
UNIVERSIDAD CARLOS III DE MADRID

Departamento de Teoría de la Señal y Comunicaciones

DOCTORAL THESIS

# BAYESIAN NONPARAMETRICS FOR TIME SERIES MODELING

Author: FRANCISCO JESÚS RODRÍGUEZ RUIZ

Supervised by: FERNANDO PÉREZ CRUZ

June 2015

**Tesis Doctoral:** BAYESIAN NONPARAMETRICS FOR TIME SERIES MODELING

**Autor:** Francisco Jesús Rodríguez Ruiz

**Director:** Fernando Pérez Cruz

**Fecha:** 30 de junio de 2015

## Tribunal

Presidente: Antonio Artés Rodríguez

Vocal: Konstantina Palla

Secretario: Juan José Murillo Fuentes

*A mis padres, Juan y Carmen,*
*porque son ellos quienes realmente*
*han hecho posible esta Tesis.*

*"It's a dangerous business, Frodo, going out your door.*
*You step onto the road, and if you don't keep your feet,*
*there's no knowing where you might be swept off to."*

J. R. R. Tolkien
(*The Lord of the Rings*)

*"Es peligroso, Frodo, cruzar tu puerta.*
*Pones tu pie en el camino, y si no cuidas tus pasos,*
*no sabes a dónde te pueden llevar."*

J. R. R. Tolkien
(*El Señor de los Anillos*)

# Agradecimientos

He de confesar que la elaboración de estos agradecimientos ha sido una tarea de magnitud comparable a la realización de Tesis en sí misma. Y no es de extrañar, teniendo en cuenta la cantidad de gente que de una manera u otra me ha ayudado a lo largo de estos años, e incluso antes de comenzar el doctorado. Soy consciente de que la mayoría de familiares, amigos y compañeros que tengan acceso a este libro rápidamente buscarán su nombre entre estas páginas, y muy probablemente sea el único capítulo que lean. Por ello, he tratado de poner especial empeño en cuidar la redacción y la ortografía, pero sobre todo en no olvidar ninguno de los nombres. No obstante, pido por adelantado la compresión y el entendimiento si cometo el terrible error de dejar a alguien en el tintero (o, en este caso, en el teclado del ordenador).

Siguiendo el protocolo, y no me refiero a TCP/IP, debo comenzar por agradecer a mi beca FPU (referencia AP-2010-5333) por cuatro años de salario mileurista. Gracias al Ministerio de Educación por no concederme ninguna ayuda adicional para realizar estancias de movilidad, eliminando así cualquier posible riesgo de concesión de la ayuda y denegación de la misma a posteriori, una vez realizada la estancia, como ya ocurrió en su día con algunos compañeros. Agradezco, eso sí, a las ayudas de movilidad de la UC3M, y al Grupo de Tratamiento de la Señal en general, por cubrir dichas estancias en su lugar.

Gracias a Fernando, mi tutor de tesis, por todo el tiempo dedicado, por la ayuda activa que me ha prestado desde el principio, y por su facilidad para generar ideas e impregnar los resultados obtenidos de acuerdo a los postulados de la Ingeniería del Whisky. Gracias también a Juanjo, profesor en la Universidad de Sevilla, y principal responsable de que yo acabara en la UC3M. Me gustaría nombrar también a Antonio, Joaquín, Ángel, Jose y David, profesores del grupo con los que en algún momento he compartido alguna discusión (científica) y de los que he tenido la oportunidad de aprender bastante. Sin olvidar a David Blei y Neil Lawrence, supervisores de las estancias que he realizado, y unos auténticos genios.

Gracias también a Isa, compañera de fatigas durante todo este tiempo y coautora de no pocos artículos; a Mélanie, con quien he compartido algunos trabajos, además de innumerables ideas y pasteles; a Pablo "bueno" o Pablo "Amazon", firme defensor de la doctrina bayesiana, de quien también he aprendido mucho; a Jesús, cuyos conocimientos, que van más allá de lo meramente técnico, me han sido indispensables en todo momento; a Víctor, con quien también he intercambiado muchas ideas y acaloradas discusiones; a Alfredo, quien también ha contribuido a esta Tesis corrigiendo algunas erratas; a Luca, con quien viví durante un breve período de tiempo, y de quien pude aprender algo de MCMC; a Pablo "malo", coautor también de unos cuantos trabajos; y a todas sus charlas de grupo. Debo también unas palabras a Tobi, gran profesor y mentor, muy querido por sus alumnos (entre los que me incluyo). Agradezco a los viejos (Paloma, Katrin, Eugenia, Jair, Vladi, Jesse, Wilton, Camilo, Concha, Luis, Sandra, Mostafa, Blanca, Niamh y Jorge), y a los nuevos (Alex, Vivian, Grace, Alberto, dos Franes y dos Gonzalos). Os merecéis un libro de agradecimientos cada uno, pero un análisis tan detallado está fuera del alcance de esta Tesis. Por supuesto, gracias también a Ana Hernando: nueve de cada diez funcionarios de la Administración pública deberían aprender de ella. Gracias a Harold, por solucionar todos nuestros problemas informáticos, y al cluster, por crearlos. Gracias también a la *Wikipedia*, que a pesar de ser la referencia que con más frecuencia he consultado en estos años, no me permiten citarla en la bibliografía, y (¡casi se me olvida!) a la Oficina de Postgrado de la UC3M, que en todo momento ha tratado concienzudamente de reforzar nuestras capacidades de auto-aprendizaje y auto-valía.

Trasladándonos ahora un poco más al sur, he de agradecer también a la gente de la ESI de Sevilla y a los maesianos con los que compartí tantos momentos. En especial, gracias a Rodolfo, con quien más contacto mantengo a pesar de la distancia y del paso de los años, pero también a Gema, Luis, Laura, Noe, Jessi, Marta, Emi, Salva, Javi, Betania, otra Marta, Celia, Elvira, Cristina, Fátima, Mercedes, Fran, Isa, y a tantos otros que conocí hace ya unos años pero que dejan huella. Gracias también a los jerezanos: Darío, Núñez, Raúl, Carou, Nai, Jorge y

Luky, con quienes he compartido muchas partidas y alguna que otra barbacoa en los últimos años. Y a Andrea, que no es sureña pero no le importaría serlo.

Continuando por la categoría de amigos almerienses, no puedo olvidar a Antonio, Álvaro, Dany, otro Antonio, Esther, Lucas, Manu, Rocío, Pedro, Laura, Loren, Anabel, Aless y Juan. Ellos son los amigos de siempre, aquellos con quienes puedo compartir bromas que nadie más entenderá, a quienes siempre intento reservar un hueco de mi tiempo cada vez que piso Almería. Algún día lograremos completar esa trilogía que nos quedó pendiente.

Corresponde ahora el turno a familiares. En primer lugar, me gustaría conceder una palabras a mis padres, Juan y Carmen, quienes han estado conmigo siempre y me han apoyado en prácticamente la totalidad de mis decisiones. Gracias a mis hermanos Juan, Jose Carlos y Carmen Mari, por ser para mí unos modelos a seguir desde que era pequeño. Sin olvidar, como no podía ser menos, a los acoplados (como solía denominarlos mi abuelo Antonio): Lola, Mónica y Jesús. Entre todos, consiguen que cada visita a Almería me resulte demasiado corta. Aunque parte de ese mérito es compartido con las nuevas incorporaciones a la familia, África y Celia, cuyas risas lo hacen todo más fácil. Estoy convencido de que serán capaces de leer estas líneas (o cualesquiera otras) sorprendentemente pronto. También me gustaría mencionar aquí a mi abuelo Diego y, por supuesto, a la memoria de Carmen, Antonio y Pepa. Gracias también a todos mis tíos y primos y, en general, a los innumerables miembros de mi familia, por brindarme la oportunidad de aprender idiomas cada vez que me pedían un árbol genealógico en clase de inglés o francés. Gracias asimismo a *Busbam*, con quien indiscutiblemente también he pasado mucho tiempo en estos últimos años.

Por último (*last but not least*), a esa intrépida persona que diariamente me ayuda y me da ánimos, que ha vivido de cerca todas mis idas y venidas, y que a pesar de ello me aguanta y dice que lo seguirá haciendo. A ti, Ana. Aún nos queda mucho por compartir.

# Abstract

In many real-world signal processing problems, an observed temporal sequence can be explained by several unobservable independent causes, and we are interested in recovering the canonical signals that lead to these observations. For example, we may want to separate the overlapping voices on a single recording, distinguish the individual players on a financial market, or recover the underlying brain signals from electroencephalography data. This problem, known as source separation, is in general highly underdetermined or ill-posed. Methods for source separation generally seek to narrow the set of possible solutions in a way that is unlikely to exclude the desired solution.

However, most classical approaches for source separation assume a fixed and known number of latent sources. This may represent a limitation in contexts in which the number of independent causes is unknown and is not limited to a small range. In this Thesis, we address the signal separation problem from a probabilistic modeling perspective. We encode our independence assumptions in a probabilistic model and develop inference algorithms to unveil the underlying sequences that explain the observed signal. We adopt a Bayesian nonparametric (BNP) approach in order to let the inference procedure estimate the number of independent sequences that best explain the data.

BNP models place a prior distribution over an infinite-dimensional parameter space, which makes them particularly useful in probabilistic models in which the number of hidden parameters is unknown *a priori*. Under this prior distribution, the posterior distribution of the hidden parameters given the data assigns higher probability mass to those configurations that best explain the observations. Hence, inference over the hidden variables is performed using standard Bayesian inference techniques, which avoids expensive model selection steps.

We develop two novel BNP models for source separation in time series. First, we propose a non-binary infinite factorial hidden Markov model (IFHMM), in which the number of parallel chains of a factorial hidden Markov model (FHMM)

is treated in a nonparametric fashion. This model constitutes an extension of the binary IFHMM, but the hidden states are not restricted to take binary values. Moreover, by placing a Poisson prior distribution over the cardinality of the hidden states, we develop the infinite factorial unbounded-state hidden Markov model (IFUHMM), and an inference algorithm that can infer both the number of chains and the number of states in the factorial model. Second, we introduce the infinite factorial finite state machine (IFFSM) model, in which the number of independent Markov chains is also potentially infinite, but each of them evolves according to a stochastic finite-memory finite state machine model. For the IFFSM, we apply an efficient inference algorithm, based on particle Markov chain Monte Carlo (MCMC) methods, that avoids the exponential runtime complexity of more standard MCMC algorithms such as forward-filtering backward-sampling.

Although our models are applicable in a broad range of fields, we focus on two specific problems: power disaggregation and multiuser channel estimation and symbol detection. The power disaggregation problem consists in estimating the power draw of individual devices, given the aggregate whole-home power consumption signal. Blind multiuser channel estimation and symbol detection involves inferring the channel coefficients and the transmitted symbol in a multiuser digital communication system, such as a wireless communication network, with no need of training data. We assume that the number of electrical devices or the number of transmitters is not known in advance. Our experimental results show that the proposed methodology can provide accurate results, outperforming state-of-the-art approaches.

# Resumen

En multitud de problemas reales de procesado de señal, se tiene acceso a una secuencia temporal que puede explicarse mediante varias causas latentes independientes, y el objetivo es la recuperación de las señales canónicas que dan lugar a dichas observaciones. Por ejemplo, podemos estar interesados en separar varias señales de voz solapadas en una misma grabación, distinguir los agentes que operan en un mismo mercado financiero, o recuperar las señales cerebrales a partir de los datos de un electroencefalograma. Este problema, conocido como separación de fuente, es en general sobredeterminado. Los métodos de separación de fuente normalmente tratan de reducir el conjunto de posibles soluciones de tal manera que sea poco probable excluir la solución deseada.

Sin embargo, en la mayoría de métodos clásicos de separación de fuente, se asume que el número de fuentes latentes es conocido. Esto puede representar una limitación en aplicaciones en las que no se conoce el número de causas independientes y dicho número no está acotado en un pequeño intervalo. En esta Tesis, consideramos un enfoque probabilístico para el problema de separación de fuente, en el que las asunciones de independencia se pueden incluir en el modelo probabilístico, y desarrollamos algoritmos de inferencia que permiten recuperar las señales latentes que explican la secuencia observada. Nos basamos en la utilización de métodos bayesianos no paramétricos (BNP) para permitir al algoritmo estimar adicionalmente el número de secuencias que mejor expliquen los datos.

Los modelos BNP nos permiten definir una distribución de probabilidad sobre un espacio de dimensionalidad infinita, lo cual los hace particularmente útiles para su aplicación en modelos probabilísticos en los que el número de parámetros ocultos es desconocido *a priori*. Bajo esta distribución de probabilidad, la distribución *a posteriori* sobre los parámetros ocultos del modelo, dados los datos, asignará una mayor densidad de probabilidad a las configuraciones que mejor expliquen las observaciones, evitando por tanto los métodos de selección de modelo, que son computacionalmente costosos.

En esta Tesis, desarrollamos dos nuevos modelos BNP para la separación de fuente en secuencias temporales. En primer lugar, proponemos un modelo oculto de Markov factorial infinito (IFHMM) no binario, en el que tratamos de manera no paramétrica el número de cadenas paralelas de un modelo oculto de Markov factorial (FHMM). Este modelo constituye una extensión del IFHMM binario, pero se elimina la restricción de que los estados ocultos sean variables binarias. Además, imponiendo una distribución de Poisson sobre la cardinalidad de los estados ocultos, desarrollamos el modelo oculto de Markov factorial infinito con estados no acotados (IFUHMM), y un algoritmo de inferencia con la capacidad de inferir tanto el número de cadenas como el número de estados del modelo factorial. En segundo lugar, proponemos un modelo de máquina de estados factorial infinita (IFFSM), en el que el número de cadenas de Markov paralelas e independientes también es potencialmente infinito, pero cada una de ellas evoluciona según un modelo de máquina de estados estocástica con memoria finita. Para el IFFSM, aplicamos un eficiente algoritmo de inferencia, basado en métodos *Markov chain Monte Carlo* (MCMC) de partículas, que evita la complejidad exponencial en tiempo de ejecución de otros algoritmos MCMC más comunes, como el de filtrado hacia adelante y muestreo hacia atrás.

A pesar de que nuestros modelos son aplicables en una amplia variedad de campos, nos centramos en dos problemas específicos: separación de energía, y estimación de canal y detección de símbolos en un sistema multi-usuario. El problema de separación de energía consiste en, dada la señal de potencia total consumida en una casa, estimar de manera individual el consumo de potencia de cada dispositivo. La estimación de canal y detección de símbolos consiste en inferir los coeficientes de canal y los símbolos transmitidos en un sistema de comunicaciones digital multi-usuario, como una red de comunicaciones inalámbrica, sin necesidad de transmitir símbolos piloto. Asumimos que tanto el número de dispositivos eléctricos como el número de transmisores es en principio desconocido y no acotado. Los resultados experimentales demuestran que la metodología propuesta ofrece buenos resultados y presenta mejoras sobre otros métodos propuestos en la literatura.

# Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| ADER | activity detection error rate |
| AP | access point |
| ARS | adaptive rejection sampling |
| AWGN | additive white Gaussian noise |
| BNP | Bayesian nonparametric |
| BP | beta process |
| BPSK | binary phase-shift keying |
| CDMA | code-division multiple access |
| CRF | Chinese restaurant franchise |
| CRP | Chinese restaurant process |
| CSI | channel state information |
| DEP | detection error probability |
| DP | Dirichlet process |
| DS-CDMA | direct-sequence code-division multiple access |
| EM | expectation maximization |
| EP | expectation propagation |
| FFBS | forward-filtering backward-sampling |

| | |
|---|---|
| FHMM | factorial hidden Markov model |
| FSM | finite state machine |
| GP | Gaussian process |
| HDP | hierarchical Dirichlet process |
| HHMM | hierarchical hidden Markov model |
| HMM | hidden Markov model |
| HSMM | hidden semi-Markov model |
| IBP | Indian buffet process |
| ICA | independent component analysis |
| IFFSM | infinite factorial finite state machine |
| IFHMM | infinite factorial hidden Markov model |
| IFUHMM | infinite factorial unbounded-state hidden Markov model |
| IHHMM | infinite hierarchical hidden Markov model |
| IHMM | infinite hidden Markov model |
| ISI | inter-symbol interference |
| M2M | machine-to-machine |
| MAP | *maximum a posteriori* |
| MCMC | Markov chain Monte Carlo |
| MIBP | Markov Indian buffet process |
| MIMO | multiple-input multiple-output |
| MSE | mean square error |
| PGAS | particle Gibbs with ancestor sampling |
| QAM | quadrature amplitude modulation |

| | |
|---|---|
| QPSK | quadrature phase-shift keying |
| RJMCMC | reversible jump Markov chain Monte Carlo |
| SER | symbol error rate |
| SIMO | single-input multiple-output |
| SMC | sequential Monte Carlo |
| SNR | signal-to-noise ratio |
| WCN | wireless communication network |

# 1 Introduction

## 1.1 Background

Machine learning algorithms attempt to perform relevant tasks by inductive learning [131]. They offer a feasible and effective approach for problems where manual programming fails. In particular, machine learning is defined as a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty [91]. Machine learning (which strongly overlaps with data mining, pattern recognition and predictive analytics) techniques are widely used in business, industry, science and government.

Some motivating examples where machine learning techniques have proven to be useful are:

1. **Topic modeling:** We have an archive of the raw text of thousands of books, which have been previously scanned and converted to text. We want to discover the themes in the texts, organize the books by subject, and build a navigator for users to explore the collection [22, 63].

2. **Clustering gene expression data:** We have measured expression levels of a large number of genes using microarray technology, yielding time-series data spanning some phenomenon such as development or disease progression, and we want to cluster the time series to identify genes that are differentially expressed between two groups, with these groups often representing normal and diseased or tumor tissue [37, 59].

3. **Face recognition:** We want to automatically detect human faces on a digital image, given a facial database consisting on a set of images with labeled faces [142].

4. **Recommendation systems:** We have data from a movie website containing millions of users' histories of ratings, and we want to automatically recommend movies to users based on this information [114, 51].

In all these examples, we have access to a real-world database and want to perform a task given the available data as input. Furthermore, there is an inherent source of uncertainty, which may come from noisy measurements, incomplete information, or from the fact that we only have access to a subset of the data from a larger population. Machine learning takes into account this uncertainty in a statistical manner.

Within machine learning, probabilistic models provide a useful approach to develop new methods in order to analyze data [72, 19, 91]. In generative probabilistic models, we explicitly encode our prior assumptions about the hidden structure of the data by including hidden variables and a probability distribution over both the hidden variables and the observed data. The model represents, often in considerably idealized form, the data-generating process. This process and the corresponding hidden structure can be represented in a probabilistic graphical

model [136].

Given a probabilistic model and the observed data, we make use of an inference algorithm to analyze the data under our assumptions and fine-tune the model. The inference method recovers the hidden structure that best explains the observations through exploration of the posterior distribution of the hidden variables given the data. In descriptive tasks, like problems #1 and #2 above, the posterior distribution helps us explore the data, with the hidden structure probabilistically "filled in". In predictive tasks, like problems #3 and #4, we use the posterior distribution to make predictions about new observations.

In probabilistic modeling, choosing the model complexity is often a nuisance because it is expensive to fit many models on large datasets. For instance, in problem #1 above, a natural question is how many themes (or topics) we should consider. Clearly, this quantity should grow as we collect more and more books. Another example arises in the context of clustering, where we are interested in finding an "informative" partition of the observations in an unsupervised manner, but we might not know the number of clusters in advance. Again, as more and more observations are collected, we should expect to find a larger number of clusters. Bayesian nonparametric (BNP) models constitute an approach to model selection and adaptation, where the complexity of the model is allowed to grow with data size, as opposed to parametric models, which use a fixed number of parameters.

Thus, BNP models provide a useful tool for problems in which the number of unknown hidden variables is itself unknown. Instead of specifying a closed model, BNP priors place probability mass on an infinite range of models and let the inference procedure choose the one that best fits the data [97, 47]. Under this prior, the number of hidden components can be learned using standard Bayesian inference techniques.

## 1.2 Motivation

Real-world processes generally produce observable outputs that can be characterized as signals. The signals can be discrete in nature (e.g., characters from a

finite alphabet), or continuous (e.g., speech or music samples). The signals are typically corrupted by other signal sources, transmission distortions, reverberation, or noise. A problem of fundamental interest is characterizing such real-world signals in terms of signal models. In particular, there are several signal processing 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 [75], separate the overlapping voices on a single recording [45] or the number of players and their strategies on a financial market [77], separate the contribution of each device to the total power consumed at a household [76], or detect the symbols sent by different transmitters in a multiuser communication channel [143]. In some of these problems, the number of independent causes 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.

The problem of signal separation, also known as source separation, was first formulated in the eighties, and is in general a highly underdetermined or ill-posed problem [60, 29]. It was closely related to independent component analysis (ICA) [28, 68, 67] until the late nineties, in which methods like sparse component analysis [6, 23] or non-negative matrix factorization [81, 98, 66] appeared. In general, methods for source separation generally seek to narrow the set of possible solutions in a way that is unlikely to exclude the desired solution and, hence, existing approaches rely on independence, sparsity or structural assumptions.

However, most classical approaches for source separation assume a fixed and known number of latent sources. In this Thesis, we address signal processing problems from a machine learning perspective. We encode our independence assumptions in a probabilistic model and develop inference algorithms to recover the underlying sequences that combine to form the observed signal. We adopt a BNP approach in order to avoid the model selection step and let the inference procedure estimate the number of independent sequences that best explain the data.

Although our models are general enough to be applied in a large variety of problems, we focus on two specific applications: power disaggregation and blind multiuser channel estimation. We describe these applications below.

- The power disaggregation problem consists in estimating the power draw of each individual device given the aggregated whole-home power consumption signal. 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 [31, 96]. 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.

  This problem has already been tackled in other works [79, 76, 71]. However, up to our knowledge, all previous works consider that the number of devices is known, which may be a limitation when applied to houses that do not fit these assumptions. We address this limitation by placing a BNP prior that controls the number of active devices. Furthermore, we also infer the number of states in which devices can be without restricting the precise number of states to be bounded.

- When digital symbols are transmitted over frequency-selective channels, inter-symbol interference (ISI) occurs, degrading the performance of the receiver. To improve the performance, channel estimation is applied to mitigate the effects of ISI. Blind channel estimation involves channel estimation (typically jointly with symbol detection) without the use of pilot symbols (training data), which allows a more efficient communication as the total bandwidth becomes available for the user's data.

  In many modern multiuser communication systems, users are allowed to enter or leave the system at any given time. Thus, the number of active users is an unknown and time-varying parameter, and the performance of the system

depends on how accurately this parameter is estimated over time. We address the problem of blind joint channel parameter and data estimation in a multiuser communication channel in which the number of transmitters is not known. In the literature, we can find several works addressing this problem [141, 55, 143, 12, 10, 11, 134]. However, a characteristic shared by all of them is the assumption of an explicit upper bound for the number of transmitters (users), which may represent a limitation in some scenarios. Our BNP approach naturally avoids this limitation by assuming instead an unbounded number of transmitters. Furthermore, we do not restrict our approach to memoryless channels, which also differs from the existing approaches.

Although machine learning and BNP techniques have already been applied to communication problems (see, e.g., [100, 26]), there are still many other problems in which these methods can help improve the performance of classical algorithms. One of the goals of this Thesis is to push in that direction.

## 1.3 Time Series Modeling

We are interested in modeling temporal sequences by including a set of hidden variables and assuming that the observations are conditionally independent given these hidden variables. We assume that the state space of the latent variables is discrete with known dimensionality.[1] These properties naturally lead us to hidden Markov models (HMMs). In this section, we briefly review the HMM, the factorial hidden Markov model (FHMM), and the finite state machine (FSM), which are the basic building blocks that we use throughout the Thesis. In Section 1.3.4, we review some nonparametric extensions of classical time series models.

### 1.3.1 Hidden Markov Models

HMMs characterize time varying sequences with a simple yet powerful latent variable model [17, 104]. HMMs have been a major success story in many fields

[1] In some applications, like power disaggregation, we may also be interested in learning the number of hidden states. See Chapter 3 for further details.

involving complex sequential data, including speech [103] and handwriting [92] recognition, computational molecular biology [16], natural language processing [80], and digital communications [124].

In HMMs, the observed discrete time sequence $\{\mathbf{y}_t\}_{t=1}^T$ is assumed to depend on a hidden sequence of variables, $\{s_t\}_{t=1}^T$, where $T$ is the number of time steps. Each hidden variable belongs to a discrete space, i.e., $s_t \in \{0, 1, \ldots, Q-1\}$, where $Q$ stands for the number of states of the Markov model. Figure 1.1 represents the corresponding graphical model.

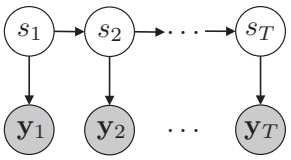



Figure 1.1: Graphical representation of the HMM.

An HMM is completely characterized by: (i) the initial state probabilities, which contain the probabilistic information of the hidden state $s_t$ at time $t = 1$; (ii) the transition probabilities, that describe how the latent states $s_t$ evolve with time; and (iii) the emission probabilities, which describe the likelihood of each observation $\mathbf{y}_t$ given the hidden state $s_t$. Thus, the probability distribution over $\{s_t, \mathbf{y}_t\}$ can be written as

$$p(\{s_t, \mathbf{y}_t\}_{t=1}^T) = \underbrace{p(s_1)}_{\text{(i)}} \times \prod_{t=2}^{T} \underbrace{p(s_t|s_{t-1})}_{\text{(ii)}} \times \prod_{t=1}^{T} \underbrace{p(\mathbf{y}_t|s_t)}_{\text{(iii)}}. \tag{1.1}$$

Due to the discrete nature of the hidden states, the initial state probabilities $p(s_1 = q)$ can be described by a vector $\boldsymbol{\pi} \in [0, 1]^Q$. The transition probabilities $p(s_t = k|s_{t-1} = q)$ are denoted by $a_{qk}$ and can be stored in a $Q \times Q$ matrix $\mathbf{A}$, whose rows are denoted by $\mathbf{a}_q$ $(q = 0, \ldots, Q-1)$. Hence, $\mathbf{a}_q = [a_{q0}, \ldots, a_{q(Q-1)}]$ corresponds to the transition probability vector from state $q$. The emission probabilities $p(\mathbf{y}_t|s_t = q)$, or likelihood terms, are application-dependent, and they typically depend on some parameters $\boldsymbol{\Phi}_q$.

Inference in HMMs involves estimating the sequence of hidden states $\{s_t\}_{t=1}^T$ given the observed signal. This is a problem that can be solved with complexity

scaling as $\mathcal{O}(TQ^2)$ using the forward-backward algorithm. When the parameters $\boldsymbol{\pi}$, $\mathbf{A}$ and $\boldsymbol{\Phi}_q$ that govern the transition and emission probabilities are also unknown, exact inference can still be carried out using an expectation maximization (EM) procedure [32], whose particularization for HMMs is also known as the Baum-Welch algorithm [17]. The EM procedure alternates between the E step, which fixes the current parameters and computes the posterior probabilities over the hidden states $s_t$ via the forward-backward algorithm, and the M step, which uses these probabilities to maximize the expected log-likelihood of the observations as a function of the parameters.

In a fully Bayesian context, the parameters $\boldsymbol{\pi}$, $\mathbf{A}$ and $\boldsymbol{\Phi}_q$ are also treated as random variables, and standard Bayesian inference techniques (e.g., Gibbs sampling) can be applied instead of the EM procedure.

### 1.3.2 Factorial Hidden Markov Models

FHMMs represent the observed time series with several independent parallel HMMs [48]. These parallel HMMs can be seen as several independent causes affecting the observations. We denote by $s_{tm}$ the state of Markov chain $m$ at time instant $t$, for $m = 1, \ldots, M$ and $t = 1, \ldots, T$. Figure 1.2 shows the corresponding graphical model.

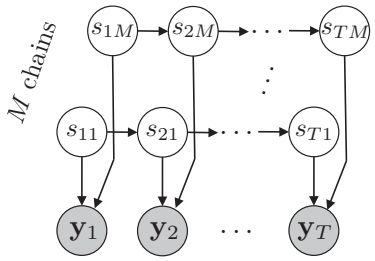



Figure 1.2: Graphical representation of the FHMM.

In an FHMM, the probability distribution over the observations and the latent states is given by

$$p(\{s_{tm}, \mathbf{y}_t\}) = \prod_{m=1}^{M} \underbrace{p(s_{1m})}_{\text{(i)}} \times \prod_{m=1}^{M} \prod_{t=2}^{T} \underbrace{p(s_{tm}|s_{(t-1)m})}_{\text{(ii)}} \times \prod_{t=1}^{T} \underbrace{p(\mathbf{y}_t|s_{t1}, \ldots, s_{tM})}_{\text{(iii)}}. \quad (1.2)$$

In this case, the independence of the parallel Markov chains ensures that the initial state probabilities (i) and the transition probabilities (ii) factorize over $m$. Each observation $\mathbf{y}_t$ depends on all the hidden chains through the likelihood term (iii), which induces dependencies across $m$ in the posterior distribution of the hidden states $\{s_{tm}\}$ given the observations.

In this model, each Markov chain $m$ can be described with a different transition probability matrix $\mathbf{A}^m$ of size $Q \times Q$, such that the probability $p(s_{tm} = k|s_{(t-1)m} = q) = a^m_{qk}$. (Note that $Q$ can also be different in each of the Markov chains, but for simplicity we assume a constant value for the number of states across chains.)

The FHMM can be interpreted as a single HMM in which each hidden state $s_t$ can take $Q^M$ different values. Under this equivalent single HMM, the corresponding transition matrix contains $Q^{2M}$ elements. In particular, this transition matrix can be obtained as the Kronecker product of the transition probability matrices of the FHMM, i.e.,

$$\mathbf{A}_{\text{HMM}} = \bigotimes_{m=1}^{M} \mathbf{A}^m_{\text{FHMM}}. \tag{1.3}$$

Hence, the FHMM can be understood as a simplification of a hidden state transition matrix with $Q^{2M}$ elements into $M$ transition matrices, each with $Q^2$ elements.

Exact inference in FHMMs cannot be carried out with an EM (or Baum-Welch) algorithm due to its computational complexity. As in standard HMMs, the M step is simple and tractable. However, the combinatorial nature of the hidden state representation makes the E step computationally intractable. The naïve exact algorithm which consists of translating the FHMM into the equivalent HMM before running the forward-backward algorithm has complexity $\mathcal{O}(TQ^{2M})$, although it can be reduced to $\mathcal{O}(TMQ^{M+1})$ by exploiting the structure of the model [48]. This exponential time complexity makes exact inference computationally unfeasible.

As in many other intractable systems, approximate inference can be carried out using either Markov chain Monte Carlo (MCMC) methods [109] or variational inference algorithms [73]. Within the MCMC approaches, Gibbs sampling or blocked Gibbs sampling are the standard methods of choice, whereas mean field or structured variational methods are the most common approaches for variational

inference [48].

### 1.3.3 Finite State Machines

FSMs have been applied to a huge variety of problems, including biology (e.g., neurological systems), artificial intelligence (e.g., speech modeling), control applications, communications and electronics [8, 137]. In fact, the HMM is a particularization of the FSM. In its more general form given in [64], an FSM is defined as an abstract machine consisting of:

- A set of states, including the initial state and final state. Variants of this include machines having multiple initial states and multiple final states.
- A set of discrete or continuous input symbols or vectors.
- A set of discrete or continuous output symbols or vectors.
- A state transition function, which takes the current input event and the previous state and returns the next state.
- An emission function, which takes the current state and the input event and returns an output event.

A deterministic FSM is one where the transition and emission functions are deterministic. That is, the next state and output events are completely determined by the current state and input event. In contrast, a stochastic FSM is one where the transition and/or emission functions are probabilistic. As an example, in the case of HMMs, the emission function is probabilistic, and the states are not directly observable through the output events. Instead, each state produces one of the possible output events with a certain probability. Similarly, in the HMM, the states evolve according to some transition probabilities.

In this Thesis, we focus on stochastic finite-memory FSMs, although we refer to this class of machines simply as FSMs for brevity. In these FSMs, the next state only depends on a finite number of previous input events, i.e., the current state can be represented as the vector containing the last input events. More

formally, the FSM relies on a finite memory $L$ and a finite alphabet $\mathcal{X}$. Each new input $x_t \in \mathcal{X}$ produces a deterministic change in the state of the FSM, and a stochastic observable output $\mathbf{y}_t$. The next state and the output depend on the current state and the input. The graphical model for an FSM with $L = 2$ is depicted in Figure 1.3.

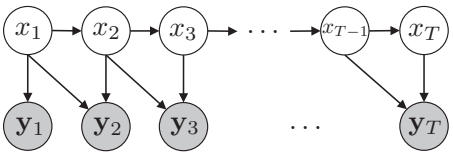



Figure 1.3: Graphical representation of an FSM with $L = 2$.

Under this model, the probability distribution over the inputs $x_t$ and the observations $\mathbf{y}_t$ can be written as

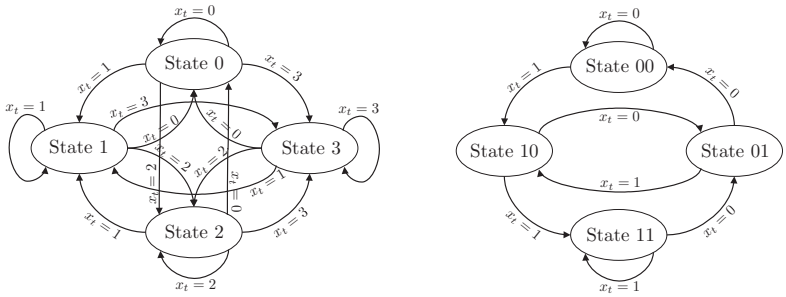
$$p(\{x_t, \mathbf{y}_t\}_{t=1}^T) = \underbrace{p(x_1)}_{\text{(i)}} \times \prod_{t=2}^{T} \underbrace{p(x_t|x_{t-1})}_{\text{(ii)}} \times \prod_{t=1}^{T} \underbrace{p(\mathbf{y}_t|x_t, \ldots, x_{t-L+1})}_{\text{(iii)}}. \tag{1.4}$$

The FSM model also requires the specification of the initial state, which is defined by the inputs $x_{2-L}, \ldots, x_0$. This model differs from the standard HMM in two ways. First, in many cases, the transition probability $p(x_t|x_{t-1})$ does not depend on the previous input $x_{t-1}$, i.e., $p(x_t|x_{t-1}) = p(x_t)$. Nevertheless, we maintain this dependency because it better fits the applications in this Thesis. Second, the likelihood of each observation $\mathbf{y}_t$ depends not only on $x_t$, but also on the previous $L - 1$ inputs.

