A Bayesian Nonparametric Method for Clustering, Imputation, and Forecasting in Multivariate Time Series

Feras A. Saad
Probabilistic Computing Project
Massachusetts Institute of Technology

Vikash K. Mansinghka
Probabilistic Computing Project
Massachusetts Institute of Technology

Abstract

This article proposes a Bayesian nonparametric method for forecasting, imputation, and clustering in sparsely observed, multivariate time series. The method is appropriate for jointly modeling hundreds of time series with widely varying, non-stationary dynamics. Given a collection of $N$ time series, the Bayesian model first partitions them into independent clusters using a Chinese restaurant process prior. Within a cluster, all time series are modeled jointly using a novel “temporally-coupled” extension of the Chinese restaurant process mixture. Markov chain Monte Carlo techniques are used to obtain samples from the posterior distribution, which are then used to form predictive inferences. We apply the technique to challenging prediction and imputation tasks using seasonal flu data from the US Center for Disease Control and Prevention, demonstrating competitive imputation performance and improved forecasting accuracy as compared to several state-of-the-art baselines. We also show that the model discovers interpretable clusters in datasets with hundreds of time series using macroeconomic data from the Gapminder Foundation.

1 Introduction

Multivariate time series data is ubiquitous, arising in domains such as macroeconomics, neuroscience, and public health. Unfortunately, forecasting, imputation, and clustering problems can be difficult to solve when there are tens or hundreds of time series. One challenge in these settings is that the data may reflect underlying processes with widely varying, non-stationary dynamics. Another challenge is that standard approaches based on auto-regressive models [4], such as AR(p), ARMA, ARIMA, and GARCH models, can become statistically unstable and computationally intensive [14]. Auto-regressive models also require users to know how to choose from a large set of possible parameter settings and appropriately transform the data. Other technical issues that arise include non-linear dependencies, missing data, the presence of outliers, redundant variables, and sparse dependencies; see e.g. [9] for a survey of empirical time series data.

This paper presents a domain-general nonparametric Bayesian model for multivariate time series which aims to address some of the above challenges. It also presents Markov chain Monte Carlo algorithms for forecasting, imputation, and clustering based on this model. The modeling approach is based on two extensions to Dirichlet process mixture models. First, we introduce an extension to the Chinese restaurant process that captures temporal dependencies in the generated data. Second, we introduce a hierarchical extension that allows us to cluster time series into groups whose underlying dynamics are modeled and forecast jointly.

We use the approach to forecast flu incidence rates in 10 US regions, based on data from the US Center for Disease Control and Prevention. Our approach uses weather and social media time series as covariates. Quantitative experiments show that the proposed approach outperforms several Bayesian and non-Bayesian baselines, including Facebook’s Prophet algorithm [31], Gaussian process models, and seasonal ARIMA models [13]. We also show competitive imputation performance with state-of-the-art time series imputation techniques such as Amelia II [12]. Furthermore, we apply the method to cluster hundreds of macroeconomic time series from the Gapminder Foundation, which detects meaningful clusters of countries whose data exhibit coherent temporal patterns.
2 The Temporally-Coupled Chinese Restaurant Process Mixture Model

We first outline the notations and basic setup used throughout this paper. Let \( \{x^n: n = 1, \ldots, N\} \) denote a collection of \( N \) discrete-time series, where the first \( T \) variables of the \( n \)th time series is \( x^n_{1:T} = (x^n_1, x^n_2, \ldots, x^n_T) \). Slice notation is used to index subsequences of variables, so that \( x^n_{t_1:t_2} = (x^n_{t_1}, \ldots, x^n_{t_2}) \) for \( t_1 < t_2 \). Superscript \( n \) will be often be omitted when discussing a single time series. The remainder of this section develops a generative process for the joint distribution of all random variables \( \{x^n: t = 1, \ldots, N, n = 1, \ldots, N\} \) in the \( N \) time series, which we proceed to describe in stages.

2.1 Background: CRP representation of Dirichlet process mixture models

Our approach is based on a temporal extension of the standard Dirichlet process mixture (DPM), which we review briefly. First consider the standard DPM in the non-temporal setting [8], with concentration \( \alpha \) and base measure \( \pi_\theta \). The joint distribution of a sequence of \( m \) exchangeable random variables \( x_1, \ldots, x_m \) is:

\[
P \sim DP(\alpha, \pi_\theta), \quad \theta_j \mid P \sim P, \quad x_j \mid \theta_j \sim F(\cdot \mid \theta_j).
\]

The DPM can be represented in terms of the Chinese restaurant process [1]. As \( P \) is almost-surely discrete, the \( m \) draws \( \{\theta_j\} \) contain repeated values, thereby inducing a clustering among data \( x_j \). Let \( \{\theta_k\} \) be the unique values among the \( \{\theta_j\} \), and let \( z_j \) denote the cluster assignment of \( x_j \) which satisfies \( \theta_{z_j} = \theta_j \). Let \( n_{jk} \) be the number of observations \( x_i \) with \( z_i = k \) for \( i < j \). Using the conditional distribution of \( z_j \) given previous cluster assignment \( z_{1:j-1} \), the joint distribution of an exchangeable data sequence \( x_1, x_2, \ldots \) in the CRP mixture model is:

\[
\{\theta_k\} \sim \pi_\theta
\]

\[
Pr[z_j = k \mid z_{1:j-1}, \alpha] \propto \begin{cases} n_{jk} & \text{if } 1 \leq k \leq \max(z_{1:j-1}) \\ \alpha & \text{if } k = \max(z_{1:j-1}) + 1 \end{cases}
\]

\[
x_j \mid z_j, \{\theta_k\} \sim F(\cdot \mid \theta_{z_j})
\]

The basic CRP mixture model (1), and algorithms for posterior inference, have been studied extensively for nonparametric modeling in a variety of statistical applications (see [32] and references therein for a survey). In the next section, we extend the CRP mixture model to the case of a single non-exchangeable, temporally-coupled data sequence \( x_1, x_2, \ldots \).

\[
\text{Figure 1: Graphical model for the temporally-coupled CRP in a single time series } x = (x_1, x_2, \ldots) \text{ with lagged window size } p = 1.
\]

2.2 The temporally-coupled CRP mixture for modeling a single time series

Our objective is to define a CRP-like process for a non-exchangeable discrete-time series \( (x_1, x_2, \ldots) \). where there is now a natural temporal ordering and dependence among the variables. Instead of having \( (x_t, z_t) \) be conditionally independent of all other data given \( z_{1:t-1} \) as in CRP mixture (1), we instead consider using previous observations \( x_{t-p:t-1} \) to make predictions about \( z_t \). The main idea in our approach is to modify the CRP prior by biasing the cluster probability \( Pr[z_t = k \mid z_{1:t-1}] \) at step \( t \) to additionally account for (i) the \( p \) most recent observations \( x_{t-p:t-1} \), and (ii) collection of lagged values \( D_{tk} := \{x_{t'-p:t'-1} \mid z_{t'} = k, 1 \leq t' < t\} \) of earlier data points \( x_{t'} \) assigned to cluster \( k \). The joint distribution of time series \( (x_1, x_2, \ldots) \) in the “temporally-coupled” CRP mixture model is:

\[
\{\theta_k\} \sim \pi_\theta
\]

\[
Pr[z_t = k \mid z_{1:t-1}, x_{t-p:t-1}, \alpha] \propto \begin{cases} n_{tk} G(x_{t-p:t-1} \mid D_{tk}, \lambda_G) & \text{if } 1 \leq k \leq \max(z_{1:t-1}) \\ \alpha G(x_{t-p:t-1} \mid \lambda_G) & \text{if } k = \max(z_{1:t-1}) + 1 \end{cases}
\]

\[
x_t \mid z_t, \{\theta_k\} \sim F(\cdot \mid \theta_{z_t})
\]

