Infinite Factorial Unbounded-State Hidden Markov Model

Isabel Valera, Francisco J.R. Ruiz, and Fernando Perez-Cruz, *Senior Member, IEEE*

Abstract—There are many scenarios in artificial intelligence, signal processing or medicine, in which a temporal sequence consists of several unknown overlapping independent causes, and we are interested in accurately recovering those canonical causes. Factorial hidden Markov models (FHMMs) present the versatility to provide a good fit to these scenarios. However, in some scenarios, the number of causes or the number of states of the FHMM cannot be known or limited a priori. In this paper, we propose an infinite factorial unbounded-state hidden Markov model (IFUHMM), in which the number of parallel hidden Markovmodels (HMMs) and states in each HMM are potentially unbounded. We rely on a Bayesian nonparametric (BNP) prior over integer-valued matrices, in which the columns represent the Markov chains, the rows the time indexes, and the integers the state for each chain and time instant. First, we extend the existent infinite factorial binary-state HMM to allow for any number of states. Then, we modify this model to allow for an unbounded number of states and derive an MCMC-based inference algorithm that properly deals with the trade-off between the unbounded number of states and chains. We illustrate the performance of our proposed models in the power disaggregation problem.

Index Terms—Time series, Bayesian nonparametrics, hidden Markov models, Gibbs sampling, slice sampling, variational inference, reversible jump Markov chain Monte Carlo

1 Introduction

There are several real-world problems in which an observed temporal sequence can be explained by several unobservable independent causes, and we are interested in describing the latent model that leads to these observations. For example, we might want to distinguish the heartbeat of twins [1], separate the overlapping voices on a single recording [2], or separate the contribution of each device to the total power consumed at a household [3]. In some of these problems, the number of independent causes and the number of states are known or limited to a small range (e.g., babies in a womb), but in others that might not be the case. For instance, the number of active devices in a house might differ by orders of magnitude, and the states used by each device can also be different. Accurate estimation of the specific device-level power consumption avoids instrumenting every individual device with monitoring equipment, and the obtained information can be used to significantly improve the power efficiency of consumers [4], [5]. Furthermore, it allows providing recommendations about their relative efficiency (e.g., a household that consumes more power in heating than the average might need better isolation) and detecting faulty equipment.

Hidden Markov models (HMMs) characterize time varying sequences with a simple yet powerful latent variable model [6]. HMMs have been a major success story in many fields involving complex sequential data, including speech [7] and handwriting [8] recognition, computational molecular biology [9] and natural language processing [10]. In most of these applications, the model topology is determined in advance and the model parameters are estimated by an expectation maximization (EM) procedure [11], whose particularization is also known as the Baum-Welch (or forward-backward) algorithm [12]. However, both the standard estimation procedure and the model definition for HMMs suffer from important limitations as not considering the complexity of the model (making it hard to avoid over or underfitting) and needing to pre-specify the model structure. In [13], the authors proposed an inference algorithm for HMMs based on reversible jump Markov chain Monte Carlo (RJMCMC) techniques [14] to address the model selection problem, which can be used to estimate both the parameters and the number of hidden states of an HMM in a Bayesian framework.

Factorial HMMs (FHMMs) model the observed time series with independent parallel HMMs [15]. These parallel HMMs can be seen as several independent causes affecting the observed time series or, alternatively, as a simplification of a hidden state transition matrix into several smaller transition matrices. However, in many cases we do not know how many causes (HMMs) there are and how many states would be needed in each Markov chain.

Bayesian nonparametric (BNP) models have appeared as a replacement of classical finite-dimensional prior distributions with general stochastic processes, allowing an

- *I. Valera is with the Max Planck Institute for Software Systems at Kaiserslautern, Germany, and the Department of Signal Processing and Communications, University Carlos III in Madrid, Spain. E-mail: ivalera@mpi-sws.org.*
- *F.J.R. Ruiz is with the Department of Computer Science, Columbia University, NY, and the Department of Signal Processing and Communications, University Carlos III in Madrid, Spain. E-mail: f.ruiz@columbia.edu.*
- *F. Perez-Cruz is with the Technical Staff at Bell Labs (Alcatel-Lucent), NJ, and the Department of Signal Processing and Communications, University Carlos III in Madrid, Spain. E-mail: Fernando.Perez-Cruz@Alcatel-Lucent.com.*

Manuscript received 13 June 2013; revised 28 Nov. 2014; accepted 23 Oct. 2015. Date of publication 8 Nov. 2015; date of current version 11 Aug. 2016. Recommended for acceptance by Y. Teh.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2015.2498931

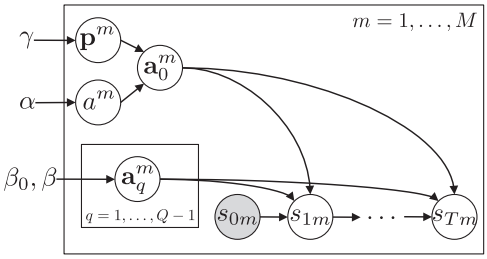


Fig. 1. Graphical model of the nonbinary finite FHMM.

open-ended number of degrees of freedom in a model [16]. In the literature, many nonparametric extensions of standard time series models can be found. The hierarchical Dirichlet process (HDP) has been proposed to define an HMM with an infinite number of latent states called HDP-HMM [17]. This model has been applied to speaker diarization in [2]. The nonparametric extension of the FHMM in [15] is the infinite factorial (binary) HMM (IFHMM) [18], which defines a probability distribution over an unbounded number of binary Markov chains. We can also find a nonparametric hierarchical HMM [19], in which the number of levels in the hierarchy is potentially infinite.

In this paper, we build a BNP generative model to deal with time series, with the capacity of finding behavioral patterns in the data and learning the number of agents from their effects on the observations, e.g., the number of devices that are active in a home. We also infer the state for every agent without limiting the precise number of states in which they can be. Our model can be understood as an IFHMM in which the number of states in each chain is not known or bounded. We hence refer to our model as infinite factorial unbounded-state HMM (IFUHMM). The extension to IFUHMM is not straightforward, as we need to balance the potentially infinite parallel chains with the number of states in each chain. We should not only be able to explain the observations, but doing it in a meaningful way, so that (for instance) the results can help people in power saving by minimizing the power consumption of the most consuming devices.

We construct the IFUHMM in two steps. We first build an FHMM in which the number of states, Q, is a random variable drawn from an infinite discrete probability distribution. Then, an unbounded number of parallel Markov chains are generated following a nonbinary Markov Indian buffet process (mIBP), similar to the binary IFHMM in [18]. Hence, we can define a distribution over integer-valued matrices satisfying three properties: 1) the potential number of columns (Markov chains) is unbounded; 2) the number of states in the Markov chains can be arbitrarily large; and, 3) the rows (representing time steps) follow independent Markov processes. We develop an MCMC inference algorithm that allows estimating not only the parameters of the model, but also the number of states and the number of parallel chains of the proposed IFUHMM.

In our experiments with power disaggregation data, we show that the IFHMM in [18] is capable of fitting the observed sequence, as well as our IFUHMM does, but the binary parallel chains do not have direct interpretation as individual devices and we would need to combine several of them to describe each device, which leads to a complex combinatorial problem in real life scenarios with a large number of causes with many states. Due to the more flexible unbounded prior, our IFUHMM is more generally applicable.

The rest of the paper is organized as follows. In Section 2, we introduce the nonbinary IFHMM with a fixed number of latent states, and a Gaussian observation model is proposed in Section 3. In Section 4, three inference algorithms for this model are developed: two MCMC methods based on Gibbs and slice sampling, and a variational inference algorithm. Section 5 introduces an infinite discrete prior distribution over the number of hidden states and an inference method that allows learning both the number of parallel chains and states. Sections 6 and 7 are respectively devoted to the experiments and conclusions.

2 Nonbinary Infinite Factorial HMM

The model proposed in this section is a nonbinary extension of the IFHMM developed in [18]. The proposed model places a prior distribution over integer-valued matrices with an infinite number of columns (each representing a Markov chain), in which the values of their elements correspond to the labels of the hidden states. Therefore, under this construction, the values of the elements of the matrix are exchangeable. This approach differs from [20], in which the authors propose a prior distribution over integer-valued matrices with an infinite number of columns, but the elements are ordered according to their cardinality.

2.1 Finite Model

We depict the graphical model for a factorial HMM in Fig. 1, in which M, Q and T stand, respectively, for the number of chains, the number of states of the Markov model, and the number of time steps. In this figure, $s_{tm} \in \{0, 1, \ldots, Q-1\}$ represents the hidden state at time instant t in the m-th chain and all the states s_{tm} are grouped together in a $T \times M$ matrix denoted by $\mathbf{S}$. For simplicity, we assume that $s_{0m} = 0$ for all the Markov chains.

For each chain m, the states s_{tm} follow an HMM with transition probabilities contained in the $Q \times Q$ matrix $\mathbf{A}^m$, whose rows are denoted by $\mathbf{a}_q^m$ $(q = 0, \ldots, Q-1)$. Hence, $\mathbf{a}_q^m$ corresponds to the transition probability vector from state q in chain m. Thus, under this model, the transition probability matrices $\mathbf{A}^m$ are independently distributed for each Markov chain $m = 1, \ldots, M$. As the variables s_{tm} follow an HMM, we can write that

$$s_{tm} | s_{(t-1)m}, \mathbf{A}^m \sim \mathbf{a}^m_{s_{(t-1)m}}. \tag{1}$$

In order to be able to extend the number of parallel chains to infinity, and similarly to the IFHMM [18], we need to consider an inactive state. When we let M go to infinity, we have to ensure that for a finite value of T, only a finite subset of the parallel chains become active, while the rest of them remain inactive and do not influence the observations. We consider that the state 0 corresponds to the inactive state and, therefore, $s_{tm} = 0$ indicates that the mth chain is not active at time t. Hence, as shown in Fig. 1, the transition probability vectors $\mathbf{a}_q^m$ are differently distributed for $q = 0$ (inactive state) than for the rest of the states. We place a beta prior over the self-transition probability of the inactive state, i.e.,

$$a^m|\alpha \sim \text{Beta}\Big(1, \frac{\alpha}{M}\Big), \tag{2}$$

and set the transition probability vector from the inactive state to

$$\mathbf{a}_0^m = \Big[a^m(1-a^m)p_1^m \ldots (1-a^m)p_{Q-1}^m\Big], \tag{3}$$

where

$$\mathbf{p}^m|Q,\gamma \sim \text{Dirichlet}(\gamma). \tag{4}$$

Under this construction, the probability distribution over the vector $\mathbf{a}_0^m$ can be easily derived by applying the linear transformation property of random variables from a^m and $\mathbf{p}^m$ to $\mathbf{a}_0^m$, yielding

$$\begin{aligned} p\big(\mathbf{a}_0^m|Q,\alpha,\gamma\big) &= p\big(a_{00}^m|\alpha\big)p\big(a_{01}^m,\ldots,a_{0(Q-1)}^m|a_{00}^m,\gamma\big) \\ &= \text{Beta}\Big(a_{00}^m\Big|1,\frac{\alpha}{M}\Big)\big(1-a_{00}^m\big)^{2-Q} \\ &\quad \times \text{Dirichlet}\left(\frac{a_{01}^m}{1-a_{00}^m},\ldots,\frac{a_{0(Q-1)}^m}{1-a_{00}^m}\middle|\gamma\right), \end{aligned} \tag{5}$$

where the elements of vector $\mathbf{a}_0^m$ are denoted by a_{0i}^m, for $i = 0,\ldots,Q-1$. In Eqs. (2) and (4), α is the concentration parameter, which controls the probability of leaving state 0, and γ incorporates *a priori* knowledge about the transition probabilities from the inactive state to any other state (i.e., $1,\ldots,Q-1$).