Similarly to the FHMM, the FSM can also be converted into a single standard HMM. In this equivalent HMM, each state $s_t$ can be expressed as the $L$-vector containing the last $L$ inputs, i.e.,

$$s_t = [x_t, x_{t-1}, \ldots, x_{t-L+1}], \tag{1.5}$$

therefore yielding $Q = |\mathcal{X}|^L$ states. In this case, the $|\mathcal{X}|^L \times |\mathcal{X}|^L$ transition probability matrix $\mathbf{A}$ of the corresponding equivalent HMM is a sparse matrix that contains $|\mathcal{X}|$ non-zero elements per row and column, since most of the transitions




(a) (b)

Figure 1.4: (a) State diagram of an HMM with $Q = 4$ states. Each state is completely determined by the last input event $x_t$, and all transitions among states are allowed. (b) State diagram of an FSM with $L = 2$ and $\mathcal{X} = \{0, 1\}$. Each state can be represented by the vector containing the last $L$ input events. Each state can only transition to other $|\mathcal{X}| = 2$ states, depending on the input event $x_t$. Hence, not all transitions among states are allowed, but it is possible to reach any other state in the graph after $L = 2$ transitions.

are not allowed (each state can only transition to other $|\mathcal{X}|$ states). See Figure 1.4 for a state diagram of an HMM and an FSM.

Exact inference in FSMs is computationally intractable. The E step of the naïve EM approach in the equivalent HMM has computational complexity scaling as $\mathcal{O}(T|\mathcal{X}|^{2L})$, although it can be reduced to $\mathcal{O}(T|\mathcal{X}|^{L+1})$ by exploiting the sparsity of the equivalent transition matrix. However, the exponential dependency on $L$ makes exact inference intractable for moderately large values of $L$.

As in FHMMs, approximate inference schemes are needed under the FSM model, being MCMC methods or variational inference algorithms the most common methods of choice.

### 1.3.4 Bayesian Nonparametrics for Time Series

BNPs have appeared as a replacement of classical finite-dimensional prior distributions with general stochastic processes, allowing an open-ended number of degrees of freedom in a model [97]. We refer to Chapter 2 for a more detailed presentation of BNP models.

In the literature, many nonparametric extensions of standard time series mod-

els can be found. Some examples of BNP models for time series are:

- An HMM with an infinite number of latent states is developed in [18, 119]. This model is known as infinite hidden Markov model (IHMM). As the authors rely on the hierarchical Dirichlet process (HDP) to define their nonparametric model, it is also referred to as HDP-HMM. In the IHMM, the number of states is treated in a nonparametric fashion, i.e., $Q$ is allowed to tend to infinity. Even though the number of states is potentially infinite, for a finite value of $T$ only a finite number of them are present.

- An extension of the IHMM is the sticky HDP-HMM [45], which includes a bias term to artificially increase the self-transition probability of states. This has the advantage of avoiding redundant states to be created during inference.

- Hidden semi-Markov models (HSMMs) can also avoid the rapid-switching problem and geometric duration of the states in HMMs. A nonparametric explicit-duration HSMM can be found in [71]. This is an extension of the IHMM that explicitly introduces the time duration of states as hidden variables of the model.

- The reversible IHMM presented in [99] makes use of a hierarchy of gamma processes, instead of the HDPs, to define a prior over an infinite transition matrix that results reversible.

- The infinite factorial hidden Markov model (IFHMM) in [130] considers an infinite number of parallel Markov chains in an FHMM and, therefore, the number of independent causes that influence the observations is treated nonparametrically. The IFHMM is based on the Markov Indian buffet process (MIBP), which assumes binary states (i.e., $Q = 2$), being state 0 the inactive state. In the IFHMM, there is an infinite number of parallel Markov chains, but most of them are in the inactive state and only a finite subset of the chains become active.

- Hierarchical hidden Markov models (HHMMs) consider a hierarchy of standard HMMs, in which latent states are themselves HHMMs, which contain substates, and can emit strings of observations [41]. The nonparametric generalization of the HHMM is the infinite hierarchical hidden Markov model (IHHMM), which considers an infinite number of levels in the hierarchy [58].

- For Markov switching processes and switching linear dynamical systems, a nonparametric construction is presented in [44]. For instance, in switching linear dynamical systems, a potentially infinite number of dynamical models can be assumed, and transitions from one dynamical model to another one can occur at any time during the observation period.

## 1.4 Contributions

The contributions of this Thesis have also been or will be partially published in [126, 125, 127, 113, 128]. They correspond to extensions of existing BNP models with applications to the problems of power disaggregation and blind multiuser joint channel estimation and symbol detection. We summarize our contributions below.

### 1.4.1 Infinite Factorial Unbounded-State HMM

The IHMM in [119] considers an HMM with a potentially infinite cardinality of the state space. The IFHMM in [130] considers instead an infinite number of binary-state HMMs. The first contribution of this Thesis is the development of a non-binary IFHMM, that is, an FHMM in which the number of Markov chains $M$ is infinite and the cardinality of the state space can take any arbitrary value, $Q$. We develop two MCMC-based inference algorithms for this model, and also a structured variational inference algorithm. We show that the non-binary IFHMM provides more accurate results than its binary counterpart in problems in which several states can be taken by the latent independent causes.

As a second contribution, we extend the non-binary IFHMM in order to be

able to additionally infer the number of states, $Q$. We develop our model without restricting $Q$ to be bounded, and hence we refer to it as infinite factorial unbounded-state hidden Markov model (IFUHMM). We develop an inference algorithm for this model based on reversible jump Markov chain Monte Carlo (RJMCMC) techniques [52], and show that the IFUHMM can properly infer both the number of parallel chains $M$ and the number of states $Q$ in the factorial model.

We apply these models to the power disaggregation and the multiuser channel estimation problems. For power disaggregation, we show that the non-binary IFHMM and the IFUHMM provide better results when compared to the standard (parametric) FHMM, and also when compared to the binary IFHMM. More importantly, our fully blind IFUHMM does not require any prior information about the number of active devices in a house, nor specific prior information about the behavior of individual devices.

For blind multiuser channel estimation, we apply the IFUHMM to detect an unbounded number of users with an unbounded channel length. Up to our knowledge, this constitutes the first attempt to simultaneously detect both the number of users and the channel length in a multiuser communication scheme. Our inference algorithm provides the dispersive channel model for each user and a probabilistic estimate for each transmitted symbol in a fully blind manner, i.e., without the need of transmitting a sequence of pilot symbols. The obtained results are promising, opening a new research challenge in applying BNP tools to digital communication problems.

### 1.4.2 Infinite Factorial Finite State Machines

When applied to communication problems, the IFUHMM suffers from several limitations. These limitations arise from the fact that the model does not take into account some additional prior knowledge of digital communication systems. For instance, the IFUHMM allows transitions among all the states of each HMM, which does not fit the channel state of transmitters, in which not all transitions among states are possible.

In order to address these limitations, we develop the infinite factorial finite state machine (IFFSM) model as another contribution of the Thesis. In the IFFSM model, we incorporate specific prior information about communication systems. The states of an FSM naturally model the channel state information (CSI) of each transmitter. The factorial extension allows us to model several transmitters simultaneously sending bursts of symbols, and the nonparametric version allows us to infer the number of users in the system. We develop an MCMC algorithm based on a combination of slice sampling [95] and particle Gibbs with ancestor sampling (PGAS) [83].

We apply our IFFSM to the multiuser channel estimation problem, showing a more accurate and flexible approach (when compared to the IFUHMM) to estimate the channel coefficients, the number of transmitters and the transmitted symbols in a fully blind manner, with no need of training data.

## 1.5 Organization

The remainder of this Thesis is organized as follows. In Chapter 2, we review the basics of some BNP models. In particular, we focus on the Dirichlet process (DP), the beta process (BP) and some of their variants that are of interest to build our models.

The rest of chapters are devoted to our contributions. Chapter 3 introduces the non-binary IFHMM and the IFUHMM, as well as the corresponding inference algorithms. In Chapter 4, we introduce the IFFSM model and the associated inference algorithm.

The applications of these models can be found in Chapters 5 and 6. In the former, we apply the IFHMM and the IFUHMM to the power disaggregation problem. In the latter, we apply the IFUHMM and the IFFSM for blind multiuser channel estimation and symbol detection.

Finally, Chapter 7 is devoted to the conclusions and future research lines.

# 2

# Review of Bayesian Nonparametrics

## 2.1 Introduction

Most of machine learning problems consist in learning an appropriate set of parameters within a model class from training data. The problem of determining appropriate model classes is referred to as model selection or model adaptation. The model selection problem is of fundamental concern for machine learning practitioners, chiefly for avoidance of over-fitting and under-fitting, but also for discovery of the causes and structures underlying data. Some examples of model selection and adaptation include: selecting the number of clusters in a clustering problem, the number of hidden states in a hidden Markov model (HMM), the number of latent variables in a latent variable model, or the complexity of features in nonlinear regression. Although some recent papers show that Bayesian nonparametric (BNP) priors are not consistent at estimating the number of components, e.g.,

the number of clusters [90], in this Thesis we are not interested in the theoretical properties of BNPs, but in practical applications in which we seek for meaningful results for our problem at hand.

Nonparametric models constitute an approach to model selection, where the model complexity is allowed to grow with data size. For example, fitting a Gaussian mixture with a fixed number of Gaussians is a parametric approach for density estimation. The nonparametric (frequentist) approach would be a Parzen window estimator, which centers a Gaussian at each observation (and hence uses one mean parameter per observation). Nonparametric methods have become popular in classical (non-Bayesian) statistics [138]. As an example, the support vector machine [30] has been widely applied for many classification problems.

BNP methods provide a Bayesian framework for model selection and adaptation using nonparametric models. A Bayesian formulation of nonparametric problems is nontrivial, since the dimensionality of the parameter space in a nonparametric approach is allowed to change with sample size. The BNP solution is to use an infinite-dimensional parameter space, ensuring that only a finite subset of the available parameters is used for any given finite dataset. This subset generally grows with the dataset size. In other words, a BNP model is a Bayesian model on an infinite-dimensional parameter space that can be evaluated on a finite sample using only a finite subset of the available parameters to explain the sample [97].

The parameter space typically consists of random functions or measures. Random functions and measures, and more generally probability distributions on infinite-dimensional random objects, are called stochastic processes. Gaussian processes (GPs), Dirichlet processes (DPs) and beta processes (BPs) are some examples of stochastic processes.

In this chapter, we provide a brief overview of some BNP priors. We focus on models based on the DP, the hierarchical Dirichlet process (HDP) and the BP.

## 2.2 Stochastic Processes

### 2.2.1 Dirichlet Process

The DP is a stochastic process whose realizations are random infinite discrete probability distributions [40]. A DP is completely specified by a base distribution $G_0$ (which is the expected value of the process) and a positive real number $\alpha$ (usually referred to as concentration parameter), which plays the role of an inverse variance.

The weak distribution[1] of the DP is as follows. Let the base distribution $G_0$ be a probability measure on a space $\Phi$. A random probability measure $G$ over $\Phi$ constitutes a DP if, for any finite measurable partition $(A_1, A_2, \ldots, A_r)$ of $\Phi$, the random vector $(G(A_1), G(A_2), \ldots, G(A_r))$ is distributed as a finite-dimensional Dirichlet distribution of the form

$$\begin{aligned}&(G(A_1), G(A_2), \ldots, G(A_r))\\ &\sim \text{Dirichlet}\left(\alpha G_0(A_1), \alpha G_0(A_2), \ldots, \alpha G_0(A_r)\right).\end{aligned} \tag{2.1}$$

We write $G \sim \text{DP}(\alpha, G_0)$ if $G$ is a random probability measure with distribution given by the DP. The first two cumulants of the DP are given by

$$\mathbb{E}[G(A)] = G_0(A), \tag{2.2}$$

and

$$\text{Var}[G(A)] = \frac{G_0(A)(1 - G_0(A))}{\alpha + 1}. \tag{2.3}$$

An explicit representation of a draw $G \sim \text{DP}(\alpha, G_0)$ from a DP can be written as

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}, \tag{2.4}$$

where $\pi_k$ are atom weights, and $\phi_k \in \Phi$ are atom locations defined in the parameter space [116]. The representation in (2.4) ensures that draws from a DP are atomic (discrete) with probability one.

Note that Eq. 2.4 defines an infinite mixture model, i.e., a mixture model with a countably infinite number of clusters. However, since the weights $\pi_k$ decrease

[1] The weak distribution of a stochastic process is the set of all its finite-dimensional marginals.

exponentially quickly, only a small number of clusters will be used to describe any finite dataset. In fact, the expected number of components grows logarithmically with the number of observations.[2] In the DP mixture model, the actual number of clusters describing the data is not fixed, and can be automatically inferred from the data using the usual Bayesian posterior inference framework. The DP mixture model has been widely studied in the literature [13, 38, 86].

We now describe the DP construction as an infinite limit of a finite mixture model with particular Dirichlet priors on mixing proportions. We then turn to the implicit construction through the culinary metaphor of the Chinese restaurant process (CRP). Finally, we give another explicit representation of DPs using the stick-breaking construction.

### Infinite Limit of Finite Mixture Models

Many BNP models can be derived as the infinite limit of finite (parametric) Bayesian models. In particular, the DP mixture model can be derived as the limit of a sequence of finite mixture models, where the number of components in the mixture is taken to infinity [93, 106].

Let us assume a finite mixture model with $K$ components. Let $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_K]$ denote the mixing proportions, and assume that we place a symmetric Dirichlet prior on $\boldsymbol{\pi}$ with parameters $(\alpha/K, \ldots, \alpha/K)$. Let $\phi_k$ denote the parameter observation vector associated with the $k$-th mixture component, and let $\phi_k$ have prior distribution $G_0$. Thus, we have the following model:

$$\begin{aligned} &\boldsymbol{\pi}|\alpha \sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K), \\ &z_i|\boldsymbol{\pi} \sim \boldsymbol{\pi}, \\ &\phi_k|G_0 \sim G_0, \\ &y_i|z_i, \{\phi_k\}_{k=1}^K \sim p(y_i|\phi_{z_i}), \end{aligned} \tag{2.5}$$

where $y_i$ denotes the $i$-th observation, and $z_i$ corresponds to its cluster allocation.

[2] This can be relaxed by replacing the DP with a Pitman-Yor process, in which the growth of the number of components follows a power-law property [102].

We define $G^{(K)}$ as the distribution defined by the $K$-component mixture model, i.e., $G^{(K)} = \sum_{k=1}^{K} \pi_k \delta_{\phi_k}$. As $K \to \infty$, for every function $f$ integrable with respect to $G_0$,

$$\int f(\phi) dG^{(K)}(\phi) \to \int f(\phi) dG(\phi), \tag{2.6}$$

which implies that the marginal distribution over the set of observations $y_1, \ldots, y_N$ tends to the distribution of the DP mixture model [69].

### Chinese Restaurant Process

Another representation of infinite dimensional models is based on de Finetti's Theorem. Any infinitely exchangeable[3] sequence $\phi_1, \ldots, \phi_N$ uniquely defines a stochastic process, called the de Finetti measure, that makes all the $\phi_i$'s iid. For some models, it is sufficient to work directly with the $\phi_i$'s and have the underlying stochastic process implicitly defined. This implicit representation of a BNP model is useful in practice, as it may lead to simple and efficient inference algorithms.

The implicit representation of the DP is the Pólya urn scheme [20], which is closely related to a distribution on partitions known as the CRP [7]. Let $\phi_1, \ldots, \phi_N$ be a sequence of iid random variables distributed according to $G$. That is, all variables $\phi_i$ are conditionally independent given $G$, and hence exchangeable. The successive conditional distributions of $\phi_i$ given $\phi_1, \ldots, \phi_{i-1}$ takes the form

$$\phi_i | \phi_1, \ldots, \phi_{i-1}, \alpha, G_0 \sim \sum_{\ell=1}^{i-1} \frac{1}{i-1+\alpha} \delta_{\phi_\ell} + \frac{\alpha}{i-1+\alpha} G_0. \tag{2.7}$$

This expression shows that $\phi_i$ has non-zero probability of being equal to one of the previous draws. This leads to a "rich gets richer" effect, in which the more often a point is drawn, the more likely it is to be drawn in the future. Let $\{\phi_k^\star\}_{k=1}^K$ denote a sequence containing the unique values of variables $\phi_i$. By defining $m_k$ as the number of values $\phi_i$ which are equal to $\phi_k^\star$, we can rewrite (2.7) as

$$\phi_i | \phi_1, \ldots, \phi_{i-1}, \alpha, G_0 \sim \sum_{k=1}^{K} \frac{m_k}{i-1+\alpha} \delta_{\phi_k^\star} + \frac{\alpha}{i-1+\alpha} G_0. \tag{2.8}$$

[3] An infinitely exchangeable sequence is a sequence whose probability is invariant under finite permutations of its first $n$ elements, for all $n \in \mathbb{N}$.

Despite this "richer gets richer" effect, the probability of $\phi_i$ being drawn from $G_0$ (and hence being different to all previous values) is always positive, and proportional to $\alpha$.

Eqs. 2.7 and 2.8 can be interpreted as a Pólya urn model, in which a ball $\phi_i$ is associated with a color $\phi_k^\star$. The balls are drawn from the urn equiprobably. When a ball is drawn, it is placed back in the urn together with a new ball of the same color. In addition, with probability proportional to $\alpha$, a new atom (color) is created by drawing from $G_0$, and a ball of that new color is added to the urn.

Alternatively, this process can also be viewed as the CRP culinary metaphor. The CRP actually defines a distribution over partitions. In the CRP, we consider a Chinese restaurant with an infinite number of tables. Each $\phi_i$ corresponds to a customer who enters the restaurant, while the distinct values $\phi_k^\star$ correspond to the tables at which the customers sit. The $i$-th customer sits at the table indexed by $\phi_k^\star$ with probability proportional to the number of customers $m_k$ already seated there (in which case we set $\phi_i = \phi_k^\star$), or sits at a new table with probability proportional to $\alpha$ (therefore increasing $K$ by one, drawing $\phi_K^\star \sim G_0$, and setting $\phi_i = \phi_K^\star$). See Figure 2.1 for a sketch of the CRP.

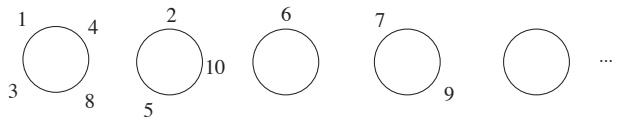



Figure 2.1: Illustration of the Chinese restaurant process. Circles correspond to tables in the restaurant, while numbers correspond to customers sitting on tables.

### Stick-Breaking Construction

Explicit representations of stochastic processes directly describe a random draw from the stochastic process, rather than describing its distribution. A prominent example of an explicit representation is the so-called stick-breaking construction of the DP [116].

The discrete random measure $G$ in (2.4) is uniquely determined by two infinite sequences, $\{\pi_k\}_{k\in\mathbb{N}}$ and $\{\phi_k\}_{k\in\mathbb{N}}$. The stick-breaking representation of the DP

generates these two sequences by drawing $\phi_k \sim G_0$ independently, and by drawing a set of auxiliary variables as

$$v_k \sim \text{Beta}(\alpha, 1) \tag{2.9}$$

for $k = 1, 2, \ldots$ The atom weights $\pi_k$ are then obtained as

$$\pi_k = v_k \prod_{\ell=1}^{k-1} (1 - v_\ell). \tag{2.10}$$

Note that the sequence $\{\pi_k\}_{k\in\mathbb{N}}$ constructed by (2.9) and (2.10) satisfies $\sum_{k=1}^{\infty} \pi_k = 1$ with probability one. We write $\boldsymbol{\pi} \sim \text{GEM}(\alpha)$ if $\boldsymbol{\pi}$ is a random probability measure over the positive integers defined by Eqs. 2.9 and 2.10 (GEM stands for Griffiths, Engen and McCloskey) [101].

We can understand the construction of the sequence $\{\pi_k\}_{k\in\mathbb{N}}$ in this way. Starting with a stick of unit length, at each iteration $k = 1, 2, \ldots$ a piece of relative length $v_k$ is broken off (relative to the current length of the stick). See Figure 2.2 for an illustration.

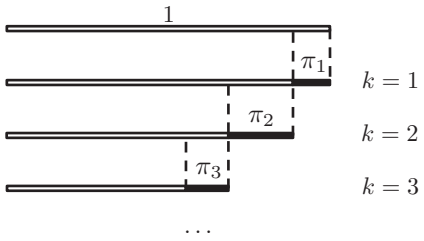



Figure 2.2: Illustration of the stick-breaking construction for the DP.

### 2.2.2 Hierarchical Dirichlet Process

The HDP is a BNP prior which is useful for modeling grouped data [119]. The HDP is a distribution over a set of random probability measures. The process defines a set of random probability measures $G_j$ (one for each group of data), and a global random probability measure $G$. The global measure $G$ is distributed as a DP with concentration parameter $\alpha_1$ and base probability measure $G_0$, i.e.,

$G \sim \mathrm{DP}(\alpha_1, G_0)$. Hence, it can be expressed as

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}. \tag{2.11}$$

The random measures $G_j$ are conditionally independent given $G$, and they are distributed as DPs with concentration parameter $\alpha_2$ and base probability measure $G$, i.e., $G_j \sim \mathrm{DP}(\alpha_2, G)$. Since $G$ is a discrete probability measure with support at the points $\{\phi_k\}_{k\in\mathbb{N}}$, each probability measure $G_j$ has support on the same set of points and, therefore, we can write

$$G_j = \sum_{k=1}^{\infty} \pi_{jk} \delta_{\phi_k}. \tag{2.12}$$

In other words, the atom weights[4] $\pi_{jk}$ are different for each group $j$, but the atom locations are shared across groups $j$.

Let $(A_1, A_2, \ldots, A_r)$ be a measurable partition of $\Phi$. Since $G_j \sim \mathrm{DP}(\alpha_2, G)$ for each $j$, we have by definition that the random vector $(G_j(A_1), G_j(A_2), \ldots, G_j(A_r))$ is distributed as

$$\begin{aligned} &(G_j(A_1), G_j(A_2), \ldots, G_j(A_r)) \\ &\sim \mathrm{Dirichlet}\left(\alpha_2 G(A_1), \alpha_2 G(A_2), \ldots, \alpha_2 G(A_r)\right). \end{aligned} \tag{2.13}$$

This will be useful to establish the connection between the weights $\pi_{jk}$ and the weights of the global measure, $\pi_k$.

Note that Eq. 2.12 defines an infinite mixture model for each group of observations $j$. Furthermore, all groups share the atom locations $\phi_k$ (which do not depend on $j$), and they also share statistical strength on the weights $\pi_{jk}$, as detailed below.

### Infinite Limit of Finite Mixture Models

Similarly to the DP, the HDP can also be derived as the infinite limit of a finite mixture model. In this section, we present a finite model that yields the HDP

[4] We use the notation $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots]$ to refer to the atom weights of the global probability measure $G$, and $\boldsymbol{\pi}_j = [\pi_{j1}, \pi_{j2}, \ldots]$ to refer to the atom weights of each group-level probability measure $G_j$.

mixture model in the infinite limit, although this is not the only finite mixture model with this property [119].

Consider the following collection of finite mixture models, where $\boldsymbol{\pi}$ is a global vector of mixing proportions of length $K$ and $\boldsymbol{\pi}_j$ is a group-specific vector of mixing proportions of the same length:

$$\begin{aligned}
&\boldsymbol{\pi}|\alpha_1 \sim \text{Dirichlet}(\alpha_1/K, \ldots, \alpha_1/K), \\
&\boldsymbol{\pi}_j|\alpha_2, \boldsymbol{\pi} \sim \text{Dirichlet}(\alpha_2\boldsymbol{\pi}), \\
&\phi_k|G_0 \sim G_0, \\
&z_{ji}|\boldsymbol{\pi}_j \sim \boldsymbol{\pi}_j, \\
&y_{ji}|z_{ji}, \{\phi_k\}_{k=1}^K \sim p(y_{ji}|\phi_{z_{ji}}),
\end{aligned} \tag{2.14}$$

where $y_{ji}$ denotes the $i$-th observation within the $j$-th group, and $z_{ji}$ represents its cluster allocation. As $K \to \infty$, the finite mixture model in (2.14) yields the HDP mixture model. The random probability measure $G^{(K)} = \sum_{k=1}^K \pi_k \delta_{\phi_k}$ satisfies Eq. 2.6 when $K \to \infty$, and using standard properties of the Dirichlet distribution, it can be shown that the relationship $G_j^{(K)} \sim \text{DP}(\alpha_2, G^{(K)})$ also holds for finite measures. Hence, as $K \to \infty$, the marginal distribution over the observations $y_{ji}$ induced by the finite mixture model with $K$ components tends to the HDP mixture model.

### Chinese Restaurant Franchise

The Chinese restaurant franchise (CRF) is the implicit distribution for the HDP, in the same way that the CRP is the implicit distribution for the DP. In the CRF, we have a restaurant franchise with as many restaurants as groups of data. All restaurants offer the same menu. At each table of each restaurant one dish is ordered from the menu by the first customer who sits there, and it is shared among all customers who sit at that table. Multiple tables in multiple restaurants can serve the same dish [119].

We denote by $\phi_{ji}$ the parameter corresponding to the $i$-th observation in the $j$-th group. Note that $\phi_{ji}$ can only take a value in $\{\phi_1^\star, \ldots, \phi_K^\star\}$. Random variables

$\phi_k^\star$ form the global menu of dishes, and they are iid distributed following $G_0$. We also introduce variables $\psi_{jt}$ to denote the dish that is served at table $t$ in restaurant $j$.

Additionally, let $t_{ji}$ be the indicator of the table in which customer $i$ in restaurant $j$ is sitting, and let $k_{jt}$ be the indicator of the dish that is served in table $t$ in restaurant $j$. In other words, customer $i$ in restaurant $j$ sat at table $t_{ji}$, while table $t$ in restaurant $j$ serves dish $k_{jt}$. Regarding the notation for the counts, we write $n_{jtk}$ to denote the number of customers in restaurant $j$ at table $t$ eating dish $k$, and $m_{jk}$ to denote the number of tables in restaurant $j$ serving dish $k$. Marginal counts are represented with dots, e.g., $n_{jt\bullet}$ represents the number of customers in restaurant $j$ at table $t$, and $m_{j\bullet}$ represents the number of occupied tables in restaurant $j$.

Following Eq. 2.8, the conditional distribution of $\phi_{ji}$ given the previous variables $\phi_{j1}, \ldots, \phi_{j(i-1)}$ and $G$ can be written as

$$\phi_{ji}|\phi_{j1}, \ldots, \phi_{j(i-1)}, \alpha_2, G \sim \sum_{t=1}^{m_{j\bullet}} \frac{n_{jt\bullet}}{i-1+\alpha_2}\delta_{\psi_{jt}} + \frac{\alpha_2}{i-1+\alpha_2}G. \tag{2.15}$$

In this expression, $G$ is assumed to be given. We now proceed to integrate out the random measure $G$. Since $G \sim \mathrm{DP}(\alpha_1, G_0)$, we can use Eq. 2.8 again to obtain the conditional distribution of $\psi_{jt}$ as

$$\begin{aligned}\psi_{jt}|\psi_{11}, \psi_{12}, \ldots, \psi_{21}, \ldots, \psi_{j(t-1)}, \alpha_1, G_0 \\ \sim \sum_{k=1}^{K} \frac{m_{\bullet k}}{m_{\bullet\bullet}+\alpha_1}\delta_{\phi_k^\star} + \frac{\alpha_1}{m_{\bullet\bullet}+\alpha_1}G_0.\end{aligned} \tag{2.16}$$

Hence, if a term in the first summation of (2.15) is chosen, then we set $\phi_{ji} = \psi_{jt}$ and let $t_{ji} = t$ for the chosen table $t$. Otherwise, we increment $m_{j\bullet}$ by one, draw a new value $\psi_{jm_{j\bullet}}$ and set $\phi_{ji} = \psi_{jm_{j\bullet}}$ and $t_{ji} = m_{j\bullet}$. In order to draw this value of $\psi_{jm_{j\bullet}}$, we use Eq. 2.16. If we draw a new $\psi_{jt}$ via choosing a term in the first summation of (2.16), we set $\psi_{jt} = \phi_k^\star$ and let $k_{jt} = k$ for the chosen $k$. If the second term is chosen, then we increment $K$ by one, draw a new $\phi_K^\star \sim G_0$ and set $\psi_{jt} = \phi_K^\star$ and $k_{jt} = K$.

### Stick-Breaking Construction

The HDP also admits an explicit representation through the stick-breaking construction [119]. In order to derive this construction, we first need to establish the connection between the global stick lengths $\pi_k$ and the group-specific stick lengths $\pi_{jk}$.

Let $K_\ell = \{k : \phi_k \in A_\ell\}$ for $\ell = 1, \ldots, r$. If $G_0$ is a non-atomic measure, then the values $\phi_k$ are almost surely distinct, so any partition of the positive integers $(K_1, \ldots, K_r)$ corresponds to some partition of $\Phi$. Thus, for each $j$, and following (2.13), we have that

$$\left( \sum_{k \in K_1} \pi_{jk}, \ldots, \sum_{k \in K_r} \pi_{jk} \right) \sim \text{Dirichlet} \left( \alpha_2 \sum_{k \in K_1} \pi_k, \ldots, \alpha_2 \sum_{k \in K_r} \pi_k \right), \tag{2.17}$$

for every finite partition of the positive integers. Hence, each $\boldsymbol{\pi}_j$ is independently distributed according to $\boldsymbol{\pi}_j \sim \text{DP}(\alpha_2, \boldsymbol{\pi})$, where $\boldsymbol{\pi}$ is interpreted here as a probability measure on the positive integers.

Variables $\pi_k$ of the global DP can be obtained through the stick-breaking construction in Eq. 2.10, yielding

$$\pi_k = v_k \prod_{\ell=1}^{k-1} (1 - v_\ell), \tag{2.18}$$

where $v_k \sim \text{Beta}(1, \alpha_1)$. Using (2.17), we can obtain that

$$v_{jk} \sim \text{Beta} \left( \alpha_2 \pi_k, \alpha_2 \left( 1 - \sum_{\ell=1}^{k} \pi_\ell \right) \right) \tag{2.19}$$

and

$$\pi_{jk} = v_{jk} \prod_{\ell=1}^{k-1} (1 - v_{j\ell}). \tag{2.20}$$

Regarding the parameters $\phi_k$ in (2.11), they are independently distributed as $G_0$. This completes the stick-breaking construction for the HDP.

## 2.2.3 Beta Process

The BP was defined in [61] for applications in survival analysis. However, it became popular within the machine learning community later on, when it was

related to the Indian buffet process (IBP) in [53]. The BP is the de Finetti mixing distribution underlying the IBP, in the same way that the DP is the de Finetti mixing distribution of the CRP [120].

A BP is a positive Lévy process whose Lévy measure depends on two parameters: $c$, which is a positive function over the space $\Phi$ that we call the concentration function, and $G_0$, which is a fixed measure on $\Phi$, called the base measure, with $\alpha = G_0(\Phi)$. If $c$ is a constant, it is also called the concentration parameter. We write $G \sim \mathrm{BP}(c, G_0)$ to denote that $G$ is a random measure distributed following a BP.

Assuming that $G_0$ is continuous, the Lévy measure of the BP is

$$\nu(d\phi, d\pi) = c(\phi)\pi^{-1}(1-\pi)^{c(\phi)-1}d\pi G_0(d\phi) \tag{2.21}$$

on $\Phi \times [0, 1]$.

To draw $G \sim \mathrm{BP}(c, G_0)$, we can draw a set of points $(\phi_m, \pi_m) \in \Phi \times [0, 1]$ from a Poisson process with base measure $\nu$, and let

$$G = \sum_{m=1}^{\infty} \pi_m \delta_{\phi_m}. \tag{2.22}$$

Hence, $G$ is a discrete random measure with probability one, similarly to the DP and the HDP. The variables $\pi_m$ are the atom weights, whilst $\phi_m$ are the atom locations. In contrast to the DP, the BP is an unnormalized random measure, which means that the weights $\pi_m$ do not add up to one. Instead, $\sum_{m=1}^{\infty} \pi_m = G(\Phi)$, which is a random variable.

The BP, and more specifically the IBP, are typically used as a BNP approach for latent feature modeling. In latent feature modeling, the properties of each object can be represented by an unobservable vector of latent features, and the observations are generated from a distribution determined by those latent feature values. Eq. 2.22 defines a latent feature model with an infinite number of features, in which the probability of objects having feature $m$ is equal to the weight $\pi_m$, and features are represented by the atom locations $\phi_m$. Since the weights $\pi_m$ decrease exponentially quickly, only a small number of features will be used to

describe any finite dataset *a priori*.[5] In fact, the expected number of features grows logarithmically with the number of observations.