The main difference between the temporally-coupled CRP mixture (2) and the standard CRP mixture (1) is the term \( G(x_{t-p:t-1} \mid D_{tk}, \lambda_G) \), which is a non-negative “cohesion function” on \( \mathbb{R}^p \), parametrized by \( D_{tk} \) and a bundle of real values \( \lambda_G \). This term measures how well the current lagged values \( x_{t-p:t-1} \) match the collection of lagged values of earlier data \( D_{tk} \) in each cluster \( k \), thereby introducing temporal dependence to the model. The smoothness of the process depends on the choice of the window size \( p \): if \( t_1 \) and \( t_2 \) are close in time (relative to \( p \)) then they have overlapping lagged values \( x_{t_1-p:t_1-1} \) and \( x_{t_2-p:t_2-1} \), so \( G \) increases the prior probability that \( \{z_t = z_k\} \). More generally, any pair of time points \( t_1 \) and \( t_2 \) that share similar lagged
1. Sample concentration parameter of CRP
   \[ \alpha \sim \text{Gamma}(1,1) \]

2. Sample model hyperparameters \((n = 1, 2, \ldots, N)\)
   \[ \lambda^n_G \sim H^n_G \]
   \[ \lambda^n_F \sim H^n_F \]

3. Sample distribution parameters of \(F\) \((n = 1, 2, \ldots, N)\)
   \[ \theta^n_1, \theta^n_2, \ldots \sim \pi(\cdot | \lambda^n_F) \]

4. Assume first \(p\) values are known \((n = 1, 2, \ldots, N)\)
   \[ x^n_{-p+1:0} \sim (x^n_{-p+1}, \ldots, x^n_0) \]

5. Sample time series observations \((t = 1, 2, \ldots)\)

   5.1 Sample temporal cluster assignment \(z_t\)
   \[ \text{Pr}[z_t = k | z_{1:t-1}, x^n_{-p:t-1}, \alpha] \sim \text{CRP}(k | a, z_{1:t-1}) \prod_{n=1}^N G(x^n_{t-p:t-1} | D^n_{tk}, \lambda^n) \]
   where \(D^n_{tk} := \{x^n_{t-p:t-1} | z_t = k, 1 \leq t' < t \}\)
   and \(k = 1, \ldots, \max(z_{1:t-1}) + 1 \)

   5.2 Sample data \(x^n_t \quad (n = 1, 2, \ldots, N)\)
   \[ x^n_t | z_t, \{\theta^n_k\} \sim F(\cdot | \theta^n_{z_t}) \]

(a) Generative process for the multivariate TCCR mixture

(b) Discovering flu season dynamics with the TCCR mixture

Figure 2: (a) Hierarchical Bayesian model for the joint distribution of \(N\) dependent time series \(\{x^n\}\), using a multivariate temporally-coupled CRP mixture. Lagged values for all the time series are used for reweighting the CRP by \(G\) in Step 5.1. The dependencies between time series are mediated by the shared cluster assignment \(z_t\), which ensures that all the time series share the same segmentation of the time course into various temporal regimes. (b) Applying the multivariate TCCR with \(p = 10\) weeks to model flu, tweets, and weather data in US Region 4. Six temporal regimes describing the seasonal behavior shared among \(x^{\text{flu}}, x^{\text{tweet}}, \text{and} x^{\text{temp}}\) are detected in this posterior sample: purple, gray, and red are the pre-peak rise, peak, and post-peak decline during the flu season; yellow, brown, and green represent the rebound in between successive seasons. In 2012, the model reports absence of the red post-peak regime, reflecting the season’s mild flu peak. See Section 5 for more details.

values are a-priori more likely to have similar distributions for generating \(x_{t1}\) and \(x_{t2}\), because \(G\) increases the probability that \(\{z_{t1} = z_{t2} = k\}\), so that \(x_{t1}\) and \(x_{t2}\) are both drawn from \(F(\cdot | \theta_k)\).

Figure 1 shows a graphical model for the TCCR mixture (2) with window size \(p = 1\). The model proceeds as follows: first assume observations of the first \(p\) observations \((x_{-p+1}, \ldots, x_0)\) are fixed or have a known initial distribution. At time step \(t\), we sample a cluster assignment \(z_t\), whose probability of joining cluster \(k\) is a product of (i) the CRP probability for \(\{z_{t} = k\}\) given all previous cluster assignments \(z_{1:t-1}\), and (ii) the cohesion term \(G(x^n_{t-p:t-1} | \lambda^G, D_{tk})\). In Figure 1, bold edges between the \(z_t\)’s denote the CRP probabilities, while edges from \(z_{t-1}\) back up to \(z_t\) represent reweighting the CRP using \(G\). Cluster assignment \(z_t\) identifies the “temporal regime” that dictates the distribution of observation \(x_t \sim F(\cdot | \theta_{z_t})\). Observe that if \(p = 0\) or \(G \propto 1\), then the model reduces to a standard CRP mixture (1) with no temporal dependence, since \(\{z_{t}, x_t\}\) are conditionally independent of the entire time series history \(x_{1:t-1}\) given assignments \(z_{1:t-1}\).

We complete the model by describing choices of \(F\) and \(G\). For the data distribution \(F\), we use a Normal distribution with Normal-InverseGamma prior \(\pi_{\Theta}\):

\[
\pi_{\Theta}(\mu_k, \sigma^2_k | m, V, a, b) = N(\mu_k | m, \sigma^2_k V) Ig(\sigma^2_k | a, b),
\]

\[
F(x_t | \mu_k, \sigma^2_k) = N(x_t | \mu_k, \sigma^2_k),
\]

where \(\theta_k = (\mu_k, \sigma^2_k)\) are the per-cluster parameters of \(F\), and \(\lambda^G = (m, V, a, b)\) the hyperparameters of \(\pi_{\Theta}\). Conjunctive of \(F\) and \(\pi_{\Theta}\) [3] implies that \(\theta_k\) need not be explicitly represented in the generative model 2a (see Appendix B). As for \(G\), it may in general be any non-negative weighting function which assigns a high potential to lagged data vectors that are “similar” in some sense. Previous approaches in the Bayesian regression setting have constructed covariate-dependent probability measures using kernel-based
It is informative to consider how 
\( z_{2012} \text{ to } 2015 \), which we explore further in Section 5.

weather activity in Region 4 of the United States from

Figure 2b shows an illustrative application to discover-

hyperpriors to the model's hyperparameters. Fig-

multivariate TCCRP mixture, where we have added

Figure 2a contains a step-by-step description of the multivariate TCCRP mixture, where we have added hyperpriors to the model's hyperparameters. Figure 2b shows an illustrative application to discovering the joint seasonal dynamics for flu, Twitter, and

where hyperparameter \( \lambda_{Gi} = (m_{ki}, V_{ki}, a_{ki}, b_{ki}) \) and data \( D_{tki} = \{x_{t'-i} : z_{t'} = k, 1 \leq t' < t \} \). Equations for the data-dependent terms \( (m_{ki}, V_{ki}, a_{ki}, b_{ki}) \) are given in Appendix A. We emphasize that \( G \) is a reweighting term only, and does not itself define a probability distribution over lagged data. Mathematically, the cohesion attracts \( x_t \) towards a cluster \( k \) that assigns \( x_{t-p:t-1} \) a high predictive density, under a product of optimal Gaussian distributions (in the Bayesian sense, see [20]) having observed \( D_{tk} \).