For the active states ($q = 1,\ldots,Q-1$), the transition probability vectors are distributed as

$$\mathbf{a}_q^m|Q,\beta_0,\beta \sim \text{Dirichlet}(\beta_0,\beta,\ldots,\beta), \tag{6}$$

where β_0 and β model the *a priori* information about the transition probabilities from states other than 0.

Similarly to the binary mIBP in [18], we can obtain the probability distribution over the matrix $\mathbf{S}$ after integrating out the transition probabilities, yielding the expression in (7), where elements of vector $\mathbf{a}_q^m$ are denoted by a_{qi}^m, containing the probability of transitioning from state q to state i in the Markov chain m. Additionally, n_{qi}^m counts the number of transitions from state q to state i in chain m, and $n_{q\bullet}^m$ represents the number of transitions from state q to any other state in chain m, namely, $n_{q\bullet}^m = \sum_{i=0}^{Q-1} n_{qi}^m$.

$$\begin{aligned} p(\mathbf{S}|Q,\alpha,\beta_0,\beta,\gamma) &= \int p(\mathbf{S}|\{\mathbf{A}^m\}_{m=1}^M)\prod_{m=1}^M (p(\mathbf{A}^m|Q,\alpha,\beta_0,\beta,\gamma)d\mathbf{A}^m) \\ &= \prod_{m=1}^M \left[\frac{\frac{\alpha}{M}\Gamma((Q-1)\gamma)}{(\Gamma(\gamma))^{Q-1}}\frac{\prod_{i=1}^{Q-1}\Gamma(n_{0i}^m+\gamma)}{\Gamma\Big(\sum_{i=1}^{Q-1}(n_{0i}^m+\gamma)\Big)}\frac{\Gamma(n_{00}^m+1)\Gamma\Big(\frac{\alpha}{M}+\sum_{i=1}^{Q-1}n_{0i}^m\Big)}{\Gamma\Big(n_{0\bullet}^m+1+\frac{\alpha}{M}\Big)}\right. \\ &\quad \left.\times\prod_{q=1}^{Q-1}\left(\frac{\Gamma(\beta_0+(Q-1)\beta)}{\Gamma(\beta_0)(\Gamma(\beta))^{Q-1}}\frac{\Gamma(n_{q0}^m+\beta_0)\prod_{i=1}^{Q-1}\Gamma(n_{qi}^m+\beta)}{\Gamma\Big(n_{q\bullet}^m+\beta_0+(Q-1)\beta\Big)}\right)\right]. \end{aligned} \tag{7}$$

2.2 Taking the Infinite Limit

As the number of independent Markov chains M tends to infinity, the probability of a single matrix $\mathbf{S}$ in Eq. (7) vanishes in this model. This is not a limitation, since we are not interested in the probability of a single matrix, but in the probability of the whole equivalence class of $\mathbf{S}$. Similarly to the results for the IBP in [21], the equivalence classes are defined with respect to a function on integer-valued matrices, called $lof(\cdot)$ (left-ordered form). In particular, $lof(\mathbf{S})$ is obtained by sorting the columns of the matrix $\mathbf{S}$ from left to right by the history of that column, which is defined as the magnitude of the base-Q number expressed by that column, taking the first row as the most significant value.

Additionally, since the elements of matrix $\mathbf{S}$ can be arbitrarily relabeled, we can also define a permutation function on the labels of the states in $\mathbf{S}$. Specifically, we say that two matrices $\mathbf{S}_1$ and $\mathbf{S}_2$ with elements in $\{0,\ldots,Q-1\}$ are in the same equivalence class if there exists a permutation function $f(\cdot)$ on the set $\{0,\ldots,Q-1\}$, subject to $f(0)=0$, such that, when applied to all the elements of $\mathbf{S}_2$ to obtain $\mathbf{S}_2'$, $lof(\mathbf{S}_1) = lof(\mathbf{S}_2')$. Roughly, two matrices are equivalent if they are equal after a particular reordering of their columns and/or relabeling of their nonzero elements. Note that the element 0 cannot be relabeled, since it represents the inactive state and therefore requires special treatment, as detailed earlier.

Let us denote by $[\mathbf{S}]$ the set of equivalent matrices to $\mathbf{S}$ as defined above. There are $\frac{(Q-1)!}{(Q-N_Q)!N_f}\frac{M!}{\prod_{h=0}^{Q^T-1}M_h!}$ matrices in this set, with M_h being the number of columns with history h, N_Q being the number of visited states in $\mathbf{S}$, including 0, and where N_f is the number of (previously defined)

$$\begin{aligned} \lim_{M\to\infty} p([\mathbf{S}]|Q,\alpha,\beta_0,\beta,\gamma) &= \lim_{M\to\infty}\frac{(Q-1)!}{(Q-N_Q)!N_f}\frac{M!}{\prod_{h=0}^{Q^T-1}M_h!}p(\mathbf{S}|Q,\alpha,\beta_0,\beta,\gamma) = \frac{(Q-1)!}{(Q-N_Q)!N_f}\frac{\alpha^{M_+}}{\prod_{h=1}^{Q^T-1}M_h!}e^{-\alpha H_T} \\ &\times\prod_{m=1}^{M_+}\left[\frac{\Gamma(n_{00}^m+1)\Gamma\Big(\sum_{i=1}^{Q-1}n_{0i}^m\Big)}{\Gamma(n_{0\bullet}^m+1)}\frac{\Gamma((Q-1)\gamma)\prod_{i=1}^{Q-1}\Gamma(n_{0i}^m+\gamma)}{\Gamma\Big(\sum_{i=1}^{Q-1}(n_{0i}^m+\gamma)\Big)(\Gamma(\gamma))^{Q-1}}\prod_{q=1}^{Q-1}\left(\frac{\Gamma(\beta_0+(Q-1)\beta)}{\Gamma(\beta_0)(\Gamma(\beta))^{Q-1}}\frac{\Gamma(n_{q0}^m+\beta_0)\prod_{i=1}^{Q-1}\Gamma(n_{qi}^m+\beta)}{\Gamma\Big(n_{q\bullet}^m+\beta_0+(Q-1)\beta\Big)}\right)\right]. \end{aligned} \tag{8}$$

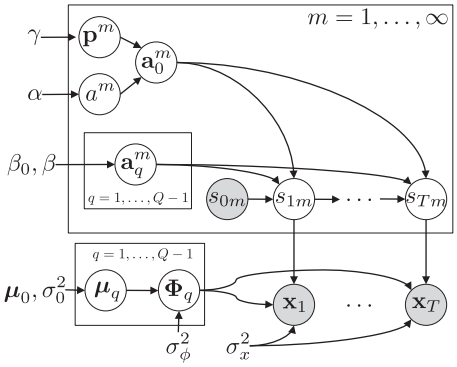


Fig. 2. Graphical observation model for the nonbinary infinite factorial HMM.

permutation functions $f(\cdot)$ such that, when applied to all the elements of $\mathbf{S}$ to obtain $\mathbf{S}'$, $lof(\mathbf{S}) = lof(\mathbf{S}')$. Since all the matrices in $[\mathbf{S}]$ have the same probability, we can easily compute $p([\mathbf{S}]|Q, \alpha, \beta_0, \beta, \gamma)$. Taking the limit as M tends to infinity, we reach Eq. (8), where M_+ stands for the number of nonzero columns, and H_T for the Tth harmonic number, i.e., $H_T = \sum_{j=1}^{T} \frac{1}{j}$.

This model is exchangeable in the columns, in the integer labels used to denote the elements of $\mathbf{S}$, and it is also Markov exchangeable in the rows. The Markov exchangeability property holds because $p([\mathbf{S}]|Q, \alpha, \beta_0, \beta, \gamma)$ only depends on the number of transitions among states n_{qi}^m, and not on the particular sequence of states. We recover the binary mIBP in [18] by setting $Q = 2$. Similarly to the IFHMM in [18], we describe a culinary metaphor analogous to the IBP in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2015.2498931.

2.3 Stick-Breaking Construction

Since the representation of the model above is similar to the binary mIBP in [18], a stick-breaking construction is also readily available. This construction allows using a combination of slice sampling and dynamic programming for inference, as detailed in Section 4.2.

The stick-breaking construction requires defining a distribution over the parameters corresponding to the transition probabilities a^m sorted in ascending order, namely, $a^{(m)}$. For convenience, we define the complementary probabilities $c^{(m)} = 1 - a^{m'}$, such that $c^{(1)} > c^{(2)} > \ldots$ Hence, following a similar procedure as in the stick breaking construction of the standard IBP in [22], we can write

$$p(c^{(1)}) = \text{Beta}(\alpha, 1), \tag{9}$$

and

$$p(c^{(m)}|c^{(m-1)}) \propto (c^{(m)})^{\alpha-1}\mathbb{I}(0 \leq c^{(m)} \leq c^{(m-1)}), \tag{10}$$

where $\mathbb{I}(\cdot)$ is the indicator function, which takes value one if its argument is true and zero otherwise.

Let $\mathbf{a}_q^{(m)}$ and $\mathbf{p}^{(m)}$ be the variables corresponding to, respectively, $\mathbf{a}_q^{m'}$ and $\mathbf{p}^{m'}$ sorted by chains according to the values of $a^{m'}$. Then, since $\mathbf{a}_q^{m'}$ and $\mathbf{p}^{m'}$ follow the distributions in Eqs. (6) and (4), respectively, which are independent of m', the sorted variables $\mathbf{a}_q^{(m)}$ and $\mathbf{p}^{(m)}$ have also the same prior distributions.