### Infinite Limit of Finite Latent Feature Models

Let us assume a finite feature model with $M$ components and $N$ objects. In this model, $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_M]$ is a vector containing the probabilities of possessing the latent features. The possession of feature $m$ by object $i$ is indicated by a binary variable $s_{im}$. We place a beta prior over $\pi_m$ and a Bernouilli prior over $s_{im}$, i.e.,

$$\begin{aligned} \pi_m|\alpha &\sim \text{Beta}\left(\frac{\alpha}{M}, 1\right), \\ s_{im}|\pi_m &\sim \text{Bernouilli}(\pi_m). \end{aligned} \tag{2.23}$$

If we denote by $\mathbf{S}$ the $N \times M$ matrix containing all the latent variables $s_{im}$, then the probability over $\mathbf{S}$ after integrating out the weights $\pi_m$ can be expressed as

$$p(\mathbf{S}) = \prod_{m=1}^{M} \frac{\frac{\alpha}{M}\Gamma\left(n_m + \frac{\alpha}{M}\right)\Gamma(N - n_m + 1)}{\Gamma\left(N + 1 + \frac{\alpha}{M}\right)}, \tag{2.24}$$

where $n_m = \sum_{i=1}^{N} s_{im}$ [54].

As $M \to \infty$, the probability $p(\mathbf{S})$ vanishes. However, we are not interested in the probability of a single matrix $\mathbf{S}$, but in the probability of any equivalent matrix to $\mathbf{S}$. Two matrices $\mathbf{S}$ and $\mathbf{S}'$ are said to be equivalent if they are equal up to a permutation of their columns. We denote by $[\mathbf{S}]$ the set of matrices that are in the same equivalence class as $\mathbf{S}$. The cardinality of $[\mathbf{S}]$ is $\frac{M!}{\prod_{h=0}^{2^N} M_h!}$, being $M_h$ the number of columns with history $h$ (the history $h$ of the $m$-th column is defined as the vector corresponding to the $m$-th column of matrix $\mathbf{S}$). Hence, the probability distribution on $[\mathbf{S}]$ can be written as

$$p([\mathbf{S}]) = \frac{M!}{\prod_{h=0}^{2^N} M_h!} \prod_{m=1}^{M} \frac{\frac{\alpha}{M}\Gamma\left(n_m + \frac{\alpha}{M}\right)\Gamma(N - n_m + 1)}{\Gamma\left(N + 1 + \frac{\alpha}{M}\right)}. \tag{2.25}$$

We can now take the limit of (2.25) as $M \to \infty$. Doing so yields the following result:

$$\lim_{M\to\infty} p([\mathbf{S}]) = \frac{\alpha^{M_+}}{\prod_{h=1}^{2^N} M_h!} e^{-\alpha H_N} \prod_{m=1}^{M_+} \frac{(N - n_m)!(n_m - 1)!}{N!}, \tag{2.26}$$

[5] This process can be more formally described through a Bernouilli process with base measure $G$.

where $M_+$ is the number of non-zero columns of $\mathbf{S}$ and $H_N$ is the $N$-th harmonic number, i.e., $H_N = \sum_{i=1}^{N} \frac{1}{i}$.

The probability distribution in (2.26) corresponds to the probability given by the IBP, described below. The underlying de Finetti mixing distribution is the BP with $c = 1$ [120].

### Indian Buffet Process

Similarly to the CRP, in this case there is also an implicit construction through culinary metaphor that yields Eq. 2.26. The IBP receives its name due to Indian restaurants in London, which offer buffets with an apparently infinite number of dishes.

In the IBP, $N$ customers enter a restaurant one after another. Each customer encounters a buffet consisting of infinitely many dishes arranged in a line. The first customer starts at the left of the buffet and takes a serving from each dish, stopping after a Poisson($\alpha$) number of dishes as his plate becomes overburdened. The $i$-th customer moves along the buffet, sampling dishes in proportion to their popularity, serving herself with probability $\frac{n_m}{i}$, where $n_m$ is the number of previous customers who have sampled a dish. Having reached the end of all previously sampled dishes, the $i$-th customer then tries a Poisson($\frac{\alpha}{i}$) number of new dishes. We can indicate which customers chose which dishes using a binary matrix $\mathbf{S}$ with $N$ rows and infinitely many columns, where $s_{im} = 1$ if the $i$-th customer sampled the $m$-th dish [53, 54]. In Figure 2.3, we show a representation of an IBP matrix $\mathbf{S}$.

$$\mathbf{S} = \underbrace{\underbrace{\begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1M_+} & 0 & 0 & \cdots \\ s_{21} & s_{22} & \cdots & s_{2M_+} & 0 & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \\ s_{N1} & s_{N2} & \cdots & s_{NM_+} & 0 & 0 & \cdots \end{bmatrix}}_{M_+ \text{ non-zero columns}}}_{M \text{ columns (features)}} \Bigg\} N \text{ observations}$$

Figure 2.3: Illustration of an IBP matrix.

Using $M_{\text{new}}^{(i)}$ to indicate the number of new dishes tried by customer $i$, the

probability of any particular matrix $\mathbf{S}$ being produced by this process is

$$p(\mathbf{S}) = \frac{\alpha^{M_+}}{\prod_{i=1}^{N} M_{\text{new}}^{(i)}!} e^{-\alpha H_N} \prod_{m=1}^{M_+} \frac{(N-n_m)!(n_m-1)!}{N!}. \tag{2.27}$$

Taking into account that in this case there are $\frac{\prod_{i=1}^{N} M_{\text{new}}^{(i)}!}{\prod_{h=1}^{2^N} M_h!}$ matrices in the set $[\mathbf{S}]$ and summing (2.27) over all the matrices in this set, we recover Eq. 2.26.

### Stick-Breaking Construction

The stick-breaking construction of the BP in [117] is an equivalent representation of the IBP prior, useful for some inference algorithms.

In this construction, a sequence of independent random variables $\{v_m\}_{m\in\mathbb{N}}$ is first generated according to

$$v_m \sim \text{Beta}(\alpha, 1), \tag{2.28}$$

and the weights $\pi_m$ are then obtained as

$$\pi_m = \prod_{\ell=1}^{m} v_\ell, \tag{2.29}$$

resulting in a decreasing sequence of probabilities $\pi_m$.

This construction can be understood with the stick-breaking process illustrated in Figure 2.4. Starting with a stick of length 1, at each iteration $m = 1, 2, \ldots$, a piece is broken off at a point $v_m$ relative to the current length of the stick. The variable $\pi_m$ corresponds to the length of the stick just broken off, and the other piece of the stick is discarded.

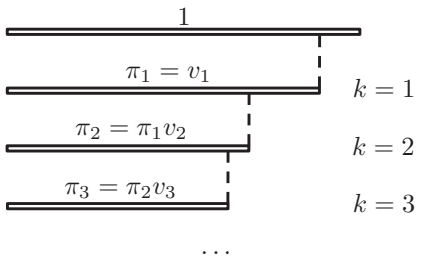



Figure 2.4: Illustration of the stick-breaking construction for the BP.

## 2.3 Markov Indian Buffet Process

The Markov Indian buffet process (MIBP) is a variant of the IBP that is useful for the construction of a factorial hidden Markov model (FHMM) with a potentially infinite number of parallel Markov chains [130]. In the MIBP, we also consider a binary matrix $\mathbf{S}$ with an infinite number of columns. Here, the $t$-th row represents time step $t$ ($t = 1, \ldots, T$), while the $m$-th column represents the states of the $m$-th Markov chain ($m = 1, \ldots, M$). Hence, each element $s_{tm} \in \{0, 1\}$ indicates whether the $m$-th Markov chain is active at time instant $t$.

Similarly to the IBP, we derive the probability over $\mathbf{S}$ as $M \to \infty$ in three ways. First, we derive this probability as the infinite limit of a finite FHMM. Second, we follow a process guided by a culinary metaphor, which in this case is known as MIBP. Third, we give a stick-breaking construction of the process.

### 2.3.1 Infinite Limit of a Finite FHMM

Consider the following FHMM model, in which each Markov chain evolves according to the transition matrix

$$\mathbf{A}^m = \begin{pmatrix} 1 - a^m & a^m \\ 1 - b^m & b^m \end{pmatrix}, \tag{2.30}$$

where

$$p(s_{tm} = j | s_{(t-1)m} = i) = (\mathbf{A}^m)_{ij}. \tag{2.31}$$

We place a beta prior over $a^m$ and $b^m$ with parameters

$$\begin{aligned} a^m &\sim \text{Beta}\left(\frac{\alpha}{M}, 1\right), \\ b^m &\sim \text{Beta}(\beta_0, \beta_1), \end{aligned} \tag{2.32}$$

being $\alpha$, $\beta_0$ and $\beta_1$ hyperparameters of the model. Each Markov chain starts with a dummy zero state $s_{0m} = 0$. The hidden state sequence for chain $m$ is generated by sampling $T$ steps from a Markov chain with transition matrix $\mathbf{A}^m$. Figure 2.5 shows the corresponding graphical model.

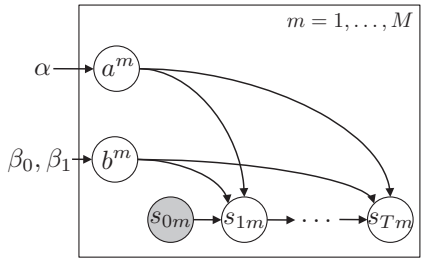



Figure 2.5: Graphical finite model for the MIBP.

Under this model, the probability over matrix $\mathbf{S}$ after integrating out $a^m$ and $b^m$ can be written as

$$p(\mathbf{S}) = \prod_{m=1}^{M} \frac{\frac{\alpha}{M}\Gamma\left(\frac{\alpha}{M} + n_{01}^m\right)\Gamma(1 + n_{00}^m)\Gamma(\beta_0 + \beta_1)\Gamma(\beta_1 + n_{10}^m)\Gamma(\beta_0 + n_{11}^m)}{\Gamma\left(\frac{\alpha}{M} + 1 + n_{00}^m + n_{01}^m\right)\Gamma(\beta_0 + \beta_1 + n_{10}^m + n_{11}^m)\Gamma(\beta_0)\Gamma(\beta_1)}, \quad (2.33)$$

where $n_{qk}^m$ denotes the number of transitions from state $q$ to state $k$ in the $m$-th Markov chain, including the transition from the dummy state to the first state.

Similarly to the IBP, the probability of a single matrix $\mathbf{S}$ is zero when $M \to \infty$. However, we are interested in the probability of the whole equivalence class of $\mathbf{S}$, denoted by $[\mathbf{S}]$, which consists on the set of matrices that are equal to $\mathbf{S}$ after applying a permutation of the columns. The number of matrices in $[\mathbf{S}]$ is equal to $\frac{M!}{\prod_{h=0}^{2^T-1} M_h!}$, being $M_h$ the number of columns with history $h$. Hence, the probability over the whole equivalence class is given by

$$p([\mathbf{S}]) = \frac{M!}{\prod_{h=0}^{2^T-1} M_h!} p(\mathbf{S}). \quad (2.34)$$

In the limit as $M \to \infty$, Eq. 2.34 yields

$$\begin{aligned} \lim_{M\to\infty} p([\mathbf{S}]) = & \frac{\alpha^{M_+}}{\prod_{h=1}^{2^T} M_h!} e^{-\alpha H_T} \\ & \times \prod_{m=1}^{M_+} \frac{(n_{01}^m - 1)!(n_{00}^m)!\Gamma(\beta_0 + \beta_1)\Gamma(\beta_1 + n_{10}^m)\Gamma(\beta_0 + n_{11}^m)}{(n_{00}^m + n_{01})!\Gamma(\beta_0)\Gamma(\beta_1)\Gamma(\beta_0 + \beta_1 + n_{10}^m + n_{11}^m)}, \end{aligned} \quad (2.35)$$

being $M_+$ the number of non-zero columns in $\mathbf{S}$, and $H_T = \sum_{t=1}^{T} \frac{1}{t}$.

### 2.3.2 Culinary Metaphor: The MIBP

We can also derive Eq. 2.35 through a stochastic process analogous to the IBP. In this process, $T$ customers enter an Indian restaurant with an infinitely long

buffet of dishes organized in a line. The first customer enters the restaurant and takes a serving from each dish, starting at the left of the buffet and stopping after a Poisson($\alpha$) number of dishes as his plate becomes overburdened. The $t$-th customer enters the restaurant and starts at the left of the buffet. At dish $m$, she looks at the customer in front of her to see whether she has served herself that dish.

- If the customer in front of her took dish $m$, then the $t$-th customer serves herself that dish with probability $\frac{n_{11}^m+\beta_0}{n_{10}^m+n_{11}^m+\beta_0+\beta_1}$, being $n_{11}^m$ the number of previous customers who took dish $m$ when the customer in front of them took dish $m$, and $n_{10}^m$ the number of previous customers who did not take dish $m$ when the customer in front of them took dish $m$.

- If the customer in front of her did not take dish $m$, then the $t$-th customer serves herself that dish with probability $\frac{n_{01}^m}{1+n_{01}^m+n_{00}^m}$, being $n_{01}^m$ the number of previous customers who took dish $m$ when the customer in front of them did not take dish $m$, and $n_{00}^m$ the number of previous customers who did not take dish $m$ when the customer in front of them did not take dish $m$.

The $t$-th customer then moves on to the next dish and does exactly the same. After the customer has passed all dishes people have previously served themselves from, she tries Poisson($\frac{\alpha}{t}$) new dishes.

If we fill in the entries of the $T \times M$ matrix $\mathbf{S}$ with the number of units that every customer took from every dish, and we denote by $M_{\text{new}}^{(t)}$ the number of new dishes tried by the $t$-th customer, the probability of any particular matrix $\mathbf{S}$ being produced by this process is given by

$$\begin{aligned} p(\mathbf{S}) =& \frac{\alpha^{M_+}}{\prod_{t=1}^{T} M_{\text{new}}^{(t)}!} e^{-\alpha H_T} \\ & \times \prod_{m=1}^{M_+} \frac{(n_{01}^m - 1)!(n_{00}^m)!\Gamma(\beta_0+\beta_1)\Gamma(\beta_1+n_{10}^m)\Gamma(\beta_0+n_{11}^m)}{(n_{00}^m+n_{01})!\Gamma(\beta_0)\Gamma(\beta_1)\Gamma(\beta_0+\beta_1+n_{10}^m+n_{11}^m)}. \end{aligned} \tag{2.36}$$

We can recover Eq. 2.35 by summing over all possible matrices that can be generated using this process that are in the same equivalence class. It is straight-

forward to check that there are exactly $\frac{\prod_{t=1}^{T} M_{\text{new}}^{(t)}!}{\prod_{h=1}^{2^T} M_h!}$ matrices in $[\mathbf{S}]$. Multiplying this by equation (2.36) yields Eq. 2.35.

Note that this construction of the MIBP shows that the effective dimension of the model $M_+$ follows a Poisson($\alpha H_T$) distribution.

### 2.3.3 Stick-Breaking Construction

We can adapt the stick-breaking construction for the IBP in [117] to the MIBP. This construction is useful because it allows simpler inference algorithms [130].

We first find the distribution of the parameters $a^m$, sorted in decreasing order. We introduce the notation $a^{(m)}$ to denote the sorted values of $a^m$, such that $a^{(1)} > a^{(2)} > a^{(3)} > \ldots$ Since the distribution over variables $a^m$ is equal to the distribution of the weights in the standard IBP, we can write that

$$a^{(1)} \sim \text{Beta}(\alpha, 1), \tag{2.37}$$

and

$$p(a^{(m)}|a^{(m-1)}) \propto (a^{(m-1)})^{-\alpha}(a^{(m)})^{\alpha-1}\mathbb{I}(0 \leq a^{(m)} \leq a^{(m-1)}), \tag{2.38}$$

being $\mathbb{I}(\cdot)$ the indicator function, which takes value one if its argument is true and zero otherwise. Note that this construction is equivalent to the stick-breaking construction for the IBP described in Section 2.2.3.

With respect to variables $b^{(m)}$, i.e., variables $b^m$ reordered to match the corresponding values of $a^{(m)}$, we remark that the Beta($\beta_0, \beta_1$) prior in (2.32) does not depend on $m$. Thus, the variables $b^{(m)}$ are also Beta($\beta_0, \beta_1$) distributed.

## 2.4 Inference in BNP Models

We have described two classes of BNP models: mixture models based on the CRP and latent factor models based on the IBP. Both types of models posit a generative probabilistic process of a set of observations that includes hidden structure. In order to analyze data with these models, we need to examine the posterior distribution of the hidden structure given the observations. This gives

us a distribution over the latent structure that tells us which latent structure is likely to have generated our data.

The main computational problem in BNP modeling (as in most of Bayesian statistics) is computing the posterior. For many interesting models including BNP models, the posterior is not available in closed form, therefore requiring an approximation. In this section, we give an overview on some of the most widely-used inference algorithms that approximate the posterior. Details on the inference algorithms for the models developed in this Thesis are given in Chapters 3 and 4.

One of the most widely used posterior inference methods in BNP models are Markov chain Monte Carlo (MCMC) methods. The idea of MCMC methods is to define a Markov chain on the hidden variables that has the posterior of interest as its equilibrium distribution. By drawing samples from this Markov chain, we eventually obtain samples from the posterior. A simple form of MCMC sampling is Gibbs sampling, where the Markov chain is constructed by considering the conditional distribution of each hidden variable given the rest of hidden variables and the observations. The CRP construction is particularly amenable to Gibbs sampling inference, as obtaining these conditional distributions is straightforward. A detailed survey of Gibbs sampling for inference in DP mixture models can be found in [94]. Gibbs sampling for the IBP is described in [54, 78]. For the HDP-HMM, Gibbs sampling can still be applied as described in [45], although beam sampling is a method with better mixing properties, since it allows running forward-filtering backward-sampling (FFBS) steps as part of the inference procedure [129]. Regarding the infinite factorial hidden Markov model (IFHMM), a slice sampling approach [95] that makes use of the stick-breaking construction can be applied to allow for FFBS sweeps [130].

An alternative approach to MCMC methods is variational inference [73]. This approach is based on the idea of approximating the posterior with a simpler family of distributions and searching for the member of that family that is closest to it. Hence, variational methods turn inference into an optimization problem. Unlike

MCMC methods, variational inference algorithms are not guaranteed to recover the posterior, but they are typically faster than MCMC, and convergence assessment is straightforward. These methods have been applied to DP mixture models [21] and BP latent feature models [36]. Variational inference usually operates on the random measure representation through the stick-breaking construction of the DP and the BP.

## 2.5 Applications of BNP Models

The choice of an appropriate stochastic process depends on the problem at hand. GPs have proven to be useful for regression and classification problems, and also for unsupervised non-linear dimensionality reduction [107]. DPs are most commonly applied to clustering problems in which the number of clusters is not known in advance. The HDP has been applied as a nonparametric version of latent Dirichlet allocation [22], in which each document corresponds to a group, and the number of topics is potentially infinite [119]. The IBP has been used as a nonparametric extension of independent component analysis (ICA) [53], as nonparametric latent feature modeling [111, 112], or as a building block for more complex models [57, 140].

Regarding time series modeling, many BNP priors have been developed in the literature (see Section 1.3.4). Two of the most common ones are the HDP-HMM [117], also known as infinite hidden Markov model (IHMM), and the IFHMM [130].

The IHMM makes use of the HDP to define infinite-length transition probability vectors. Under this model, each vector $\boldsymbol{\pi}_j$ corresponds to the transition probability vector from state $j$. All vectors $\boldsymbol{\pi}_j$ share statistical strength through the global weights $\boldsymbol{\pi}$, therefore ensuring that only a finite subset of the available states in the HMM are used for any finite number of observations.

The IFHMM is based on the MIBP matrix, in which columns are Markov chains and rows are time steps. The MIBP places a prior over binary matrices with an infinite number of columns. In the IFHMM, state 0 corresponds to the inactive state, while state 1 is the active state. For a finite dataset, the MIBP construction

ensures that only a finite subset of the parallel Markov chains become active, while the rest of them remain in the inactive state and do not influence the observations.

# 3

# Infinite Factorial Unbounded-State Hidden Markov Model

## 3.1 Introduction

Hidden Markov models (HMMs) characterize time varying sequences with a simple yet powerful latent variable model [104]. HMMs have been a major success story in many fields involving complex sequential data, including speech [103] and handwriting [92] recognition, computational molecular biology [16] and natural language processing [80]. In most of these applications, the model topology is determined in advance and the model parameters are estimated by an expectation maximization (EM) procedure [32], whose particularization is also known as the Baum-Welch algorithm [17]. However, both the standard estimation procedure and the model definition for HMMs suffer from important limitations as not con-

sidering the complexity of the model (making it hard to avoid over or underfitting) and needing to pre-specify the model structure. In [110], the authors proposed an inference algorithm for HMMs based on reversible jump Markov chain Monte Carlo (RJMCMC) techniques [52] 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 hidden Markov models (FHMMs) model the observed time series with independent parallel HMMs [48]. 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.

In this chapter, we build a Bayesian nonparametric (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 or the number of transmitters in a multiuser communication scenario. 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 infinite factorial hidden Markov model (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 hidden Markov model (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 that, for instance, help investors and policy makers understand how the market operates, or 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 [130]. 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 a Markov chain Monte Carlo (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.

## 3.2 Nonbinary Infinite Factorial HMM

The model proposed in this section is a nonbinary extension of the IFHMM developed in [130]. 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 [121], 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.

### 3.2.1 Finite Model

We depict the graphical model for a FHMM in Figure 3.1, 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

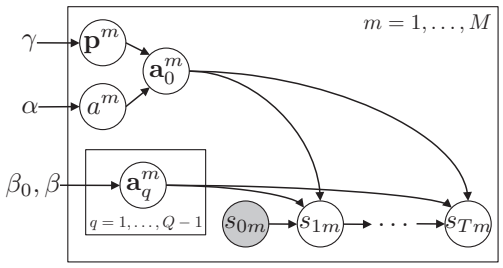



Figure 3.1: Graphical model of the nonbinary finite FHMM.

independently distributed for each Markov chain $m = 1, \ldots, M$. As the variables $s_{tm}$ follow an HMM, we can write that

$$s_{tm}|s_{(t-1)m}, \mathbf{A}^m \sim \mathbf{a}^m_{s_{(t-1)m}}. \tag{3.1}$$

In order to be able to extend the number of parallel chains to infinity, and similarly to the IFHMM [130], 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 $m$-th chain is not active at time $t$. Hence, as shown in Figure 3.1, 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.,

$$a^m|\alpha \sim \text{Beta}\left(1, \frac{\alpha}{M}\right), \tag{3.2}$$

and set the transition probability vector from the inactive state to

$$\mathbf{a}_0^m = \begin{bmatrix} a^m & (1-a^m)p_1^m & \ldots & (1-a^m)p_{Q-1}^m \end{bmatrix}, \tag{3.3}$$

where

$$\mathbf{p}^m|Q, \gamma \sim \text{Dirichlet}(\gamma). \tag{3.4}$$

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

$$\begin{aligned} p(\mathbf{a}_0^m|Q,\alpha,\gamma) &= p(a_{00}^m|\alpha)p(a_{01}^m,\ldots,a_{0(Q-1)}^m|a_{00}^m,\gamma) \\ &= \text{Beta}\left(a_{00}^m\middle|1,\frac{\alpha}{M}\right)\ (1-a_{00}^m)^{2-Q} \\ &\quad\times \text{Dirichlet}\left(\frac{a_{01}^m}{1-a_{00}^m},\ldots,\frac{a_{0(Q-1)}^m}{1-a_{00}^m}\middle|\gamma\right), \end{aligned} \tag{3.5}$$

where the elements of vector $\mathbf{a}_0^m$ are denoted by $a_{0i}^m$, for $i = 0,\ldots,Q-1$. In Eqs. 3.2 and 3.4, $\alpha$ is the concentration parameter, which controls the probability of leaving state 0, and $\gamma$ 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

$$\mathbf{a}_q^m|Q,\beta_0,\beta \sim \text{Dirichlet}(\beta_0,\beta,\ldots,\beta), \tag{3.6}$$

where $\beta_0$ and $\beta$ model the *a priori* information about the transition probabilities from states other than 0.

Similarly to the binary MIBP in [130], we can obtain the probability distribution over the matrix $\mathbf{S}$ after integrating out the transition probabilities, yielding the expression in (3.7), 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, $n_{q\bullet}^m = \sum_{i=0}^{Q-1} n_{qi}^m$.

### 3.2.2 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. 3.7 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 Indian buffet process (IBP) in [54], 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

$$
\begin{aligned}
p(\mathbf{S}|Q,\alpha,\beta_0,\beta,\gamma) &= \int p(\mathbf{S}|\{\mathbf{A}^m\}_{m=1}^M) \prod_{m=1}^{M} (p(\mathbf{A}^m|Q,\alpha,\beta_0,\beta,\gamma) d\mathbf{A}^m) \\
&= \prod_{m=1}^{M} \left[ \frac{\frac{\alpha}{M}\Gamma\left((Q-1)\gamma\right)}{\left(\Gamma(\gamma)\right)^{Q-1}} \frac{\prod_{i=1}^{Q-1}\Gamma(n_{0i}^m+\gamma)}{\Gamma\left(\sum_{i=1}^{Q-1}(n_{0i}^m+\gamma)\right)} \frac{\Gamma(n_{00}^m+1)\Gamma\left(\frac{\alpha}{M}+\sum_{i=1}^{Q-1} n_{0i}^m\right)}{\Gamma\left(n_{0\bullet}^m+1+\frac{\alpha}{M}\right)} \right. \\
&\quad \left. \times \prod_{q=1}^{Q-1} \left( \frac{\Gamma\left(\beta_0+(Q-1)\beta\right)}{\Gamma(\beta_0)\left(\Gamma(\beta)\right)^{Q-1}} \frac{\Gamma(n_{q0}^m+\beta_0)\prod_{i=1}^{Q-1}\Gamma(n_{qi}^m+\beta)}{\Gamma\left(n_{q\bullet}^m+\beta_0+(Q-1)\beta\right)} \right) \right].
\end{aligned} \tag{3.7}
$$

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) 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. 3.8, where $M_+$ stands for the number of nonzero columns, and $H_T$ for the $T$-th harmonic

$$\lim_{M\to\infty} p([\mathbf{S}]|Q,\alpha,\beta_0,\beta,\gamma) = \lim_{M\to\infty} \frac{(Q-1)!}{(Q-N_Q)!N_f} \frac{M!}{\prod_{h=0}^{Q^T-1} M_h!} p(\mathbf{S}|Q,\alpha,\beta_0,\beta,\gamma)$$

$$= \frac{(Q-1)!}{(Q-N_Q)!N_f} \frac{\alpha^{M_+}}{\prod_{h=1}^{Q^T-1} M_h!} e^{-\alpha H_T}$$

$$\times \prod_{m=1}^{M_+} \left[ \frac{\Gamma(n_{00}^m+1)\Gamma\left(\sum_{i=1}^{Q-1} n_{0i}^m\right)}{\Gamma(n_{0\bullet}^m+1)} \frac{\Gamma\left((Q-1)\gamma\right)\prod_{i=1}^{Q-1}\Gamma(n_{0i}^m+\gamma)}{\Gamma\left(\sum_{i=1}^{Q-1}(n_{0i}^m+\gamma)\right)\left(\Gamma(\gamma)\right)^{Q-1}} \right. \tag{3.8}$$

$$\left. \times \prod_{q=1}^{Q-1} \left( \frac{\Gamma\left(\beta_0+(Q-1)\beta\right)}{\Gamma(\beta_0)\left(\Gamma(\beta)\right)^{Q-1}} \frac{\Gamma(n_{q0}^m+\beta_0)\prod_{i=1}^{Q-1}\Gamma(n_{qi}^m+\beta)}{\Gamma\left(n_{q\bullet}^m+\beta_0+(Q-1)\beta\right)} \right) \right].$$

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.[1] 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 [130] by setting $Q=2$.

### 3.2.3 Culinary Metaphor

Following a similar procedure as in [130] for the IFHMM, we can derive the expression for $\lim_{M\to\infty} p([\mathbf{S}]|Q,\alpha,\beta_0,\beta,\gamma)$ in Eq. 3.8 from a culinary metaphor analogous to the IBP. In this process, $T$ customers enter sequentially a restaurant with an infinitely long buffet of dishes. The first customer starts at the left of the buffet and takes a serving from each dish, taking (possibly different) quantities for each one and stopping after a Poisson($\alpha$) number of dishes. The number of units $q \in \{1,\ldots,Q-1\}$ she takes is independently sampled for each dish from a uniform

[1] A sequence is Markov exchangeable if its distribution is invariant under permutations of the transitions.

distribution.

The $t$-th customer enters the restaurant and starts at the left of the buffet. At dish $m$, she looks at the customer in front of her to see how many units she has taken from that dish and proceeds as follows:

- If the $(t-1)$-th customer did not take the $m$-th dish, she serves herself that dish with probability $\frac{\sum_{i=1}^{Q-1} n_{0i}^m}{n_{0\bullet}^m+1}$, where $n_{0i}^m$ is the number of previous customers who took $i$ units from dish $m$ when the person in front of them did not take the dish $m$. If she does, the number of units she takes is given by $i$ with probability $\frac{\gamma+n_{0i}^m}{\sum_{j=1}^{Q-1}(\gamma+n_{0j}^m)}$ $(i=1,\ldots,Q-1)$.
- If the $(t-1)$-th customer took $q$ units from the $m$-th dish, the $t$-th customer either serves herself $i$ units with probability given by $\frac{\beta+n_{qi}^m}{\beta_0+(Q-1)\beta+\sum_{j=0}^{Q-1}(n_{qj}^m)}$ (with $i=1,\ldots,Q-1$), or she does not take that dish with probability $\frac{\beta_0+n_{q0}^m}{\beta_0+(Q-1)\beta+\sum_{j=0}^{Q-1}(n_{qj}^m)}$, where $n_{qi}^m$ is the number of previous customers who took $i$ units from dish $m$ when the person in front of them took $q$ units.

The $t$-th customer then moves on to the next dish and repeats the above procedure. After having passed all dishes people have previously served themselves from, she takes independent quantities $q \sim \text{Uniform}\left(\frac{1}{Q-1},\ldots,\frac{1}{Q-1}\right)$ from a Poisson$(\frac{\alpha}{t})$ number of new dishes.

If we fill in the entries of the $T \times M$ matrix $\mathbf{S}$ with the number of units that every customer took from every dish, and we denote with $M_{\text{new}}^{(t)}$ the number of new dishes tried by the $t$-th customer, the probability of any particular matrix $\mathbf{S}$ being produced by this process is given in Eq. 3.9.

There are $\frac{(Q-1)!}{(Q-N_Q)!N_f}\prod_{t=1}^{T} M_{\text{new}}^{(t)}!/\prod_{h=1}^{Q^T-1} M_h!$ matrices in the same equivalence class as $\mathbf{S}$, and therefore we can recover Eq. 3.8 by summing over all possible matrices lying in the set $[\mathbf{S}]$ generated by this process.

### 3.2.4 Stick-Breaking Construction

Since the representation of the model above is similar to the binary MIBP in [130], a stick-breaking construction is also readily available. This construction allows

$$p(\mathbf{S}|Q,\alpha,\beta_0,\beta,\gamma) = \frac{\alpha^{M_+}}{\prod_{t=1}^{T} M_{\text{new}}^{(t)}!} e^{-\alpha H_T} \times \prod_{m=1}^{M_+} \left[ \frac{\Gamma(n_{00}^m+1)\Gamma\left(\sum_{i=1}^{Q-1} n_{0i}^m\right)}{\Gamma(n_{0\bullet}^m+1)} \frac{\Gamma\left((Q-1)\gamma\right)\prod_{i=1}^{Q-1}\Gamma(n_{0i}^m+\gamma)}{\Gamma\left(\sum_{i=1}^{Q-1}(n_{0i}^m+\gamma)\right)\left(\Gamma(\gamma)\right)^{Q-1}} \times \prod_{q=1}^{Q-1} \left( \frac{\Gamma\left(\beta_0+(Q-1)\beta\right)}{\Gamma(\beta_0)\left(\Gamma(\beta)\right)^{Q-1}} \frac{\Gamma(n_{q0}^m+\beta_0)\prod_{i=1}^{Q-1}\Gamma(n_{qi}^m+\beta)}{\Gamma\left(n_{q\bullet}^m+\beta_0+(Q-1)\beta\right)} \right) \right]. \tag{3.9}$$

using a combination of slice sampling and dynamic programming for inference.

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 [117], we can write

$$p(c^{(1)}) = \text{Beta}(\alpha, 1), \tag{3.10}$$

and

$$p(c^{(m)}|c^{(m-1)}) \propto (c^{(m)})^{\alpha-1}\mathbb{I}(0 \leq c^{(m)} \leq c^{(m-1)}), \tag{3.11}$$

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. 3.6 and 3.4, repectively, which are independent of $m'$, the sorted variables $\mathbf{a}_q^{(m)}$ and $\mathbf{p}^{(m)}$ have also the same prior distributions.

## 3.3 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 $\mathbf{Y}$, composed of $T$ vectors $\mathbf{y}_t$ 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 $\mathbf{y}_t$ 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 $\mathbf{S}$. These are straightforward conditions that do not limit the applicability of the proposed model.

We propose two similar Gaussian observation models. The choice of either of them should depend on the specific application. These two likelihood models are depicted in Figures 3.2 and 3.3. In both of them, $\mathbf{Y}$ is distributed as a Gaussian random matrix with independent elements, each with variance $\sigma_y^2$, i.e.,

$$\begin{aligned} p(\mathbf{Y}|\mathbf{S}, \mathbf{\Phi}_1, \ldots, \mathbf{\Phi}_{Q-1}, \sigma_y^2) = \frac{1}{(2\pi\sigma_y^2)^{\frac{TD}{2}}} \exp\left\{ -\frac{1}{2\sigma_y^2} \right. \\ \left. \times \operatorname{trace}\left[ \left( \mathbf{Y} - \sum_{q=1}^{Q-1} \mathbf{Z}_q \mathbf{\Phi}_q \right)^\top \left( \mathbf{Y} - \sum_{q=1}^{Q-1} \mathbf{Z}_q \mathbf{\Phi}_q \right) \right] \right\}, \end{aligned} \tag{3.12}$$

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 $\mathbf{\Phi}_q$ are $M \times D$ matrices, with $M$ being the number of columns in $\mathbf{S}$. Thus, the mean value for $\mathbf{y}_t$ depends on the additive contribution of all chains at time instant $t$.