2.3 Extending the TCCRP mixture to multiple dependent time series

We now generalize the univariate TCCRP mixture (2) to handle a collection of \( N \) time series \( \{x^n : n = 1, \ldots, N\} \), assumed for now to all be dependent.

The main idea is to let the temporal regime assignment \( z_t \) be shared among all the time series, and use lagged values of all \( N \) time series when reweighting the CRP probabilities by the cohesion function \( G \):

\[
Pr[z_t = k \mid z_{1:t-1}, x_{1:p:t-1}, G] \propto CRP(k \mid \alpha) \prod_{n=1}^{N} G(x^n_{t-p:t-1}; D^n_{tk}, \lambda^n_G),
\]

where \( k = 1, \ldots, \max(z_{1:t-1}) + 1 \).

Figure 2a contains a step-by-step description of the multivariate TCCRP mixture, where we have added hyperpriors to the model's hyperparameters. Figure 2b shows an illustrative application to discovering the joint seasonal dynamics for flu, Twitter, and weather activity in Region 4 of the United States from 2012 to 2015, which we explore further in Section 5. It is informative to consider how \( z_t \) mediates dependences between \( x^{1:N} \). First, the model requires all time series to be in the same regime \( z_t \) at time \( t \). However, each time series has its own set of per-cluster parameters \( \{\theta^n_k\} \). Therefore, all the time series share the same segmentation \( z_{1:T} \) of the time course into various temporal regimes, even though the parametric distributions \( F(\cdot | \theta^n_k), n = 1, \ldots, N \) within each temporal regime \( k \in z_{1:T} \) differ. Second, the model assumes that data \( \{x^n_t : n = 1, \ldots, N\} \) at time \( t \) are independent given \( z_t \), and the cohesion (5) factors as a product. This characteristic is essential for numerical stability of the method in high dimensional and sparse regimes, while still maintaining the ability to recover complex distributions due to the infinite CRP mixture.

2.4 Learning the dependence structure between multiple time series

The multivariate TCCRP mixture in Figure 2a makes the restrictive assumption that all time series \( x^{1:N} \) are dependent with one another. However, with dozens or hundreds of time series whose temporal regimes are not well-aligned, forcing a single segmentation sequence \( z_{1:T} \) to apply to all \( N \) time series is likely to provide a poor fit to the data. We relax this modeling assumption by introducing a sparsity-inducing hierarchical prior, so that the model can determine which subsets of the \( N \) time series are probably well-described by a joint TCCRP model. In particular, we nonparametrically partition the \( N \) time series using a CRP, and then model all time series assigned to the same cluster jointly using a multivariate TCCRP mixture from Figure 2a:

\[
(c^1, c^2, \ldots, c^N) \sim CRP(\cdot | \alpha_0) \\
\{x^n : c^n = k\} \sim \text{Multivariate TCCRP} \\
k = 1, \ldots, \max(c^{1:N}),
\]

where \( c^n \) is the cluster assignment of \( x^n \). Figure 3 shows a visual illustration of the structure learning prior applied to an illustrative population of 5 EEG time series. In cluster 2 of Figure 3c, the final yellow segment illustrates two time series that share the latent regime at each time step, but have different distributional characteristics within each regime.

3 Related Work

Our modeling approach can be viewed as a temporal extension of previous nonparametric Bayesian models for cross-sectional data. Several authors have built covariate-based clustering models that employ a reweighted Chinese restaurant process [28, 23, 17, 19, 11]. These techniques are applied to nonparametric Bayesian regression problems involving exchangeable data. The TCCRP mixture prior (2) extends the idea of covariate-based clustering to the case of time series data, where exchangeability assumptions do not hold and standard algorithms for inference in DPMs do not apply. Moreover, the hierarchical model from this paper, shown in (6), coincides with the CrossCat model from [16] when all temporal dependencies are removed (by setting \( p = 0 \) or \( G \propto 1 \)).

In univariate time series, DP mixtures of autoregressive models have been proposed to learn tran-
sition distributions \( x_t | x_{t-1:t-p} \) in non-stationary time series, using both Polya-urn [18, 29, 30, 15] and stick-breaking [6, 2] representations. Unlike these methods, our approach uses simple mixtures of normals, which improves numerical stability when scaled to high dimensions. Nonparametric Bayesian frameworks for modeling multivariate time series data have also been proposed. For example, [26] introduce a dynamic density estimation method based on the dependent Dirichlet process, and [22] show how to cluster time series using a Pitman-Yor mixture of linear state-space models with a nonlinear observation function. However, both these approaches require explicit, dataset-specific specifications of the underlying temporal dynamics such as seasonality and trends. Moreover, neither approach has been quantitatively compared to non-Bayesian baseline techniques on challenging multivariate forecasting and imputation tasks. To the best of our knowledge, this paper presents the first nonparametric Bayesian model that can be applied to a broad class of time series data for multivariate forecasting and imputation tasks. To the best of our knowledge, this paper presents the first nonparametric Bayesian model that can be applied to a broad class of time series data for multivariate forecasting and imputation, without needing to specify custom temporal dynamics on a per-problem basis.

4 Posterior Inferences via Markov Chain Monte Carlo

In this section, we outline the full model likelihood, and briefly describe MCMC algorithms for inference in the TCCRP mixture with hierarchical prior \( \beta \). Since the model learns \( M = \max(c^{1:N}) \) separate TCCRP mixture models (one for each cluster of time series) we superscript latent variables of Figure 2a by \( m = 1, \ldots, M \). Namely, \( \alpha^m \) is the CRP concentration, and \( z_{1:T}^m \) the latent regime vector, shared by all time series in cluster \( m \). Further, let \( K_m = \max(z_{1:T}^m) \) denote the number of unique regimes in \( z_{1:T}^m \). Given model order \( p \) and observations \( \{ x_{n:p+1:T}^n : n = 1, \ldots, N \} \), we have:

\[
\begin{align*}
    & P(\alpha_0, c^{1:N}, \alpha^{1:M}, \lambda_G^{1:N}, \lambda_F^{1:N}, \{ \theta_j^m : 1 \leq j \leq K, n = 1 \}^N_{n=1}, \\
    & z_{1:T}^1, \ldots, z_{1:T}^N | x_{1:p+1:T} \\
    &= p(\alpha_0) \text{CRP}(\alpha^{1:N} | \alpha_0) \\\n    &= \left( \prod_{n=1}^N \prod_{t=1}^T N \right) \frac{1}{\prod_{n=1}^N H_G(\lambda_G^n) \prod_{n=1}^N H_F(\lambda_F^n)} \prod_{n=1}^N \prod_{j=1}^{K_n} \pi_{\theta_j}(\theta_j^n) \prod_{m=1}^M \left( \prod_{t=1}^T \prod_{n=1}^N b^m_n \text{CRP}(z_t^n | z_{1:t-1}^m, \alpha^m) \right) \prod_{n=1}^N G(x_{n:p+1:T}^n | z_{1:T}^m, \lambda^n) F(x_t^n | \theta_t^n), \quad (7)
\end{align*}
\]

where \( b^m_n \) normalizes the term between the square brackets, summing over \( z_t^m = 1, \ldots, \max(z_{1:T}^m) + 1 \). Eq (7) defines the unnormalized posterior distribution of all latent variables given the data. Appendix B contains detailed algorithms for posterior inference. Briefly, temporal regime assignments \( (z_t^m | z_{1:T}^m, \alpha^m, \lambda^n) \) are sampled using a variant of Algorithm 3 from [21], taking care to handle the temporal-coupling term \( b^m_n \) which is not found in traditional DPM samplers. We also outline an alternative particle-learning scheme [5] to sample \( (z_{1:T}^m, \alpha^m, \lambda^n) \) jointly as a block. Time series cluster assignments \( (c^n | c^{1:N}) \) are sampled using an empirical Bayes approach [25] based on the “griddy Gibbs” [24] sampler.