3 Gaussian Observation Model

We use the nonbinary mIBP as a building block for a full probabilistic model, in which $\mathbf{S}$ can be interpreted as an arbitrarily large set of parallel Markov chains. We add a likelihood model which describes the distribution over the $T \times D$ observation matrix $\mathbf{X}$, composed of T vectors $\mathbf{x}_t$ of length D corresponding to the available observations at time instants $t = 1, \ldots, T$. Note that there are three conditions for the likelihood model to be valid as M tends to infinity: i) the likelihood must be invariant to permutations of the Markov chains; ii) it must also be invariant to the particular labeling of the nonzero elements of $\mathbf{S}$; and iii) the distribution on $\mathbf{x}_t$ cannot depend on any parameter of chain m if $s_{tm} = 0$. Roughly, the likelihood must be invariant for any matrix in the set of equivalent classes of $\mathbf{S}$.

Our choice for the likelihood model is shown in Fig. 2, in which $\mathbf{X}$ is distributed as a Gaussian random matrix with independent elements, each with variance σ_x^2, i.e.,

$$p(\mathbf{X}|\mathbf{S}, \boldsymbol{\Phi}_1, \ldots, \boldsymbol{\Phi}_{Q-1}, \sigma_x^2) = \frac{1}{(2\pi\sigma_x^2)^{\frac{TD}{2}}} \exp\left\{ -\frac{1}{2\sigma_x^2} \times \text{trace}\left[\left(\mathbf{X} - \sum_{q=1}^{Q-1} \mathbf{Z}_q \boldsymbol{\Phi}_q \right)^\top \left(\mathbf{X} - \sum_{q=1}^{Q-1} \mathbf{Z}_q \boldsymbol{\Phi}_q \right) \right] \right\}, \tag{11}$$

where $\mathbf{Z}_q$ is defined as a binary $T \times M$ matrix with elements $(\mathbf{Z}_q)_{tm} = 1$ if $s_{tm} = q$ and zero otherwise, and $\boldsymbol{\Phi}_q$ are $M \times D$ matrices, with M being the number of columns in $\mathbf{S}$. Thus, the mean value for $\mathbf{x}_t$ depends on the additive contribution of all chains at time instant t.

We place a Gaussian prior with independent elements over the matrices $\boldsymbol{\Phi}_q$, i.e.,

$$p(\boldsymbol{\Phi}_q|\boldsymbol{\mu}_q, \sigma_\phi^2) = \frac{1}{\left(2\pi\sigma_\phi^2\right)^{\frac{DM}{2}}} \exp\left\{ -\frac{1}{2\sigma_\phi^2} \times \text{trace}\left[(\boldsymbol{\Phi}_q - \mathbf{1}_M \boldsymbol{\mu}_q)^\top (\boldsymbol{\Phi}_q - \mathbf{1}_M \boldsymbol{\mu}_q) \right] \right\}, \tag{12}$$

where $\mathbf{1}_M$ represents a column vector of length M with all elements equal to one and $\boldsymbol{\mu}_q$ are D-dimensional Gaussian distributed row vectors with mean $\boldsymbol{\mu}_0$ and covariance matrix $\sigma_0^2\mathbf{I}_D$, i.e., $p(\boldsymbol{\mu}_q|\sigma_0^2) = \mathcal{N}(\boldsymbol{\mu}_0, \sigma_0^2\mathbf{I}_D)$, where $\mathbf{I}_D$ stands for the identity matrix of size D. We include the hyperparameter σ_ϕ^2 to control the variance of the parameters corresponding to different chains within every state q $(q = 1, \ldots, Q-1)$. For a small value of σ_ϕ^2/σ_0^2, $\boldsymbol{\Phi}_q$ is close to its mean and therefore the parameters for any particular state q are similar through all the chains. For larger values of σ_ϕ^2/σ_0^2, the parameters may seem decorrelated for the same state at different chains.

4 Inference

Inference in BNP models is typically addressed by Markov chain Monte Carlo (MCMC) methods, such as Gibbs

sampling [2], [17] or beam sampling [23]. Additionally, variational inference has appeared as a complementary alternative to MCMC methods as a general source of approximation methods for inference in large-scale statistical models [15], [24], [25]. In the spirit of describing a general learning algorithm, we have developed both MCMC and variational inference algorithms, as they have different properties.

First, we put forward two MCMC methods: one consists of Gibbs sampling and the other is a blocked sampler based on a forward-filtering backward-sampling algorithm. Second, we propose a variational inference algorithm, which can be viewed as a combination of the main ideas from the finite variational approach for the IBP in [24] and the variational inference proposed for infinite HMMs in [25].

4.1 Gibbs Sampling

MCMC methods have been broadly applied to infer the latent structure $\mathbf{S}$ from a given observation matrix $\mathbf{X}$ (see, e.g., [18], [21]). We focus on Gibbs sampling for posterior inference over the mIBP matrix. The algorithm iteratively samples the value of each element s_{tm} given the remaining variables, i.e., it samples from

$$p(s_{tm} = k|\mathbf{X}, \mathbf{S}_{\neg tm}) \propto p(s_{tm} = k|\mathbf{S}_{\neg tm})p(\mathbf{X}|\mathbf{S}), \tag{13}$$

where $\mathbf{S}_{\neg tm}$ represents the matrix $\mathbf{S}$ without the element s_{tm}. For clarity, throughout this subsection we drop the dependence on the hyperparameters in the notation.

Hence, for $t = 1, \ldots, T$, the Gibbs sampler proceeds as follows:

1) For $m = 1, \ldots, M_+$, sample element s_{tm} from (13). Then, if the mth chain remains inactive for all the time instants, remove that chain and update M_+.
2) Draw M_{new} columns of $\mathbf{S}$ with states s_{tm} ($m = M_+ + 1, \ldots, M_+ + M_{new}$) from a distribution where the prior is $\text{Poisson}(M_{new}|\frac{\alpha}{T}) \times \frac{1}{(Q-1)^{M_{new}}}$, and update M_+. For each value of M_{new}, we try all the possible states in which the new chains can be at time t, and we restrict the possible values of M_{new} to a finite set (as in [21]).

For conciseness, let us denote the previous and the following states to s_{tm} as $j = s_{(t-1)m}$ and $\ell = s_{(t+1)m}$, respectively. Hence, we can compute the first term in Eq. (13) as detailed in Appendix B, available in the online supplemental material.

In order to compute the second term in Eq. (13), we first need to integrate out $\boldsymbol{\mu}_q$ as

$$\begin{aligned} p(\boldsymbol{\Phi}_q|\mathbf{S}) &= \int p(\boldsymbol{\Phi}_q|\mathbf{S}, \boldsymbol{\mu}_q)p(\boldsymbol{\mu}_q)d\boldsymbol{\mu}_q \\ &= \frac{1}{(2\pi)^{DM_+/2}\sigma_\phi^{(M_+-1)D}(\sigma_0^2 M_+ + \sigma_\phi^2)^{D/2}} \\ &\quad \times \exp\left\{-\frac{1}{2\sigma_\phi^2}\text{trace}\Big[(\boldsymbol{\Phi}_q - \mathbf{M}_\Phi)^\top \boldsymbol{\Sigma}_\Phi^{-1}(\boldsymbol{\Phi}_q - \mathbf{M}_\Phi)\Big]\right\}, \end{aligned} \tag{14}$$

where $\boldsymbol{\Sigma}_\Phi = (\mathbf{I}_{M_+} - \frac{\sigma_0^2}{\sigma_0^2 M_+ + \sigma_\phi^2}\mathbf{1}_{M_+}\mathbf{1}_{M_+}^\top)^{-1}$ and $\mathbf{M}_\Phi = \frac{\sigma_\phi^2}{\sigma_0^2 M_+ + \sigma_\phi^2}\boldsymbol{\Sigma}_\Phi \mathbf{1}_{M_+}\boldsymbol{\mu}_0$. Then, $p(\mathbf{X}|\mathbf{S})$ can be computed integrating out all matrices $\boldsymbol{\Phi}_q$, yielding

$$\begin{aligned} p(\mathbf{X}|\mathbf{S}) &= \frac{1}{(2\pi\sigma_x^2)^{TD/2}|\boldsymbol{\Sigma}_{Q-1}|^{D/2}} \\ &\quad \times \exp\left\{-\frac{1}{2\sigma_x^2}\text{trace}\Big[(\mathbf{X} - \mathbf{M}_X)^\top \boldsymbol{\Sigma}_{Q-1}^{-1}(\mathbf{X} - \mathbf{M}_X)\Big]\right\}, \end{aligned} \tag{15}$$

being $\mathbf{M}_X = \sum_{q=1}^{Q-1}\boldsymbol{\Sigma}_q\mathbf{M}_q$, and where the $T \times T$ matrix $\boldsymbol{\Sigma}_{Q-1}^{-1}$ and the $T \times D$ matrices $\mathbf{M}_q$ can be iteratively computed as

$$\boldsymbol{\Sigma}_q^{-1} = \boldsymbol{\Sigma}_{q-1}^{-1} - \boldsymbol{\Sigma}_{q-1}^{-1}\mathbf{Z}_q\mathbf{W}_q\mathbf{Z}_q^\top\boldsymbol{\Sigma}_{q-1}^{-1} \tag{16}$$

and

$$\mathbf{M}_q = \frac{\sigma_x^2}{\sigma_0^2 M_+ + \sigma_\phi^2}\boldsymbol{\Sigma}_{q-1}^{-1}\mathbf{Z}_q\mathbf{W}_q\mathbf{1}_{M_+}\boldsymbol{\mu}_0, \tag{17}$$

with $\mathbf{W}_q$ given by

$$\mathbf{W}_q^{-1} = \mathbf{Z}_q^\top\boldsymbol{\Sigma}_{q-1}^{-1}\mathbf{Z}_q + \frac{\sigma_x^2}{\sigma_\phi^2}\boldsymbol{\Sigma}_\Phi^{-1}, \tag{18}$$

for $q = 1, \ldots, Q-1$. For the first iteration, $\boldsymbol{\Sigma}_0$ is the identity matrix of size M_+.