The difference between both models is the prior over the matrices $\mathbf{\Phi}_q$ and the noise variance $\sigma_y^2$. In Model #1 (Figure 3.2), $\sigma_y^2$ is a fixed hyperparameter, and

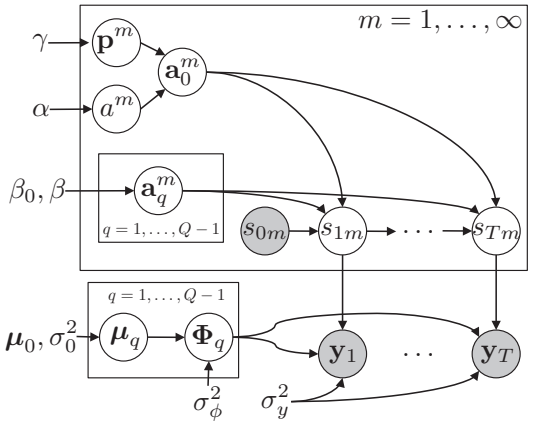



Figure 3.2: Graphical observation model #1 for the nonbinary IFHMM.

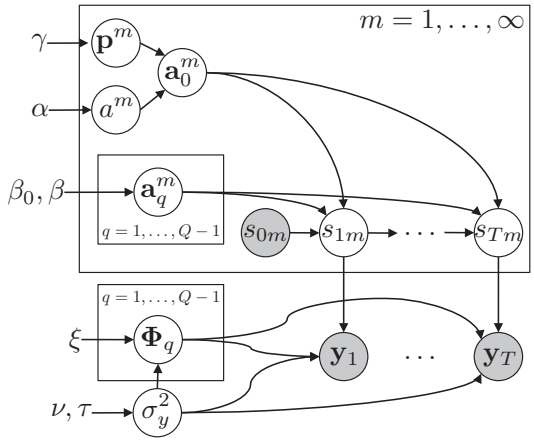



Figure 3.3: Graphical observation model #2 for the nonbinary IFHMM.

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

$$\begin{aligned} p(\boldsymbol{\Phi}_q|\boldsymbol{\mu}_q, \sigma_\phi^2) = \frac{1}{(2\pi\sigma_\phi^2)^{\frac{DM}{2}}} \exp\left\{ -\frac{1}{2\sigma_\phi^2} \right. \\ \left. \times \operatorname{trace}\left[ \left( \boldsymbol{\Phi}_q - \mathbf{1}_M \boldsymbol{\mu}_q^\top \right)^\top \left( \boldsymbol{\Phi}_q - \mathbf{1}_M \boldsymbol{\mu}_q^\top \right) \right] \right\}, \end{aligned} \tag{3.13}$$

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 column vectors with mean $\boldsymbol{\mu}_0$ and covariance matrix $\sigma_0^2 \mathbf{I}_D$, i.e.,

$$p(\boldsymbol{\mu}_q|\sigma_0^2) = \mathcal{N}(\boldsymbol{\mu}_0, \sigma_0^2 \mathbf{I}_D), \tag{3.14}$$

where $\mathbf{I}_D$ stands for the identity matrix of size $D$. We include the hyperparameter

$\sigma_\phi^2$ 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 $\sigma_\phi^2/\sigma_0^2$, $\mathbf{\Phi}_q$ is close to its mean and therefore the parameters for any particular state $q$ are similar through all the chains. For larger values of $\sigma_\phi^2/\sigma_0^2$, the parameters may seem decorrelated for the same state at different chains.

In Model #2 (Figure 3.3), we place a normal-inverse-gamma prior over $\sigma_y^2$ and the set of matrices $\{\mathbf{\Phi}_q\}_{q=1}^{Q-1}$, i.e.,

$$\begin{aligned} p(\{\mathbf{\Phi}_q\}_{q=1}^{Q-1}, \sigma_y^2 | \xi, \nu, \tau) = \frac{\nu^\tau}{\Gamma(\tau)} \left(\frac{1}{\sigma_y^2}\right)^{\tau+1} \exp\left\{-\frac{\nu}{\sigma_y^2}\right\} \\ \times \prod_{q=1}^{Q-1} \frac{1}{\left(2\pi\sigma_y^2/\xi\right)^{\frac{DM}{2}}} \exp\left\{ -\frac{\xi}{2\sigma_y^2} \operatorname{trace}\left[\mathbf{\Phi}_q^\top \mathbf{\Phi}_q\right] \right\}. \end{aligned} \tag{3.15}$$

Note that, in Model #2, the parameters $\mathbf{\Phi}_q$ for any particular state $q$ through all the chains are assumed to be independent, and we have also placed a prior over the observation noise variance $\sigma_y^2$.

## 3.4 Inference

Inference in BNP models is typically addressed by MCMC methods, such as Gibbs sampling [118, 45] or beam sampling [129]. 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 [36, 48, 35]. 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 (FFBS) 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 [36] and the variational inference proposed for infinite hidden Markov models (IHMMs) in [35]. Both of them are applicable when the number of states $Q$ is known.

### 3.4.1 Gibbs Sampling

MCMC methods have been broadly applied to infer the latent structure $\mathbf{S}$ from a given observation matrix $\mathbf{Y}$ (see, e.g., [54, 130]). 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

$$p(s_{tm} = k|\mathbf{Y}, \mathbf{S}_{\neg tm}) \propto p(s_{tm} = k|\mathbf{S}_{\neg tm})p(\mathbf{Y}|\mathbf{S}), \tag{3.16}$$

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 (3.16). Then, if the $m$-th 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 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 [54]).

We now derive the specific form of Eq. 3.16. The details of the computation of the first term in (3.16) can be found in Appendix A (see Section A.1). Regarding the second term, its form depends on the considered likelihood model:

- For the Gaussian observation Model #1, we first need to integrate out $\boldsymbol{\mu}_q$ as

$$\begin{aligned} p(\boldsymbol{\Phi}_q|\mathbf{S}) &= \int p(\boldsymbol{\Phi}_q|\mathbf{S}, \boldsymbol{\mu}_q)p(\boldsymbol{\mu}_q)d\boldsymbol{\mu}_q \\ &= \frac{1}{(2\pi)^{DM_+/2}\sigma_\phi^{(M_+-1)D}(\sigma_0^2 M_+ + \sigma_\phi^2)^{D/2}} \\ &\quad \times \exp\left\{-\frac{1}{2\sigma_\phi^2}\,\mathrm{trace}\left[(\boldsymbol{\Phi}_q - \mathbf{M}_\Phi)^\top \boldsymbol{\Sigma}_\Phi^{-1}(\boldsymbol{\Phi}_q - \mathbf{M}_\Phi)\right]\right\}, \end{aligned} \tag{3.17}$$

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

$$\begin{aligned} p(\mathbf{Y}|\mathbf{S}) =& \frac{1}{(2\pi\sigma_y^2)^{TD/2}|\mathbf{\Sigma}_{Q-1}|^{D/2}} \\ & \times \exp\left\{-\frac{1}{2\sigma_y^2}\,\mathrm{trace}\left[(\mathbf{Y}-\mathbf{M}_Y)^\top \mathbf{\Sigma}_{Q-1}^{-1}(\mathbf{Y}-\mathbf{M}_Y)\right]\right\}, \end{aligned} \tag{3.18}$$

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

$$\mathbf{\Sigma}_q^{-1} = \mathbf{\Sigma}_{q-1}^{-1} - \mathbf{\Sigma}_{q-1}^{-1}\mathbf{Z}_q\mathbf{W}_q\mathbf{Z}_q^\top\mathbf{\Sigma}_{q-1}^{-1} \tag{3.19}$$

and

$$\mathbf{M}_q = \frac{\sigma_y^2}{\sigma_0^2 M_+ + \sigma_\phi^2}\mathbf{\Sigma}_{q-1}^{-1}\mathbf{Z}_q\mathbf{W}_q\mathbf{1}_{M_+}\boldsymbol{\mu}_0^\top, \tag{3.20}$$

with $\mathbf{W}_q$ given by

$$\mathbf{W}_q^{-1} = \mathbf{Z}_q^\top\mathbf{\Sigma}_{q-1}^{-1}\mathbf{Z}_q + \frac{\sigma_y^2}{\sigma_\phi^2}\mathbf{\Sigma}_\Phi^{-1}, \tag{3.21}$$

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

- For the Gaussian likelihood Model #2, we have

$$p(\mathbf{Y}|\mathbf{S}) = \frac{\nu^\tau}{(2\pi)^{\frac{TD}{2}}|\mathbf{\Sigma}_{Q-1}|^{\frac{D}{2}}}\frac{\Gamma\left(\frac{TD}{2}+\tau\right)}{\Gamma(\tau)}\frac{1}{\nu+\frac{1}{2}\,\mathrm{trace}\left[\mathbf{Y}^\top\mathbf{\Sigma}_{Q-1}^{-1}\mathbf{Y}\right]}, \tag{3.22}$$

where the $T\times T$ matrix $\mathbf{\Sigma}_{Q-1}^{-1}$ can be iteratively computed as in Eq. 3.19, with matrix $\mathbf{W}_q$ given in this case by

$$\mathbf{W}_q^{-1} = \mathbf{Z}_q^\top\mathbf{\Sigma}_{q-1}^{-1}\mathbf{Z}_q + \xi\mathbf{I}_{M_+}, \tag{3.23}$$

for $q = 1,\ldots,Q-1$. Again, $\mathbf{\Sigma}_0$ is the identity matrix of size $T$.

### 3.4.2 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 [115, 130]. 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 FFBS algorithm, conditional on keeping all other Markov chains fixed. In order to apply this dynamic programming step, we also need a slice sampling algorithm [95] which adaptively truncates our model into a finite FHMM, performing exact inference without assuming alternative approximate models [130, 129].

Here, we make use of the stick-breaking construction of the model, presented in Section 3.2.4, and introduce an auxiliary slice variable $\vartheta$ distributed as

$$\vartheta|\mathbf{S}, \{c^{(m)}\} \sim \text{Uniform}\left(0, \min_{m:\exists t, s_{tm}\neq 0} c^{(m)}\right), \tag{3.24}$$

resulting in the joint distribution

$$p(\vartheta, \mathbf{S}, \{c^{(m)}, \mathbf{p}^{(m)}, \mathbf{a}_q^{(m)}\}) = p(\vartheta|\mathbf{S}, \{c^{(m)}\})p(\mathbf{S}, \{c^{(m)}, \mathbf{p}^{(m)}, \mathbf{a}_q^{(m)}\}). \tag{3.25}$$

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

From (3.25), 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 $\vartheta$, we have that

$$\begin{aligned} &p(\mathbf{S}|\mathbf{Y}, \vartheta, \{c^{(m)}, \mathbf{p}^{(m)}, \mathbf{a}_q^{(m)}\}) \\ &\propto p(\vartheta|\mathbf{S}, \{c^{(m)}\})p(\mathbf{S}|\mathbf{Y}, \{c^{(m)}, \mathbf{p}^{(m)}, \mathbf{a}_q^{(m)}\}), \end{aligned} \tag{3.26}$$

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. 3.24 does not need to be uniform, and a flexible Beta distribution can be used instead [130].

Unlike the Gibbs sampler, the blocked sampling algorithm does not allow us to integrate out the matrices $\mathbf{\Phi}_q$ or the noise variance $\sigma_y^2$, and they have to be sampled from their corresponding posterior distributions. In the case of the likelihood Model #1, the variables $\boldsymbol{\mu}_q$ can still be integrated out. Hence, the blocked sampling algorithm iteratively applies these steps:

1. Sample the slice variable $\vartheta$ from (3.24). This step may also involve adding new chains.

2. For each represented chain $m$, sample the $m$-th column of $\mathbf{S}$ via dynamic programming. Compact the representation by removing all chains in the all zero state.

3. For each active chain,[2] sample $c^{(m)}$, $\mathbf{p}^{(m)}$ and $\{\mathbf{a}_q^{(m)}\}$.

4. Sample the matrices $\mathbf{\Phi}_q$ (and the noise variance $\sigma_y^2$, if model #2 is considered).

In Step 1, $\vartheta$ is first sampled from (3.24). Then, starting from $m = M_+ + 1$, new variables $c^{(m)}$ are iteratively sampled from

$$\begin{aligned} p(c^{(m)}|c^{(m-1)}) \propto \exp\left(\alpha \sum_{t=1}^{T} \frac{1}{t}(1-c^{(m)})^t\right) \\ \times (c^{(m)})^{\alpha-1}(1-c^{(m)})^T \mathbb{I}(0 \leq c^{(m)} \leq c^{(m-1)}) \end{aligned} \tag{3.27}$$

until $c^{(m)} < \vartheta$. Since Eq. 3.27 is log-concave in $\log c^{(m)}$ [117], we can apply adaptive rejection sampling (ARS) [49]. 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 either the rest of rows of matrices $\mathbf{\Phi}_q$ (for model #1) or the noise variance $\sigma_y^2$ (for model #2). 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 FFBS sweep on one column of $\mathbf{S}$, having fixed the rest of columns [130].

In Step 3, for each chain, $c^{(m)}$ is sampled from [117]

$$\begin{aligned} p(c^{(m)}|\mathbf{S}, c^{(m-1)}, c^{(m+1)}) \propto (c^{(m)})^{n_{0\bullet}^{(m)} - n_{00}^{(m)} - 1} \\ \times (1-c^{(m)})^{n_{00}^{(m)}} \mathbb{I}(c^{(m+1)} \leq c^{(m)} \leq c^{(m-1)}), \end{aligned} \tag{3.28}$$

[2] An active chain is a chain in which not all states are zero.

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 $m$-th chain, considering the ordering given by the stick-breaking construction.

In Step 4, under Model #2, we first sample $\sigma_y^2$ for its inverse gamma posterior distribution given the data and the rest of hidden variables. Then, for both likelihood models, all matrices $\mathbf{\Phi}_q$ can be simultaneously sampled from the corresponding Gaussian posterior distribution given $\mathbf{S}$, $\mathbf{Y}$ and $\sigma_y^2$.

### 3.4.3 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 [73]. 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 [35, 48]. In [35], a variational inference algorithm for the IHMM is proposed. In [48] the authors develop several inference algorithms for the standard FHMM 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 [48] 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 3.2.1. The hyperparameters of the model[3] are gathered in the set $\mathcal{H} = \{Q, \alpha, \gamma, \beta_0, \beta, \sigma_0^2, \sigma_\phi^2, \sigma_y^2, \boldsymbol{\mu}_0\}$ and, similarly, we denote the set of unobserved variables by $\Psi = \{\mathbf{S}, \mathbf{a}_j^m, a^m, \mathbf{p}^m, \boldsymbol{\Phi}_k, \boldsymbol{\mu}_k\}$, for $j, k = 1, \ldots, Q-1$ and $m = 1, \ldots, M$.

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

$$\begin{aligned} p_M(\Psi, \mathbf{Y}|\mathcal{H}) = &\left(\prod_{k=1}^{Q-1} \left(p_M(\boldsymbol{\Phi}_k|\boldsymbol{\mu}_k, \sigma_\phi^2) p_M(\boldsymbol{\mu}_k|\sigma_0^2)\right)\right) \\ &\times \left(\prod_{m=1}^{M}\prod_{t=1}^{T} p_M(s_{tm}|s_{(t-1)m}, \mathbf{A}^m)\right) \\ &\times \left(\prod_{m=1}^{M} \left(\prod_{j=1}^{Q-1} p_M(\mathbf{a}_j^m|Q, \beta_0, \beta)\right) p_M(\mathbf{p}^m|Q, \gamma) p_M(a^m|\alpha)\right) \\ &\times p_M(\mathbf{Y}|\mathbf{S}, \boldsymbol{\Phi}_1, \ldots, \boldsymbol{\Phi}_{Q-1}). \end{aligned} \tag{3.29}$$

We approximate $p_M(\Psi|\mathbf{Y}, \mathcal{H})$ with the variational distribution $q(\Psi)$ given in Eq. 3.30, which is completely factorized except for the state matrix $\mathbf{S}$. We use the structured variational distribution for $q(\mathbf{S})$ developed in [48], 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

$$\begin{aligned} q(\Psi) = q(\mathbf{S}) &\left(\prod_{k=1}^{Q-1} \left(q(\boldsymbol{\Phi}_k) q(\boldsymbol{\mu}_k)\right)\right) \\ &\times \left(\prod_{m=1}^{M} \left(q(\mathbf{p}^m) q(a^m) \prod_{j=1}^{Q-1} q(\mathbf{a}_j^m)\right)\right), \end{aligned} \tag{3.30}$$

[3]For brevity, in this section we focus on the Gaussian observation model #1 in Figure 3.2.

being

$$q(\mathbf{S}) = \prod_{m=1}^{M} \frac{1}{Z_Q^m} \prod_{t=1}^{T} q(s_{tm}|s_{(t-1)m}), \tag{3.31}$$

where $Z_Q^m$ are the constants that ensure that $q(\mathbf{S})$ is properly normalized. The specific form for every term in Eqs. 3.30 and 3.31 is given by

$$q(s_{tm} = k|s_{(t-1)m} = j) \propto P_{jk}^m \cdot b_{kt}^m, \tag{3.32}$$

$$\begin{aligned} q(\mathbf{\Phi}_k) = & \frac{1}{(2\pi)^{MD/2}|\mathbf{\Lambda}_k|^{D/2}} \\ & \times \exp\left\{-\frac{1}{2}\,\mathrm{trace}\left[(\mathbf{\Phi}_k - \mathbf{L}_k)^\top \mathbf{\Lambda}_k^{-1}(\mathbf{\Phi}_k - \mathbf{L}_k)\right]\right\}, \end{aligned} \tag{3.33}$$

$$q(\boldsymbol{\mu}_k) = \mathcal{N}(\boldsymbol{\omega}_k, \mathbf{\Omega}_k), \tag{3.34}$$

$$q(\mathbf{p}^m) = \mathrm{Dirichlet}(\varepsilon_1^m, \ldots, \varepsilon_{Q-1}^m), \tag{3.35}$$

$$q(a^m) = \mathrm{Beta}(\nu_1^m, \nu_2^m), \tag{3.36}$$

and

$$q(\mathbf{a}_j^m) = \mathrm{Dirichlet}(\tau_{j0}^m, \ldots, \tau_{j(Q-1)}^m). \tag{3.37}$$

Inference involves optimizing the variational parameters of $q(\Psi)$ to minimize the Kullback-Leibler divergence of $p_M(\Psi|\mathbf{Y}, \mathcal{H})$ 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 A (see Section A.2).

## 3.5 Prior over the Number of States

The model in Section 3.2, as well as the inference algorithms in Section 3.4, 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 $\lambda$, namely,

$$p(Q|\lambda) = \frac{\lambda^{Q-2}e^{-\lambda}}{(Q-2)!}, \qquad Q = 2, \ldots, \infty. \tag{3.38}$$

As shown in Eq. 3.8, 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

$$p([\mathbf{S}]|\alpha, \beta_0, \beta, \gamma) = \sum_{Q=2}^{\infty} p([\mathbf{S}]|Q, \alpha, \beta_0, \beta, \gamma)p(Q|\lambda). \tag{3.39}$$

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. 3.39 is finite, as the series is convergent. To show this, it suffices to check that

$$\lim_{Q\to\infty} \frac{p([\mathbf{S}]|Q+1, \alpha, \beta_0, \beta, \gamma)p(Q+1|\lambda)}{p([\mathbf{S}]|Q, \alpha, \beta_0, \beta, \gamma)p(Q|\lambda)} < 1. \tag{3.40}$$

This condition holds since the limit can be simplified[4] to $\lim_{Q\to\infty} \frac{p(Q+1|\lambda)}{p(Q|\lambda)}$, which is less than one for every $\lambda > 0$ (indeed, the limit is equal to 0).

### 3.5.1 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 $p([\mathbf{S}], Q|\mathbf{Y}, \mathcal{H}')$, where $\mathcal{H}'$ is defined as the set of hyperparameters of the model.

We propose an MCMC inference algorithm that obtains samples from the target distribution $p([\mathbf{S}], Q|\mathbf{Y}, \mathcal{H}')$. An MCMC method dealing with HMMs can be found in [110], where a RJMCMC algorithm is used to estimate not only the parameters of the model, but also the number of states $Q$ of the HMM. RJMCMC

[4]Note that, according to Eq. 3.8, in the limit when $Q \to \infty$, $\frac{p([\mathbf{S}]|Q+1,\alpha,\beta_0,\beta,\gamma)}{p([\mathbf{S}]|Q,\alpha,\beta_0,\beta,\gamma))} = 1$.

methods, which were first introduced in [52] 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 over $[\mathbf{S}]$, $Q$ and the rest of latent variables of the model, given the observations $\mathbf{Y}$ and the hyperparameters $\mathcal{H}'$. Due to the multiplicity of Markov chains and the high dimensionality of the proposed IFHMM, the acceptance probabilities for transdimensional jumps under RJMCMC techniques turn out to be extremely low, which makes convergence too slow to be practical.

Since we can obtain the marginalized distribution $p([\mathbf{S}], Q|\mathbf{Y}, \mathcal{H}')$, where dimension-changing variables have been integrated out, RJMCMC methods are not needed and we apply a standard Metropolis-Hastings algorithm instead [88, 56]. Nevertheless, we adapt the procedure in [110] to develop our inference algorithm. Hence, our MCMC sampler proceeds iteratively as follows:

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 3.4.1, or a sweep of the blocked sampling described in Section 3.4.2. In the latter case, the transition probabilities, the matrices $\mathbf{\Phi}_q$ and the noise variance $\sigma_y^2$ must be sampled (Steps 3 and 4 in Section 3.4.2) before performing Step 1.

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 (DP) mixture model proposed in [70]. 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

$$R = \frac{p([\tilde{\mathbf{S}}], Q+1|\mathbf{Y}, \mathcal{H}')}{p([\mathbf{S}], Q|\mathbf{Y}, \mathcal{H}')} \frac{d_{Q+1} P^d_{select}}{b_Q P^b_{select} P_{alloc}}, \tag{3.41}$$

which ensures that the detailed balance condition is satisfied. In (3.41), $P^d_{select}$ 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 [70]) for those states in $\mathbf{S}$ taking value $q_*$. Rearranging and simplifying the factors in Eq. 3.41, $R$ can be expressed for the split and merge moves as

$$R = \frac{p(\mathbf{Y}|[\tilde{\mathbf{S}}], \mathcal{H}')}{p(\mathbf{Y}|[\mathbf{S}], \mathcal{H}')} \frac{p([\tilde{\mathbf{S}}]|Q+1, \mathcal{H}')}{p([\mathbf{S}]|Q, \mathcal{H}')} \frac{p(Q+1|\lambda)}{p(Q|\lambda)} \frac{d_{Q+1} 2/Q}{b_Q P_{alloc}}. \tag{3.42}$$

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

$$R = \frac{p([\tilde{\mathbf{S}}]|Q+1, \mathcal{H}')}{p([\mathbf{S}]|Q, \mathcal{H}')} \frac{p(Q+1|\lambda)}{p(Q|\lambda)} \frac{d_{Q+1}}{b_Q(Q_0+1)}, \tag{3.43}$$

| From \ To | State 0 | State 1 | State 2 |
|---|---|---|---|
| State 0 | 0.5964 | 0.1530 | 0.2506 |
| State 1 | 0.2973 | 0.6738 | 0.0289 |
| State 2 | 0.2463 | 0.2208 | 0.5329 |

(a) Chain 1

| From \ To | State 0 | State 1 | State 2 |
|---|---|---|---|
| State 0 | 0.6321 | 0.0466 | 0.3213 |
| State 1 | 0.3205 | 0.4947 | 0.1848 |
| State 2 | 0.2413 | 0.1262 | 0.6325 |

(b) Chain 2

Table 3.1: Transition probabilities for the synthetic toy dataset.

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 [108, 70] for further details), the sampler behaves as desired in terms of converging to a realization from the marginalized posterior distribution $p([\mathbf{S}], Q|\mathbf{Y}, \mathcal{H}')$.

## 3.6 Toy Example

We now design an example to show that the proposed model works as expected. We randomly generate a FHMM with two chains and three states (the inactive state and two active states). For this purpose, we randomly sample the corresponding transition probability matrices, increasing the self-transition probabilities and the probability of transitioning to the inactive state, yielding the matrices shown in Table 3.1.

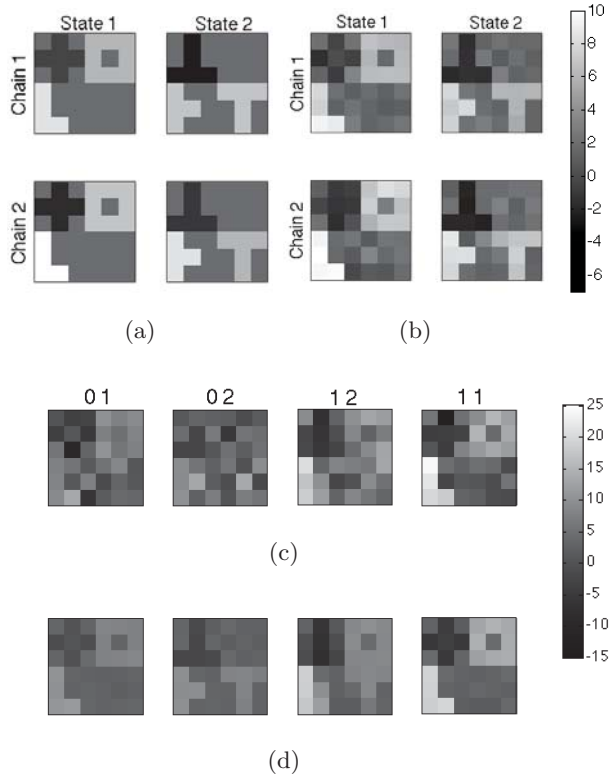



Figure 3.4: (a) The four base images. (b) Inferred posterior of the base images. (c) Four observed images with the corresponding states in each of the two chains. (d) Inferred posterior of the four images above.

The observation vector at each time instant is a 6-by-6-pixel image generated as a linear combination of the corresponding base images shown in Figure 3.4a, which represent the two active states in each chain. The observations are corrupted with Gaussian additive noise with zero mean and variance $\sigma_y^2 = 10$ (examples can be found in Figure 3.4c). We generate 200 examples to learn the IFUHMM model. We use the Gaussian observation Model #1 in Section 3.3. We initialize the sampler described in Section 3.5.1 with $M_+ = 1$, $Q = 2$, setting each $s_{tm} = q$ ($q \in \{0, \ldots, Q-1\}$) with probability $1/Q$, and we set the hyperparameters to $\alpha = 0.5$, $\gamma = 1$, $\beta_0 = \beta = 1$, $\sigma_0^2 = 100$, $\sigma_\phi^2 = 5$, $\boldsymbol{\mu}_0 = \mathbf{0}$ and $\lambda = 5$.

After running $5,000$ iterations of the inference algorithm (using Gibbs sampling for the first step), we reach a solution with $Q = 3$ states and $M_+ = 3$ chains. The

extra chain appears due to the effect of noise, and it is active only at three time instants. Therefore, we remove it in the plots. Figure 3.4b shows the inferred base images for the two active states in both chains, and in Figure 3.4d we plot the mean value of the inferred posterior probability for the four images in Figure 3.4c.

# 4

# Infinite Factorial Finite State Machine

## 4.1 Introduction

The hidden Markov model (HMM) is one of the most widely and successfully applied statistical models for the description of discrete-time series data. Its success mainly lies on two facts. First, it provides a model to capture non-trivial correlations across the observed sequence, in such a way that the observations become conditionally independent given a hidden sequence of states. Second, there are algorithms that allow an efficient calculation of the relevant quantities needed for statistical inference, like the expectation maximization (EM) or Baum-Welch algorithm [17]. This algorithm makes use of a forward-backward recursion for the E step of the inference, which yields complexity $\mathcal{O}(TQ^2)$, being $T$ the length of the

sequence and $Q$ the cardinality of the hidden states.

The factorial hidden Markov model (FHMM) [48] is an extension of the HMM in which $M$ parallel hidden chains evolve independently, and cooperatively generate the observed data. If the cardinality of each hidden state is $Q$, then the FHMM can alternatively be represented as a single HMM in which each hidden state can take $Q^M$ different values. However, the main challenge in FHMMs is posterior inference. Exact inference can be performed with a computational cost of $\mathcal{O}(TQ^{2M})$, although it can be reduced to $\mathcal{O}(TMQ^{M+1})$ by exploiting the independence of the parallel chains [48]. This exponential dependency makes exact inference intractable and, therefore, approximate inference methods are applied instead.

The stochastic finite state machine (FSM) with finite memory is another extension of the HMM. Under this model, the next state depends only on a finite number of previous inputs. This FSM relies on a finite memory $L$ and a finite alphabet $\mathcal{X}$. Each new input $x_t \in \mathcal{X}$ produces a deterministic change in the state of the FSM and an observable output. The next state and the output solely depend on the current state and the input. The FSM can also be represented as a single HMM, in which each state can be represented as the vector containing the last $L$ inputs. Performing exact inference on this model has complexity $\mathcal{O}(T|\mathcal{X}|^{2L})$, although it can be reduced to $\mathcal{O}(T|\mathcal{X}|^{L+1})$ by exploiting the model structure. Hence, approximate inference methods are required to avoid the exponential dependency on the memory length $L$.

In this chapter, we build a generative model to deal with time sequences in which several independent causes affect the observed data, as in the FHMM, and each of the hidden chains can be represented as an FSM with finite memory. We rely on Bayesian nonparametric (BNP) techniques in order to allow for a potentially infinite number of parallel chains and, therefore, we refer to our model as infinite factorial finite state machine (IFFSM). Our IFFSM model builds on the infinite factorial hidden Markov model (IFHMM) in [130], which considers an infinite number of parallel Markov chains with binary hidden states. We develop

an inference algorithm, based on a combination of Markov chain Monte Carlo (MCMC) and sequential Monte Carlo (SMC) techniques, that avoids the exponential complexity of both the FHMM and the FSM.

Although our model can be easily generalized and applied in a broad range of real-world problems, we motivate our model on the problem of blind multiuser channel estimation and symbol detection in a digital communication scenario. In this problem, each user (transmitter) sends sequences of symbols to a single receiver, and the receiver observes the superposition of the symbols transmitted by all the users in the system. Furthermore, due to physical properties of the communication channel, such as multipath, a symbol transmitted at time $t$ may affect the observations received at a future time instant $t' > t$. This is equivalent to assuming that the communication channel has memory, as it can "remember" the last symbols sent by the transmitters. Thus, the FSM can naturally model such communication system.

## 4.2 Infinite Factorial Finite State Machine

The model proposed here is an extension of the IFHMM in [130] that allows considering a potentially infinite number of FSMs that evolve independently of each other.

In order to be able to deal with an infinite number of FSMs, we need to consider an inactive state, such that the observations do not depend on those FSMs that are inactive. While active, the input symbol to the $m$-th FSM at time instant $t$, denoted by $x_{tm}$, is assumed to belong to the set $\mathcal{A}$, with finite cardinality $|\mathcal{A}|$. While inactive, we can assume that $x_{tm} = 0$ and, therefore, each input $x_{tm} \in \mathcal{X}$, with $\mathcal{X} = \mathcal{A} \bigcup \{0\}$. We introduce the binary auxiliary variables $s_{tm}$ to denote whether the $m$-th FSM is active at time $t$, such that

$$x_{tm}|s_{tm} \sim \begin{cases} \delta_0(x_{tm}) & \text{if } s_{tm} = 0, \\ \mathcal{U}(\mathcal{A}) & \text{if } s_{tm} = 1, \end{cases} \tag{4.1}$$

where $\delta_0(\cdot)$ denotes a point mass located at 0, and $\mathcal{U}(\mathcal{A})$ denotes the uniform

distribution over the set $\mathcal{A}$. Note that, conditioned on the auxiliary variables $s_{tm}$, the input symbols $x_{tm}$ are independent and identically distributed.

As in the IFHMM, we place a BNP prior over the binary matrix $\mathbf{S}$ that contains all variables $s_{tm}$. This prior is known as the Markov Indian buffet process (MIBP), and we write $\mathbf{S} \sim \text{MIBP}(\alpha, \beta_0, \beta_1)$ to denote that the matrix $\mathbf{S}$ is distributed according to a MIBP with parameters $\alpha$, $\beta_0$ and $\beta_1$ (the role of each hyperparameter is explained below). The MIBP places a prior distribution over binary matrices with a finite number of rows $T$ and an infinite number of columns $M$, in which each element $s_{tm} = (\mathbf{S})_{tm} \in \{0, 1\}$. Each row represents a time instant, whilst each column represents a Markov chain. The MIBP ensures that, for any finite value of $T$, only a finite value of columns in $\mathbf{S}$ become active, while the rest of them remain in the all-zero state and do not influence the observations. We refer to Chapter 2 for a more detailed presentation of the MIBP.