4.1 Making predictive inferences

Given a collection of approximate posterior samples \( \{ \hat{\xi}_1, \ldots, \hat{\xi}_S \} \) of all latent variables produced by \( S \) runs...
of Markov chain Monte Carlo, we can draw a variety of predictive inferences about the time series \( x^{1:N} \) which form the basis of the applications in Section 5.

**Forecasting** For out-of-sample time points, forecasting on an \( h \) step horizon \( T < t < T + h \) is achieved by ancestral sampling: first draw \( \tilde{s} \sim \text{Uniform}[1 \ldots S] \), then simulate Step 5 of Figure 2a using the latent variables in \( \xi^s \) for \( t = T, \ldots, T + h \).

**Clustering** For a pair of time series \( (x^t, x^s) \), the posterior probability that they are dependent is the fraction of samples in which they are in the same cluster:

\[
P(c^i = c^k \mid \{ x^{1:N} \}) \approx \frac{1}{S} \sum_{s=1}^{S} I [c^{i,s} = c^{k,s}].
\]  

(8)

**Imputation** Posterior inference yields samples of each temporal regime \( \tilde{z}_t^s \) for all in-sample time points \( 1 \leq t \leq T \); the posterior distribution of a missing value is:

\[
P(x^n_t \in B \mid \{ x^{1:N} \}) \approx \frac{1}{S} \sum_{s=1}^{S} F(B \mid \hat{\theta}^{n,s}_{t}).
\]  

(9)

5 Applications

In this section, we apply the proposed model to (i) forecasting and imputation tasks on seasonal flu data from the US Center for Disease Control and Prevention (CDC), and (ii) clustering hundreds of time series using macroeconomic data from the Gapminder Foundation. We describe the setup in the text below, with further details in the captions of Figures 4, 5, 6 and 8. Experimental methods are given in Appendix C.

Predicting flu rates is a fundamental task in public health policy. The CDC has an extensive dataset of flu rates and associated time series such as weather and vaccinations, with measurements taken weekly from January 1998 to June 2015. Figure 4a shows the influenza-like-illness rate (ILI, or simply flu), tweets, and minimum temperature time series in US Region 6, as well as six temporal regimes detected by one posterior sample of the TCCRP mixture model (\( p = 10 \) weeks). To quantitatively investigate the forecasting abilities of the model, we hold out the 2015 season for 10 US regions and formulate forecasts on a rolling ba-
(a) Four representative flu time series imputed jointly

(b) Example imputations in R09

(c) Mean absolute imputation errors in ten United States flu regions

| Method                  | R01 | R02 | R03 | R04 | R05 | R06 | R07 | R08 | R09 | R10 |
|-------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Mean Imputation         | 0.65(0.04) | 0.85(0.11) | 0.91(0.04) | 1.07(0.06) | 0.66(0.04) | 1.20(0.08) | 1.17(0.10) | 0.75(0.04) | 0.80(0.05) | 1.10(0.10) |
| Linear Interpolation    | 0.49(0.07) | 0.63(0.08) | 0.57(0.06) | 0.42(0.04) | 0.44(0.04) | 0.71(0.08) | 0.71(0.09) | 0.35(0.03) | 0.43(0.05) | 0.72(0.06) |
| Cubic Splines           | 0.61(0.15) | 0.72(0.09) | 0.61(0.06) | 0.89(0.10) | 0.69(0.06) | 1.68(0.21) | 1.42(0.22) | 0.63(0.05) | 0.99(0.22) | 1.47(0.13) |
| Amelia II [12]          | 0.20(0.03) | 0.52(0.11) | 0.25(0.02) | 0.45(0.03) | 0.20(0.03) | 0.53(0.04) | 0.53(0.05) | 0.37(0.03) | 0.39(0.04) | 0.51(0.03) |
| Multivariate TCCRP      | 0.23(0.03) | 0.47(0.09) | 0.23(0.02) | 0.49(0.04) | 0.31(0.03) | 0.55(0.05) | 0.75(0.07) | 0.84(0.03) | 0.37(0.03) | 0.67(0.07) |

**Figure 5**: Jointly imputing missing data in ten flu populations over eight seasons. (a) Imputations and standard errors in four of the time series. The TCCRP mixture model accurately captures both seasonal behavior as well as non-recurrent characteristics, such as the very mild flu season in 2012. (c) Comparing imputation quality with several baseline methods; the TCCRP mixture (using p = 10 weeks) achieves competitive performance. Cubic splines are completely ineffective due to long sequences without any observations. (b) While linear interpolation may seem to be a good performer given its simplicity and mean errors, unlike the TCCRP it cannot predict non-linear behavior when an entire flu season is unobserved. In these regimes, linear interpolation misses seasonality, resulting in poor imputations and miscalibrated uncertainty for downstream analyses.

sis. Namely, for each week t = 2014.40, ..., 2015.20 we forecast x_{flu}^{t+h} given x_{flu}^{t-2} and all available covariate data up to time t, with horizon h = 10. A core challenge in formulating forecasts is the two-week delay in obtaining the latest flu data: when forecasting x_{flu}^{t+h}, the most recent flu measurement is x_{flu}^{t-2}. Second, covariate time series are themselves sparsely observed in the training data; for instance, all tweet time series are missing before June 2013 (see Figure 4a). Experimental outcomes are summarized in Figure 4. We benchmark against domain-general baselines which do not require detailed custom modeling for obtaining forecasts. The univariate TCCRP only models the flu, while the multivariate TCCRP additionally includes tweet and weather signals. While methods such as seasonal ARIMA [13] support covariate data in principle, they cannot handle missing covariates in the training set or over the course of the forecast horizon (without significant, customized subroutines for imputation and forecasting as prerequisites for the main forecasting task). Both Facebook Prophet [31] and ARIMA incorrectly forecast the peak behavior (top row of Figure 4a), and illustrate high bias in the post-peak regime (bottom row of Figure 4). The Gaussian processes show wide posterior uncertainty, even in the relatively noiseless post-peak regime. Quantitatively, the univariate TCCRP achieves comparable performance to the GP methods. The multivariate TCCRP, which uses sparsely observed flu and weather covariates (when they are available), consistently produces the most accurate forecasts at all horizon lengths.

Next, we investigated the performance of the proposed model on a multivariate imputation task. We dropped windows of length 10 at a rate of 5% from flu series in US Regions 1-10, and built a joint model over all 10 time series. The top panel of Figure 5a shows a subset of the model’s imputations on four populations using Eq (9). Quantitative comparisons of imputation quality to several baseline methods are given in Table 5c. The TCCRP mixture achieves performance comparable to Amelia II [12], a specialized imputation package, with neither method clearly dominating outright.