4.2 Blocked Sampling

It is common knowledge that Gibbs sampling may present slow mixing when applied to time series models, due to potentially strong couplings between successive time steps [18], [26]. A typical approach to circumvent this limitation consists on blocked sampling the latent states s_{tm} for each chain, i.e., sampling a whole Markov chain using a forward-filtering backward-sampling algorithm, conditional on keeping all other Markov chains fixed. In order to apply this dynamic programming step, we also need a slice sampling algorithm [27] which adaptively truncates our model into a finite FHMM, performing exact inference without assuming alternative approximate models [18], [23].

Here, we make use of the stick-breaking construction of the model, presented in Section 2.3, and introduce an auxiliary slice variable ϑ distributed as

$$\vartheta|\mathbf{S}, \{c^{(m)}\} \sim \text{Uniform}\Big(0, \min_{m:\exists t, s_{tm}\neq 0} c^{(m)}\Big), \tag{19}$$

resulting in the joint distribution

$$\begin{aligned} &p(\vartheta, \mathbf{S}, \{c^{(m)}, \mathbf{p}^{(m)}, \mathbf{a}_q^{(m)}\}) \\ &= p(\vartheta|\mathbf{S}, \{c^{(m)}\})p(\mathbf{S}, \{c^{(m)}, \mathbf{p}^{(m)}, \mathbf{a}_q^{(m)}\}). \end{aligned} \tag{20}$$

Again, the dependence on the hyperparameters has been dropped in the notation.

From (20), it is clear that the original model has not been altered, since it can be recovered after integrating out the slice variable. However, when we condition the posterior over $\mathbf{S}$ on ϑ, we have that

$$\begin{aligned} &p(\mathbf{S}|\mathbf{X}, \vartheta, \{c^{(m)}, \mathbf{p}^{(m)}, \mathbf{a}_q^{(m)}\}) \\ &\propto p(\vartheta|\mathbf{S}, \{c^{(m)}\})p(\mathbf{S}|\mathbf{X}, \{c^{(m)}, \mathbf{p}^{(m)}, \mathbf{a}_q^{(m)}\}), \end{aligned} \tag{21}$$

which forces all columns of $\mathbf{S}$ for which $c^{(m)} < \vartheta$ to be zero. Our model ensures that there can only be a finite number of columns for which $c^{(m)} > \vartheta$ and, therefore, conditioning on

the slice variable effectively truncates the model into a finite FHMM. Note that the distribution in Eq. (19) does not need to be uniform, and a flexible Beta distribution can be used instead [18].

Unlike the Gibbs sampler, the blocked sampling algorithm does not allow us to integrate out the matrices $\mathbf{\Phi}_q$, and they have to be sampled from the corresponding Gaussian posterior distributions. The variables $\boldsymbol{\mu}_q$ can still be integrated out. Hence, the blocked sampling algorithm iteratively applies these steps:

1) Sample the slice variable ϑ from (19). This step may also involve adding new chains.
2) For each represented chain m, sample the mth column of $\mathbf{S}$ via dynamic programming. Compact the representation by removing all chains in the all zero state.
3) For each active chain,[1] sample $c^{(m)}$, $\mathbf{p}^{(m)}$ and $\{\mathbf{a}_q^{(m)}\}$.
4) Sample the matrices $\mathbf{\Phi}_q$.

In Step 1, ϑ is first sampled from (19). Then, starting from $m = M_+ + 1$, new variables $c^{(m)}$ are iteratively sampled from

$$p(c^{(m)}|c^{(m-1)}) \propto \exp\left(\alpha \sum_{t=1}^{T} \frac{1}{t}(1-c^{(m)})^t\right) \times (c^{(m)})^{\alpha-1}(1-c^{(m)})^T \mathbb{I}(0 \leq c^{(m)} \leq c^{(m-1)}) \tag{22}$$

until $c^{(m)} < \vartheta$. Since Eq. (22) is log-concave in $\log c^{(m)}$ [22], we can apply adaptive rejection sampling (ARS) [28]. Let M_{new} be the number of new variables $c^{(m)}$ that are greater than the slice variable. If $M_{new} > 0$, then we update M_+, expand the representation of matrix $\mathbf{S}$ by adding M_{new} zero columns, and we sample the values of the new rows of matrices $\mathbf{\Phi}_q$ from the corresponding Gaussian conditional distributions, given the rest of rows of matrices $\mathbf{\Phi}_q$. For each new chain, we also draw the new variables $\mathbf{p}^{(m)}$ and $\{\mathbf{a}_q^{(m)}\}_{q=1}^{Q-1}$ from the prior.

Step 2 consists on a blocked sampler, which runs a forward-filtering backward-sampling sweep on one column of $\mathbf{S}$, having fixed the rest of columns [18].

In Step 3, for each chain, $c^{(m)}$ is sampled from [22]

$$p(c^{(m)}|\mathbf{S}, c^{(m-1)}, c^{(m+1)}) \propto (c^{(m)})^{T-n_{00}^{(m)}-1} \times (1-c^{(m)})^{n_{00}^{(m)}} \mathbb{I}(c^{(m+1)} \leq c^{(m)} \leq c^{(m-1)}), \tag{23}$$

while the posteriors for $\mathbf{p}^{(m)}$ and $\mathbf{a}_q^{(m)}$ (given $\mathbf{S}$) are, respectively, Dirichlet distributions with parameters

$$\gamma + n_{01}^{(m)}, \ldots, \gamma + n_{0(Q-1)}^{(m)},$$

and

$$\beta_0 + n_{q0}^{(m)}, \beta + n_{q1}^{(m)}, \ldots, \beta + n_{q(Q-1)}^{(m)},$$

where we denote by $n_{qi}^{(m)}$ the number of transitions from state q to state i in the mth chain, considering the ordering given by the stick-breaking construction.

1. An active chain is a chain in which not all states are zero.

In Step 4, all matrices $\mathbf{\Phi}_q$ can be simultaneously sampled from the corresponding Gaussian posterior distribution given $\mathbf{S}$ and $\mathbf{X}$.

4.3 Variational Inference

Variational inference provides a complementary alternative to MCMC methods as a general source of approximation methods for inference in large-scale statistical models [29]. Variational inference algorithms are in general computationally less expensive compared to MCMC methods, but they involve solving a non convex optimization problem, which implies that the algorithm may get trapped in local optima.

HMM-specific variational inference algorithms can be found in [15], [25]. In [25], a variational inference algorithm for the infinite HMM is proposed. In [15] the authors develop several inference algorithms for the standard factorial HMM where they include two variational methods: a completely factorized and a structured variational algorithm. While the former method uses a completely factorized variational distribution to approximate the posterior probability of the model by assuming independence among the state variables, the structured variational method preserves much of the probabilistic structure of the original system by considering the dependencies among the states. Structured variational methods are generally preferred since they allow reducing the number of variational parameters and, therefore, they correspond to coordinate-wise optimization over bigger coordinate blocks than the completely factorized approaches. The structured variational algorithm in [15] also requires a forward-backward algorithm within each Markov chain to implement an efficient and exact inference.

We develop a variational inference algorithm for a finite (and large enough) value of the number of chains, M. Thus, we consider the finite model in Section 2.1. The hyperparameters of the model are gathered in the set $\mathcal{H} = \{Q, \alpha, \gamma, \beta_0, \beta, \sigma_0^2, \sigma_\phi^2, \sigma_x^2, \boldsymbol{\mu}_0\}$ and, similarly, we denote the set of unobserved variables in the model by $\Psi = \{\mathbf{S}, \mathbf{a}_j^m, a^m, \mathbf{p}^m, \mathbf{\Phi}_k, \boldsymbol{\mu}_k\}$, for $j,k = 1, \ldots, Q-1$ and $m = 1, \ldots, M$.

The joint probability distribution over all the variables is given by $p_M(\Psi, \mathbf{X}|\mathcal{H})$, where the subscript M indicates that the probability distribution has been truncated to M Markov chains. From the definition of the model, $p_M(\Psi, \mathbf{X}|\mathcal{H})$ can be factorized as follows:

$$\begin{aligned} p_M(\Psi,\mathbf{X}|\mathcal{H}) = &\left(\prod_{k=1}^{Q-1}\left(p_M(\mathbf{\Phi}_k|\boldsymbol{\mu}_k, \sigma_\phi^2)p_M(\boldsymbol{\mu}_k|\sigma_0^2)\right)\right) \\ &\times\left(\prod_{m=1}^{M}\prod_{t=1}^{T} p_M(s_{tm}|s_{(t-1)m}, \mathbf{A}^m)\right) \\ &\times\left(\prod_{m=1}^{M}\left(\prod_{j=1}^{Q-1} p_M(\mathbf{a}_j^m|Q, \beta_0, \beta)\right)p_M(\mathbf{p}^m|Q,\gamma)p_M(a^m|\alpha)\right) \\ &\times p_M(\mathbf{X}|\mathbf{S}, \mathbf{\Phi}_1, \ldots, \mathbf{\Phi}_{Q-1}). \end{aligned} \tag{24}$$

We approximate $p_M(\Psi|\mathbf{X}, \mathcal{H})$ with the variational distribution $q(\Psi)$ given in Eq. (25), which is completely factorized

except for the state matrix $\mathbf{S}$. We use the structured variational distribution for $q(\mathbf{S})$ developed in [15], which preserves much of the probabilistic structure of the original model while maintaining the tractability of the inference. Thus, the variational distribution can be written as

$$q(\Psi) = q(\mathbf{S})\left(\prod_{k=1}^{Q-1}(q(\mathbf{\Phi}_k)q(\boldsymbol{\mu}_k))\right) \times \left(\prod_{m=1}^{M}\left(q(\mathbf{p}^m)q(a^m)\prod_{j=1}^{Q-1}q\left(\mathbf{a}_j^m\right)\right)\right), \tag{25}$$

being

$$q(\mathbf{S}) = \prod_{m=1}^{M}\frac{1}{Z_Q^m}\prod_{t=1}^{T}q(s_{tm}|s_{(t-1)m}), \tag{26}$$

where Z_Q^m are the constants that ensure that $q(\mathbf{S})$ is properly normalized. The specific form for every term in Eqs. (25) and (26) is given by

$$q(s_{tm} = k|s_{(t-1)m} = j) \propto P_{jk}^m \cdot b_{kt}^m, \tag{27}$$

$$q(\mathbf{\Phi}_k) = \frac{1}{(2\pi)^{MD/2}|\mathbf{\Lambda}_k|^{D/2}} \times \exp\left\{-\frac{1}{2}\mathrm{trace}\left[(\mathbf{\Phi}_k - \mathbf{L}_k)^\top \mathbf{\Lambda}_k^{-1}(\mathbf{\Phi}_k - \mathbf{L}_k)\right]\right\}, \tag{28}$$

$$q(\boldsymbol{\mu}_k) = \mathcal{N}(\boldsymbol{\omega}_k, \mathbf{\Omega}_k), \tag{29}$$

$$q(\mathbf{p}^m) = \mathrm{Dirichlet}\left(\varepsilon_1^m, \ldots, \varepsilon_{Q-1}^m\right), \tag{30}$$

$$q(a^m) = \mathrm{Beta}\left(\nu_1^m, \nu_2^m\right), \text{ and} \tag{31}$$

$$q(\mathbf{a}_j^m) = \mathrm{Dirichlet}\left(\tau_{j0}^m, \ldots, \tau_{j(Q-1)}^m\right). \tag{32}$$

Inference involves optimizing the variational parameters of $q(\Psi)$ to minimize the Kullback-Leibler divergence of $p_M(\Psi|\mathbf{X}, \mathcal{H})$ from $q(\Psi)$, i.e., $D_{KL}(q||p_M)$. This optimization can be performed by iteratively applying the fixed-point set of equations given in Appendix C, available in the online supplemental material.