We make use of the stick-breaking construction of the MIBP, which is particularly useful to develop practical inference algorithms [117, 130]. Under the stick-breaking construction, two hidden variables for each Markov chain are introduced, representing the transition probabilities between the active and inactive states. In particular, we denote with $a^m$ the self-transition probability of the inactive state, and with $b^m$ the transition probability from active to inactive of the $m$-th chain. Hence, the transition probability matrix of the $m$-th Markov chain can be written as

$$\mathbf{A}^m = \begin{pmatrix} a^m & 1 - a^m \\ b^m & 1 - b^m \end{pmatrix}, \tag{4.2}$$

and the binary auxiliary states $s_{tm}$ evolve according to

$$p(s_{tm} = 0 | s_{(t-1)m} = 0, a^m) = a^m, \tag{4.3}$$

and

$$p(s_{tm} = 0 | s_{(t-1)m} = 1, b^m) = b^m. \tag{4.4}$$

We sort the columns of $\mathbf{S}$ (chains of the IFHMM) according to their values of $a^m$, such that $a^{(1)} < a^{(2)} < a^{(3)} < \ldots$, and we work instead with the complementary

probabilities $c^m = 1 - a^m$, such that $c^{(1)} > c^{(2)} > c^{(3)} > \dots$ The probability distribution over variables $c^{(m)}$ is given by

$$c^{(1)} \sim \text{Beta}(\alpha, 1), \tag{4.5}$$

and

$$p(c^{(m)}|c^{(m-1)}) \propto (c^{(m)})^{\alpha-1}\mathbb{I}(0 \leq c^{(m)} \leq c^{(m-1)}), \tag{4.6}$$

being $\mathbb{I}(\cdot)$ the indicator function [117]. With respect to variables $b^{(m)}$, i.e., variables $b^m$ reordered accordingly to decreasing values of $a^m$, they are distributed as

$$b^{(m)} \sim \text{Beta}(\beta_0, \beta_1). \tag{4.7}$$

The MIBP allows for a potentially infinite number of parallel FSMs in our IFFSM. Eq. 4.1 and the MIBP prior over $\mathbf{S}$ ensure that only a finite subset of the FSMs become active during the observation period. In the IFFSM model, each input symbol $x_{tm}$ does not only influence the observation $\mathbf{y}_t$ at time instant $t$, but it has also an impact on the future $L-1$ observations, $\mathbf{y}_{t+1}, \dots, \mathbf{y}_{t+L-1}$. Therefore, the likelihood function for $\mathbf{y}_t$ depends on the last $L$ input symbols of all FSMs, yielding

$$p(\mathbf{y}_t|\mathbf{X}) = p(\mathbf{y}_t|\{x_{tm}, x_{(t-1)m}, \dots, x_{(t-L+1)m}\}_{m=1}^{\infty}), \tag{4.8}$$

with $\mathbf{X}$ being the $T \times M$ matrix that contains all symbols $x_{tm}$. In our model, we assume dummy input symbols $x_{tm} = 0$ for $t \leq 0$.

The resulting IFFSM model, particularized for $L = 2$, is shown in Figure 4.1. Note that this model can be equivalently represented as a single HMM, as shown in Figure 4.2, using the extended states $s_{tm}^{(e)}$, with

$$s_{tm}^{(e)} = \left[ \; x_{tm}, \quad s_{tm}, \quad x_{(t-1)m}, \quad s_{(t-1)m}, \quad \dots, \quad x_{(t-L+1)m}, \quad s_{(t-L+1)m} \; \right]. \tag{4.9}$$

However, we maintain the representation in Figure 4.1 because it allows us to derive an efficient inference algorithm.

## 4.3 Generalization of the Model

The model in Section 4.2 can be straightforwardly generalized in order to capture additional properties of the underlying structure. We consider two ways of gener-

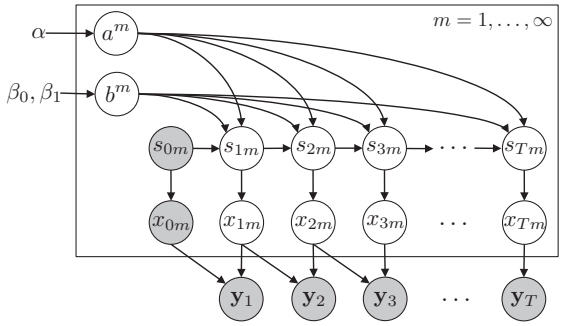



Figure 4.1: Graphical model of the IFFSM with $L = 2$.

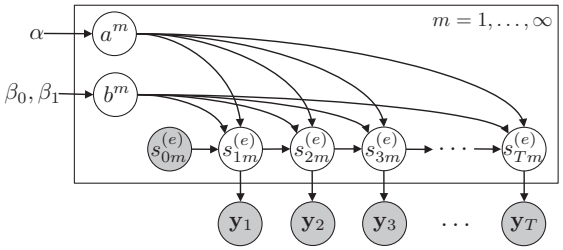



Figure 4.2: Equivalent graphical model for the IFFSM.

alization that our inference algorithm in Section 4.5 can handle with minor or no modifications.

First, we have assumed so far that the input symbols $x_{tm}$ belong to a finite set $\mathcal{X}$, therefore yielding $|\mathcal{X}|^L$ possible states in each parallel chain. However, we can also consider that the set $\mathcal{X}$ is either countably or uncountably infinite, implying that the input symbols $x_{tm}$ are not necessarily discrete-valued. The resulting model is no longer an IFFSM, but an infinite factorial model in which the hidden variables affect not only the current observation, but also the future ones.

Second, regarding the temporal evolution of the input symbols $x_{tm}$, Eq. 4.1 implies that $x_{tm}$ is independent of $x_{(t-1)m}$ given $s_{tm}$, which may constitute a limitation in some applications. We can easily extend our model by letting $x_{tm}$ depend on both $s_{tm}$ and $x_{(t-1)m}$, i.e., by removing the constraint that $p(x_{tm}|s_{tm}, x_{(t-1)m})$ does not depend on $x_{(t-1)m}$. The corresponding graphical model that considers this generalization is depicted in Figure 4.3. Note that this model can still be represented as shown in Figure 4.2, but Figure 4.3 explicitly shows the relationships

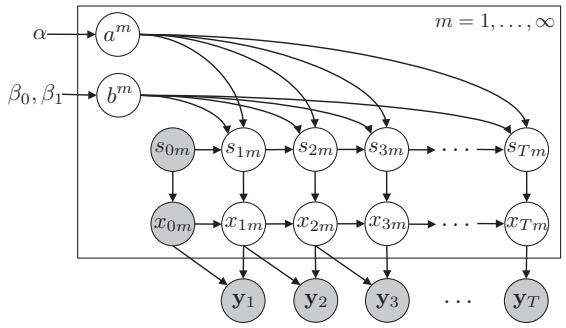



Figure 4.3: Graphical model of the general infinite factorial model with $L = 2$.

among the hidden variables of the model.

Our model in Figure 4.3 generalizes some other models that have been propose in the literature. We can obtain some particularizations as detailed below (Table 4.1 summarizes this information):

- We trivially recover the IFFSM model in Section 4.2 if we let $\mathcal{X}$ be a finite set and we let $p(x_{tm}|s_{tm}, x_{(t-1)m}) = p(x_{tm}|s_{tm})$.

- We recover the IFHMM in [130] by choosing $\mathcal{X} = \{0, 1\}$ (or, equivalently, $\mathcal{A} = \{1\}$), $x_{tm} = s_{tm}$ and $L = 1$.

- We obtain a non-binary IFHMM if we assume that $\mathcal{X}$ is a discrete set with cardinality greater than 2 and that $L = 1$. This model is not equivalent to the non-binary IFHMM described in Chapter 3 due to the transition probability from active to inactive. In the IFHMM in Chapter 3, the transition probability from active to inactive depends on the current (active) state, while under the model in Figure 4.3, this probability is given by $b^m$, which does not depend on $x_{tm}$. In this sense, the non-binary IFHMM in Chapter 3 can capture some additional information about the hidden states.

- We obtain the independent component analysis (ICA) IFHMM model for source separation proposed in [130] if we set $\mathcal{X} = \mathbb{R}$, we let $x_{tm}$ be Gaussian distributed and we also choose $p(x_{tm}|s_{tm}, x_{(t-1)m}) = p(x_{tm}|s_{tm})$ and $L = 1$.

- We obtain an "infinite factorial linear dynamical system" [74] with on/off

| Model | $\mathcal{X}$ | $p(x_{tm}\|s_{tm}=1, x_{(t-1)m})$ | $L$ |
|---|---|---|---|
| IFFSM | $\mathcal{A}\bigcup\{0\}$ | $\mathcal{U}(\mathcal{A})$ | $\in \mathbb{N}$ |
| Binary IFHMM | $\{0,1\}$ | $\delta_1(x_{tm})$ | 1 |
| Non-binary IFHMM | $\{0,1,\ldots,Q-1\}$ | $a^m_{jk} = p(x_{tm}=k\|x_{(t-1)m}=j)$ | 1 |
| ICA IFHMM | $\mathbb{R}$ | $\mathcal{N}(x_{tm}\|\mu,\sigma^2)$ | 1 |
| Infinite factorial linear dynamical system | $\mathbb{R}$ | $\mathcal{N}(x_{tm}\|bx_{(t-1)m},\sigma^2)$ | $\in \mathbb{N}$ |

Table 4.1: Particularizations of the general infinite factorial model.

states by assuming that the variables $x_{tm}$ are Gaussian distributed given $x_{(t-1)m}$, $\mathcal{X} = \mathbb{R}$ and $L = 1$. Moreover, if we let $L$ be greater than 1, the resulting likelihood takes into account, e.g., the echo that may be present in the observed sequence.

## 4.4 A Gaussian Observation Model

The IFFSM model in Section 4.2 (and also its extension in Section 4.3) can be applied as a building block for a full probabilistic model, in which an arbitrarily large set of parallel FSMs influence the observations. Note that there are two conditions for the likelihood model to be valid as the number of FSMs $M$ tends to infinity: i) the likelihood must be invariant to permutations of the chains, and ii) the distribution on $\mathbf{y}_t$ cannot depend on any parameter of the $m$-th FSM if $x_{\tau m} = 0$ for $\tau = t - L + 1, \ldots, t$.

In this section, we propose a Gaussian likelihood model specifically designed for our motivating application of a digital communication system. The corresponding graphical model is represented in Figure 4.4. In such system, we have access to $T$ observation vectors $\mathbf{y}_t$ of length $D$, which we collect in a $T \times D$ matrix $\mathbf{Y}$. The received sequence is the linear combination of the symbols transmitted by all users in the system, weighted by the channel coefficients. In particular, the observation vector for each time instant can be written as

$$\mathbf{y}_t = \sum_{m=1}^{\infty} \sum_{\ell=1}^{L} \mathbf{h}_m^{\ell} x_{(t-\ell+1)m} + \mathbf{n}_t, \tag{4.10}$$

being $x_{tm}$ the input symbol of the $m$-th FSM at time instant $t$, $\mathbf{h}_m^{\ell}$ the channel coefficients or emission parameters, and $\mathbf{n}_t$ the additive noise. Both $\mathbf{h}_m^{\ell}$ and $\mathbf{n}_t$ are

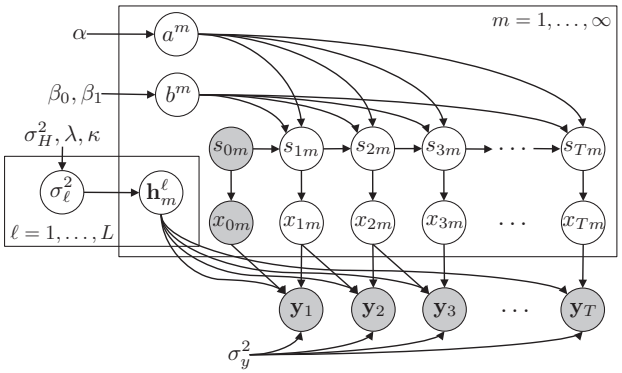



Figure 4.4: Graphical Gaussian observation model for an IFFSM with $L = 2$.

vectors of length $D$.

In digital communication systems, the observations and the transmitted symbols can be complex-valued. Hence, the finite set $\mathcal{A}$ may contain complex-valued elements, and we need to consider a distribution over complex numbers for the observations $\mathbf{y}_t$. For that purpose, we place a circularly symmetric complex Gaussian prior distribution[1] with independent elements over the observation parameters $\mathbf{h}_m^\ell$ and the noise $\mathbf{n}_t$, of the form

$$\mathbf{n}_t | \sigma_y^2 \sim \mathcal{CN}(\mathbf{0}, \sigma_y^2 \mathbf{I}_D, \mathbf{0}), \tag{4.11}$$

and

$$\mathbf{h}_m^\ell | \sigma_\ell^2 \sim \mathcal{CN}(\mathbf{0}, \sigma_\ell^2 \mathbf{I}_D, \mathbf{0}), \tag{4.12}$$

with $\mathbf{I}_D$ being the identity matrix of size $D$. Given Eqs. 4.10 and 4.11, the probability distribution over $\mathbf{y}_t$ is also a complex Gaussian, i.e.,

$$p(\mathbf{y}_t | \{\mathbf{h}_m^\ell\}, \{x_{tm}\}, \sigma_y^2) = \mathcal{CN}\left( \sum_{m=1}^{\infty} \sum_{\ell=1}^{L} \mathbf{h}_m^\ell x_{(t-\ell+1)m}, \sigma_y^2 \mathbf{I}_D, \mathbf{0} \right). \tag{4.13}$$

Here, the noise variance $\sigma_y^2$ is a hyperparameter of the model.

[1]The complex Gaussian distribution $\mathcal{CN}(\boldsymbol{\mu}, \boldsymbol{\Gamma}, \mathbf{C})$ over a vector $\mathbf{x}$ of length $D$ is given by $p(\mathbf{x}) = \frac{1}{\pi^D \sqrt{\det(\boldsymbol{\Gamma}) \det(\mathbf{P})}} \exp\left\{ -\frac{1}{2} \left[ (\mathbf{x}-\boldsymbol{\mu})^{\mathsf{H}}, (\mathbf{x}-\boldsymbol{\mu})^{\top} \right] \begin{bmatrix} \boldsymbol{\Gamma} & \mathbf{C} \\ \mathbf{C}^{\mathsf{H}} & \boldsymbol{\Gamma}^\star \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{x}-\boldsymbol{\mu} \\ (\mathbf{x}-\boldsymbol{\mu})^\star \end{bmatrix} \right\}$, where $\mathbf{P} = \boldsymbol{\Gamma}^\star - \mathbf{C}^{\mathsf{H}} \boldsymbol{\Gamma}^{-1} \mathbf{C}$, $(\cdot)^\star$ denotes the complex conjugate, and $(\cdot)^{\mathsf{H}}$ denotes the conjugate transpose. A circularly symmetric complex Gaussian distribution has $\boldsymbol{\mu} = \mathbf{0}$ and $\mathbf{C} = \mathbf{0}$.

Regarding the variances $\sigma_\ell^2$, we place an inverse-gamma prior over each variable $\sigma_\ell^2$ with mean $\mathbb{E}[\sigma_\ell^2] = \sigma_H^2 e^{-\lambda(\ell-1)}$ and standard deviation $\mathrm{Std}[\sigma_\ell^2] = \kappa\mathbb{E}[\sigma_\ell^2]$, being $\sigma_H^2$, $\lambda$ and $\kappa$ hyperparameters of the model. By defining $\nu_\ell = (\tau-1)\sigma_H^2 e^{-\lambda(\ell-1)}$ and $\tau = 2 + \kappa^{-2}$, the probability distribution over $\sigma_\ell^2$ can be written as

$$p(\sigma_\ell^2|\sigma_H^2, \lambda, \kappa) = \frac{(\nu_\ell)^\tau}{\Gamma(\tau)} \left(\frac{1}{\sigma_\ell^2}\right)^{\tau+1} \exp\left\{-\frac{\nu_\ell}{\sigma_\ell^2}\right\}. \tag{4.14}$$

The choice of this particular prior is based on the assumption that the channel coefficients (or observation parameters) $\mathbf{h}_m^\ell$ are *a priori* expected to decay with the index $\ell$, since they model the multipath effect. However, if the data contains enough evidence against this assumption, the posterior distribution will assign high probability mass to larger values of $\sigma_\ell^2$.

## 4.5 Inference via Blocked Sampling

One of the main challenges in Bayesian probabilistic models is posterior inference, which involves the computation of the posterior distribution over the hidden variables in the model given the data. In many models of interest, including BNP models, the posterior distribution cannot be obtained in closed form, and an approximate inference algorithm is used instead.

In BNP time series models, inference is typically carried out using either MCMC methods, such as Gibbs sampling [118, 45], beam sampling [129], or slice sampling [130], or variational methods, with a mean field or a structured approximation [48, 35]. In this section, we develop an inference algorithm that combines two standard tools used for Monte Carlo statistical inference: MCMC and SMC.

For IFHMMs, typical approaches rely on a blocked Gibbs sampling algorithm that alternates between sampling the global variables (number of parallel chains, emission parameters and transition probabilities) conditioned on the current value of matrices $\mathbf{S}$ and $\mathbf{X}$, and sampling matrices $\mathbf{S}$ and $\mathbf{X}$ conditioned on the current value of the global variables. More specifically, the algorithm proceeds iteratively as follows:

- **Step 1:** Add $M_{\text{new}}$ new inactive chains[2] using an auxiliary slice variable and a slice sampling method. In this step, the number of considered parallel chains in increased from its initial value $M_+$ to $M^\ddagger = M_+ + M_{\text{new}}$ (we do not update $M_+$ because the new chains are in the all-zero state).

- **Step 2:** Sample the states $s_{tm}$ and the input symbols $x_{tm}$ of all the considered chains (FSMs). Compact the representation by removing those chains that remain inactive in the entire observation period, consequently updating $M_+$.

- **Step 3:** Sample the global variables, i.e., the transition probabilities $a^m$ and $b^m$ and the observation parameters $\mathbf{h}_m^\ell$ for each active chain ($m = 1, \ldots, M_+$), as well as the variances $\sigma_\ell^2$.

In **Step 1**, we follow the slice sampling scheme for inference in BNP models based on the Indian buffet process (IBP) [117, 130], which effectively transforms the model into a finite factorial model with $M^\ddagger = M_+ + M_{\text{new}}$ parallel chains. We first sample an auxiliary slice variable $\vartheta$, which is distributed as

$$\vartheta | \mathbf{S}, \{c^{(m)}\} \sim \text{Uniform}\,(0, c_{\min})\,, \tag{4.15}$$

where $c_{\min} = \min_{m:\exists t, s_{tm} \neq 0} c^{(m)}$, and we can replace the uniform distribution with a more flexible scaled beta distribution. Then, starting from $m = M_+ + 1$, new variables $c^{(m)}$ are iteratively sampled from

$$\begin{aligned} p(c^{(m)} | c^{(m-1)}) \propto \exp\left(\alpha \sum_{t=1}^{T} \frac{1}{t}(1 - c^{(m)})^t\right) \\ \times (c^{(m)})^{\alpha-1}(1 - c^{(m)})^T \mathbb{I}(0 \leq c^{(m)} \leq c^{(m-1)}), \end{aligned} \tag{4.16}$$

with $c^{(M_+)} = c_{\min}$, until the resulting value is less than the slice variable, i.e., until $c^{(m)} < \vartheta$. Since Eq. 4.16 is log-concave in $\log c^{(m)}$ [117], we can apply adaptive rejection sampling (ARS) [49] in this step. Let $M_{\text{new}}$ be the number of new variables $c^{(m)}$ that are greater than the slice variable. If $M_{\text{new}} > 0$, then we expand the representation of matrices $\mathbf{S}$ and $\mathbf{X}$ by adding $M_{\text{new}}$ zero columns, and

[2] An inactive chain is an chain in which all elements $s_{tm} = 0$.

we sample the corresponding per-chain global variables (i.e., $\mathbf{h}^{\ell}_{(m)}$ and $b^{(m)}$) from the prior, given in Eqs. 4.12 and 4.7, respectively.

**Step 2** consists in sampling the elements of the matrices $\mathbf{S}$ and $\mathbf{X}$ given the current value of the global variables. In this step, several approaches can be taken. A naïve Gibbs sampling algorithm that sequentially samples each element $x_{tm}$ (jointly with $s_{tm}$) is simple and computationally efficient, but it presents poor mixing properties due to the strong couplings between successive time steps [115, 130]. An alternative to Gibbs sampling is blocked sampling, which sequentially samples each parallel chain, conditioned on the current value of the remaining ones. This approach requires a forward-filtering backward-sampling (FFBS) sweep in each of the chains, yielding runtime complexity of $\mathcal{O}(TM^{\ddagger} \cdot 2^2)$ in standard binary IFHMMs, or $\mathcal{O}(TM^{\ddagger}|\mathcal{X}|^2)$ in memoryless ($L = 1$) FSMs. However, for FSMs with $L > 1$, the complexity of the FFBS sweeps increases to $\mathcal{O}(TM^{\ddagger}|\mathcal{X}|^{L+1})$. The exponential dependency on $L$ makes this step computationally intractable.

In order to address this problem, we propose to jointly sample matrices $\mathbf{S}$ and $\mathbf{X}$ using particle Gibbs with ancestor sampling (PGAS), an algorithm recently developed for inference in state-space models and non-Markovian latent variable models [83]. If $P$ particles are used for the PGAS kernel, the runtime complexity of the algorithm is $\mathcal{O}(PTM^{\ddagger}L^2)$ for the IFFSM model. Details on the PGAS approach are given in Section 4.5.1.

Besides its non-exponential time complexity, the PGAS approach presents two additional advantages when compared to the FFBS sweeps. First, it can be directly applied to the general model in Section 4.3, regardless of $\mathcal{X}$ being a finite or infinite set. Second, it has better mixing properties. The reason is that FFBS fixes all but one chain at each step, therefore removing the contribution of these $M^{\ddagger} - 1$ chains from the observations. In contrast, the PGAS method allows sampling simultaneously the $M^{\ddagger}$ chains or FSMs for each time instant $t = 1, \ldots, T$, which avoids getting trapped in local modes of the posterior in which a chain is splitted into several ones.

After running PGAS, we remove those chains that remain inactive in the whole

observation period. This implies removing some columns of $\mathbf{S}$ and $\mathbf{X}$ as well as the corresponding variables $\mathbf{h}_m^\ell$, $a^m$ and $b^m$, and updating $M_+$.

In **Step 3**, we sample the global variables in the model from their complete conditional distributions.[3] The complete conditional distribution over the transition probabilities $a^m$ under the semi-ordered stick-breaking construction [117] is given by

$$p(a^m|\mathbf{S}) = \text{Beta}\left(1 + n_{00}^m, n_{01}^m\right), \tag{4.17}$$

being $n_{ij}^m$ the number of transitions from state $i$ to state $j$ in the $m$-th column of $\mathbf{S}$. For the transition probabilities from active to inactive $b^m$, we have

$$p(b^m|\mathbf{S}) = \text{Beta}\left(\beta_0 + n_{10}^m, \beta_1 + n_{11}^m\right). \tag{4.18}$$

The complete conditional distributions over the emission parameters $\mathbf{h}_m^\ell$ for all chains $m = 1, \ldots, M_+$ and for all taps $\ell = 1, \ldots, L$ are given by complex Gaussians of the form

$$p(\mathbf{h}^{(d)}|\mathbf{Y}, \mathbf{X}, \{\sigma_\ell^2\}) = \mathcal{CN}\left(\boldsymbol{\mu}_{\text{POST}}^{(d)}, \boldsymbol{\Gamma}_{\text{POST}}, \mathbf{0}\right), \tag{4.19}$$

for $d = 1, \ldots, D$. Here, we have defined for notational simplicity $\mathbf{h}^{(d)}$ as the vector that contains the $d$-th component of vectors $\mathbf{h}_m^\ell$ for all $m$ and $\ell$, as given by

$$\mathbf{h}^{(d)} = \left[(\mathbf{h}_1^1)_d, \ldots, (\mathbf{h}_1^L)_d, (\mathbf{h}_2^1)_d, \ldots, (\mathbf{h}_2^L)_d, \ldots, (\mathbf{h}_{M_+}^1)_d, \ldots, (\mathbf{h}_{M_+}^L)_d\right]^\top. \tag{4.20}$$

By additionally defining the extended matrix $\mathbf{X}^{\text{ext}} = \left[\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(M_+)}\right]$ of size $T \times LM_+$, with

$$\mathbf{X}^{(m)} = \begin{bmatrix} x_{1m} & 0 & 0 & \cdots & 0 \\ x_{2m} & x_{1m} & 0 & \cdots & 0 \\ x_{3m} & x_{2m} & x_{1m} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{Tm} & x_{(T-1)m} & x_{(T-2)m} & \cdots & x_{(T-L+1)m} \end{bmatrix}, \tag{4.21}$$

$\boldsymbol{\Sigma}$ as the $L \times L$ diagonal matrix containing all variables $\sigma_\ell^2$, and $\mathbf{y}^{(d)}$ as the $T$-vector containing the $d$-th element of each observation $\mathbf{y}_t$, the posterior parameters in

[3] The complete conditional is the conditional distribution of a hidden variable, given the observations and the rest of hidden variables.

Eq. 4.19 are given by

$$\mathbf{\Gamma}_{\text{POST}} = \left( (\mathbf{I}_{M_+} \otimes \mathbf{\Sigma})^{-1} + \frac{1}{\sigma_y^2} (\mathbf{X}^{\text{ext}})^{\mathsf{H}} \mathbf{X}^{\text{ext}} \right)^{-1} \tag{4.22}$$

and

$$\boldsymbol{\mu}_{\text{POST}}^{(d)} = \frac{1}{\sigma_y^2} \mathbf{\Gamma}_{\text{POST}} (\mathbf{X}^{\text{ext}})^{\mathsf{H}} \mathbf{y}^{(d)}, \tag{4.23}$$

being $(\cdot)^{\mathsf{H}}$ the conjugate transpose, '$\otimes$' the Kronecker product, and $\mathbf{I}_{M_+}$ the identity matrix of size $M_+$.

Regarding the complete conditionals of the variances $\sigma_\ell^2$, they are given by inverse-gamma distributions of the form

$$p(\sigma_\ell^2 | \{\mathbf{h}_m^\ell\}_{m=1}^{M_+}) \propto \left( \frac{1}{\sigma_\ell^2} \right)^{1+\tau+DM_+} \exp\left\{ -\frac{\nu_\ell + \sum_{m=1}^{M_+} ||\mathbf{h}_m^\ell||_2^2}{\sigma_\ell^2} \right\}, \tag{4.24}$$

being $||\mathbf{h}_m^\ell||_2^2$ the squared $\text{L}^2$-norm of the complex vector $\mathbf{h}_m^\ell$, $\tau = 2 + \kappa^{-2}$ and $\nu_\ell = (\tau - 1)\sigma_H^2 e^{-\lambda(\ell-1)}$.

### 4.5.1 Particle Gibbs with Ancestor Sampling

We rely on PGAS [83] for Step 2 of our inference algorithm, in order to obtain a sample of the matrices $\mathbf{S}$ and $\mathbf{X}$. PGAS is a method within the framework of particle MCMC [9], which is a systematic way of combining SMC and MCMC to take advantage of the strengths of both techniques.

PGAS builds on the particle Gibbs sampler in [9], in which a Markov kernel is constructed by running an SMC sampler in which one particle trajectory is set deterministically to a reference trajectory that is specified *a priori*. After a complete run of the SMC algorithm, a new reference trajectory is obtained by selecting one of the particle trajectories with probabilities given by their importance weights. In this way, the resulting Markov kernel leaves its target distribution invariant, regardless of the number of particles used in the SMC algorithm. In order to improve the mixing properties of the particle Gibbs sampler by alleviating the so-called path degeneracy effect, a method denoted as particle Gibbs with backward simulation can be applied [139, 84]. PGAS has the same purpose as particle Gibbs

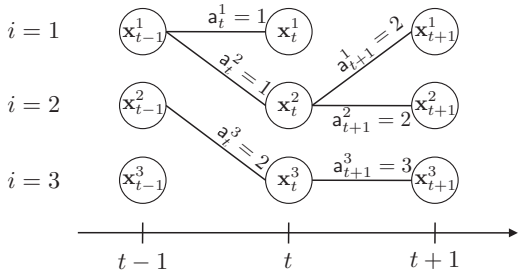



Figure 4.5: Example of the connection of particles in PGAS. We represent $P = 3$ particles $\mathbf{x}_\tau^i$ for $\tau = \{t-1, t, t+1\}$. The index $\mathsf{a}_\tau^i$ denotes the ancestor particle of $\mathbf{x}_\tau^i$. It can be seen that, e.g., the trajectories $\mathbf{x}_{1:t+1}^1$ and $\mathbf{x}_{1:t+1}^2$ only differ at time instant $t+1$.

with backward simulation, but it is not restricted to state-space models, and can also be applied to non-Markovian latent variable models.

In this section, we introduce the required formulation and describe the PGAS algorithm for application under our general model in Section 4.3. We aim at providing the necessary equations and algorithm steps, but we refer to [83] for further details on the theoretical justification of the algorithm and rigorous analysis of its properties.

In PGAS, we assume a set of $P$ particles for each time instant $t$, each representing the hidden states $\{x_{tm}\}_{m=1}^{M^\ddagger}$ (hence, they also represent $\{s_{tm}\}_{m=1}^{M^\ddagger}$). We denote the state of the $i$-th particle at time $t$ by the vector $\mathbf{x}_t^i$ of length $M^\ddagger$. We also introduce the ancestor indexes $\mathsf{a}_t^i \in \{1, \ldots, P\}$ in order to denote the particle that precedes the $i$-th particle at time $t$. That is, $\mathsf{a}_t^i$ corresponds to the index of the ancestor particle of $\mathbf{x}_t^i$. Let also $\mathbf{x}_{1:t}^i$ be the ancestral path of particle $\mathbf{x}_t^i$, i.e., the particle trajectory that is recursively defined as

$$\mathbf{x}_{1:t}^i = (\mathbf{x}_{1:t-1}^{\mathsf{a}_t^i}, \mathbf{x}_t^i). \tag{4.25}$$

Figure 4.5 shows an example to clarify the notation.

PGAS also requires an input reference particle that is held fixed. This reference particle is given by the output of the previous iteration of the PGAS algorithm, possibly extended to account for new inactive chains. For each time instant $t$, we denote this particle as $\mathbf{x}_t'$ (this is also a vector of length $M^\ddagger$). Hence, $P-1$ particles are sampled during the algorithm execution, but the $P$-th particle is kept fixed,

with $\mathbf{x}_t^P = \mathbf{x}'_t$ for all $t$. While the particle $\mathbf{x}_t^P$ is fixed, the corresponding ancestor indexes $\mathsf{a}_t^P$ are not fixed, but they are randomly drawn instead.

We need to specify how particles are propagated across time, i.e., we need to choose a distribution $r_t(\mathbf{x}_t|\mathbf{x}_{1:t-1}^{\mathsf{a}_t})$. For simplicity, we assume that

$$r_t(\mathbf{x}_t|\mathbf{x}_{1:t-1}^{\mathsf{a}_t}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}^{\mathsf{a}_t}) = \prod_{m=1}^{M^\ddagger} p(x_{tm}|s_{tm}, x_{(t-1)m}^{\mathsf{a}_t}) p(s_{tm}|s_{(t-1)m}^{\mathsf{a}_t}), \tag{4.26}$$

i.e., that particles are propagated using the transition model in Figure 4.3 in a simple bootstrap particle filter manner.[4] Note that, under the IFFSM model in Section 4.2, Eq. 4.26 can be further simplified, since $p(x_{tm}|s_{tm}, x_{(t-1)m}) = p(x_{tm}|s_{tm})$.

The resulting procedure is summarized in Algorithm 1. The algorithm involves, for each time instant $t$, resampling the particles at time $t-1$ according to their importance weights $w_{t-1}^i$, and then propagating the selected particles from instant $t-1$ to $t$ according to the distribution $r_t(\mathbf{x}_t|\mathbf{x}_{1:t-1}^{\mathsf{a}_t})$. The reference particle $\mathbf{x}_t^P$ is held fixed, but the corresponding ancestor indexes $\mathsf{a}_t^P$ are sampled at each time instant $t$ according to some weights $\widetilde{w}_{t-1|T}^i$.

We now focus on the computation on the importance weights $w_t^i$ and the ancestor weights $\widetilde{w}_{t-1|T}^i$. For the former, the particles are weighted according to $w_t^i = W_t(\mathbf{x}_{1:t}^i)$, where

$$\begin{aligned} W_t(\mathbf{x}_{1:t}) &= \frac{p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})}{p(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1}) r_t(\mathbf{x}_t|\mathbf{x}_{1:t-1})} \\ &\propto \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{1:t}) p(\mathbf{x}_{1:t})}{p(\mathbf{y}_{1:t-1}|\mathbf{x}_{1:t-1}) p(\mathbf{x}_{1:t-1}) p(\mathbf{x}_t|\mathbf{x}_{t-1})} \\ &= p(\mathbf{y}_t|\mathbf{x}_{t-L+1:t}), \end{aligned} \tag{4.27}$$

being $\mathbf{y}_{\tau_1:\tau_2}$ the set of observations $\{\mathbf{y}_t\}_{t=\tau_1}^{\tau_2}$. We have applied (4.26) to derive this expression. Eq. 4.27 implies that, in order to obtain the importance weights, it suffices to evaluate the likelihood at time $t$.

The weights $\widetilde{w}_{t-1|T}^i$ used to draw a random ancestor for the reference particle

[4] For clarity, in this section we remove the dependency on the global variables from the notation.

are given by

$$\begin{aligned} \widetilde{w}^i_{t-1|T} &= w^i_{t-1} \frac{p(\mathbf{x}^i_{1:t-1}, \mathbf{x}'_{t:T}|\mathbf{y}_{1:T})}{p(\mathbf{x}^i_{1:t-1}|\mathbf{y}_{1:t-1})} \\ &\propto w^i_{t-1} \frac{p(\mathbf{y}_{1:T}|\mathbf{x}^i_{1:t-1}, \mathbf{x}'_{t:T}) p(\mathbf{x}^i_{1:t-1}, \mathbf{x}'_{t:T})}{p(\mathbf{y}_{1:t-1}|\mathbf{x}^i_{1:t-1}) p(\mathbf{x}^i_{1:t-1})} \\ &\propto w^i_{t-1} p(\mathbf{x}'_t|\mathbf{x}^i_{t-1}) \prod_{\tau=t}^{t+L-2} p(\mathbf{y}_\tau|\mathbf{x}^i_{1:t-1}, \mathbf{x}'_{t:T}). \end{aligned} \tag{4.28}$$

In order to obtain this expression, we have made use of the Markov property of the model, and we have also ignored factors that do not depend on the particle index $i$. Note that the transition probability $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ factorizes across the parallel chains of the factorial model, as shown in (4.26). We also note that, for memoryless models (i.e., $L = 1$), Eq. 4.28 can be simplified, since the product in the last term is not present and, therefore, $\widetilde{w}^i_{t-1|T} \propto w^i_{t-1} p(\mathbf{x}'_t|\mathbf{x}^i_{t-1})$. For $L > 1$, the computation of the weights $\widetilde{w}^i_{t-1|T}$ in (4.28) for $i = 1, \ldots, P$ has computational time complexity scaling as $\mathcal{O}(PM^\ddagger L^2)$. Since this computation needs to be performed for each time instant (and this is the most expensive calculation), the resulting algorithm complexity scales as $\mathcal{O}(PTM^\ddagger L^2)$.