We then applied the hierarchical prior (6) to cluster countries in the Gapminder dataset, which contains
A Bayesian Nonparametric Method for Multivariate Time Series

Figure 6: Given GDP per capita data for 170 countries from 1960-2010, the TCCRP mixture detects several qualitatively distinct temporal patterns. The top panel shows an overlay of all the time series. Nine representative clusters averaged over 60 posterior samples are shown below. Countries within each cluster, of which a subset are labeled, share similar political, economic, and/or geographic characteristics. For instance, cluster 1 contains Western democracies with stable economic growth over 50 years (the dip in 2008 is the financial crash). Cluster 2 includes China and India, whose GDP growth rates have greatly outpaced those of industrialized nations since the 1990s. Cluster 3 contains former communist nations, whose economies tanked after fall of the Soviet Union. Outliers such as Samoa, Equatorial Guinea, and North Korea can be seen in clusters 8 and 9.

dozens of macroeconomic time series for 170 countries spanning 50 years. Because fluctuations due to events such as natural disasters, financial crises, or healthcare epidemics are poorly described by parametric or hand-designed causal models, a key objective is to automatically discover the number and types of patterns underlying the temporal structure. Figure 6 shows structure discovery in GDP time series using a TCCRP mixture ($p = 5$ years), which produces several commonsense, qualitatively distinct clusters. All countries in a cluster have a posterior dependence probability exceeding 80% according to Eq (8). Countries within each cluster share similar political, economic, and/or geographic characteristics; see caption for additional details. An expanded set of clusterings illustrating changepoint detection in cell phone subscription time series is given in Appendix D.

6 Discussion

This paper has presented a domain-general nonparametric Bayesian method for building joint models of multivariate time series. Experiments show promising quantitative and qualitative results on real-world forecasting, imputation, and clustering tasks. Two limitations of this approach suggest avenues for future research. First, although our approach does not require users to specify prior knowledge about trends or seasonality, it also does not make it easy for users to incorporate this knowledge when it is available. Second, our inference algorithms are not optimized for online analysis of streaming data sources. A scalable implementation of sequential Monte Carlo inference algorithms could have broad applications in forecasting, clustering, and statistical outlier detection.
References

[1] D. J. Aldous. *Exchangeability and related topics*, pages 1–198. Springer, 1985.

[2] I. Antoniano-Villalobos and S. G. Walker. A non-parametric model for stationary time series. *Journal of Time Series Analysis*, 37(1):126–142, 2016. ISSN 1467-9892.

[3] J. Bernardo and A. Smith. *Bayesian Theory*. Wiley Series in Probability & Statistics. Wiley, 1994. ISBN 9780471924166.

[4] P. Brockwell and R. Davis. *Time Series: Theory and Methods*. Springer Series in Statistics. Springer, 2009.

[5] C. M. Carvalho, M. S. Johannes, H. F. Lopes, and N. G. Polson. Particle learning and smoothing. *Statistical Science*, 25(1):88–106, 02 2010.

[6] M. DeYoreo and A. Kottas. A Bayesian nonparametric Markovian model for non-stationary time series. *Statistics and Computing*, Oct 2016. ISSN 1573-1375.

[7] D. B. Dunson, N. Pillai, and J.-H. Park. Bayesian density regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):163–183, 2007.

[8] M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[9] B. D. Fulcher and N. S. Jones. Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):3026–3037, 2014.

[10] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[11] L. A. Hannah, D. M. Blei, and W. B. Powell. Dirichlet process mixtures of generalized linear models. *Journal of Machine Learning Research*, 12(Jun):1923–1953, 2011.

[12] J. Honaker, G. King, M. Blackwell, et al. Amelia ii: A program for missing data. *Journal of statistical software*, 45(7):1–47, 2011.

[13] R. Hyndman and Y. Khandakar. Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software, Articles*, 27(3):1–22, 2008. ISSN 1548-7660.

[14] G. M. Koop. Forecasting with medium and large Bayesian VARS. *Journal of Applied Econometrics*, 28(2):177–203, 2013. ISSN 1099-1255. doi: 10.1002/jeae.1270.

[15] J. W. Lau and M. K. So. Bayesian mixture of autoregressive models. *Computational Statistics and Data Analysis*, 53(1):38 – 60, 2008. ISSN 0167-9473.

[16] V. Mansinghka, P. Shafto, E. Jonas, C. Petschulat, M. Gasner, and J. B. Tenenbaum. Crosscat: A fully Bayesian nonparametric method for analyzing heterogeneous, high dimensional data. *Journal of Machine Learning Research*, 17(138):1–49, 2016.

[17] P. Mueller and F. Quintana. Random partition models with regression on covariates. *Journal of statistical planning and inference*, 140(10):2801–2808, 2010.

[18] P. Mueller, M. West, and S. MacEachern. Bayesian models for non-linear autoregressions. *Journal of Time Series Analysis*, 18(6):593–614, 1997.

[19] P. Mueller, F. Quintana, and G. L. Rosner. A product partition model with regression on covariates. *Journal of Computational and Graphical Statistics*, 20(1):200–278, 2011.

[20] K. P. Murphy. Conjugate Bayesian analysis of the Gaussian distribution. Technical report, University of British Columbia, 2007.

[21] R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

[22] L. E. Nieto-Barajas and A. Contreras-Cristan. A Bayesian nonparametric approach for time series clustering. *Bayesian Analysis*, 9(2):147–170, 03 2014.

[23] J.-H. Park and D. B. Dunson. Bayesian generalized product partition model. *Statistica Sinica*, pages 1203–1226, 2010.

[24] C. Ritter and M. Tanner. The griddy Gibbs sampler. Technical Report 878, University of Wisconsin-Madison, 1991.

[25] H. Robbins. The empirical Bayes approach to statistical decision problems. *The Annals of Mathematical Statistics*, 35(1):1–20, 1964.

[26] A. Rodriguez and E. ter Horst. Bayesian dynamic density estimation. *Bayesian Analysis*, 3(2):339–365, 6 2008.

[27] U. Schaechtle, B. Zinberg, A. Radul, K. Stathis, and V. Mansinghka. Probabilistic programming with Gaussian process memoization. *arXiv preprint*, arXiv:1512.05665, 2015.

[28] B. Shahbaba and R. Neal. Nonlinear models using Dirichlet process mixtures. *Journal of Machine Learning Research*, 10(Aug):1829–1850, 2009.
[29] Y. Tang and S. Ghosal. Posterior consistency of Dirichlet mixtures for estimating a transition density. *Journal of Statistical Planning and Inference*, 137(6):1711–1726, 2007.

[30] Y. Tang and S. Ghosal. A consistent nonparametric Bayesian procedure for estimating autoregressive conditional densities. *Computational Statistics and Data Analysis*, 51(9):4424 – 4437, 2007.

[31] S. J. Taylor and B. Letham. Forecasting at scale. *The American Statistician*, To appear, 2017.

[32] Y. W. Teh. Dirichlet process. In *Encyclopedia of machine learning*, pages 280–287. Springer, 2011.
Appendices

A Data-dependent parameters for Student-T cohesion function

Following Eq (4), the cohesion function $G$ is a product of $p$ Student-T distributions whose location, scale and degrees of freedom are data-dependent [20]:

$$G(x_{t-p:t-1}; D_{tk}, \lambda_G) = \prod_{i=1}^{p} G_i(x_{t-i}; D_{tki}, \lambda_{G_i}) = \prod_{i=1}^{p} T_{z_{tki}} \left( x_{t-i}; m_{tki}, b_{tki} \frac{V_{tki} + 1}{\sigma_{tki}} \right)$$  \hspace{1cm} (10)

where $m_{tki} = |D_{tki}|$, $b_{tki} = b_k + \frac{1}{2}(m_{tki}V_{tki}^{-1} + \sum_{t'} x_{t'-i}^2 - m_{tki}V_{tki}^{-1})$.