5 Prior on the Number of States

The model in Section 2, as well as the inference algorithms in Section 4, assumes that the number of states Q in the Markov chains is known. We now deal with the case where Q is unknown and it must also be inferred from the data. Specifically, we develop an MCMC inference method to infer both the number of states Q and the number of parallel chains M_+ that constitute the matrix $\mathbf{S}$.

Let us assume that Q is a random variable and we place a prior over it, e.g., a Poisson distribution with parameter λ, namely,

$$p(Q|\lambda) = \frac{\lambda^{Q-2}e^{-\lambda}}{(Q-2)!}, \qquad Q = 2, \ldots, \infty. \tag{33}$$

As shown in Eq. (8), the probability of the whole equivalent class of the mIBP matrix $\mathbf{S}$, denoted by $[\mathbf{S}]$, is conditioned on the number of states Q. In order to obtain the marginalized (with respect to the number of states Q) probability distribution over $[\mathbf{S}]$, variable Q can be integrated out, yielding

$$p([\mathbf{S}]|\alpha, \beta_0, \beta, \gamma) = \sum_{Q=2}^{\infty} p([\mathbf{S}]|Q, \alpha, \beta_0, \beta, \gamma)p(Q|\lambda). \tag{34}$$

We remark that the term $p([\mathbf{S}]|Q, \alpha, \beta_0, \beta, \gamma)$ vanishes if $\mathbf{S}$ contains any element not included in the set $\{0, \ldots, Q-1\}$.

The summation in Eq. (34) is finite, as the series is convergent. To show this, it suffices to check that

$$\lim_{Q\to\infty}\frac{p([\mathbf{S}]|Q+1, \alpha, \beta_0, \beta, \gamma)p(Q+1|\lambda)}{p([\mathbf{S}]|Q, \alpha, \beta_0, \beta, \gamma)p(Q|\lambda)} < 1. \tag{35}$$

This condition holds since the limit can be simplified[2] to $\lim_{Q\to\infty}\frac{p(Q+1|\lambda)}{p(Q|\lambda)}$, which is less than one for every $\lambda > 0$.

5.1 Inference

Due to the flexibility of the proposed model, the inference algorithm involves a trade-off between the number of chains and the number of states. We need to find out a likely combination of the values of both variables given the observed data through the search of the mIBP matrix $\mathbf{S}$ and value of Q from the joint probability $p([\mathbf{S}], Q|\mathbf{X}, \mathcal{H}')$, where $\mathcal{H}'$ is defined as the set of hyperparameters of the model, i.e., $\mathcal{H}' = \{\alpha, \gamma, \beta_0, \beta, \sigma_0^2, \sigma_\phi^2, \sigma_x^2, \boldsymbol{\mu}_0, \lambda\}$.

We propose an MCMC inference algorithm that obtains samples from the target distribution $p([\mathbf{S}], Q|\mathbf{X}, \mathcal{H}')$. An MCMC method dealing with HMMs can be found in [13], where a reversible jump MCMC (RJMCMC) algorithm is used to estimate not only the parameters of the model, but also the number of states Q of the HMM. RJMCMC methods, which were first introduced in [14] for model selection, allow the sampler to jump between parameter subspaces of differing dimensionality.

The RJMCMC algorithm for HMMs can be almost readily applied to our model to obtain samples from the full posterior $p([\mathbf{S}], Q, \{\mathbf{A}^m\}, \{\boldsymbol{\mu}_q\}, \{\mathbf{\Phi}_q\}|\mathbf{X}, \mathcal{H}')$. Due to the multiplicity of Markov chains and the high dimensionality of the proposed IFHMM, the acceptance probabilities for transdimensional jumps under RJMCMC techniques turns out to be extremely low, which makes convergence too slow to be practical.

Since we can obtain the marginalized distribution $p([\mathbf{S}], Q|\mathbf{X}, \mathcal{H}')$, where dimension-changing variables have been integrated out, RJMCMC methods are not needed and we apply a standard Metropolis-Hastings algorithm instead. Nevertheless, we adapt the procedure in [13] to develop our inference algorithm. Hence, our MCMC sampler proceeds iteratively as follows:

2. Note that, according to Eq. (8), in the limit when Q tends to infinity $\frac{p([\mathbf{S}]|Q+1,\alpha,\beta_0,\beta,\gamma)}{p([\mathbf{S}]|Q,\alpha,\beta_0,\beta,\gamma))} = 1$.

1) Update the allocation matrix $\mathbf{S}$ for a given value of Q.
2) Consider splitting a component into two or merging two into one.
3) Consider the birth of a new state or the death of an empty state (i.e., a state that is not assigned in $\mathbf{S}$).

The number of active parallel Markov chains is updated in the first step, as the nonparametric nature of the model allows the sampler to infer this quantity. The two latter steps allow increasing or decreasing the number of states Q by one.

The first step involves either a sweep of the Gibbs sampler as detailed in Section 4.1, or a sweep of the blocked sampling described in Section 4.2. In the latter case, the transition probabilities and the matrices $\mathbf{\Phi}_q$ must be sampled (Steps 3 and 4 in Section 4.2) before performing Step 1. In our experiments we apply the blocked sampling because it is faster than the Gibbs sampler and also presents better mixing properties.

In the second step, we choose to split with probability b_Q and to merge with probability $d_Q = 1 - b_Q$. Naturally, $d_2 = 0$, and we use $b_Q = d_Q = 1/2$ for $Q = 3, \ldots, \infty$. This procedure is similar to the split/merge move for the Dirichlet process mixture model proposed in [30]. In the merge move, we start from a matrix $\tilde{\mathbf{S}}$ and $Q+1$ states and we randomly select two of the nonzero states, q_1 and q_2, and try to combine them into a single state q_*, thus creating a matrix $\mathbf{S}$ with Q states. In the split move, in which we start from a matrix $\mathbf{S}$ and Q states, a nonzero state q_* is randomly chosen and split into two new ones, q_1 and q_2, ending with a new matrix $\tilde{\mathbf{S}}$ and $Q+1$ states. The acceptance probabilities for the split and merge moves are given by $\min(1, R)$ and $\min(1, R^{-1})$, respectively, where

$$R = \frac{p([\tilde{\mathbf{S}}], Q+1|\mathbf{X}, \mathcal{H}')}{p([\mathbf{S}], Q|\mathbf{X}, \mathcal{H}')} \frac{d_{Q+1} P^d_{select}}{b_Q P^b_{select} P_{alloc}}, \tag{36}$$

which ensures that the detailed balance condition is satisfied. In (36), P^d_{select} denotes the probability of selecting two specific components in the merge move and is given by $2/(Q(Q-1))$, P^b_{select} denotes the probability of selecting a specific component in the split move and is given by $1/(Q-1)$, and P_{alloc} denotes the probability of making the particular allocation of the elements in matrix $\tilde{\mathbf{S}}$. Therefore, P_{alloc} depends on how the elements in $\mathbf{S}$ taking value q_* are split into q_1 and q_2. Although the simplest allocation method could consist on splitting completely at random, other methods can be used to increase the acceptance probability. We choose to apply a restricted Gibbs sampling scheme (as in [30]) for those states in $\mathbf{S}$ taking value q_*. Rearranging and simplifying the factors in Eq. (36), R can be expressed for the split and merge moves as

$$\begin{aligned} R \;&= \frac{p(\mathbf{X}|[\tilde{\mathbf{S}}], \sigma_0^2, \sigma_\phi^2, \sigma_x^2, \boldsymbol{\mu}_0)}{p(\mathbf{X}|[\mathbf{S}], \sigma_0^2, \sigma_\phi^2, \sigma_x^2, \boldsymbol{\mu}_0)} \frac{p([\tilde{\mathbf{S}}]|Q+1, \alpha, \beta_0, \beta, \gamma)}{p([\mathbf{S}]|Q, \alpha, \beta_0, \beta, \gamma)} \\ &\quad \times \frac{p(Q+1|\lambda)}{p(Q|\lambda)} \frac{d_{Q+1} 2/Q}{b_Q P_{alloc}}. \end{aligned} \tag{37}$$

In the third step, we first choose at random between the birth or the death of a state with probabilities b_Q and d_Q, respectively. The removal of a state is accomplished by randomly selecting an empty component and deleting it, thereby jumping from $Q+1$ states to Q. Matrix $\tilde{\mathbf{S}}$ is relabeled so that its elements belong to the set $\{0, \ldots, Q-1\}$, resulting in matrix $\mathbf{S}$. In the birth move, we start from a model with Q states and we want to create a new empty component. Matrix $\mathbf{S}$ is unaltered in this process, i.e., $\tilde{\mathbf{S}} = \mathbf{S}$. The acceptance probabilities for the birth and death moves are $\min(1, R)$ and $\min(1, R^{-1})$, respectively, where in this case R can be simplified as

$$R = \frac{p([\tilde{\mathbf{S}}]|Q+1, \alpha, \beta_0, \beta, \gamma)}{p([\mathbf{S}]|Q, \alpha, \beta_0, \beta, \gamma)} \frac{p(Q+1|\lambda)}{p(Q|\lambda)} \frac{d_{Q+1}}{b_Q(Q_0+1)}, \tag{38}$$

with Q_0 being the number of empty components before the birth of a new empty state. Note that, although the birth and the split moves seem similar, both of them are useful. In the birth step we allow the sampler to create a new empty state (which implicitly involves to have also new observation parameters for this state) that may help to explain data points that could not be explained by the existent states, while in the split move we are explaining the data in more detail by splitting a state into two new states.