## 4.6 Comparison of FFBS and PGAS

In this section, we design a simple experiment to compare the behavior of FFBS and PGAS when applied in Step 2 of our inference algorithm in Section 4.5. We show that the FFBS approach is more likely to split one underlying chain into several ones, since it estimates the parallel chains in a sequential manner, conditioned on the remaining ones. In contrast, the PGAS kernel allows sampling all parallel chains simultaneously for each time instant, which alleviates this problem. This limitation of FFBS for FHMMs has already been addressed in [122], where an alternative sampling procedure with better mixing properties is proposed.

For this comparison, we generate $T = 500$ 10-dimensional observations using 5 underlying parallel chains, each becoming active at a random initial time instant, uniformly sampled in the set $\{1, 2, \ldots, 250\}$. After activation, each chain continues in the active state for 250 consecutive time instants, with the symbols $x_{tm}$

**Algorithm 1** Particle Gibbs with ancestor sampling

**Input** : Reference particle $\mathbf{x}'_t$ for $t = 1, \dots, T$, and global variables (transition probabilities and emission parameters)

**Output**: Sample $\mathbf{x}^{\text{out}}_{1:T}$ from the PGAS Markov kernel

Draw $\mathbf{x}^i_1 \sim r_1(\mathbf{x}_1)$ for $i = 1, \dots, P-1$ (Eq. 4.26)

Set $\mathbf{x}^P_1 = \mathbf{x}'_1$

Compute the weights $w^i_1 = W_1(\mathbf{x}^i_1)$ for $i = 1, \dots, P$ (Eq. 4.27)

**for** $t = 2, \dots, T$ **do**

`// Resampling and ancestor sampling`

Draw $\mathsf{a}^i_t \sim \text{Categorical}(w^1_{t-1}, \dots, w^P_{t-1})$ for $i = 1, \dots, P-1$

Compute $\widetilde{w}^i_{t-1|T}$ for $i = 1, \dots, P$ (Eq. 4.28)

Draw $\mathsf{a}^P_t \sim \text{Categorical}(\widetilde{w}^1_{t-1|T}, \dots, \widetilde{w}^P_{t-1|T})$

`// Particle propagation`

Draw $\mathbf{x}^i_t \sim r_t(\mathbf{x}_t | \mathbf{x}^{\mathsf{a}^i_t}_{1:t-1})$ for $i = 1, \dots, P-1$ (Eq. 4.26)

Set $\mathbf{x}^P_t = \mathbf{x}'_t$

Set $\mathbf{x}^i_{1:t} = (\mathbf{x}^{\mathsf{a}^i_t}_{1:t-1}, \mathbf{x}^i_t)$ for $i = 1, \dots, P$ (Eq. 4.25)

`// Weighting`

Compute the weights $w^i_t = W_t(\mathbf{x}^i_{1:t})$ for $i = 1, \dots, P$ (Eq. 4.27)

Draw $k \sim \text{Categorical}(w^1_T, \dots, w^P_T)$

**return** $\mathbf{x}^{\text{out}}_{1:T} = \mathbf{x}^k_{1:T}$

being uniformly sampled from the set $\mathcal{A} = \left\{ \frac{1+\sqrt{-1}}{\sqrt{2}}, \frac{1-\sqrt{-1}}{\sqrt{2}}, \frac{-1+\sqrt{-1}}{\sqrt{2}}, \frac{-1-\sqrt{-1}}{\sqrt{2}} \right\}$, and becoming inactive afterwards. The coefficients $\mathbf{h}^\ell_m$ are drawn from their prior in Eq. 4.12, assuming $\sigma^2_\ell = 1$ for all $\ell$. The observations are generated according to the model in Eq. 4.10, with noise variance $\sigma^2_y = 1$.

We set the hyperparameters as $\sigma^2_H = 1$, $\lambda = 0.5$, $\kappa = 1$, $\beta_0 = 0.1$ and $\beta_1 = 2$, and initialize the sampler with $\sigma^2_\ell = \sigma^2_H e^{-\lambda(\ell-1)}$ and $M_+ = 0$, starting the inference procedure by proposing new parallel chains as detailed in Step 1 in Section 4.5. For each value of the memory length $L \in \{1, 2, 3\}$, we run 50 independent simulations, each with different data. For each simulation, we run $10,000$ iterations of both an

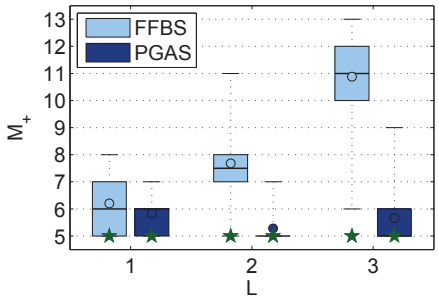



Figure 4.6: Box-plot representation of the inferred number of chains for the FFBS and PGAS approaches. We depict the 25-th, 50-th and 75-th percentiles in the standard format, as well as the most extreme values. The mean value is represented with a circle, and the true number of chains is represented with a green star.

FFBS-based algorithm and a PGAS-based algorithm (Steps 1 and 3 of the inference procedure are common for both methods). We obtain the inferred symbols $x_{tm}$ as the component-wise *maximum a posteriori* (MAP) solution, considering only the last $2,000$ iterations of the sampler.

We show in Figure 4.6 the box-plot representation of the inferred number of chains for the three considered values of $L$ and for both inference methods. The Figure shows that the FFBS approach is more likely to infer a larger value for the number of chains. For the memoryless model ($L = 1$), 14 out of the 50 simulations inferred the true number of chains under the PGAS method, being this quantity 13 for the FFBS approach. However, the effect of FFBS inferring more chains is exacerbated for $L = 2$, where the FFBS-based algorithm finds the true number of chains for only 1 out of the 50 simulations, while the PGAS-based algorithm performs well for 40 cases. For this value of $L$, the FFBS and the PGAS methods infer up to 11 and 7 chains, respectively, as shown in the Figure. For $L = 3$, the FFBS method did not recover $M_+ = 5$ for any of the 50 runs, while the PGAS approach found the true number of chains in 31 cases.

This effect is due to the fact that the FFBS algorithm gets trapped in local modes of the posterior, in which several of the inferred chains jointly explain a single underlying chain. This is a consequence of the sequential sampling method,

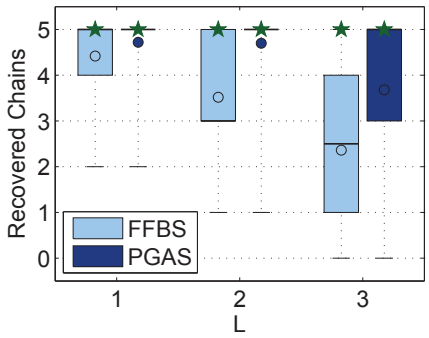



Figure 4.7: Box-plot representation of the recovered number of chains for the FFBS and PGAS approaches (i.e., number of chains with SER $< 0.1$). We depict the 25-th, 50-th and 75-th percentiles in the standard format, as well as the most extreme values. The mean value is represented with a circle, and the true number of chains is represented with a green star.

which requires conditioning on all but one of the hidden chains, making it very unlikely to merge two of them. In contrast, under the PGAS approach, each particle is a vector containing the states of all chains for each time instant, which allows simultaneously estimating all chains when running the inference algorithm.

In order to show that the extra chains are not just spurious chains that become active for a few time instants, we also include Figure 4.7, which shows a box-plot representation of the number of recovered chains. For each independent run, we compute the number of recovered chains as the number of inferred chains which exhibit a symbol error rate (SER) below an arbitrary threshold of 0.1. As the memory length $L$ increases, the FFBS-based algorithm gets more easily trapped in a local mode of the posterior in which one chain has been split into several ones and, hence, the number of recovered chains with error rate below 0.1 decreases.

# 5

# Power Disaggregation

## 5.1 Introduction

The power disaggregation problem consists in, given the aggregated whole-home power consumption signal, estimating both the number of active devices in the house and the power draw of each individual device. 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 [31, 96]. 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.

This problem has been recently addressed in [76] by applying a factorial hidden semi-Markov model (HSMM) and using an expectation maximization (EM)

algorithm, and in [71] using an explicit-duration HDP-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.

All previous works in the literature assume a fixed number of devices, and specific prior information about each individual device. This may represent a limitation in houses that do not fit these assumptions. For instance, the number of active devices in a house might differ by orders of magnitude, and the states of devices may also be different. 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 each home in each city.

In this chapter, we apply Bayesian nonparametric (BNP) models to the power disaggregation problem in order to infer both the power draws of devices and the number of active devices. We use the non-binary infinite factorial hidden Markov model (IFHMM) and the infinite factorial unbounded-state hidden Markov model (IFUHMM), and compare them with the binary IFHMM in [130] and with a standard (parametric) factorial hidden Markov model (FHMM).

In our experiments with real power disaggregation datasets, we show that the binary IFHMM 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.

## 5.2 Experimental Considerations

In Section 5.3, we first design a small scale experiment in which we evaluate the mixing properties of the Markov chain Monte Carlo (MCMC) inference algorithm described in Section 3.5.1, and compare the results with the binary IFHMM in [130] (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 (Section 5.4).

**Databases.** We validate the performance of the proposed IFUHMM applied to the power disaggregation problem in two different real databases, which are described below.

- The Reference Energy Disaggregation Dataset (REDD) [79] 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.

- The Almanac of Minutely Power (AMP) Dataset [87] 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 8 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

$$\text{acc} = 1 - \frac{\sum_{t=1}^{T}\sum_{m=1}^{M}|y_t^{(m)} - \hat{y}_t^{(m)}|}{2\sum_{t=1}^{T}\sum_{m=1}^{M} y_t^{(m)}}, \tag{5.1}$$

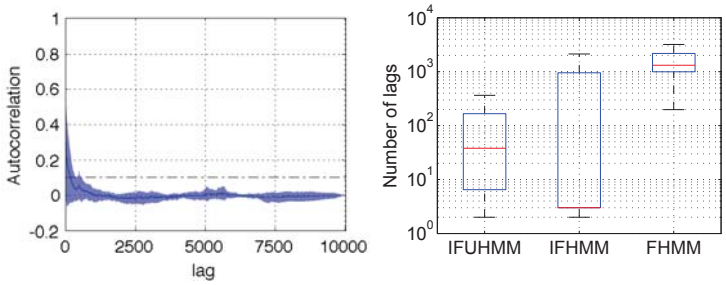
where $y_t^{(m)}$ and $\hat{y}_t^{(m)}$ are, respectively, the true and the estimated power consumption by device $m$ at time $t$ [79]. If the inferred number of devices $M_+$ is smaller than the true number of devices, we use $\hat{y}_t^{(m)} = 0$ for the undetected devices. If $M_+$ is larger than the true number of devices, we group all the extra chains as an "unknown" device and use $y_t^{(\text{unk})} = 0$ 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 #1 in Section 3.3 and, furthermore, the FHMM considers that $\mathbf{a}_0^m$ follows the prior distribution in Eq. 3.6. We set the hyperparameters to $\alpha = 1$, $\gamma = 1$, $\beta_0 = \beta = 1$, $\sigma_0^2 = 0$, $\sigma_\phi^2 = 10$, $\sigma_y^2 = 0.5$, $\boldsymbol{\mu}_0 = 15$ 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 inference algorithm, we estimate the states and observation parameters as $\widehat{s}_{tm} = \arg\max_k q(s_{tm} = k)$ and $\widehat{\boldsymbol{\Phi}}_k = \mathbf{L}_k$.

## 5.3 Small Scale Experiment

In order to evaluate the mixing properties of the inference algorithm in Section 3.5.1, 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 binary 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 (for the IFHMM) and also the number of states (for the IFUHMM) as latent variables, changes the mixing properties of the inference algorithm.

In order 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




(a) (b)

Figure 5.1: Autocorrelation plots for the small scale experiment. (a) Autocorrelation plot for the IFUHMM. (b) Number of samples for the autocorrelation to fall below 0.1.

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. 5.1 and compute it for the last $10,000$ samples of each algorithm.

We show in Figure 5.1a 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 Figure 5.1b 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% of the cases, the IFUHMM needs only a few tens of samples, while for the remaining 50% 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% of the simulations. Finally, the FHMM presents the poorest mixing properties, needing thousands of samples for 75% of the cases.

Now, we evaluate the goodness of fit of the three models. To this end, we show in Figure 5.2 the best (among the 20 samplers) achieved log-likelihood for the three models. In accordance with Figure 5.1b, the IFUHMM converges faster than the IFHMM and the FHMM algorithms. Furthermore, the IFUHMM presents the

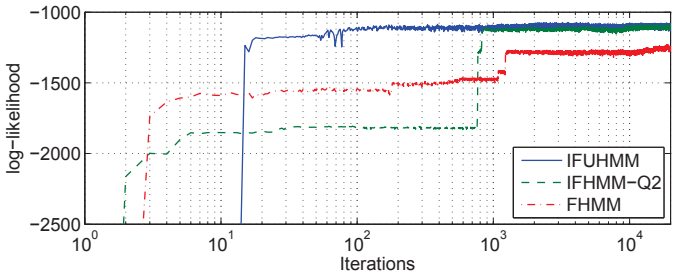



Figure 5.2: Evolution of the log-likelihood for the small scale experiment.

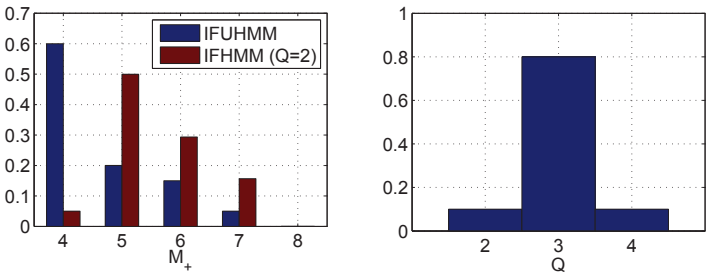
highest log-likelihood score, being the IFHMM almost as good. In addition, we show in Table 5.1 the mean and standard deviation (over the 20 samplers) of the accuracy provided by the three approaches, obtained after averaging the accuracy values of the last $10,000$ samples. We can see that although 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 Figure 5.3a the histogram for the number of inferred chains under the IFUHMM and the IFHMM, and in Figure 5.3b 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% 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 Figure 5.2, 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

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

Table 5.1: Accuracy for the small scale experiment.




(a) (b)

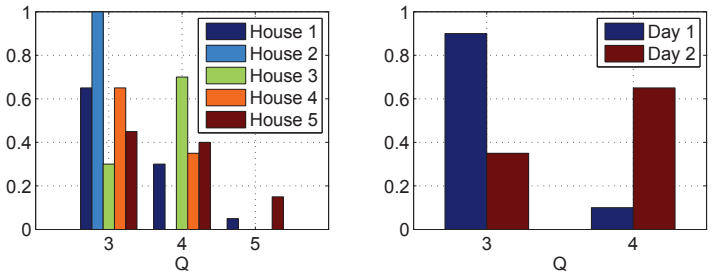
Figure 5.3: Histograms for the small scale experiment. (a) Histogram of the inferred values of $M_+$. (b) Histogram of the inferred values of $Q$ under the IFUHMM.

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.

## 5.4 Experiments with AMP and REDD Datasets

Now, we focus on solving more realistic power disaggregation problems. For the AMP database, we consider two 24-hour segments and the 8 devices detailed above. For the REDD database, we consider a 24-hour segment across 5 houses, with the 6 devices mentioned above. For both databases, we compare the results provided




(a) REDD database. (b) AMP database.

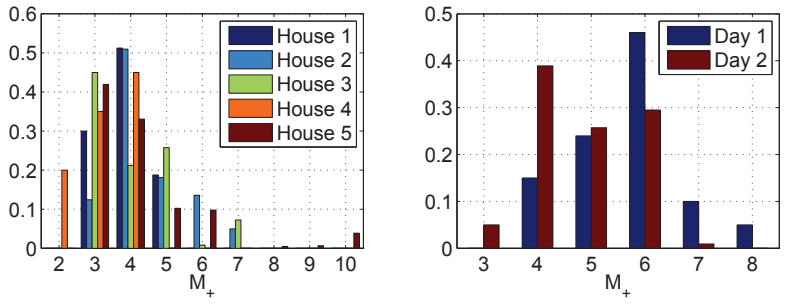
Figure 5.4: Histogram of the inferred values of $Q$ under the IFUHMM.

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 3.2.2, using the variational algorithm in Section 3.4.3 truncated to $M = 15$ Markov chains (Var-Q4).
- The IFHMM with $Q = 4$ states in Section 3.2.2, using the blocked sampling algorithm detailed in Section 3.4.2 (IFHMM-Q4).
- The proposed IFUHMM in Section 3.5.

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.

Figure 5.4 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 the reason 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 [79]. We also show in Figure 5.5 the histograms of the inferred number of chains obtained under the IFUHMM.




(a) REDD database. (b) AMP database.

Figure 5.5: Histogram of the inferred values of $M_+$ under the IFUHMM.

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

Table 5.2: Mean accuracy broken down by house (REDD database).

Tables 5.2 and 5.3 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.

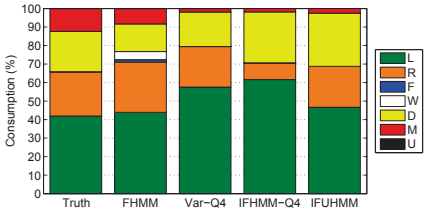
| | Day 1 | Day 2 |
|---|---|---|
| FHMM ($M = 8$, $Q = 4$) | $0.36 \pm 0.05$ | $0.37 \pm 0.05$ |
| Var-Q4 | $0.48 \pm 0.06$ | $0.51 \pm 0.06$ |
| IFHMM-Q4 | $0.58 \pm 0.11$ | $0.58 \pm 0.07$ |
| IFUHMM | $\mathbf{0.69 \pm 0.10}$ | $\mathbf{0.67 \pm 0.11}$ |

Table 5.3: Mean accuracy broken down by day (AMP database).

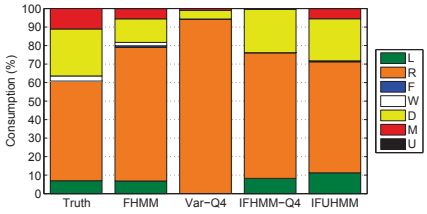
Finally, we depict in Figures 5.6 and 5.7 the true percentage of total power consumed by each device, compared to the inferred percentages by each approach, for both the REDD and AMP databases. Note that assuming a fixed number of chains can be harmful if some of the devices are not switched on at least once during the observation period (see, e.g., the second day of the AMP database in Figure 5.7b). If we now compare these figures to the histograms of the inferred number of chains in Figure 5.5, we can observe that the IFUHMM always captures the most consuming devices (see, e.g., house 1 in Figure 5.5a, which shows that the IFUHMM captures in more than 50% of the cases the true number of devices in Figure 5.6a, where each device consumes more than 10% of the total power). However, when dealing with less consuming devices (see, e.g., the washer-dryer 'W' of house 2 in Figure 5.6b), 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.

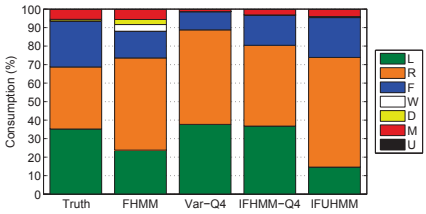
To sum up, our IFUHMM properly detects the active devices in the time series, and indicates that, in general, 3 or 4 states are enough to describe the behavior of the electrical devices. The IFUHMM does not make use of specific prior information to model each individual device but, even so, it is able 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.



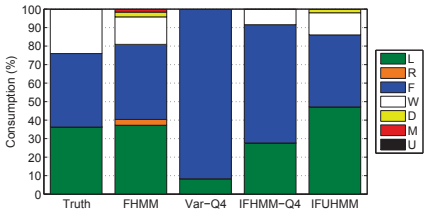

(a) House 1.



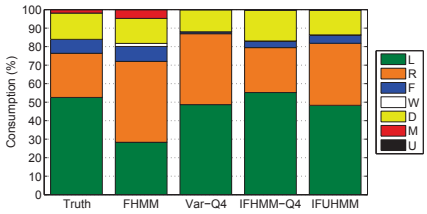

(b) House 2.




(c) House 3.

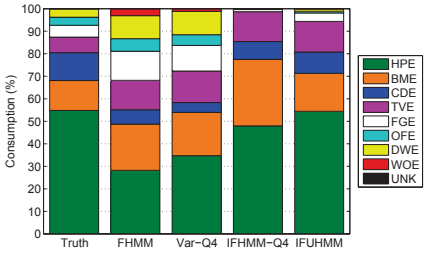



(d) House 4.

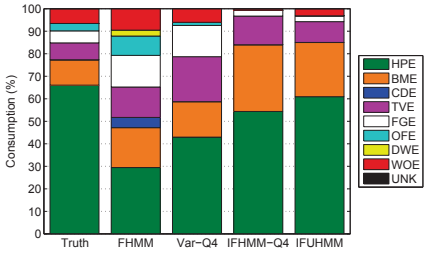



(e) House 5.

Figure 5.6: Percentage of total power consumed by each device (REDD database).




(a) Day 1.




(b) Day 2.

Figure 5.7: Percentage of total power consumed by each device (AMP database).

## 5.5 Discussion

In order to show the proper performance of the proposed inference algorithms in Chapter 3, 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 [130] 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.

# 6

# Blind Multiuser Channel Estimation

## 6.1 Introduction

One of the trends in wireless communication networks (WCNs) is the increase of heterogeneity [4]. It is not new that users of WCNs are no longer only humans talking, and the number of services and uses are booming. Machine-to-machine (M2M) communications and the Internet of Things will shape the traffic in WCNs in the years to come [1, 2, 3, 82]. While there are millions of M2M cellular devices already using second, third and fourth generation cellular networks, the industry expectation is that the number of devices will increase ten-fold in the coming years [33].

M2M traffic, which also includes communication between a sensor/actuator and a corresponding application server in the network, is distinct from consumer traffic, which has been the main driver for the design of fourth generation communication

systems. First, while current consumer traffic is characterized by small number of long lived sessions, M2M traffic involves a large number of short-lived sessions, typically involving transactions of a few hundred bytes. The short payloads involved in M2M communications make it highly inefficient to establish dedicated bearers for data transmission. Therefore, in some cases it is better to transmit small payloads in the random access request itself [27]. Second, a significant number of battery powered devices are expected to be deployed at adverse locations such as basements and tunnels, e.g., underground water monitors and traffic sensors, that demand superior link budgets. Motivated by this need for increasing link budget for M2M devices, transmission techniques that minimize the transmit power for short burst communication are needed [34]. Third, the increasing number of M2M devices requires new techniques on massive access management [123, 62]. Due to these differences, there is a strong motivation to optimize WCNs specifically for M2M communications [33].

The nature of M2M traffic leads to multiuser communication systems in which a large numbers of users may aim to enter or leave the system (i.e., start or stop transmitting) at any given time. In this context, we need a method that allows the users to access the system in a way that the signaling overhead is reduced. We advocate for Bayesian nonparametric (BNP) models, because they can adapt to heterogeneous structures and can incorporate the available information about M2M traffic in their prior.

In this chapter, we focus on the problem of determining the number of users transmitting in a communication system jointly with the channel estimation and the detection of the transmitted data. This problem appears in several specific applications. For instance, in the context of wireless sensor networks, where the communication nodes can often switch on and off asynchronously during operation. It also appears in massive multiple-input multiple-output (MIMO) multiuser communication systems [65, 85], in which the base station has a very large number of antennas and the mobile devices use a single antenna to communicate within the network. In a code-division multiple access (CDMA) context, a set of terminals

randomly access the channel to communicate with a common access point, which receives the superposition of signals from the active terminals only [134].

Our proposed BNP models become flexible enough to account for any number of transmitters, without the need of additional previous knowledge or bounds, due to their nonparametric nature. Moreover, they allow us to solve the problem in a fully unsupervised way, with no signaling data and, therefore, they are suitable for applicability on the random access channel, in which more than one terminal may decide to transmit data. We assume a potentially infinite number of transmitters that might start transmitting short bursts of symbols at any time, such that only a finite subset of the transmitters become active during any finite observation period, while the remaining (infinite) transmitters remain in an idle state (i.e., they do not transmit). Our approach consists in modeling all transmitters as an unbounded number of independent chains in an infinite factorial hidden Markov model (IFHMM), in which each chain (transmitter) has high probability of remaining in its current state (either active or idle). Under this model, the symbols sent by each transmitter can be viewed as a hidden sequence that the receiver needs to reconstruct from the received sequence. Our experimental results show that the proposed approach efficiently solves user identification, channel estimation and data detection in a jointly and fully blind way and, as a consequence, they shed light on the suitability of BNPs applied to signal processing for communications.

We focus on two BNP models for this specific application. We first apply the infinite factorial unbounded-state hidden Markov model (IFUHMM) in Chapter 3, and then the infinite factorial finite state machine (IFFSM) in Chapter 4. The advantages and limitations of both models are described throughout this Chapter.

## 6.2 MIMO Channel

When digital symbols are transmitted over frequency-selective channels, inter-symbol interference (ISI) occurs, degrading the performance of the receiver in terms of symbol detection error probability. To improve the performance, channel estimation is applied to mitigate the effects of ISI. Before detecting the transmitted

symbols, the channel state information (CSI) needs to be estimated at the receiver by sending pilots. Blind channel estimation involves symbol detection without the use of training data, which allows a more efficient communication as the total bandwidth becomes available for the user's data. This can be accomplished either by joint symbol detection and channel estimation or without explicit estimation of the CSI.

We address the problem of blind joint channel parameter and data estimation in a multiuser single-input multiple-output (SIMO) communication channel, which can be treated as a MIMO system. Specifically, we tackle the case where the number of transmitters is unknown. In a MIMO system with $N_t$ transmitters and $N_r$ receiving antennas, each receiver observes a linear combination of all the transmitted data sequences, under additive white Gaussian noise. More specifically, the $N_r$-dimensional observation vector at time instant $t$ is given by

$$\mathbf{y}_t = \sum_{m=1}^{N_t} \sum_{\ell=1}^{L} \mathbf{h}_m^\ell x_{(t-\ell+1)m} + \mathbf{n}_t, \tag{6.1}$$

where $x_{tm}$ denotes the symbol transmitted by the $m$-th transmitter at time instant $t$ (it can also take value 0), $\mathbf{h}_m^\ell$ is the $N_r$-vector that contains the channel coefficients corresponding to tap $\ell$ (with $\ell \in \{1, \ldots, L\}$, being $L$ the channel length for all the transmitter-receiver pairs), and $\mathbf{n}_t$ is the $N_r$-dimensional noise vector. We consider that the noise $\mathbf{n}_t$ is Gaussian distributed with zero mean and covariance matrix $\sigma_n^2 \mathbf{I}_{N_r}$, being $\mathbf{I}_{N_r}$ the identity matrix of size $N_r$. We define the signal-to-noise ratio (SNR) of the MIMO system as

$$\text{SNR(dB)} = -10 \log(\sigma_n^2). \tag{6.2}$$

In Figure 6.1, we show a general scheme of a flat MIMO channel.

Our goal is to infer both the number of transmitters and the transmitted symbols (as well as the channel coefficients $\mathbf{h}_m^\ell$) using the observations collected during $T$ time steps, i.e., the observation vectors $\mathbf{y}_t$ for $t = 1, \ldots, T$. This is a general scenario that represents several specific applications:

- In a CDMA context where a set of terminals wish to communicate with a common access point (AP). Each terminal accesses the channel randomly,

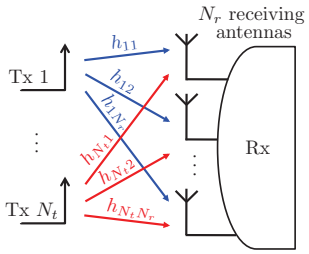



Figure 6.1: MIMO flat channel ($L = 1$) scheme. 'Tx' and 'Rx' are the abbreviations for 'transmitter' and 'receiver', respectively. Each channel coefficient $h_{ij} \triangleq (\mathbf{h}_i^1)_j$ represents the gain between the $i$-th transmitter and the $j$-th receiving antenna.

and the AP receives the superposition of signals from the active terminals only. The AP is interested in determining both the active terminals and the transmitted symbols.

- In the context of wireless sensor networks, where the communication nodes can often switch on and off asynchronously during operation, and a fusion center collects the signals from a subset of them. Again, the fusion center faces the problem of determining both the number of active sensors and the symbols that each sensor transmits [143].

- In cooperation schemes, such as interference alignment [24], in which the reuse of frequencies in nearby cells creates an interference channel between the users and the base stations, being the number of users and the channel they face unknown to the network.

- Massive MIMO [65] assumes that the base station has a very large number of antennas and the mobile devices use a single antenna to communicate within the network.

### 6.2.1 Related Work

A common assumption in frequency-selective MIMO channel estimation is that the channel length is known [89]. As this is not true in general, the usual approach consists in overestimating it, consequentially increasing the complexity of

the receiver, but also introducing a performance degradation that becomes more relevant as the assumed channel length moves away from the actual one [132]. To solve this limitation, many recent papers have addressed the problem of detecting the channel length. In [135, 14, 39], different techniques are proposed to cope with time-invariant channels, assuming either a unique channel length for the complete system or one channel length per transmitting antenna. Similarly, time-varying channels are considered in [105, 50], where a unique channel length for the full system is assumed, or in [133], where one channel length per transmitting-receiving antenna pair is considered. All of these works rely on the fact that the number of transmitters is known and does not vary on time.

For a known channel length (and typically equal to one), several recent papers addressing the problem of user activity and identification can be found in the literature. In [141], a multiuser detector that separates the identification of the active users from the detection of the symbols is proposed. In [55], users are allowed to enter or leave the system only at the beginning of a data frame and, moreover, only one user is allowed to enter at a given frame. The authors in [10] propose a method to identify the number and identity of the active users in a direct-sequence code-division multiple access (DS-CDMA) system, by using a set of training data. Therefore, no symbol detection is performed in this stage. In [143], a Bayesian approach, restricted to the case where the channel has been previously estimated, is presented. A characteristic shared by all these methods is the assumption of an explicit upper bound for the number of transmitters, which makes sense in a DS-CDMA system but may represent a limitation in other scenarios.

All these works consider that either the channel length or the number of active users is fixed and known. Our first contribution is the development of a BNP model, namely, the IFUHMM, that can simultaneously address the channel length estimation and the user activity detection without assuming any upper bounds for these parameters. Our second contribution, namely, the IFFSM model, considers a fixed value for the channel length (which can differ from $L = 1$), but solves many of the limitations of the IFUHMM, as discussed below.

## 6.3 Application of the IFUHMM

In this section, we apply the IFUHMM introduced in Chapter 3 to address the problem of user activity detection and blind channel estimation. The symbols sent by each transmitter can be viewed as a hidden sequence that the receiver needs to reconstruct from the observations, naturally leading to a hidden Markov model (HMM) [103]. Our approach consists on modeling all the transmitters as an infinite number of independent chains in a factorial hidden Markov model (FHMM) [48].

However, the presence of ISI, which occurs when the channel length $L$ in Eq. 6.1 is greater than 1, makes each observation depend not only on the current transmitted symbols, but also on the previous ones. We address this issue by considering for each transmitter the equivalent extended single HMM. Thus, in the IFUHMM, each parallel chain represents a transmitter, and the state at each time instant in the Markov chain corresponds to the state of the channel between that transmitter and all the receivers, being the state of the channel determined by the set of the last $L$ symbols sent by the transmitter. Hence, the set of unknowns is composed of the number of transmitters $N_t$, the symbols sent by each transmitter, the channel length $L$ and the channel coefficients $\mathbf{h}_m^\ell$. Due to its flexibility, our model becomes flexible enough to account for any number of transmitters and channel length in any communication scenario, without the need of additional previous knowledge or bounds.

Under this model, the emission parameters (matrices $\mathbf{\Phi}_q$ in Chapter 3) are closely related to the channel coefficients. Specifically, each row of the matrix $\mathbf{\Phi}_q$ corresponds to the linear combination of the $L$ channel coefficients corresponding to a particular state $q$ of the channel between one transmitter and all the receivers, with $q = 1, \ldots, |\mathcal{X}|^L - 1$. Note that the number of coefficients that are linearly combined to obtain the matrices $\mathbf{\Phi}_q$ coincides with the channel length $L$, which is represented in this model by the number of states $Q$, such that $Q = |\mathcal{X}|^L$. If we assume for simplicity a binary phase-shift keying (BPSK) constellation, i.e., each active symbol $x_{tm} = \pm 1$ with equal probability, then the number of states is $Q = 3^L$. We should expect to recover $Q = 2^L + 1$ (one state to model the inactivity

of the transmitter and $2^L$ for the transmission of the BPSK symbols). Note that, among the $3^L$ possible states of the channel, we are ignoring the $(3^L - 2^L - 1)$ states corresponding to the transition from the activity to the inactivity (and vice-versa) of a transmitter, since in the transmission of a burst of symbols, only two of these transition states appear and, therefore, the inference algorithm will interpret them as noise in most cases, instead of adding additional states.