B Markov chain Monte Carlo methods for posterior inference

Here, we provide the details of the MCMC method for posterior simulation from the nonparametric mixture model developed in Section 2. As discussed in the main text, conjugacy of $F$ and $\pi_0$ in Eq (3) means we can analytically marginalize parameters $\{\theta_k^q\}$ when defining the generative process of the multivariate TCRP mixture from Figure 2a. The model therefore becomes:

$$\alpha \sim \text{Gamma}(1, 1)$$

$$\lambda^n_G \sim H^n_G$$

$$\lambda^n_F \sim H^n_F$$

$$x^n_{t-p+1:0} \sim (x^n_{p+1}, \ldots, x^n_0)$$

Pr[$z_t = k \mid x^n_{t-p:t-1}, x^n_{t-p:t-1}, \alpha$] $t = 1, 2, \ldots, T$

$\propto \text{CRP}(k | \alpha, z_{1:t-1}) \prod_{n=1}^{N} G(x^n_{t-p:t-1}; D^n_{tk}, \lambda^n_G)$

where $D^n_{tk} := \{x^n_{t-p:t-1} \mid z_t = k, 1 \leq t' < t \}$ and $k = 1, \ldots, \text{max} (z_{1:t-1}) + 1$

$x^n_{t} | z_{t} = k, x^n_{t-1:N} \sim \int_F(\theta)p_{\alpha}(\theta)D_{tk}^n, \lambda^n_F) d\theta$

where $D^n_{tk} := \{x^n_{t} \mid z_t = k, 1 \leq t' < t \}$.

The integration of $F$ against $\pi_0(\theta | D^n_{tk})$ in the right hand-side of the final line evaluates to a Student-T distribution as in Eq (10), whose updates given $D^n_{tk}$ and $\lambda^n_F$ are identical to those in Eq (11) with $i = 0$.

Inference on temporal regime assignments $(z_t | z_{1:T} \setminus \ldots)$. We first describe how to transition $z_{1:T}$, assuming the collapsed version of the multivariate TCRP (12) with $N$ time series. Note that since the structure learning prior Eq (6) specifies exactly $M = \text{max}(c^{1:N})$ independent TCRP mixtures (conditioned on the assignment vector), it suffices to describe inference on $z_{1:T}$ in one of the mixtures (which keeps notation significantly simpler). Given observations $x^{1:N}_{t-p+1:T}$, the joint likelihood of model (12) is:

$$P\left(\alpha, \lambda^n_G, \lambda^n_F, z_{1:T}, x^{1:N}_{t-p+1:T} \mid p \right) = p(\alpha_0) \prod_{n=1}^{N} H^n_G(\lambda^n_G) \prod_{n=1}^{N} H^n_F(\lambda^n_F)$$

$$\prod_{t=1}^{T} \left[ b_{tk}\text{CRP}(z_t \mid z_{1:t-1}, \alpha) \right]$$

$$\prod_{n=1}^{N} G(x^n_{t-p:t-1}; D^n_{tk}, \lambda^n_G) F(x^n_{t} \mid D^n_{tk}, \lambda^n_F)$$  \hspace{1cm} (13)

The normalizer at time $t$ is given by:

$$b_{tk}(x^n_{1:T}, z_{1:t-1})$$

$$= \left( \sum_{k=1}^{K_t} \text{CRP}(k |\alpha, z_{1:t-1}) \prod_{n=1}^{N} G(x^n_{t-p:t-1}; D^n_{tk}, \lambda^n_G) \right)_{-1}$$

where $K_t = \text{max}(z_{1:t-1}) + 1$. Note that the normalizer $b_t(x^{1:N}_{t-1}, z_{1:t-1})$ ensures the reweighted cluster probabilities sum to one. It will also be convenient to define the predictive density $q_t$ at time $t$ of data $x^{1:N}_{t}$, which sums out all possible values of $z_t$:

$$q_t(x^{1:N}_{t}; z_{1:t-1})$$

$$= b_t(x^{1:N}_{t}; z_{1:t-1}) \left( \sum_{k=1}^{K_t} \text{CRP}(k |\alpha, z_{1:t-1}) \prod_{n=1}^{N} G(x^n_{t-p:t-1}; D^n_{tk}, \lambda^n_G) \right)$$

Let the current state of the Markov chain be $(\alpha, \lambda^n_G, \lambda^n_F, z_{1:T})$. We present two algorithms for sampling the latent regimes assignments. Algorithm 1 is a single-site Metropolis-Hastings procedure that targets $(z_t | z_{1:T} \setminus \ldots)$ at each step, where we assume that all data in $x^{1:N}_{1:T}$ are fully observed. Algorithm 2 is an SMC scheme to block sample $(z_{1:T} \setminus \ldots)$ using particle learning [5]. Arbitrary observations may be missing, as they are imputed over the course of inference.
Algorithm 1: single-site Metropolis-Hastings. This algorithm proposes \((z_t|\mathbf{z}_{1:T\setminus t}, \ldots)\) at each step, assuming fully observed data \(\mathbf{x}_1^{1:N}\). Repeat for \(t = 1, 2, \ldots, T\):

1. Propose \(z'_t\) from the multinomial distribution:

\[
\Pr[z'_t = k | \mathbf{z}_{1:T\setminus t}, \mathbf{x}^{1:N}] \propto \text{CRP}(k|\alpha, \mathbf{z}_{1:T\setminus t}) \prod_{n=1}^{N} G(x^n_t | D^n_{T_k} \setminus \{x^n_t\}, \lambda^n_G, \alpha),
\]

where \(k \in \text{unique}(\mathbf{z}_{1:T\setminus t}) \cup \{\max(\mathbf{z}_{1:T\setminus t}) + 1\}\).

2. Compute the MH acceptance ratio \(r(z_t \rightarrow z'_t)\), using \(b_t\) defined in (14):

\[
r(z_t \rightarrow z'_t) = \frac{\Pi_{n' > t} b_{n'}(\mathbf{z}_{1:t'-1} \cup z'_t, x^{1:N})}{\Pi_{n > t} b_n(\mathbf{z}_{1:t'}, x^{1:N})}.
\]

3. Set \(z_t \leftarrow z'_t\) with probability \(\min(1, r)\), otherwise leave \(z_t\) unchanged.

Algorithm 2: block sampling with particle-learning. This algorithm block samples \(\mathbf{z}_{1:T}\) without any assumptions on missingness of observations. Let \(o^n_t\) be the “observation indicator” so that \(o^n_t = 1\) if \(x^n_t\) is observed, and \(0\) if it is missing \((n = 1, 2, \ldots, N\) and \(t = 1, 2, \ldots, T)\). Let \(J > 0\) be the number of particles. Since we will be simulating missing values over the course of inference, we superscript all data with \(j\) to indicate the inclusion of any imputed values by particle \(j\).

1. Set \(w^j \leftarrow 1\) for \(j = 1, 2, \ldots, J\)

2. Repeat for \(t = 1, 2, \ldots, T\):

2.1. Repeat for \(j = 1, 2, \ldots, J\)

2.1.1. Sample \(z^j_t\) from the multinomial distribution:

\[
\Pr[z^j_t = k | \mathbf{z}^{1:N,j}_{1:t-1}, \mathbf{x}^{1:N,j} \setminus o_{t-1}^{j}] \propto \text{CRP}(k|\alpha, \mathbf{z}^{1:N,j}_{1:t-1}) \prod_{n=1}^{N} G(x^n_t | D^n_{T_k} \setminus \{x^n_t\}, \lambda^n_G, \alpha) \prod_{n=1}^{N} \left(F(x^n_t | D^n_{T_k}, \lambda^n_F)\right)^{o^n_t},
\]

where \(k = 1, 2, \ldots, \max(\mathbf{z}^{1:N,j}_{1:t-1}) + 1\).