Since the detailed balance, irreducibility and aperiodicity properties are satisfied (see [30], [31] for further details), the sampler behaves as desired in terms of converging to a realization from the marginalized posterior distribution $p([\mathbf{S}], Q|\mathbf{X}, \mathcal{H}')$.

6 Experimental Validation

We now validate our IFUHMM and proposed inference algorithm on two real datasets on power disaggregation. To this end, we first design a small scale experiment in which we evaluate the mixing properties of the MCMC-based inference algorithm described in Section 5.1, and compare the results with the binary IFHMM in [18] (i.e., the IFHMM with $Q = 2$ states) and with the standard FHMM. We then evaluate the performance of the IFUHMM in solving the power disaggregation problem under more realistic scenarios.

The power disaggregation problem consists in, given the aggregate whole-home power consumption signal, estimating both the number of active devices in the house and the power draw of each individual device. Recently, this problem has been addressed in [3] by applying a factorial hidden semi-Markov model (HSMM) and using an expectation maximization algorithm, and in [32] using an explicit-duration hierarchical Dirichlet process HSMM. In both works, the number of devices in the house is assumed to be known. Furthermore, the former uses training data to learn the device models, and the latter includes prior knowledge to model each specific device and ensures that all the devices are switched on at least once in the time series.

Our method is fully unsupervised, as it does not use any training data to build device-specific models, and it assumes an unknown number of devices. We believe this is the more general approach to address the power disaggregation problem, because if we want to apply this algorithm widely, it is unrealistic to think that we can obtain training information for all households and we should not expect to have a model for each potential device plugged in any home.

We validate the performance of the proposed IFUHMM applied to the power disaggregation problem in two different real databases:

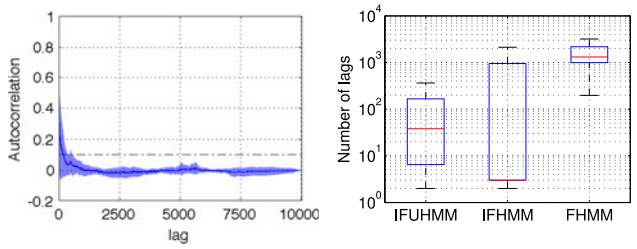


Fig. 3. Small scale experiment. (a) Autocorrelation plot for the IFUHMM. (b) Number of samples for the autocorrelation to fall below 0.1.

REDD database. The Reference Energy Disaggregation Data Set (REDD) [33] monitors several homes at low and high frequency for large periods of time. We consider 24-hour segments across 5 houses and choose the low-frequency power consumption of 6 devices: refrigerator (R), lighting (L), dishwasher (D), microwave (M), washer-dryer (W) and furnace (F). We apply a 30-second median filter and scale the data dividing by 100.

AMP database. The Almanac of Minutely Power Dataset [34] records the power consumption of a single house using 21 sub-meters for an entire year (from April 1st, 2012 to March 31st, 2013) at one minute read intervals. We consider two 24-hours segments and choose eight devices: basement plugs and lights (BME), clothes dryer (CDE), clothes washer (DWE), kitchen fridge (FGE), heat pump (HPE), home office (OFE), entertainment-TV, PVR, AMP (TVE) and wall oven (WOE). We scale the data by a factor of $1/100$.

Metric. In order to evaluate the performance of the different algorithms, we compute the mean accuracy of the estimated consumption of each device, which is measured as

$$\text{acc} = 1 - \frac{\sum_{t=1}^{T}\sum_{m=1}^{M}|x_t^{(m)} - \hat{x}_t^{(m)}|}{2\sum_{t=1}^{T}\sum_{m=1}^{M} x_t^{(m)}}, \tag{39}$$

where $x_t^{(m)}$ and $\hat{x}_t^{(m)}$ are, respectively, the true and the estimated power consumption by device m at time t [33]. If the inferred number of devices M_+ is smaller than the true number of devices, we use $\hat{x}_t^{(m)} = 0$ for the undetected devices. If M_+ is larger than the true number of devices, we group all the extra chains as an "unknown" device and use $x_t^{(\text{unk})} = 0$ to compute the accuracy. In order to compute the accuracy, as our algorithm is unsupervised, we need to assign each estimated chain to a device. We do that by sorting the estimated chains so that the accuracy is maximized.

Experimental setup. In our experiments, we consider the Gaussian observation model in Section 3 and, furthermore, the FHMM considers that $\mathbf{a}_0^m$ follows the prior distribution in Eq. (6). We set the hyperparameters to $\alpha = 1$, $\gamma = 1$, $\beta_0 = \beta = 1$, $\sigma_0^2 = 0$, $\sigma_\phi^2 = 10$, $\sigma_x^2 = 0.5$, $\boldsymbol{\mu}_0 = 15$ and $\lambda = 1$. For the IFUHMM, we speed up the inference by considering the split/merge and birth/death moves once every several iterations. We average the results provided by 20 independent runs of the samplers (or the variational algorithm), with different random initializations. For the variational algorithm, we estimate the states and observation parameters as $\widehat{s}_{tm} = \arg\max_k q(s_{tm} = k)$ and $\widehat{\boldsymbol{\Phi}}_k = \mathbf{L}_k$.

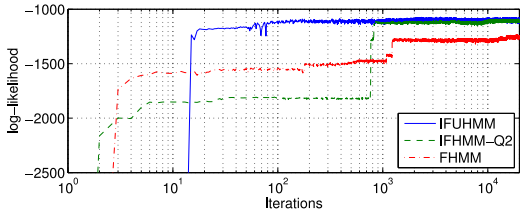


Fig. 4. Evolution of the log-likelihood.

6.1 Mixing Properties

In order to evaluate the mixing properties of the inference algorithm in Section 5.1, we aggregate the power signals of four devices of the AMP database (BME, CDE, DWE and HPE) for a 24-hour segment. Then, we apply our IFUHMM, the infinite binary FHMM (IFHMM), and the FHMM (with $M = 4$ chains and $Q = 4$ states). Our objective in this section is to analyze how increasing the flexibility of the model, by including the number of chains (IFHMM) and also the number of states (IFUHMM) as latent variables, changes the mixing properties of the algorithm.

To evaluate the mixing properties of the MCMC-based inference algorithms for the three models we need to define a function that depends on all the latent variables in the model, and that can be applied for any given number of states and chains. We choose the accuracy defined in Eq. (39) and compute it for the last $10{,}000$ samples of each algorithm.

We show in Fig. 3a the autocorrelation plot for the IFUHMM. The thick line corresponds to the mean of the autocorrelation plot for the 20 samplers, while the shaded area covers twice the standard deviation. In this figure, we observe that (on average) the autocorrelation falls below a threshold of 0.1 after a few tens of iterations. Moreover, we plot in Fig. 3b how many samples of the Markov chain under each model we should collect until the autocorrelation falls below 0.1. We show the median and the 10th, 25th, 75th and 90th percentiles in the standard box-plot format. For 50 percent of the cases, the IFUHMM needs only a few tens of samples, while for the remaining 50 percent of the simulations it needs at most a few hundreds of iterations. Although the median number of samples for the IFHMM is the smallest one (below 10), it needs hundreds or even thousands of samples for the remaining 50 percent of the simulations. Finally, the FHMM presents the poorest mixing properties, needing thousands of samples for 75 percent of the cases.

Now, we evaluate the goodness of fit of the three models. To this end, we show in Fig. 4 the best (among the 20 samplers) achieved log-likelihood for the three models. In accordance with Fig. 3b, the IFUHMM converges faster than the IFHMM and the FHMM algorithms. Furthermore, the IFUHMM presents the highest log-likelihood score, being the IFHMM almost as good. In addition, we show in Table 1 the mean and standard deviation (over the 20 samplers) of the accuracy provided by the three approaches, obtained

TABLE 1
Accuracy for the Small Scale Experiment

FHMM ($Q = 4$, $M = 4$)	0.47 ± 0.06
IFHMM ($Q = 2$)	0.67 ± 0.10
IFUHMM	$\mathbf{0.79 \pm 0.08}$

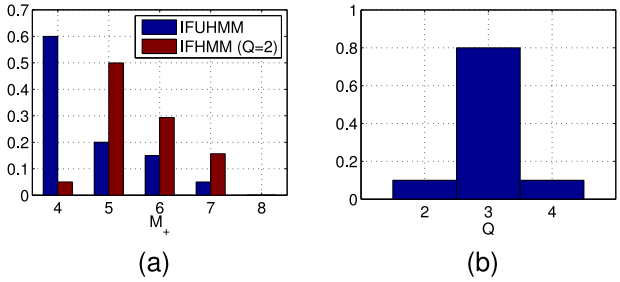


Fig. 5. Small scale experiment. Histograms of (a) M_+ and (b) Q under the IFUHMM.

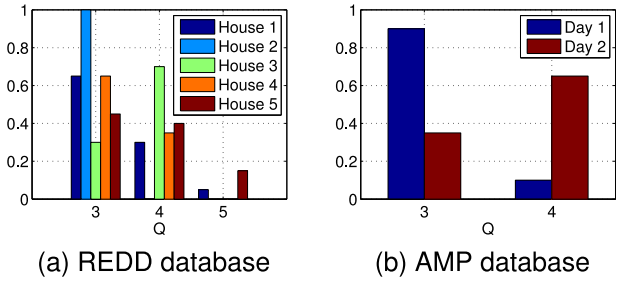


Fig. 6. Histogram of Q under the IFUHMM.

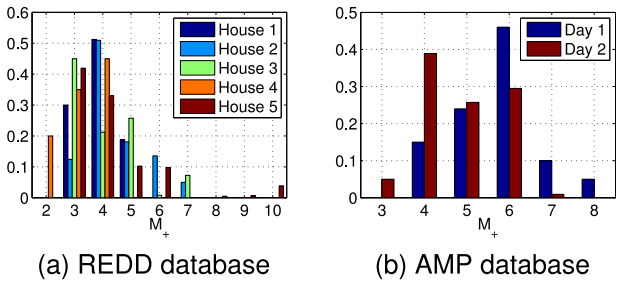


Fig. 7. Histogram of M_+ under the IFUHMM.

after averaging the accuracy values of the last $10{,}000$ samples. We can see that although both the IFUHMM and the IFHMM reach similar log-likelihood values (i.e., they can explain the observed data), in terms of accuracy, the IFUHMM is significantly better than the IFHMM.

