy. Switching Kalman filters. 1998.

J. Nassar, S. Linderman, M. Bugallo, and I. M. Park. Tree-structured recurrent switching linear dynamical systems for multi-scale modeling. *arXiv preprint arXiv:1811.12386*, 2018.

L. E. Nieto-Barajas, A. Contreras-Cristán, et al. A bayesian nonparametric approach for time series clustering. *Bayesian Analysis*, 9(1):147–170, 2014.

F. Saad and V. Mansinghka. Temporally-reweighted chinese restaurant process mixtures for clustering, imputing, and forecasting multivariate time series. In *International Conference on Artificial Intelligence and Statistics*, pages 755–764, 2018.

S. Siddiqi, B. Boots, and G. Gordon. Reduced-rank hidden markov models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 741–748, 2010.

N. Städler, S. Mukherjee, et al. Penalized estimation in high-dimensional hidden markov models with state-specific graphical models. *The Annals of Applied Statistics*, 7(4):2157–2179, 2013.

S. J. Taylor and B. Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *J. Amer. Statist. Assoc.*, 101, 2006.

M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244, 2001.

I. Valera, F. Ruiz, L. Svensson, and F. Perez-Cruz. Infinite factorial dynamical model. *Advances in Neural Information Processing Systems*, 28:1666–1674, 2015.

M. West and J. Harrison. *Bayesian Forecasting and Dynamic Models (2Nd Ed.)*. Springer-Verlag New York, Inc., New York, NY, USA, 1997. ISBN 0-387-94725-6.

H. Zhang, R. Ayoub, and S. Sundaram. Sensor selection for kalman filtering of linear dynamical systems: Complexity, limitations and greedy algorithms. *Automatica*, 78:202–210, 2017.

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

M. Zhou and L. Carin. Negative binomial process count and mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2): 307–320, 2013.

M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin. Non-parametric Bayesian dictionary learning for sparse image representations. In *NIPS*, 2009.