2.1.2. Update particle weight using predictive density \(q_t\) defined in (15):

\[
w^j \leftarrow w^j q_t(\mathbf{x}^{1:N,j}_{1:t-1} \cup \{x^n_t\}, \mathbf{z}^{1:N,j}_{1:t-1}).
\]

2.1.3. For each \(n\) such that \(o^n_t = 0\), simulate a value \(x^n_{t} \sim F(\cdot | D^n_{t-1}, \lambda^n_F)\).

2.2. If resampling criterion met, then:

2.2.1. Resample \((\mathbf{z}^{1:N,j}_{1:t}, \mathbf{x}^{1:N,j})\) proportionally to \(w^j\), \(j = 1, 2, \ldots, J\).

2.2.2. Renormalize weights \(w^j \leftarrow w^j / \sum_{j'} w^{j'}\), \(j = 1, 2, \ldots, J\).

3. Resample \(j \sim \text{Categorical}(w^1, \ldots, w^J)\) and return \((\mathbf{z}^{1:N,j}_{1:T}, \mathbf{x}^{1:N,j})\).
It is worth discussing the computational trade-offs between MH Algorithm 1 and SMC Algorithm 2. The key computational overhead in the MH algorithm is computing the acceptance ratio (17) in Step 2, which requires revisiting $O(T)$ data points (where we are treating the number of time series $N$ and window size $p$ as constants). Therefore, a full pass through all $T$ observations $x_{1:T}^n$ requires $O(T^2)$ computations. Note that it is not necessary to sum over $K_t$ in (14) when computing the $b_i$ terms in (17), since the data in at most two clusters will change when proposing $z_t$ to $z_{t'}$. The sufficient statistics can be updated in constant time using a simple dynamic programming approach.

In practice, we consider several computational approximations that simplify the scaling properties of the single-site MH Algorithm 1. For missing data, rather than evaluate the full model likelihood (13) for a single TC-CRP mixture model, with latent sequence $z_{1:T}$, data $x_{1:T}^n$, and CRP concentration $\alpha_m$. Second, let $A^m = \{n \mid c^n = m\}$ be the indices of the time series currently assigned to cluster $m$.

For notational simplicity, let $B \subseteq [N]$ and define:

$$L^m(z_{1:T}, x_{1:T}^B) = \prod_{t=1}^T b_t \text{CRP}(z_t \mid z_{1:t-1}, \alpha^m)$$

$$\prod_{n=1}^N G(x_{t-p:t-1}^n; D^m_{it}, \lambda^n) F(x_1^n \mid D^m_{it}, \lambda^n) .$$

(21)

$L^m$ is a short-hand for the product from $t = 1$ to $T$ in the full model likelihood (13) for a single TC-CRP mixture model with latent sequence $z_{1:T}$, data $x_{1:T}^n$, and CRP concentration $\alpha_m$. Second, let $A^m = \{n \mid c^n = m\}$ be the indices of the time series currently assigned to cluster $m$.

Algorithm 3: Sampling time series cluster assignments. Suppose the current state of the Markov chain is $(a_0, c^{1:N}, \alpha^{1:M}, \lambda^1, \lambda^1_F, \lambda^1_B, \lambda^1_T)$ with observations $x_{1:T}^n$. This algorithm resamples $(c^1, c^{1:N} \backslash n, \ldots)$. Repeat for $n = 1, 2, \ldots, N$:

1. If $c^n$ is not a singleton cluster, i.e. $|A^c^n| > 1$, then generate a proposal sequence by forward sampling $z_{1:T}^{M+1}$ from model prior (12), holding the data $x_{1:T}^n$ fixed at the observed values.

2. If $c^n$ is a singleton, i.e. $|A^c^n| = 1$, then re-use the current latent regime sequence by setting $z_{1:T}^{M+1} = z_{1:T}^n$.

3. For $m \in \text{unique}(c^{1:N} \backslash n)$, compute

$$p^m = \begin{cases} |A^m| L^m(z_{1:T}^m, x_{1:T}^n) & \text{if } c^n \neq m, \\ |A^m| - 1 & \text{if } c^n = m. \end{cases}$$

4. Compute the singleton proposal probability:

$$p^M = a_0 L_{M+1}(z_{1:T}^{M+1}, x_{1:T}^n)$$

5. Sample $c' \sim \text{Categorical} \{p^m\}$.

6. Compute the MH acceptance ratio

$$r(c^n \rightarrow c') = \frac{L_{c'}(z_{1:T}^{c'}, x_{1:T}^n \cup x_{1:T}^n) L^{c'}(z_{1:T}^n, x_{1:T}^n) L^{c'}(z_{1:T}^{c'}, x_{1:T}^n)}{L_{c'}(z_{1:T}^{c'}, x_{1:T}^n) L^{c'}(z_{1:T}^n, x_{1:T}^n)} .$$

(22)

7. Set $c^n \leftarrow c'$ with probability $\min(1, r)$, otherwise leave $c^n$ unchanged.
By proposing the latent regime singleton from the (conditional) prior in Step 2 of Algorithm 3, trans-dimensional adjustments [10] need not be considered. Second, when computing the MH acceptance ratio (22) in Step 6, it is not necessary to recompute all the \( L^n \) terms at each iteration. First, writing out the full products (21) results in cancellation of several terms in the numerator and denominator of (22). Second the \( b^n \) terms that do not cancel contain several duplicated components, which can be reused from one transition to the other.

In practice, we find that a similar heuristic to the one described for Algorithm 1 provides good transitions in the state space, given the similarities between Algorithm 3 and the Gibbs Algorithm 8 from [21].

**Inference on model hyperparameters**

\( \{\alpha_0,\{\alpha^n\},\{\lambda^n_1\},\{\lambda^n_p\} \ldots \} \). This section describes an empirical Bayes approach [25] for transitioning the model hyperparameters using the “griddy Gibbs” approach from [24]. For each hyperparameter, we construct a grid of 30 data-dependent logarithmically-spaced bins as follows:

- **Outer CRP concentration**
  
  \[ \text{grid}(\alpha_0) = \text{logspace}(1/N, N) \]

- **TCRCP concentration**
  
  \[ \text{grid}(\alpha^n) = \text{logspace}(1/T, T) \]

- **Normal-InverseGamma hyperparameters**
  
  \[
  \begin{align*}
  \text{grid}(m^n_0) &= \text{logspace}(\min(x^n_{1:T}) - 5, \max(x^n_{1:T}) + 5) \\
  \text{grid}(V^n_0) &= \text{logspace}(1/T, T) \\
  \text{grid}(a^n_0) &= \text{logspace}(\text{ssqdev}(x^n_{1:T})/100, \text{ssqdev}(x^n_{1:T})) \\
  \text{grid}(b^n_0) &= \text{logspace}(1, T).
  \end{align*}
  \]

Grids for the Normal-InverseGamma hyperparameters apply to both \( \lambda_F \) (\( n = 1, 2, \ldots, N \)) and \( \lambda_C \) (windows \( i = 1, 2, \ldots, p \)). We cycle through the grid points of each hyperparameter, and assess the conditional likelihood at each bin using Eq (7). We find that this method is both computationally reasonable and finds good hyperparameter settings. However, alternative approaches based on slice sampling offer a promising alternative to achieve fully Bayesian inference over hyperparameters.

**C Experimental Methods**

This section describes the experimental methods used for forecasting, clustering, and imputation pipelines from Section 5.

The full CDC flu datasets used in this paper are available at [https://github.com/GaloisInc/ppaml-cp7/tree/master/data](https://github.com/GaloisInc/ppaml-cp7/tree/master/data). Flu populations were constructed from the files `USA-flu.csv`, `USA-tweets.csv`, and `USA-weather.csv`.