To better understand this result, we depict in Fig. 5a the histogram for the number of inferred chains under the IFUHMM and the IFHMM, and in Fig. 5b the histogram for the inferred number of states under the IFUHMM. These histograms were obtained considering the last $10{,}000$ samples of the 20 samplers. We observe that the IFUHMM infers four chains 60 percent of the times, which corresponds to the true number of devices, also inferring that the number of states of the devices is $Q = 3$. The binary IFHMM mostly infers between $M_+ = 5$ and $M_+ = 7$ chains.

This explains why, although the IFUHMM and the IFHMM present similar log-likelihood scores in Fig. 4, the IFUHMM provides better accuracy. While the IFUHMM is recovering the underlying process that generates the total power consumption (allowing us to interpret each inferred chain as a device), the IFHMM needs to aggregate several of the inferred chains to construct the power consumption of each device, leading to a deterioration in the resulting accuracy. We could improve the accuracy of the IFHMM by combining several chains to fit each device. However, it would lead to a complex combinatorial problem in real life scenarios with a large number of devices with many states. Moreover, in a real scenario in which we did not have the ground truth, this solution for the poor accuracy of the IFHMM would not help to know which devices consume most. This is a typical example in which we have two nonparametric models that can explain the observed data similarly well, but while one of them (the IFUHMM) is recovering the latent structure of the data, the other one (the IFHMM) is just using its flexibility to explain the data but it does not have etiological interpretation.

Regarding the FHMM, the sampler gets trapped in a local optima. This explains its low log-likelihood and accuracy, even though it has *a priori* knowledge of the true number of devices.

6.2 Power Disaggregation

Now, we focus on solving more realistic power disaggregation problems. For the AMP database, we consider two 24-hour segments and the eight devices detailed above. For the REDD database, we consider a 24-hour segment across five houses, with the six devices mentioned above. For both databases, we compare the results provided by:

- A standard FHMM with $Q = 4$ states and perfect knowledge of the selected number of devices.
- The IFHMM with $Q = 4$ states in Section 2.2, using the variational algorithm in Section 4.3 truncated to $M = 15$ Markov chains (Var-Q4).
- The IFHMM with $Q = 4$ states in Section 2.2, using the blocked sampling algorithm detailed in Section 4.2 (IFHMM-Q4).
- The proposed IFUHMM in Section 5.

As discussed in the previous section, the binary IFHMM tends to overestimate the number of devices, sometimes growing above what our code can handle, specially when computing the accuracy. As a consequence, we do not report the results with the binary IFHMM, as it would lead to similar conclusions than in the previous section.

TABLE 2
REDD Database

	H1	H2	H3	H4	H5
FHMM ($M = 6, Q = 4$)	0.54 ± 0.05	0.67 ± 0.04	$\mathbf{0.57 \pm 0.06}$	0.45 ± 0.05	0.47 ± 0.04
Var-Q4	0.53 ± 0.04	0.60 ± 0.05	0.49 ± 0.06	0.43 ± 0.03	0.50 ± 0.05
IFHMM-Q4	0.57 ± 0.06	$\mathbf{0.75 \pm 0.02}$	0.53 ± 0.08	0.46 ± 0.07	0.57 ± 0.08
IFUHMM	$\mathbf{0.64 \pm 0.06}$	$\mathbf{0.77 \pm 0.03}$	$\mathbf{0.58 \pm 0.07}$	$\mathbf{0.55 \pm 0.07}$	$\mathbf{0.61 \pm 0.09}$

Mean accuracy broken down by house.

TABLE 3
AMP Database

	Day 1	Day 2
FHMM ($M = 8, Q = 4$)	0.36 ± 0.05	0.37 ± 0.05
Var-Q4	0.48 ± 0.06	0.51 ± 0.06
IFHMM-Q4	0.58 ± 0.11	0.58 ± 0.07
IFUHMM	$\mathbf{0.69 \pm 0.10}$	$\mathbf{0.67 \pm 0.11}$

Mean accuracy broken down by day.

Fig. 6 shows the histograms of the inferred number of states obtained with the IFUHMM. This figure shows that the required number of states in both databases is between three and five. This is why we set the number of states for the FHMM and the IFHMM to four, which in turn is a typical value of the number of states considered in the literature [33]. We also show in Fig. 7 the histograms of the inferred number of chains obtained under the IFUHMM.

Tables 2 and 3 show the mean and standard deviation of the accuracy provided by the four approaches . We observe that the IFUHMM presents the largest accuracy for both databases and for all days and houses. The FHMM is as good as the IFUHMM for house 3 of the REDD database, while for house 2 the IFHMM-Q4 provides a similar accuracy to the IFUHMM. If we now compare the two inference algorithms, the blocked sampler (IFHMM-Q4) and the variational algorithm (Var-Q4), we can observe that the IFHMM-Q4 presents in general better accuracy. Hence, although the variational algorithm runs faster than the blocked sampler, it provides less accurate results, in accordance with typical results the literature.

Finally, we depict in Figs. 8 and 9 the true percentage of total power consumed by each device, compared to the inferred percentages by each approach, for both the REDD and AMP databases. Note that assuming a fixed number of chains can be harmful if some of the devices are not switched on at least once during the observation period (see, e.g., the second day of the AMP database in Fig. 9b). If we now compare these figures to the histograms of the inferred number of chains in Fig. 7, we can observe that the IFUHMM always captures the most consuming devices (see, e.g., house 1 in Fig. 7a, which shows that the IFUHMM captures in more than 50 percent of the cases the true number of devices in Fig. 8a, where each device consumes more than 10 percent of the total power). However, when dealing with less consuming devices (see, e.g., the washer-dryer 'W' of house 2 in Fig. 8b), it tends to underestimate the number of devices, assigning the power of these less consuming devices to other more consuming devices.

From these results, we can conclude that the IFUHMM performs much better because it can adapt the number of states and chains to fit the data. For different houses or days it may choose different number of components, while the other methods stick to a value that might not be the best in some cases. Using a nonparametric prior allows for the flexibility enough to change the number of components for each scenario, providing a significant improvement over fixed models, even when they use the ground truth for the number of devices or a typical number of states.

To sum up, our IFUHMM properly detects the active devices in the time series, and indicates that, in general, three or four states are enough to describe the behavior of the electrical devices. The IFUHMM does not make use of specific prior information to model each individual device but, even so, it is able to recover the number of devices and their powers draws accurately, providing a good estimation of the percentage of the total power that each device consumes.

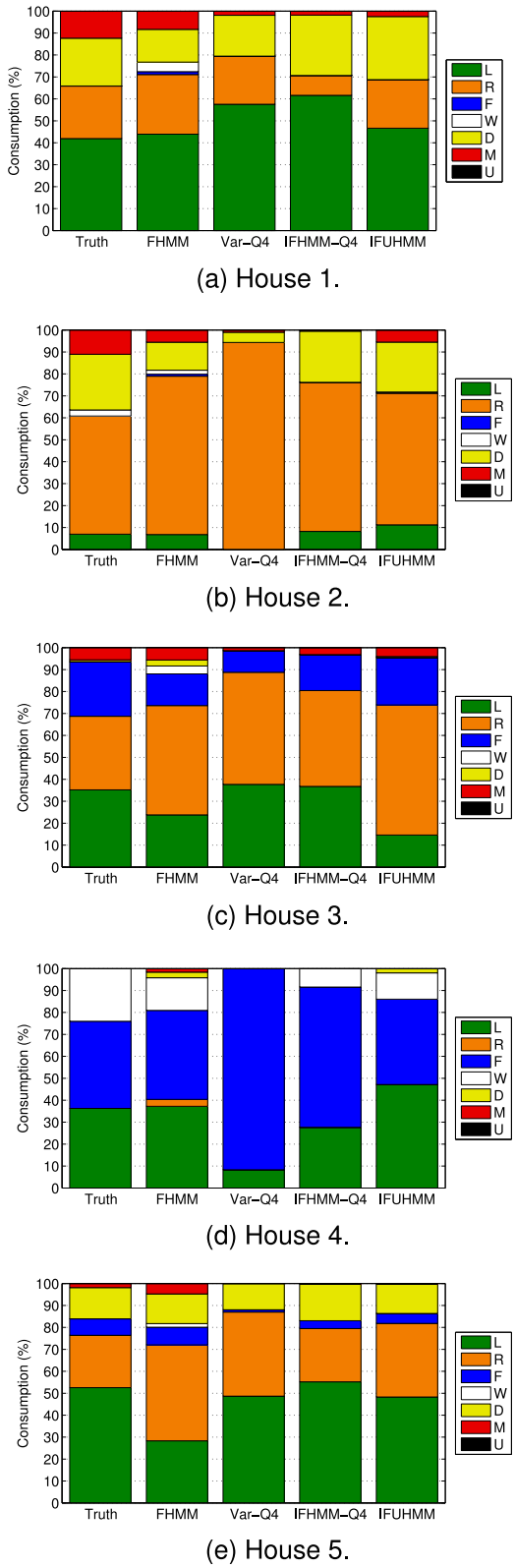
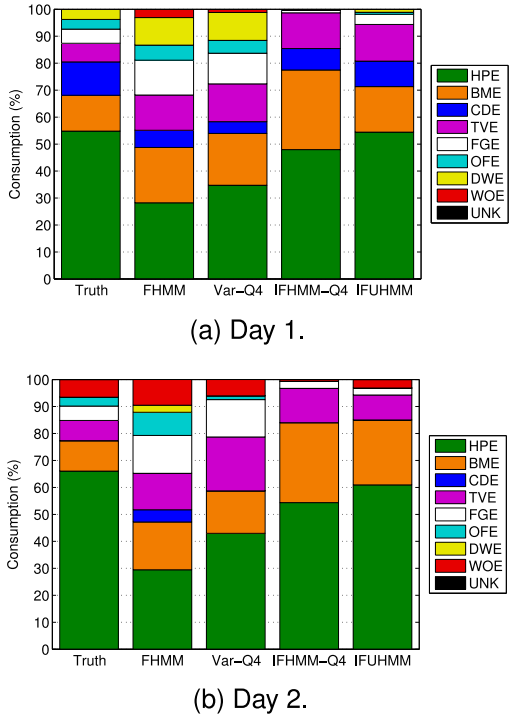


Fig. 8. REDD database. Percentage of total power consumed by each device.



(a) Day 1.

(b) Day 2.

Fig. 9. AMP database. Percentage of total power consumed by each device.