### 6.3.1 Synthetic Data: Experiments and Results

We now generate a series of examples to illustrate the performance of the proposed IFUHMM. To this end, we simulate a MIMO system for different scenarios, i.e., taking different values for the number of transmitters $N_t$, the number of receivers $N_r$, the channel length $L$, and the SNR, which is defined in Eq. 6.2. In particular, we consider four multiuser communication scenarios:

- *Scenario A*: Flat channel ($L = 1$) with two transmitters ($N_t = 2$) and $N_r = 9$ receivers, for different values of the SNR.
- *Scenario B*: Channel length $L = 2$, $N_t = 2$ and SNR $= 0$ dB, for varying values of $N_r$.
- *Scenario C*: Channel length $L = 2$, $N_t = 3$ and SNR $= 0$ dB, for varying values of $N_r$.
- *Scenario D*: Channel length $L = 2$, $N_r = 25$, and SNR $= 0$ dB, for varying values of $N_t$.

To generate the observations, we assume a number of transmitters $N_t$, each sending a burst of BPSK symbols during the observation period $T = 150$ (i.e., when the transmitter is active, each symbol $x_{tm} = \pm 1$ with equal probability). We assume that the transmitters sequentially become active with random initial instant and burst duration, ensuring that the burst consists in the transmission of at least 30 symbols since shorter bursts are unusual in a real communication system. As we described in Section 6.2, the channel is assumed to be Rayleigh,

i.e., the channel coefficients are Gaussian distributed with zero mean and unit variance, and the observations are corrupted by Gaussian additive noise with zero mean and variance $\sigma_n^2$. For the Scenarios A, B and C we run 500 independent simulations for each combination of the SNR and $N_r$ values, and for Scenario D we run 100 simulations for each value of $N_t$.

We evaluate the performance of the model in terms of detection error probability (DEP), defined as the error probability of detecting both the true number of transmitters and the true channel length. We account for an error either when $M_+$ differs from $N_t$ or when $Q$ is different than $2^L + 1$. Additionally, for those cases where the true values for the number of transmitters and channel length are recovered, we also evaluate the symbol error rate (SER), the activity detection error rate (ADER), and the mean square error (MSE) of the channel coefficient estimates. When computing the SER, an error is computed at time $t$ whenever the estimated symbol for a transmitter differs from the actual transmitted symbol, given that the transmitter is active. Regarding the ADER, it is the probability of detecting activity (inactivity) in a transmitter while that transmitter is actually inactive (active). Additionally, if we denote by $\hat{\mathbf{h}}_m^\ell$ the inferred channel coefficients, we compute the MSE as

$$\text{MSE} = \frac{1}{LN_tN_r} \sum_{m,k,\ell} \left( (\mathbf{h}_m^\ell)_k - (\hat{\mathbf{h}}_m^\ell)_k \right)^2, \tag{6.3}$$

where we estimate the coefficients using the *maximum a posteriori* (MAP) solution of the matrices $\mathbf{\Phi}_q$.

We assume the Gaussian observation Model #2 in Section 3.3. For each experiment, we run 50 iterations of our inference algorithm[1] presented in Section 3.5.1, using Gibbs sampling to infer the latent matrix $\mathbf{S}$. Note that, although we have adapted the observation model to properly fit MIMO systems, the proposed model still suffers from several limitations because, although the number of possibles active states in the channel is a power of two (i.e., $2^L$, ignoring the effects of the inactive state), our model allows any integer value above 1. Then, we resort to an

[1]The hyperparameters are set to $\alpha = 1$, $\gamma = 1$, $\beta_0 = 0.1$, $\beta = 10$, $\lambda = 1$, $\tau = 1$, $\nu = 0.1$ and $\xi = 2\sigma_n^2$ (the SNR is known at the receiver because it usually has a SNR estimator device).

additional post-processing of the inference results to account for the prior knowledge of the communication system. Specifically, we rearrange the elements of the inferred matrix $\mathbf{S}$, so that the inferred matrices $\mathbf{\Phi}_q$ properly recover the channel coefficients. We repeat the previous procedure, consisting on the 50 iterations of the inference algorithm and post-processing, initialized with the results of the first post-processing.

We compare our algorithm, denoted by IFUHMM in the plots, with two genie-aided methods, in which the model parameters, i.e., number of transmitters and channel parameters[2] (including the channel length and coefficients), are known:

- We run the optimum BCJR algorithm [15], denoted by BCJR in the plots, over a single HMM with a number of states equal to $3^{LN_t}$, i.e., where the constellation includes the symbol 0 (corresponding to the inactivity of a transmitter). As the complexity of this algorithm increases exponentially with $L$ and $N_t$, it can be only run for Scenarios A and B.

- We run 1000 iterations of a forward-filtering backward-sampling (FFBS) method [46, 25], sequentially applied in each chain of an FHMM after removing the contribution of the remaining chains in the observations. We assume that the number of states in each Markov chain is $3^L$, i.e., the inactive symbol is included in the constellation.

For the Scenario A, we show in Fig. 6.2 the DEP, the ADER, the SER and the MSE as functions of the SNR. As expected, in all these plots we observe that the performance of the proposed algorithm (initialized with $Q = 2$ states) improves as the SNR increases. Note also that the performance of our algorithm is not far from the optimum BCJR and is comparable to the FFBS approach. The DEP and MSE are not reported for BCJR or FFBS algorithms, since the CSI is known in both cases.

For the Scenario B, we show in Fig. 6.3 the DEP, the ADER, the SER and the MSE as functions of the number of receivers $N_r$, initializing the algorithm

[2]The probability of a user remaining active or inactive is set to 0.8, being the two active symbols equally probable.

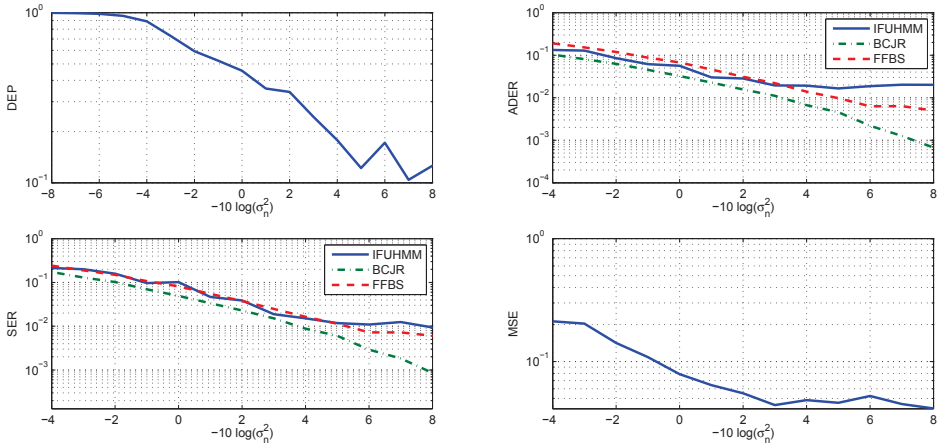



Figure 6.2: IFUHMM results for the Scenario A ($L = 1$, $N_t = 2$ and $N_r = 9$).

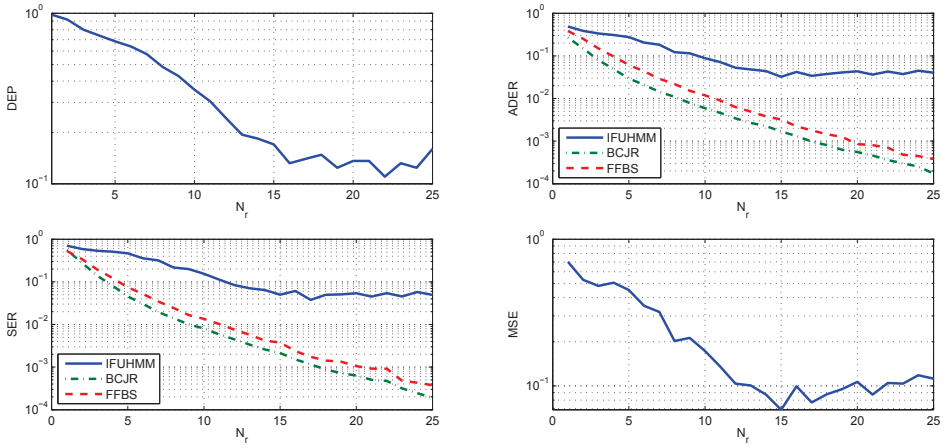



Figure 6.3: IFUHMM results for the Scenario B ($L = 2$, $N_t = 2$ and SNR $= 0$ dB).

with $Q = 6$ states. Note that the behavior of our inference improves as the number of receivers increases until $N_r = 15$, but for higher values of $N_r$ the DEP is around 10% and the ADER, the SER and the MSE also remain approximately constant. As in this scenario there are more parameters to be inferred, the genie-aided methods outperform our fully blind algorithm.

Let us analyze the performance of the inference algorithm in the Scenario C. To this end, in Fig. 6.4 we plot the DEP, the ADER, the SER and the MSE as functions of the number of receivers $N_r$ for several initializations of the number of states $Q$ (denoted by $Q_{ini}$ in the plots). The top plot shows that the DEP is much

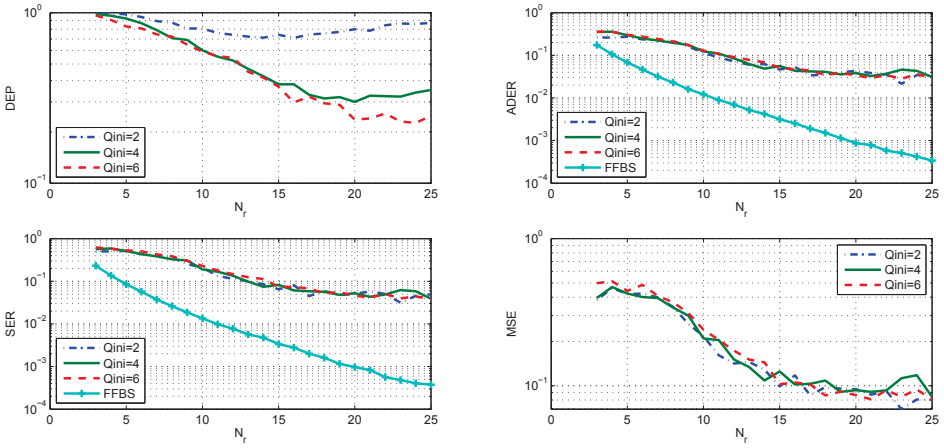



Figure 6.4: IFUHMM results for the Scenario C ($L = 2$, $N_t = 3$ and SNR $= 0$ dB).

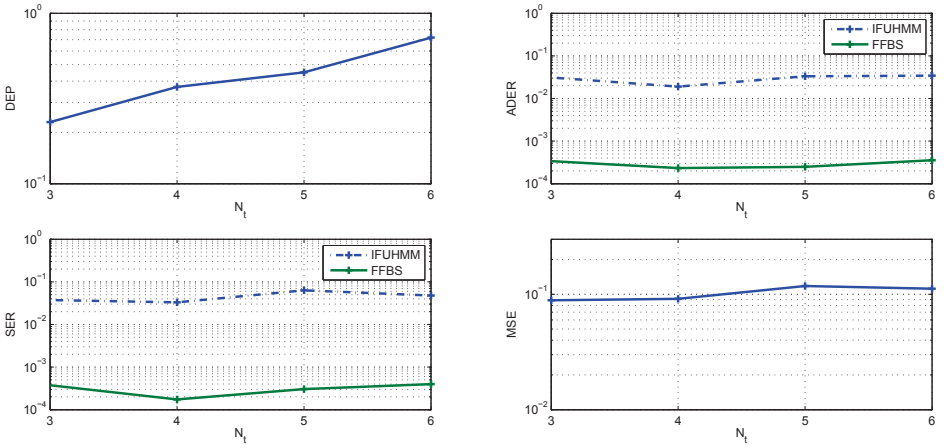



Figure 6.5: IFUHMM results for the Scenario D ($L = 2$, $N_r = 25$ and SNR $= 0$ dB).

higher when the algorithm is initialized with $Q_{ini} = 2$ states. This behavior is due to the fact that the sampler gets trapped in a local optimum different from the ground truth. In the cases in which $Q$ is initialized to 4 or 6, we find similar results, being $Q_{ini} = 6$ slightly better in terms of DEP. However, once the algorithm finds the true values of both the number of transmitters and the number of states, the performance of the model is similar regardless of the initialization.

Finally, for the Scenario D, we depict in Fig. 6.5 the DEP, the ADER, the SER and the MSE as functions of the number of transmitters $N_t$ (varying from 3 to 6), initializing the sampler with $Q = 6$ states. As the number of unknowns

grows with the number of transmitters, our algorithm provides better results in terms of DEP for $N_t = 3$. However, when the ground truth is recovered, we can see that the ADER, the SER and the MSE do not highly depend on the number of transmitters. The FFBS algorithm also exhibits this behavior.

Under Scenarios B, C and D, we observe in the SER and the ADER plots the presence of an error floor (above $10^{-2}$) corresponding to the errors caused by the active-to-inactive (and inactive-to-active) transitions, which are not taken into account in our model. These error floors can be decreased by transmitting larger bursts of symbols, i.e., they are less significant when the number of transitions becomes negligible compared to the total number of transmitted bits.

## 6.4 Application of the IFFSM

In this section, we apply the IFFSM model in Chapter 4 to address the problem of user activity detection and blind channel estimation. Similarly to the methodology above, we also adopt a BNP approach and consider a factorial model with a potentially infinite number of hidden Markov chains, representing transmitters. The key difference with respect to the IFUHMM is that we explicitly encode our prior knowledge about the CSI through finite state machines (FSMs). Under this model, each input $x_{tm}$ to the FSMs directly represents the symbol transmitted by the $m$-th user at time instant $t$. Hence, the set of unknowns is composed of the number of transmitters $N_t$, the symbols sent by each transmitter, the channel coefficients $\mathbf{h}_m^\ell$, and the variances $\sigma_\ell^2$.

### 6.4.1 Synthetic Data: Experiments and Results

We run a battery of experiments to illustrate the performance of the proposed IFFSM and the corresponding inference algorithm based on particle Gibbs with ancestor sampling (PGAS) described in Chapter 4. To this end, we simulate different scenarios of a multiuser communication system, considering different values for the number of transmitters $N_t$, the number of receivers $N_r$, the SNR, the channel memory $L$ and the constellation order.

To generate the observations, we assume that each of the $N_t$ transmitters sends a burst of symbols during the observation period of length $T = 1000$. Transmitters use quadrature amplitude modulation (QAM) with cardinality $|\mathcal{A}|$, being the symbols in the constellation normalized to yield unit energy. We assume that each transmitter becomes active at a random instant, uniformly sampled in the interval $[1, T/2]$, being the burst duration $T/2$. A Rayleigh additive white Gaussian noise (AWGN) channel is assumed, i.e., the channel coefficients and the noise are circularly symmetric complex Gaussian distributed with zero mean, being the covariances matrices $\sigma_\ell^2\mathbf{I}$ and $\sigma_n^2\mathbf{I}$, respectively. We assume $\sigma_\ell^2 = 1$ for all $\ell$, while $\sigma_n^2$ depends on the considered SNR, as given in Eq. 6.2.

We choose the following parameters to run our experiments: $N_t = 5$ transmitters, $N_r = 20$ receiving antennas, $|\mathcal{A}| = 4$ symbols in the constellation and SNR $= -3$ dB. Using this base configuration, we vary one of the parameters while holding the rest fixed. We report results with $L = 1$ (i.e., no memory), as well as for higher values of $L$. The hyperparameters are set as $\sigma_y^2 = \sigma_n^2$, $\sigma_H^2 = 1$, $\lambda = 0.5$, $\kappa = 1$, $\alpha = 1$, $\beta_0 = 0.1$ and $\beta_1 = 2$. The choice of $\beta_0$ and $\beta_1$ is based on the fact that we expect the active Markov chains to remain active and, therefore, the transition probabilities from active to inactive $b^m$, which are Beta$(\beta_0, \beta_1)$ distributed, are a priori expected to be small.

**Tempering procedure.** We observed in our experiments that performance (in terms of error rates) degrades with the increase of the SNR. This counter-intuitive effect can be easily understood by taking into account the posterior distribution and the inference procedure. When the SNR is high enough, the noise variance is too small compared to the variance of the channel coefficients, which makes the posterior get narrow around the true value of these coefficients. In other words, the posterior uncertainty on the channel coefficients becomes small, and similarly for the transmitted symbols. As a consequence, an inference algorithm based on random exploration of the posterior needs more iterations to find the peaks of the posterior distribution. In practice, we cannot afford such large number of iterations.

Instead, we propose a solution based on an heuristic to artificially widen the posterior distribution. For that purpose, we add artificial noise to the observations, consequently decreasing the SNR. From an "exploration versus exploitation" perspective, this method eases exploration of the posterior. At each iteration of the algorithm, we slightly increase the SNR by reducing the variance of the artificial noise, and we repeat this procedure until we reach the actual value of the SNR. After that, we run additional iterations to favor exploitation.

In our experiments, we initialize the inference algorithm with SNR $= -12$ dB, increasing this number by 0.002 dB at each iteration of the algorithm.

**Evaluation.** In a realistic digital communication system, the transmitted symbols are protected with redundancy codes that allow detection and correction of transmission errors at the receiver side (e.g., low density parity check codes). For that reason, we assume that the receiver can detect and ignore those inferred transmitters with high SER. For the recovered transmitters (those with SER below a threshold of 0.1), we evaluate the performance in terms of the ADER, the SER, and the MSE of the channel coefficient estimates. The ADER is the probability of detecting activity (inactivity) in a transmitter while that transmitter is actually inactive (active). When computing the SER, an error is computed at time $t$ whenever the estimated symbol for a transmitter differs from the actual transmitted symbol, considering that the transmitted symbol while inactive is $x_{tm} = 0$. We compute the MSE for each transmitter as

$$\text{MSE}_m = \frac{1}{LN_r} \sum_{k,\ell} \left|\left|(\mathbf{h}_m^\ell)_k - (\hat{\mathbf{h}}_m^\ell)_k\right|\right|^2. \tag{6.4}$$

We compare our approach (denoted by IFFSM in the plots) with three genie-aided methods which have perfect knowledge of the true number of transmitters and channel coefficients.[3] In particular, we run: (i) The PGAS algorithm that we use in Step 2 of our inference algorithm (G-PGAS); (ii) the FFBS algorithm over the equivalent FHMM with $N_t$ Markov chains and $Q = |\mathcal{A} \bigcup \{0\}|^L$ states (G-FFBS); and (iii) the optimum BCJR algorithm [15], over an equivalent single

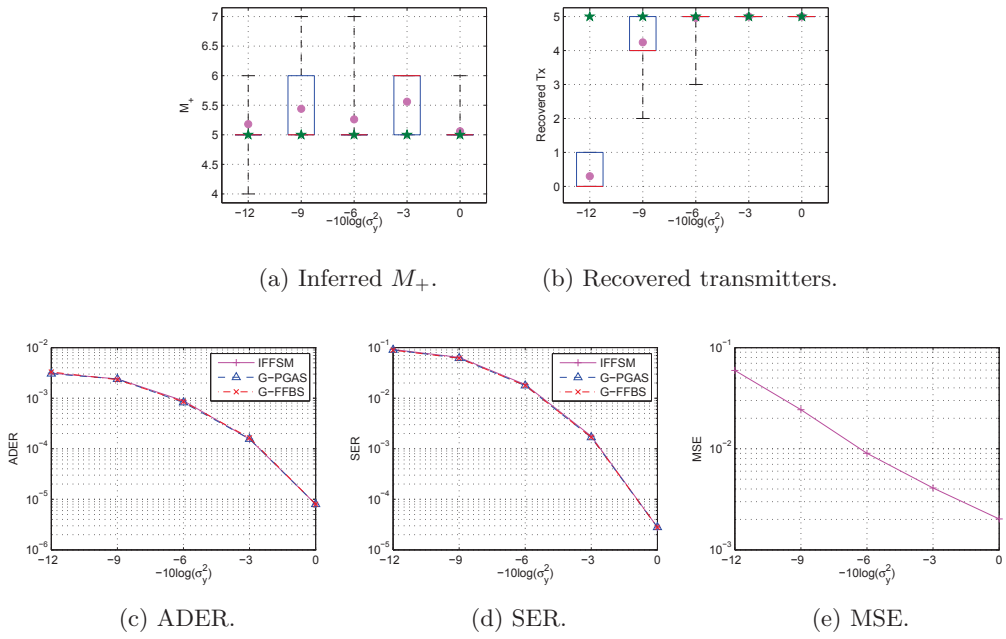
[3]For the genie-aided methods, we use $a^m = 0.998$ and $b^m = 0.002$.

HMM with a number of states equal to $|\mathcal{A} \bigcup \{0\}|^{LN_t}$ (G-BCJR). Due to exponential time complexity limitations, we only run the BCJR algorithm in scenarios with $|\mathcal{A} \bigcup \{0\}|^{2LN_t} \leq 10^6$, and the FFBS in scenarios with $|\mathcal{A} \bigcup \{0\}|^{2L} \leq 10^6$.

For each considered scenario, we run 50 independent simulations, each with different simulated data. We run $20,000$ iterations of our inference algorithm, finally obtaining the inferred symbols $\hat{x}_{tm}$ as the component-wise *maximum a posteriori* (MAP) solution, only considering the last $2,000$ iterations of the sampler. The estimates of the channel coefficients $\hat{\mathbf{h}}_m^\ell$ are then obtained as the MAP solution, conditioned on the data and the inferred symbols $\hat{x}_{tm}$. For the BCJR algorithm, we obtain the symbol estimates according to the component-wise MAP solution for each transmitter $m$ and each instant $t$. For the genie-aided PGAS and FFBS methods, we follow a similar approach by running the algorithms for $10,000$ iterations and considering the last $2,000$ samples to obtain the symbol estimates. Unless otherwise specified, we use $P = 3,000$ particles for the PGAS kernel.

**Results for memoryless channels.** We first evaluate the performance of our model and inference procedure for memoryless channels, i.e., considering $L = 1$. Figure 6.6 shows the results when the SNR varies from $-12$ dB ($\sigma_n^2 \approx 15.85$) to 0 dB ($\sigma_n^2 = 1$). Specifically, we show the ADER, the SER, the MSE, a box-plot representation[4] of the inferred number of transmitters $M_+$, and also a box-plot representation of the number of recovered transmitters (i.e., how many transmitters we recover with a SER below the threshold of 0.1). As expected, we obtain a better performance as the SNR increases. For low values of the SNR, transmitters are more likely to be masked by the noise and, therefore, we tend to recover a lower number of transmitters. We also observe that the performance (in terms of ADER and SER) of the proposed IFHMM reaches similar values to the methods with perfect knowledge of the number of transmitters and channel coefficients.

[4]We depict the 25-th, 50-th and 75-th percentiles in the standard format, as well as the most extreme values. Moreover, the mean value is represented with a pink circle, and the true number of transmitters $N_t$ is represented with a green star.



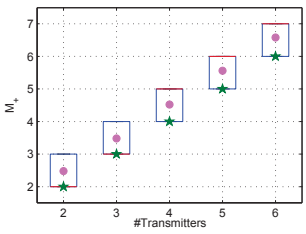

(a) Inferred $M_+$. (b) Recovered transmitters.

(c) ADER. (d) SER. (e) MSE.

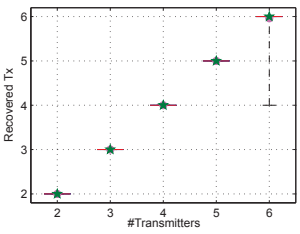
Figure 6.6: IFFSM results for different SNRs ($L = 1$).

Figure 6.7 shows the results when the true number of transmitters $N_t$ changes from 2 to 6. Although a higher value of $N_t$ implies a higher number of parameters to be estimated, we observe that the performance is approximately constant. The IFHMM recovered all the transmitters in nearly all the simulations, with performance similar to the genie-aided methods.

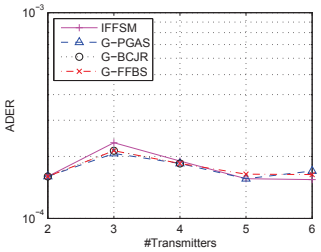
Figure 6.8 shows the results when the number of receiving antennas $N_r$ varies from 2 to 30. In this figure, we observe that we need at least 8 receivers in order to properly recover the transmitted symbols of all the transmitters. As expected, the performance in terms of ADER and SER improves when the number of receiving antennas increases, as the diversity in the observations helps to recover the transmitted symbols and channel coefficients. This behaviour is similar to the obtained by the genie-aided PGAS and FFBS, as shown in this figure. Note also that the MSE reaches a plateau for 15-20 receivers. After this value, adding more receivers does not improve the average MSE, but the extra redundancy in the observed sequences helps improve the ADER and SNR.



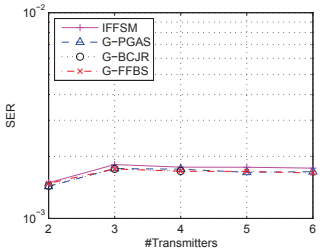

(a) Inferred $M_+$.



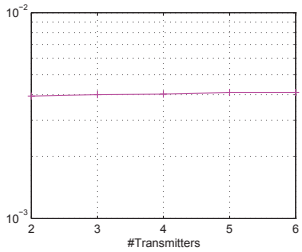

(b) Recovered transmitters.




(c) ADER.




(d) SER.

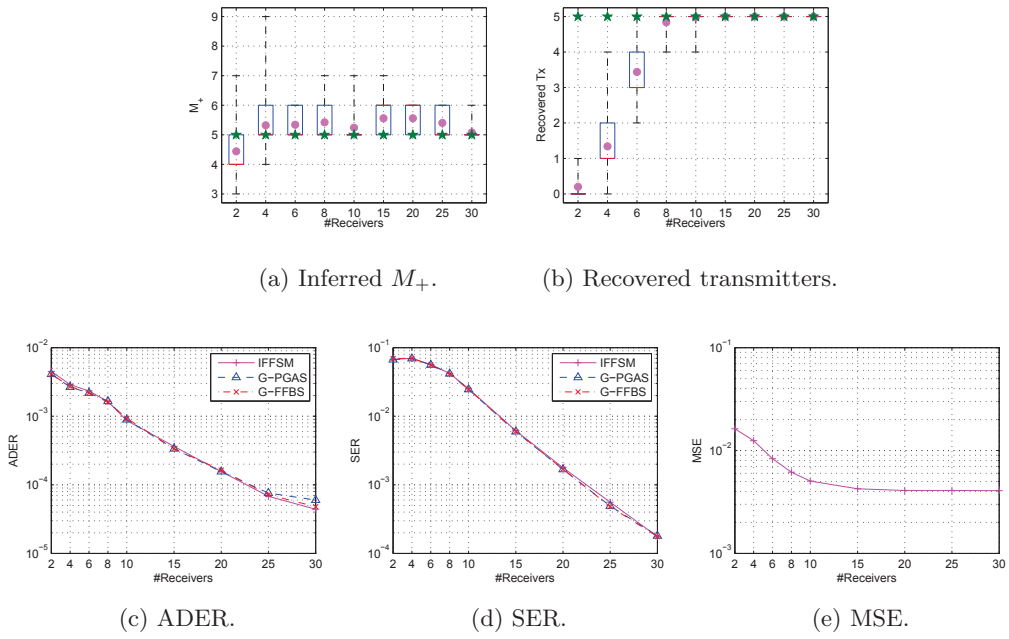



(e) MSE.

Figure 6.7: IFFSM results for different number of transmitters ($L = 1$).

**Sensitivity to the number of particles.** As the effective dimensionality of the hidden space increases, we should expect a larger number of particles to be required in order to properly estimate the transmitted symbols. To see this, we design an experiment with $N_t = 10$ transmitters and SNR $= -3$ dB. Figure 6.9 shows the log-likelihood trace plot for $10,000$ iterations of the inference algorithm, with a number of particles ranging from 300 to $30,000$. Although these results are based on a single run of the algorithm, it can be seen that the best performance is achieved with the largest number of considered particles. Additionally, this plot suggests that $P = 10,000$ particles are enough for this scenario.

We also show in Figure 6.10 the number of inferred transmitters $M_+$, as well as the number of recovered transmitters, for each value of $P$. In this figure, we represent with a green star the true value of $N_t$. (Again, these results are obtained after a single run of the algorithm.) Although we infer $M_+ = 10$ transmitters with only $P = 3,000$ particles, Figure 6.10b shows that only 8 of them exhibit a SER below the threshold of 0.1. In agreement with Figure 6.9, increasing the number




(a) Inferred $M_+$. (b) Recovered transmitters.

(c) ADER. (d) SER. (e) MSE.

Figure 6.8: IFFSM results for different number of receiving antennas ($L = 1$).

of particles from $P = 10,000$ to $30,000$ does not seem to improve performance: in both cases our algorithm is able to recover all the transmitters. Even the genie-aided PGAS algorithm, which has perfect knowledge of the channel coefficients, needs a large value of $P$ (above $3,000$) in order to recover all the transmitters.

We can conclude from these plots that we should adjust the number of particles based on the number of transmitters. However, the number of transmitters is an unknown quantity that we need to infer. There are two heuristic ways to overcome this apparent limitation. A straightforward solution is to adaptively adapt the number of particles $P$ as a function of the current number of active transmitters, $M_+$. In other words, as we gather evidence for the presence of more transmitters, we consequently increase $P$. A second approach, which is computationally less demanding but may present poorer mixing properties, consists in running the PGAS inference algorithm sequentially over each chain, conditioned on the current value of the remaining transmitters, similarly to the standard FFBS procedure for IFHMMs [130]. Alternatively, we can apply the PGAS algorithm over fixed-sized

blocks of randomly chosen transmitters.

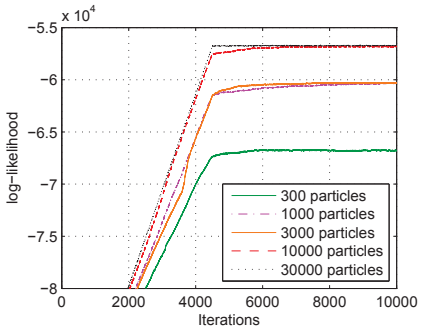
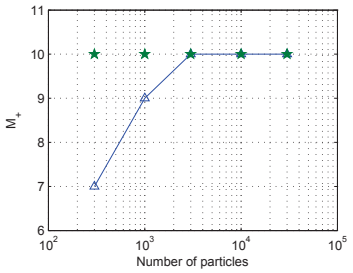
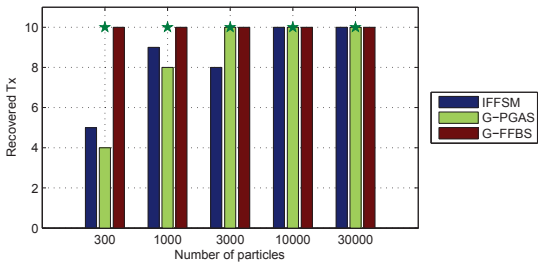



Figure 6.9: Log-likelihood for varying number of particles ($L = 1$). The initial slope is due to the tempering procedure, in which we slightly increase the SNR at each iteration.




(a) Inferred $M_+$.



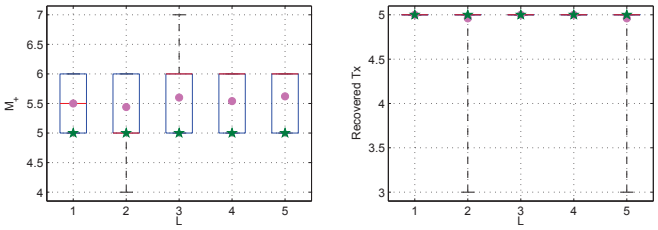

(b) Recovered transmitters.

Figure 6.10: Number of inferred and recovered transmitters for varying number of particles ($L = 1$).

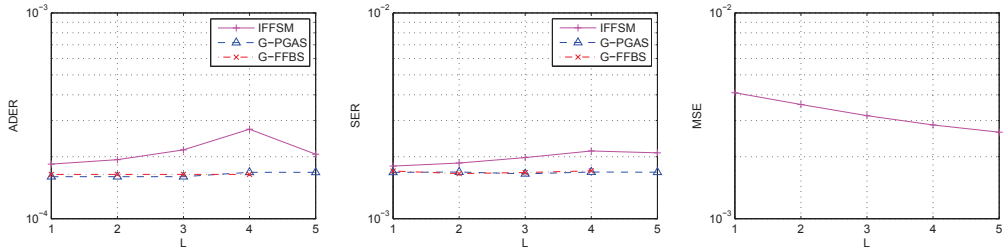
**Results for inaccurate values of the channel length.** So far, we have used $L = 1$ to generate the data, and we have assumed this value is known at the receiver side. We now run an experiment to show that we can properly estimate the transmitted symbols and the channel coefficients as long as our inference algorithm considers a sufficiently large value of $L$. For this purpose, we use our base experimental setup with $N_t = 5$ transmitters and $P = 3,000$ particles, and generate data using $L = 1$ (i.e., memoryless channel). However, we use different values for the channel length $L$ for inference.

In Figure 6.11, we show the obtained results for $L$ ranging from 1 to 5. The obtained ADER and SER do not significantly degrade with increasing values of $L$,

and we are able to recover the five transmitters in nearly all the cases. Interestingly, the MSE improves as $L$ increases. This is a consequence of the way we measure it when $L$ is larger than the ground truth, as we compare our channel estimates with zero. The fact that the MSE becomes lower indicates that we obtain better estimates for the zero coefficients than for the non-zero ones, which in turn implies that our inference algorithm can properly reduce the channel variances $\sigma_\ell^2$ when needed.




(a) Inferred $M_+$. (b) Recovered transmitters.