### C.1 Flu forecasting

In each of US Regions 1 through 10, we held out data from weeks 2014.40 through 2015.20, and produced forecasts with a 10 week horizon on a rolling basis. Tweet and minimum temperature covariates were used. More precisely, for a population \( p \) (such as US Region 10) a forecaster for week \( t \) extending \( h \) weeks into the future is a function:

\[
F_{p,t,h} : \{x_{t-2}^{\text{flu}}, x_{t-2}^{\text{cov}} \} \rightarrow \{x_{t+h}^{\text{flu}}\},
\]

which we evaluated at \( t = 2014.40 \) through 2015.20. Note the two week delay in the latest flu data is expressed by only having data up to \( t - 2 \) when forecasting at week \( t \). Second, \( x^{\text{cov}} \) contains arbitrary missing values (see tweets time series in Figure 2b). When forecasting, covariates are only available up to the current week \( t \), not the entire course of the forecast horizon.

Eight forecasting methods were used in the paper, shown in Table 4b. Below are further details on each forecaster:

- **Constant**. This method returns a constant prediction based on the most recently observed flu value \( x_{t-2}^{\text{flu}} \).

- **Linear extrapolation**. This method fits a line through the three most recently observed flu values, \( x_{t-5}^{\text{flu}}, x_{t-3}^{\text{flu}}, x_{t-2}^{\text{flu}} \), and returns predictions by extrapolating the line for \( h \) weeks.

- **GP (SE+PER+WN)**. This method is a Gaussian process whose covariance kernel is a sum of squared exponential, periodic, and white noise components. Hyperparameter inference was conducted using the open-source implementation from [27]. MH sampling on data-dependent hyperparameter grids were run for a burn-in period of 10000 iterations. Predictions were obtained by drawing 500 independent curves from the posterior predictive distribution evaluated jointly at the forecast weeks.

- **GP (SE×PER+WN)**. As above, using a covariance kernel with a product of squared exponential and periodic components, plus white noise. The change in covariance kernel resulted in little quantitative and qualitative differences.

**Facebook Prophet**. We used the open-source python implementation of Facebook Prophet [31] available at [https://facebook.github.io/prophet/](https://facebook.github.io/prophet/). The method requires no tuning, other than providing the data sampling rate which we specified as weekly. The predictor returns point estimates, as well as upper and lower confidence intervals at the held-out weeks.
**Seasonal ARIMA.** We used the open source R implementation of seasonal ARIMA from the forecast package [13], available at https://cran.r-project.org/web/packages/forecast. The model is parameterized as ARIMA\((p, d, q)(P, D, Q)_m\), where \(p\) is the non-seasonal AR order, \(d\) is the non-seasonal differencing, \(q\) is the non-seasonal MA order, \(P\) is the seasonal AR order, \(D\) is the seasonal differencing, \(Q\) is the seasonal MA order, and \(m\) is the sampling frequency per period. For each of the 10 flu seasons, we used auto.arima to perform model selection. However, we manually set \(m = 52\) (denoting weekly data) in the time series object, and set \(D = 1\) (denoting 1 flu season per year) and let the program optimize all other parameters. We used non-stepwise grid search, which is significantly slower to fit than stepwise search, but is both more extensive and more appropriate for data with seasonal behavior according to the package documentation. While auto.arima can in principle support covariate data using the \(\text{xreg}\) parameter, we were unable to successfully use \(\text{xreg}\) due to missing data in the matrix of external regressors (tweets and weather). The predictor returns point estimates, as well as upper and lower confidence intervals at the held-out weeks.

**Univariate TCCRP mixture.** This method only considered the flu time series using model (2). We used a window size of \(p = 10\) weeks, and \(S = 64\) parallel MCMC runs with a burn-in period of 5000 iterations. Predictions were obtained by drawing 500 independent curves from the posterior predictive distribution evaluated at the forecast weeks.

**Multivariate TCCRP mixture.** This method only considered flu, weather and tweet time series using the model in Figure 2a. We used a window size of \(p = 10\) weeks, and \(S = 64\) parallel MCMC runs with a burn-in period of 5000 iterations. Missing covariate data was handled by using the approximation given in (20). Note that using the hierarchical structure prior (6) resulted in little to no quantitative difference. The three time series are dependent, which was reflected in their posterior dependence probability (8) being 1 across all 64 chains. Predictions were obtained by sampling 500 independent curves from the posterior predictive distribution evaluated at the forecast weeks.

**C.2 Flu imputation**

We constructed a single population of 10 flu time series for US Regions 1 through 10. Missing data was dropped independently in each time series by removing consecutive windows of length 10 at a rate of 5\%. The full and dropped datasets used for benchmarking are shown in Figure 7. Below are further on details on each of the five imputation methods:

**Mean imputation.** This method returns the per-series mean as the imputed value for each data point.

**Linear interpolation.** This method constructs a straight line between every pair of time points \(t_1 < t_2\) which have at least one missing observation between them. The interpolation method used was pandas.Series.interpolate from the python pandas package at https://pandas.pydata.org.

**Cubic interpolation.** The cubic interpolation routine used was scipy.interpolate.interp1d from the python scipy package at https://scipy.org.

**Amelia II.** We used the R package amelia from http://cran.r-project.org/web/packages/Amelia, which implements the technique described by [12]. As this method performs multiple imputation, we used 100 samples per missing data point. Imputation errors were averaged over the multiple imputations.

**Multivariate TCCRP.** A window of \(p = 10\) weeks was used, with \(S = 64\) parallel MCMC runs and a burn-in period of 5000 iterations. 100 predictive samples from each of the chains were obtained using Eq (9), and imputation errors were averaged over the multiple imputations. Joint imputations of Regions 1 through 10 are are shown Figure 7c.

**C.3 Gapminder clustering**

The clustering results from Figure 6 were obtained by using a TCCRP with a window of \(p = 5\) years. The 9 clusters that are shown were obtained by averaging dependence probabilities over \(S = 60\) posterior samples (using a burn-in period of 5000 iterations), and extracting groups of variables whose dependence probabilities exceeded 80\%. The time series in Figure 6 are linearly rescaled to \(\left[0, 1\right]\) for plotting purposes only.

**D Expanded results on clustering macroeconomic variables**

In addition to clustering GDP series from Figure 6 in Section 5, we applied the multivariate TCCRP prior with hierarchical extension (6) to cluster historical cell phone subscription data. The outcome of the clustering is shown in Figure 8, where we show all 170 time series in the left most figure, along with three representative clusters from one posterior sample. Each cluster corresponds to countries whose change point in cell phone subscribers from zero to non-zero fell in a distinct window: 1985-1995 in cluster 1, 1995-2000 in cluster 2, and 2000-2005 in cluster 3. We also compare renderings of the the pairwise dependence probability matrix with the pairwise cross-correlation matrix. Refer to the caption of Figure 8 for additional details.
Figure 7: Full, missing, and imputed flu time series over eight years in US Regions 1 through 10.
(a) Three posterior clusters in the TCCRP mixture correspond to three non-overlapping change point windows.

(b) Pairwise dependence probability heatmap

(c) Pairwise cross-correlation heatmap

Figure 8: Discovering changepoint patterns in cell phone subscriptions for 170 countries in the Gapminder dataset. (a) The three clusters (extracted from one posterior sample) correspond to three regimes each with non-overlapping change point windows, annotated by red boxes. The representative countries in each cluster have similar adoption times of cell phone technology, a feature which differs across the clusters. (b) and (c) The matrix of dependence probabilities (averaged over 60 posterior samples using Eq (8)) and the matrix of pairwise cross-correlations (bottom) between all pairs 170 time series. Each row and column is a time series, and the color of a cell (a value between 0,1) indicates the posterior dependence probability, resp. cross-correlation coefficient (significant at the 0.05 level with Bonferroni correction). The TCCRP mixture detects more refined dependence structures than those captured by linear statistics.