7 Conclusions and Future Work

We have extended the existent binary IFHMM [18] to allow for any number of states in the Markov chains and developed two MCMC-based algorithms and a variational inference algorithm for this model. Additionally, by placing an infinite discrete prior distribution over the number of states, we have derived an inference algorithm that learns both the number of parallel chains and the number of hidden states in a FHMM. This algorithm resembles the RJMCMC techniques for HMMs but, since all the dimension-changing variables can be integrated out under our model, we resort instead to a standard Metropolis-Hastings algorithm. Therefore, our algorithm effectively deals with the trade-off problem between the number of chains and the number of states, avoiding the model selection, and can be useful to find the Markov structure in the data and to explain the latent causes of the observations in a meaningful way.

In order to show the proper performance of the proposed algorithms, we have focused on solving the power disaggregation problem on two real datasets. In these experiments, we have found that the number of devices in the power disaggregation problem, as well as their parameters, can be inferred in a fully blind manner. We have also obtained that inferring the number of chains and states in the FHMM, instead of fixing them *a priori*, improves performance. Hence, the proposed IFUHMM appears as a more generally applicable model than the existing binary IFHMM [18] to find the hidden canonical causes in a time series.

One of the limitations of the proposed approach, when used over a significant proportion of the power grid of any city, is to find the correspondence of each estimated chain with a specific device, as the model is blind and we do not have individual information for each house. There are two complementary ways around it. First, we can use statistical properties from the inferred chains: if a chain is active for minutes or hours consuming a significant amount of power, we could believe it represents the lighting in that house; if a chain is only active for a few minutes consuming much power, we can think of it as a microwave; if it were on all day long with a periodic power signal it would be the fridge; and if it were only used for around an hour a few days per week, it might be the washing machine. Second, we can also augment our model by considering a hierarchy, in which the chains are shared across the houses, but their activation is individually computed for each house. In this way, we only need to infer some representative devices that are shared among several houses.

Acknowledgments

I. Valera is currently supported by the Humboldt research fellowship for postdoctoral researchers program and acknowledges the support of Plan Regional-Programas I+D of Comunidad de Madrid (AGES-CM S2010/BMD-2422). F.J.R. Ruiz is supported by an FPU fellowship from the Spanish Ministry of Education (AP2010-5333). This work is also partially supported by Ministerio de Economía of Spain (projects COMPREHENSION, id. TEC2012-38883-C02-01, and ALCIT, id. TEC2012-38800-C03-01), by Comunidad de Madrid (project CASI-CAM-CM, id. S2013/ICE-2845), by the Office of Naval Research (ONR N00014-11-1-0651), and by the European Union 7th Framework Programme through the Marie Curie Initial Training Network 'Machine Learning for Personalized Medicine' (MLPM2012, Grant No. 316861). Isabel Valera and Francisco J.R. Ruiz contributed equally in this paper.

References

[1] M. Keralapura, M. Pourfathi, and B. Sirkeci-Mergen, "Impact of contrast functions in fast-ICA on twin ECG separation," *IAENG Int. J. Comput. Sci.*, vol. 38, no. 1, p. 38, 2011.

[2] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "A sticky HDP-HMM with application to speaker diarization," *Ann. Appl. Statist.*, vol. 5, no. 2A, pp. 1020–1056, 2011.

[3] H. Kim, M. Marwah, M. F. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *Proc. SIAM Int. Conf. Data Mining*, 2011, pp. 747–758.

[4] S. Darby, "The effectiveness of feedback on energy consumption: A review for DEFRA of the literature on metering, billing and direct displays," Environ. Change Inst., University of Oxford, Oxford, U.K., 2006.

[5] B. Neenan and J. Robinson, "Residential electricity use feedback: A research synthesis and economic framework," Elect. Power Res. Inst., Palo Alto, CA, USA, Tech. Rep. 1016844, 2009.

[6] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[7] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *IEEE ASSP Mag.*, vol. ASSPM-3, no. 1, pp. 4–16, Jan. 1986.

[8] R. Nag, K. Wong, and F. Fallside, "Script recognition using hidden Markov models," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1986, vol. 11, pp. 2071–2074.

[9] P. Baldi, Y. Chauvin, T. Hunkapiller, and M. A. McClure, "Hidden Markov models of biological primary sequence information," *Proc. Nat. Acad. Sci. USA*, vol. 91, no. 3, pp. 1059–1063, 1994.

[10] J. Kupiec, "Robust part-of-speech tagging using a hidden Markov model," *Comput. Speech Language*, vol. 6, no. 3, pp. 225–242, 1992.

[11] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statistical Soc. Series B (Methodological)*, vol. 39, pp. 1–38, 1977.
[12] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Statist.*, vol. 37, pp. 1554–1563, 1966.
[13] C. P. Robert, T. Rydén, and D. M. Titterington, "Bayesian inference in hidden Markov models through reversible jump Markov chain Monte Carlo," *J. Roy. Statistical Soc. Series B*, vol. 62, pp. 57–75, 2000.
[14] P. J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 82, pp. 711–732, 1995.
[15] Z. Ghahramani and M. I. Jordan, "Factorial hidden Markov models," *Mach. Learn.*, vol. 29, nos. 2/3, pp. 245–273, 1997.
[16] M. I. Jordan, *Hierarchical Models, Nested Models and Completely Random Measures*, M.-H. Chen, D. Dey, P. Mueller, D. Sun, and K. Ye, Eds. New York, NY, USA: Springer, 2010.
[17] Y. W. Teh and M. I. Jordan, "Hierarchical Bayesian nonparametric models with applications," in *Bayesian Nonparametrics: Principles and Practice*, N. Hjort, C. Holmes, P. Müller, and S. Walker, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2010.
[18] J. Van Gael, Y. W. Teh, and Z. Ghahramani, "The infinite factorial hidden Markov model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, vol. 21, pp. 1697–1704.
[19] K. A. Heller, Y. W. Teh, and D. Görür, "Infinite hierarchical hidden Markov models," in *Proc. 12th Int. Conf. Artif. Intell. Statist.*, 2009, vol. 12, pp. 224–231.
[20] M. Titsias, "The infinite Gamma-Poisson feature model," *Proc. Adv. Neural Inf. Process. Syst.*, 2007, vol. 19, pp. 1513–1520.
[21] T. L. Griffiths and Z. Ghahramani, "The Indian buffet process: An introduction and review," *J. Mach. Learn. Res.*, vol. 12, pp. 1185–1224, 2011.
[22] Y. W. Teh, D. Görür, and Z. Ghahramani, "Stick-breaking construction for the Indian buffet process," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2007, vol. 11, pp. 556–563.
[23] J. Van Gael, Y. Saatci, Y. W. Teh, and Z. Ghahramani, "Beam sampling for the infinite hidden Markov model," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, vol. 25, pp. 1088–1095.
[24] F. Doshi-Velez, K. T. Miller, J. Van Gael, and Y. W. Teh, "Variational inference for the Indian buffet process," in *Proc. 12th Int. Conf. Artif. Intell. Stat.*, 2009, pp. 137–144.
[25] N. Ding and Z. Ou, "Variational nonparametric Bayesian hidden Markov model," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2010, pp. 2098–2101.
[26] S. L. Scott, "Bayesian methods for hidden Markov models: Recursive computing in the 21st century," *J. Amer. Statistical Assoc.*, vol. 97, no. 457, pp. 337–351, 2002.
[27] R. Neal, "Slice sampling," *Ann. Statist.*, vol. 31, pp. 705–767, 2000.
[28] W. R. Gilks and P. Wild, "Adaptive rejection sampling for Gibbs sampling," *J. Roy. Statistical Soc. Series C (Appl. Statist.)*, vol. 41, no. 2, pp. 337–348, 1992.
[29] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, no. 2, pp. 183–233, Nov. 1999.
[30] S. Jain and R. Neal, "A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model," *J. Comput. Graphical Statist.*, vol. 13, pp. 158–182, 2000.
[31] S. Richardson and P. J. Green, "On Bayesian analysis of mixtures with an unknown number of components (with discussion)," *J. Roy. Statistical Soc., Series B (Methodological)*, vol. 59, no. 4, pp. 731–792, 1997.
[32] M. J. Johnson and A. S. Willsky, "Bayesian nonparametric hidden semi-Markov models," *J. Mach. Learn. Res.*, vol. 14, pp. 673–701, Feb. 2013.
[33] J. Z. Kolter and M. J. Johnson, "REDD: A public data set for energy disaggregation research," in *Proc. SustKDD Workshop Data Mining Appl. Sustainability*, San Diego, CA, vol. 25, pp. 59–62, Aug. 2011.
[34] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I. V. Bajic, "AMPds: A public dataset for load disaggregation and eco-feedback research," in *Proc. IEEE Elect. Power Energy Conf.*, 2013, pp. 1–6.

Isabel Valera received the BSc degree in electrical engineering from Technical University of Cartagena (Spain), and the MSc degree in multimedia and communications from the University Carlos III in Madrid, Spain, in 2009 and 2012, respectively. She received the PhD degree in electrical engineering in 2014 at the University Carlos III in Madrid, and then joined the MPI-SWS as postdoctoral researcher, where she is currently funded by the Humboldt research postdoctoral fellowship. Her research turns around the development probabilistic models to solve real world problems in diverse areas such as bioengineering, communication systems and social media. She is working on Bayesian nonparametric models as well as discrete- and continuous-time probabilistic models, and inference methods based on MCMC and variational inference.

Francisco J.R. Ruiz received the BSc degree in electrical engineering from the University of Seville, Spain, and the MSc degree in multimedia and communications from the University Carlos III in Madrid, Spain, in 2010 and 2012, respectively. He received the PhD degree in electrical engineering in 2015 and then joined the Department of Computer Science, Columbia University as a postdoctoral researcher, where he works with David Blei. His main research interests include applications of Bayesian nonparametric models to signal processing, as well as the associated inference methods (MCMC methods, variational inference, and expectation propagation).

Fernando Perez-Cruz (SM'06) received the MSc/BSc in electrical engineering from the University of Sevilla in 1996, and the PhD degree in electrical engineering in 2000 from the Technical University of Madrid. He is a member of the Technical Staff at Bell Labs and an associate professor with the Department of Signal Theory and Communication, University Carlos III in Madrid. He has been a visiting professor at Princeton University under the sponsorship of a Marie Curie Fellowship. He has held position at the Gatsby Unit (London), Max Planck Institute for Biological Cybernetics (Tuebingen), BioWulf Technologies (New York) and the Technical University of Madrid and Alcala University (Madrid). His current research interest lies in machine learning and information theory and its application to signal processing and communications. He has authored over 90 contributions to international journals and conferences. He has also coauthored a book in digital communications. He is a senior member of the IEEE.