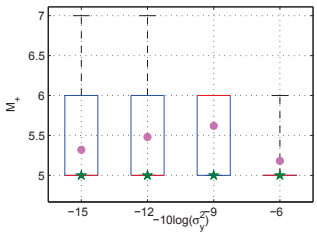



(c) ADER. (d) SER. (e) MSE.

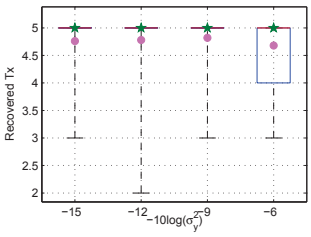
Figure 6.11: IFFSM results for different channel lengths ($L_{\text{true}} = 1$).

**Results for larger channel lengths.** We now evaluate the performance of our model and inference procedure for channels with memory, i.e., considering $L > 1$. Figure 6.12 shows the results when the SNR varies from $-15$ dB ($\sigma_n^2 \approx 31.62$) to $-6$ dB ($\sigma_n^2 \approx 3.98$), considering $L = 5$ to generate the data. We use the true value of the channel length $L$ for inference. In the figure, we show the ADER, the SER, the MSE, a box-plot representation of the inferred number of transmitters $M_+$, and also a box-plot representation of the number of recovered transmitters. As in the memoryless case, we obtain a better performance as the SNR increases. In contrast

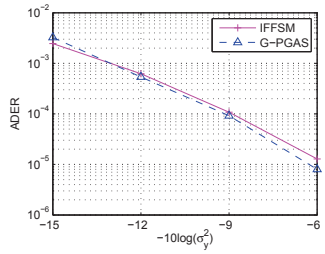
to the memoryless case, in most experiments we recover the five transmitters even for SNR $= -15$ dB. This makes sense, because the channel memory adds more redundancy in the observed sequence. Our inference algorithm is able to exploit such redundancy to better estimate the transmitted symbols, despite the fact that more channel coefficients need to be estimated. Note that the performance in terms of ADER and SER is similar to the genie-aided PGAS-based method.



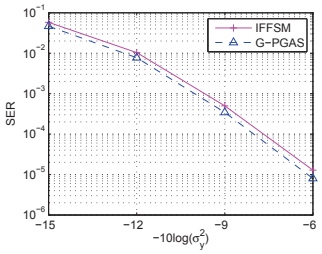

(a) Inferred $M_+$.



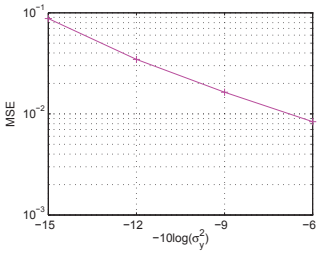

(b) Recovered transmitters.




(c) ADER.




(d) SER.

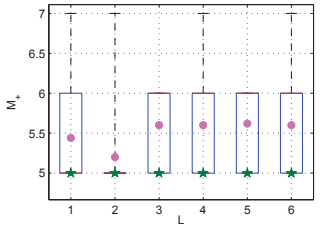



(e) MSE.

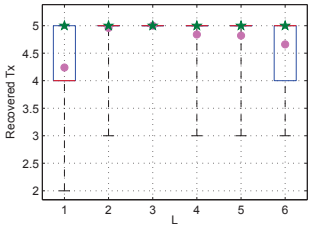
Figure 6.12: IFFSM results for different SNRs ($L = 5$).

In Figure 6.13, we show the obtained results for different values of the parameter $L$, ranging from 1 to 6. We use the true value of the channel length $L$ for inference, and we consider SNR $= -9$ dB in these experiments. The figure shows the ADER, the SER, the MSE, and box-plot representations of the inferred number of transmitters $M_+$ and the number of recovered transmitters. Here, it becomes clear that our model can exploit the redundancy introduced by the channel memory, as the performance in terms of SER and ADER improves as $L$ increases. The MSE also improves with $L$, although it reaches a constant value for $L > 3$, similarly to the experiments in which we increase the number of receivers (although

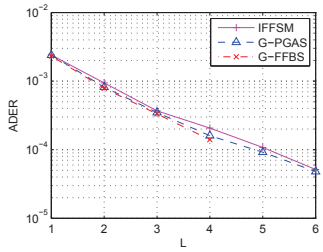
differently, in both cases we add redundancy to the observations). We can also observe that the performance is similar to the genie-aided methods (we do not run the FFBS algorithm for $L \geq 5$ due to its computational complexity).



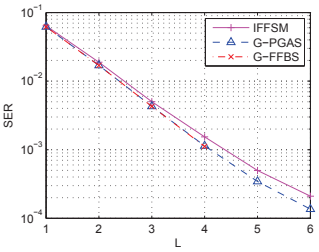

(a) Inferred $M_+$.



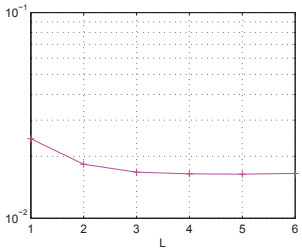

(b) Recovered transmitters.




(c) ADER.




(d) SER.




(e) MSE.

Figure 6.13: IFFSM results for different values of $L$.

### 6.4.2 Real Data: Experiments and Results

With the aim of considering a more realistic communication scenario, we use WISE software [42] to design an indoor wireless system. This software tool, developed at Bell Laboratories, includes a 3D ray-tracing propagation model, as well as algorithms for computational geometry and optimization, to calculate measures of radio-signal performance in user-specified regions. Its predictions have been validated with physical measurements.

Using WISE software and the map of an office located at Bell Labs Crawford Hill, we place $N_r = 12$ receivers and $N_t = 6$ transmitters across the office, intentionally placing the transmitters together in order to ensure that interferences occur in the nearby receivers. Figure 6.14 shows the considered map.

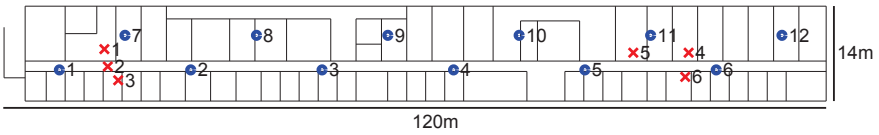



Figure 6.14: Plane of the considered office building. Circles represent receivers, and crosses represent transmitters. All transmitters and receivers are placed at a height of 2 metres.

We consider a Wi-Fi transmission system with a bandwidth of 20 MHz or, equivalently, 50 ns per channel tap. We simulate the transmission of $1,000$-symbol bursts over this communication system, using a quadrature phase-shift keying (QPSK) constellation normalized to yield unit energy. We scale the channel coefficients by a factor of 100, and we consequently scale the noise variance by $10^4$, yielding $\sigma_n^2 \approx 7.96 \times 10^{-9}$. We set the transmission power to 0 dBm. Each transmitter becomes active at a random point, uniformly sampled in the interval $[1, T/2]$, and we consider an observation period of $T = 2,000$. This ensures overlapping among all the transmitted signals.

Wi-Fi systems are not limited by the noise level, which is typically small enough, but by the users' interferences, which can be avoided by using a particular frequency channel for each user. Our goal is to show that cooperation of receivers in a Wi-Fi communication system can help recover the symbols transmitted by several users even when they simultaneously transmit over the same frequency channel, therefore allowing for a larger number of users in the system.

In our experiments, we vary $L$ from 1 to 5. Five channel taps correspond to the radio signal travelling a distance of 750 m, which should be enough given the dimensions of the office (the signal suffers attenuation when it reflects on the walls, so we should expect it to be negligible in comparison to the line-of-sight ray after a 750-m travelling distance). Following the tempering procedure as above, we initialize the algorithm with $\sigma_y^2 \approx 15.85$ and we linearly increase the SNR for around $26,600$ iterations, running $3,400$ additional iterations afterwards. We compare our IFFSM with a non-binary IFHMM model with state space cardinality $|\mathcal{X}| = 5^L$ using FFBS sweeps for inference (we do not run the FFBS algorithm for $L = 5$ due to its computational complexity). We set the hyperparameters as

| Model | $L$ | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| IFFSM | 6/6 | 6/6 | 6/6 | 6/6 | 6/6 |
| IFHMM | 3/11 | 3/11 | 3/8 | 1/10 | – |

(a) # Recovered transmitters / Inferred $M_+$.

| Model | $L$ | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| IFFSM | 2.58 | 2.51 | 0.80 | 0.30 | 0.16 |
| IFHMM | 2.79 | 1.38 | 5.53 | 1.90 | – |

(b) MSE of the channel coefficients ($\times 10^{-6}$).

Table 6.1: Results for the Wi-Fi experiment.

$\sigma_y^2 = \sigma_n^2$, $\sigma_H^2 = 0.01$, $\lambda = 0.5$, $\kappa = 1$, $\alpha = 1$, $\beta_0 = 0.1$ and $\beta_1 = 2$.

We show in Table 6.1a the number of recovered transmitters (defined as the number of transmitters for which we recover all the transmitted symbols with no error) found after running the two inference algorithms, together with the inferred value of $M_+$, averaged for the last $2,000$ iterations. We see that the IFHMM tends to overestimate the number of transmitters, which deteriorates the overall symbol estimates and, as a consequence, not all the transmitted symbols are recovered. We additionally report in Table 6.1b the MSE of the first channel tap, i.e., $\frac{1}{6\times 12}\sum_m ||\mathbf{h}_m^1 - \widehat{\mathbf{h}}_m^1||^2$, averaged for the last $2,000$ iterations, being $\widehat{\mathbf{h}}_m^\ell$ the inferred channel coefficients. We sort the transmitters so that the MSE is minimized, and ignore the extra inferred transmitters. As expected, for our IFFSM, the MSE decreases as we consider a larger value of $L$, since the model better fits the actual radio propagation model. However, the IFHMM fails in estimating the channel coefficients, yielding in most of the cases a worse estimation than our proposed model due to the poor mixing properties of the FFBS algorithm, as discussed above.

## 6.5 Discussion

WCNs are becoming heterogeneous and not only at the mobile terminals, but also at the base station end. This heterogeneity will lead to complex networks in which we cannot assume how many users will be active or which channels they will face or how long the communications will last. In this chapter, we have applied our BNP models to blindly learn how many users are active and the channel they

face in short periods of time and without requiring extremely long bursts of data. Even in these adversarial conditions, which are typical in M2M communications, our proposed algorithms are able to recover the network structure and provides meaningful estimation of the transmitted sequences. We compare our results with schemes that have complete knowledge of the network structure and parameters and, even though we do not outperform these genie-aided methods, it is remarkable that we are able to mimic their behavior.

When applied to this problem, the IFUHMM can infer the channel length $L$, but it suffers from several limitations. For instance, each $Q \times Q$ transition matrix is not assumed to be a sparse matrix and, therefore, all transitions among states are allowed in the model. Furthermore, $Q$ is allowed to take any integer value above 1, which does not adequately model the MIMO channel, in which the number of states can only take values equal to the constellation order raised to some (positive) power. Another limitation that appears in the experimental section is the mismatch in modeling the effects of the inactive state: the active-to-inactive and inactive-to-active transitions cause an error floor in the ADER and the SER.

In contrast, the IFFSM model assumes that the channel length $L$ is a fixed and known parameter, but it successfully circumvents the aforementioned limitations of the IFUHMM. Furthermore, our inference algorithm based on PGAS avoids the exponential runtime complexity with respect to the parameter $L$.

In spite of the many open issues that arise before we can consider a BNP solution to this problem as viable, the obtained results are promising, opening a new research challenge in applying BNP tools to communication problems. Although it is outside the scope of this Chapter, the prior information about how WCNs are built and their typical overhead (each transmission starts in a predefined way) could be incorporated in the model to solve the identifiability of each user and further reduce the error rate.

# 7 Conclusions

## 7.1 Summary

In this chapter, we summarize the contributions of this Thesis, and also describe some possible lines for future research.

The contributions of this Thesis are twofold. On the one hand, regarding technical aspects, we have developed new Bayesian nonparametric (BNP) priors for time series modeling, as well as the necessary inference algorithms to approximate the posterior distribution. On the other hand, we have provided a new approach for the power disaggregation problem and the blind multiuser channel estimation problem.

Regarding the technical contributions, we have extended the existent binary infinite factorial hidden Markov model (IFHMM) [130] to allow for any number of states in the Markov chains and developed two inference algorithms based on

Markov chain Monte Carlo (MCMC) 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 cardinality of the hidden states in a factorial hidden Markov model (FHMM). This algorithm resembles the reversible jump Markov chain Monte Carlo (RJMCMC) techniques for hidden Markov models (HMMs) but, since all the dimension-changing variables can be integrated out, we opt instead for a standard Metropolis-Hastings algorithm. 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.

We have also developed the infinite factorial finite state machine (IFFSM), a factorial model with a potentially infinite number of parallel Markov chains in which each one evolves independently according to a finite-memory stochastic finite state machine (FSM) model. Each FSM can be understood as a single HMM in which the states are represented by the last $L$ input symbols, and hence only a few of the transitions among states are allowed. We have also shown that the IFFSM model can be easily extended to a factorial model in which the hidden variables can be either discrete or continuous. In addition, we have proposed an inference algorithm based on particle Gibbs with ancestor sampling (PGAS) that can be applied to this general factorial model, and that has better mixing properties than the more standard forward-filtering backward-sampling (FFBS) approach when applied to the IFFSM model. More importantly, our PGAS-based inference algorithm does not suffer from exponential runtime complexity with respect to the parameter $L$.

Regarding the application side, we have applied our non-binary IFHMM and the infinite factorial unbounded-state hidden Markov model (IFUHMM) to the power disaggregation problem. By making use of two real-world datasets (the AMP and the REDD datasets), we have shown that the number of devices in a

house, as well as their parameters, can be inferred in a fully blind manner, with no prior information about the behavior of specific devices. We have also obtained that inferring the number of chains and states in the FHMM, instead of fixing them *a priori*, improves performance.

We have applied both the IFUHMM and the IFFSM model to the blind multiuser channel estimation and symbol detection problem, showing that we can properly recover the true number of transmitters and the underlying symbol sequences in a fully blind way, with no need of training data. The IFUHMM can infer both the number of transmitter and the length of the channel impulse response, but it suffers from several limitations, e.g., considering that all transitions among the states are allowed. The IFFSM specifically takes into account the properties of the communication channel, therefore bypassing all these limitations, although it requires the specification of the channel length, $L$.

## 7.2 Future Work

Our work also suggests several paths for further research, both in the technical and application sides. We provide below a list with some of the potential future research lines.

**Other applications.** Our BNP time series models are general enough to be applied to other problems besides power disaggregation and multiuser communications. An interesting research line consists in evaluating the performance and limitations of the IFUHMM or the IFFSM on other blind signal separation problems, e.g., for financial time series, where there is an unknown number of traders in the market and we only observe the quotes placed by other participants. Our models might require specific tuning or improvements for other considered applications.

**Hierarchy of electrical devices.** For the power disaggregation problem, a potential improvement of our IFUHMM consists of considering a hierarchy, in which the chains are shared across the houses, but their activation is individually com-

puted for each house. In this way, we only need to infer some representative devices that are shared among several houses.

**Doubly nonparametric IFUHMM.** Another extension of the IFUHMM consists in developing a doubly nonparametric model, in which both the number of states and the number of parallel chains are infinite. This model would combine the benefits of the infinite hidden Markov model (IHMM) and the IFHMM, allowing an inference algorithm without split and merge moves.

**Semi-Markov extension of the IFFSM.** Regarding the blind multiuser channel estimation and symbol detection problem, an extension of the IFFSM based on semi-Markov models is also of potential interest. In this way, transmitters are assumed to send only a burst of symbols during the observation period, and a changepoint detection algorithm may detect the activation/deactivation instants.

**Header and coding schemes.** In a practical implementation of a communication system, further considerations can be taken. For instance, each transmitter can add a fixed header at the beginning (or end) of its message. In this way, the receiver would know in advance a subset of the transmitted symbols and would need to focus only on estimating the payload (as well as the channel coefficients). Furthermore, a channel coding scheme can be used in order to add redundancy to the user's data, effectively decreasing the resulting bit error probability.

**Online inference algorithm.** Furthermore, in practice, the receiver does not have access to a fixed window of observations, but data arrives instead as a never-ending stream. Hence, we would have to adapt our inference algorithms in order to efficiently recover the symbols. A sensible approach may rely on a sliding window that runs over time. For each particular position of the sliding window, we can run a few iterations of a sampler restricted to the observations in this window, initialized with the hidden structured inferred for the previous time window.

**Time-varying channels.** Our current approach for the blind multiuser channel estimation problem is restricted to static channels, i.e., the channel coefficients do not vary over time. A potentially useful research line may consist in taking into

account the temporal evolution of the channel coefficients.

**Scalable inference algorithms.** One of the limitations of our inference algorithms, as for most BNP models, is scalability. Hence, developing scalable inference algorithms for our models would also be a significant contribution. For this purpose, approaches may rely on stochastic variational algorithms [63], which have been recently adapted to HMMs [43]. The adaptation of the algorithm in [43] to our models is not straightforward, as they require forward-backward sweeps, which present exponential complexity for FHMMs and FSMs.

**Mixing of the FFBS.** Along the same lines, improving mixing of our inference algorithms based on FFBS also constitutes an interesting research line. In this case, the recently developed Hamming ball auxiliary sampling scheme [122] may be useful. Again, the adaptation of this method to our models is not straightforward, due to their nonparametric nature. Furthermore, the adaptation of the method in [122] to (factorial) FSMs is challenging.

**EP-based inference.** Another research line concerning inference consists in adapting the expectation propagation (EP) algorithm for multiple-input multiple-output (MIMO) communication channels in [26] to be used as part of our inference method. The adaptation of this algorithm should consider both the memory of the channel and the high self-transition probabilities of the inactive and active states.

# A

# Inference Details for the Non-Binary Infinite Factorial HMM

## A.1 Assignment Probabilities for the Gibbs Sampler

We now derive the probability $p(s_{tm} = k|\mathbf{S}_{\neg tm})$, needed in Section 3.4.1 of the main text. This expression can be expressed, up to a proportionality constant, as shown in Eq. A.1. Let $n_{qi}^{\neg tm}$ be the number of transitions from state $q$ to state $i$ in chain $m$, excluding the transitions from state $s_{(t-1)m}$ to $s_{tm}$ and from state $s_{tm}$ to $s_{(t+1)m}$. Similarly, let $n_{q\bullet}^{\neg tm}$ be the total number of transitions from state $q$ in chain $m$ without taking into account state $s_{tm}$, namely, $n_{q\bullet}^{\neg tm} = \sum_{i=0}^{Q-1} n_{qi}^{\neg tm}$. The expression in Eq. A.1 takes different forms depending on the values of $j = s_{(t-1)m}$ and $\ell = s_{(t+1)m}$, yielding Eq. A.2 for $j = 0$ and Eq. A.3 for $j \neq 0$.

$$p(s_{tm}=k|\mathbf{S}_{\neg tm}) \propto \begin{cases} \int_{\mathbf{a}_j^m} p(s_{tm}=k|\mathbf{a}_j^m)p(\mathbf{a}_j^m|\{s_{\tau m}|s_{(\tau-1)m}=j,\tau\neq t,t+1\})d\mathbf{a}_j^m \times \\ \quad \times \int_{\mathbf{a}_k^m} p(s_{(t+1)m}=\ell|\mathbf{a}_k^m)p(\mathbf{a}_k^m|\{s_{\tau m}|s_{(\tau-1)m}=k,\tau\neq t,t+1\})d\mathbf{a}_k^m, & \text{if } k\neq j \\ \\ \int_{\mathbf{a}_k^m} p(s_{(t+1)m}=\ell,s_{tm}=k|\mathbf{a}_k^m)p(\mathbf{a}_k^m|\{s_{\tau m}|s_{(\tau-1)m}=k,\tau\neq t,t+1\})d\mathbf{a}_k^m, & \text{if } k=j. \end{cases} \tag{A.1}$$

a) If $j=0$:

$$p(s_{tm}=k|\mathbf{S}_{\neg tm}) \propto \begin{cases} \frac{n_{00}^{\neg tm}+1}{(n_{0\bullet}^{\neg tm}+1)(n_{0\bullet}^{\neg tm}+2)}\left(\delta_{\ell 0}(n_{00}^{\neg tm}+2)+(1-\delta_{\ell 0})\frac{(\gamma+n_{0\ell}^{\neg tm})\sum_{i=1}^{Q-1}n_{0i}^{\neg tm}}{\sum_{i=1}^{Q-1}(\gamma+n_{0i}^{\neg tm})}\right), & \text{if } k=0 \\ \frac{\left(\delta_{\ell 0}\beta_0+(1-\delta_{\ell 0})\beta+n_{k\ell}^{\neg tm}\right)(\gamma+n_{0k}^{\neg tm})\left(\sum_{i=1}^{Q-1}n_{0i}^{\neg tm}\right)}{(1+n_{0\bullet}^{\neg tm})\left(\beta_0+(Q-1)\beta+n_{k\bullet}^{\neg tm}\right)\left(\sum_{i=1}^{Q-1}(\gamma+n_{0i}^{\neg tm})\right)}, & \text{if } k=1,\ldots,Q-1. \end{cases} \tag{A.2}$$

b) If $j\neq 0$:

$$p(s_{tm}=k|\mathbf{S}_{\neg tm}) \propto \begin{cases} \frac{(\beta_0+n_{j0}^{\neg tm})}{(n_{0\bullet}^{\neg tm}+1)\left(\beta_0+(Q-1)\beta+n_{j\bullet}^{\neg tm}\right)} \\ \quad \times\left(\delta_{\ell 0}(n_{00}^{\neg tm}+1)+(1-\delta_{\ell 0})\frac{(\gamma+n_{0\ell}^{\neg tm})\sum_{i=1}^{Q-1}n_{0i}^{\neg tm}}{\sum_{i=1}^{Q-1}(\gamma+n_{0i}^{\neg tm})}\right), & \text{if } k=0 \\ \frac{\left(\delta_{\ell 0}\beta_0+(1-\delta_{\ell 0})\beta+n_{k\ell}^{\neg tm}+\delta_{k\ell}\delta_{kj}\right)(\beta+n_{jk}^{\neg tm})}{\left(\beta_0+(Q-1)\beta+n_{k\bullet}^{\neg tm}+\delta_{kj}\right)\left(\beta_0+(Q-1)\beta+n_{j\bullet}^{\neg tm}\right)}, & \text{if } k=1,\ldots,Q-1. \end{cases} \tag{A.3}$$

## A.2 Update Equations for the Variational Algorithm

Here, we provide the update equations for the variational inference algorithm in Section 3.4.3. The variational inference algorithm involves optimizing the variational parameters of $q(\Psi)$ to minimize the Kullback-Leibler divergence of $p_M(\Psi|\mathbf{Y},\mathcal{H})$ from $q(\Psi)$, i.e., $D_{KL}(q||p_M)$. This optimization can be performed by iteratively applying the following fixed-point set of equations:

$$P_{jk}^m = \begin{cases} \exp\left\{\psi(\tau_{jk}^m)-\psi\left(\sum_{i=0}^{Q-1}\tau_{ji}^m\right)\right\}, & \text{if } j\neq 0, \\ \exp\left\{\psi(\nu_1^m)-\psi(\nu_1^m+\nu_2^m)\right\}, & \text{if } j=0,k=0, \\ \exp\left\{\psi(\nu_2^m)-\psi(\nu_1^m+\nu_2^m)+\psi(\varepsilon_k^m)\right. \\ \qquad \left.-\psi\left(\sum_{i=1}^{Q-1}\varepsilon_i^m\right)\right\}, & \text{if } j=0,k\neq 0, \end{cases} \tag{A.4}$$

$$b_{kt}^{m} = \exp\Bigg\{ -\frac{1}{2\sigma_y^2}(\mathbf{L}_k)_m(\mathbf{L}_k)_m^\top + \frac{1}{\sigma_y^2}(\mathbf{L}_k)_m\Bigg(\mathbf{y}_t - \sum_{\ell\neq m}\sum_{i=1}^{Q-1}(\mathbf{L}_i)_\ell q(s_{t\ell}=i)\Bigg)^\top\Bigg\}, \tag{A.5}$$

$$\tau_{jk}^{m} = \delta_{k0}\beta_0 + (1-\delta_{k0})\beta + \sum_{t=1}^{T} q(s_{(t-1)m}=j, s_{tm}=k), \tag{A.6}$$

$$\nu_1^m = 1 + \sum_{t=1}^{T} q(s_{(t-1)m}=0, s_{tm}=0), \tag{A.7}$$

$$\nu_2^m = \frac{\alpha}{M} + \sum_{t=1}^{T} q(s_{(t-1)m}=0, s_{tm}>0), \tag{A.8}$$

$$\varepsilon_k^m = \gamma + \sum_{t=1}^{T} q(s_{(t-1)m}=0, s_{tm}=k), \tag{A.9}$$

$$\mathbf{\Omega}_k = \left(\frac{1}{\sigma_0^2} + \frac{M}{\sigma_\phi^2}\right)^{-1}\mathbf{I}_D, \tag{A.10}$$

$$\boldsymbol{\omega}_k = \mathbf{\Omega}_k\left(\frac{1}{\sigma_\phi^2}\mathbf{L}_k^\top\mathbf{1}_M + \frac{1}{\sigma_0^2}\boldsymbol{\mu}_0\right), \tag{A.11}$$

$$\mathbf{\Lambda}_k = \left(\frac{1}{\sigma_\phi^2}\mathbf{I}_M + \frac{1}{\sigma_y^2}\mathbf{C}_k\right)^{-1}, \tag{A.12}$$

and

$$\mathbf{L}_k = \mathbf{\Lambda}_k\left(\frac{1}{\sigma_\phi^2}\boldsymbol{\omega}_k\mathbf{1}_M^\top + \frac{1}{\sigma_y^2}\mathbf{Q}_k^\top\left(\mathbf{Y} - \sum_{j\neq k}\mathbf{Q}_j\mathbf{L}_j\right)\right), \tag{A.13}$$

where $(\mathbf{L}_k)_m$ denotes the $m$-th row of matrix $\mathbf{L}_k$, $\psi(\cdot)$ stands for the digamma function [5, p. 258–259], $\delta_{ii'}$ denotes the Kronecker delta function (which takes value one if $i = i'$ and zero otherwise), and the elements of the $T \times M$ matrices $\mathbf{Q}_k$ and $M \times M$ matrices $\mathbf{C}_k$ are, respectively, given by

$$(\mathbf{Q}_k)_{tm} = q(s_{tm}=k) \tag{A.14}$$

and

$$(\mathbf{C}_k)_{mm'} = \begin{cases} \sum_{t=1}^{T} q(s_{tm} = k) q(s_{tm'} = k), & \text{if } m \neq m' \\ \sum_{t=1}^{T} q(s_{tm} = k), & \text{if } m = m'. \end{cases} \tag{A.15}$$

The probabilities $q(s_{tm})$ and $q(s_{tm}, s_{(t-1)m})$ can be obtained through a standard forward-backward algorithm for hidden Markov models (HMMs) within each chain, in which the variational parameters $P^m_{jk}$ and $b^m_{kt}$ play respectively the role of the transition probabilities and the observation probability associated with state variable $s_{tm}$ taking value $k$ in the Markov chain $m$ [48].

# References


[1] M2M white paper: the growth of device connectivity. Technical report, The FocalPoint Group, 2003.

[2] Global machine to machine communication. Technical report, Vodafone, 2010.

[3] Device connectivity unlocks value. Technical report, Ericsson, January 2011.

[4] Small cell market status. Technical report, Informa Telecoms and Media. Issue 4, December 2012.

[5] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover Publications, New York, 1972.

[6] F. Abrard, Y. Deville, and P. White. From blind source separation to blind source cancellation in the underdetermined case: A new approach based on time-frequency analysis. In *Proceedings of the 3rd International Conference on Independent Component Analysis and Signal Separation*, pages 734–739, December 2001.

[7] D. Aldous. Exchangeability and related topics. In *École d'été de probabilités de Saint-Flour, XIII—1983*, pages 1–198. Springer, 1985.

[8] J. A. Anderson and T. J. Head. *Automata Theory with Modern Applications*. Cambridge University Press, 2006.

[9] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society Series B*, 72(3):269–342, 2010.

[10] A. Angelosante, E. Biglieri, and M. Lops. Multiuser detection in a dynamic environment - Part II: Joint user identification and parameter estimation. *IEEE Transactions on Information Theory*, 55:2365–2374, May 2009.

[11] D. Angelosante, E. Biglieri, and M. Lops. Low-complexity receivers for multiuser detection with an unknown number of active users. *Signal Processing*, 90:1486–1495, May 2010.

[12] D. Angelosante, E. Grossi, G. B. Giannakis, and M. Lops. Sparsity-aware estimation of CDMA system parameters. In *IEEE 10th Workshop on Signal Processing Advances in Wireless Communications '09*, pages 697–701, June 2009.

[13] C. E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.

[14] M. Arroyo, J. Via, and I. Santamaria. Deterministic MIMO channel order estimation based on canonical correlation analysis. In *5th IEEE Sensor Array and Multichannel Signal Processing Workshop*, pages 18–22, july 2008.

[15] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transactions on Information Theory*, 20(2):284–287, March 1974.

[16] P. Baldi, Y. Chauvin, T. Hunkapiller, and M. A. McClure. Hidden markov models of biological primary sequence information. *Proceedings of the National Academy of Sciences*, 91(3):1059–1063, 1994.

[17] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, 1966.

[18] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden Markov model. In *Advances in Neural Information Processing Systems*, volume 14, 2002.

[19] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[20] D. Blackwell and J. MacQueen. Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, 1:353–355, 1973.

[21] D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–144, 2006.

[22] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003.

[23] P. Bofill and M. Zibulevsky. Underdetermined blind source separation using sparse representations. *Signal Processing*, 81(11):2353–2362, 2001.

[24] V. R. Cadambe and S. A. Jafar. Interference alignment and degrees of freedom of the $K$-user interference channel. *IEEE Transactions on Information Theory*, 54(8):3425–3441, August 2008.

[25] C. K. Carter and R. Kohn. On Gibbs sampling for state space models. *Biometrika*, 81(3):541–553, August 1994.

[26] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Perez-Cruz. Expectation propagation detection for high-order high-dimensional MIMO systems. *IEEE Transactions on Communications*, 62(8):2840–2849, August 2014.

[27] Y. Chen and W. Wang. Machine-to-machine communication in LTE-A. In *Proceedings of 2010 IEEE Vehicular Technology Conference*, September 2010.

[28] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, April 1994.

[29] P. Comon and C. Jutten, editors. *Handbook of blind source separation: Independent component analysis and applications*. Communications engineering. Elsevier, Amsterdam, Boston (Mass.), 2010.

[30] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995.

[31] S. Darby. The effectiveness of feedback on energy consumption: A review for DEFRA of the literature on metering, billing and direct displays. Technical report, Environmental Change Institute, University of Oxford, 2006.

[32] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[33] H. S. Dhillon, H. Huang, H. Viswanathan, and R. A. Valenzuela. Fundamentals of throughput maximization with random arrivals for M2M communications. *IEEE Transactions on Communications*, 62(11):4094–4109, November 2014.

[34] H. S. Dhillon, H. C. Huang, H. Viswanathan, and R. A. Valenzuela. Power-efficient system design for cellular-based machine-to-machine communications. *CoRR*, abs/1301.0859, 2013.

[35] N. Ding and Z. Ou. Variational nonparametric Bayesian hidden Markov model. In *ICASSP'10*, pages 2098–2101, 2010.

[36] F. Doshi-Velez, K. T. Miller, J. Van Gael, and Y. W. Teh. Variational inference for the Indian buffet process. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 12, 2009.

[37] D. B. Dunson. Nonparametric Bayes applications to biostatistics. In

*Bayesian Nonparametrics: Principles and Practice*. Cambridge University Press, 2010.

[38] M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the american statistical association*, 90(430):577—588, 1995.

[39] C. Estêvão, R. Fernandes, P. Comon, and G. Favier. Blind identification of MISO-FIR channels. *Signal Processing*, 90(2):490–503, February 2010.

[40] T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.

[41] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden Markov model: Analysis and applications. *Machine Learning*, 32(1):41–62, July 1998.

[42] S. J. Fortune, D. M. Gay, B. W. Kernighan, O. Landron, R. A. Valenzuela, and M. H. Wright. WISE design of indoor wireless systems: Practical computation and optimization. *IEEE Computing in Science & Engineering*, 2(1):58–68, March 1995.

[43] N. Foti, J. Xu, D. Laird, and E. B. Fox. Stochastic variational inference for hidden Markov models. In *Advances in Neural Information Processing Systems 27*, 2014.

[44] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. Bayesian nonparametric methods for learning Markov switching processes. *IEEE Signal Processing Magazine*, 27(6):43–54, 2010.

[45] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. A sticky HDP-HMM with application to speaker diarization. *Annals of Applied Statistics*, 5(2A):1020–1056, 2011.

[46] S. Fruhwirth-Schnatter. Data augmentation and dynamic linear models. *Journal of Time Series Analysis*, 15(2):183–202, March 1994.

[47] S. J. Gershman and D. M. Blei. A tutorial on Bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1–12, February 2012.

[48] Z. Ghahramani and M. I. Jordan. Factorial hidden Markov models. *Machine Learning*, 29(2–3):245–273, 1997.

[49] W. R. Gilks and P. Wild. Adaptive rejection sampling for Gibbs sampling. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 41(2):337–348, 1992.

[50] S. G. Glisic and P. A. Leppanen. *Wireless Communications: TDMA Versus CDMA*. Kluwer Academic Publishers, Norwell, MA, USA, 1997.

[51] P. Gopalan, F. J. R. Ruiz, R. Ranganath, and D. M. Blei. Bayesian nonparametric Poisson factorization for recommendation systems. In *Artificial Intelligence and Statistics*, 2014.

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

[53] T. L. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. In *Advances in Neural Information Processing Systems 18*, 2006.

[54] T. L. Griffiths and Z. Ghahramani. The Indian buffet process: An introduction and review. *Journal of Machine Learning Research*, 12:1185–1224, 2011.

[55] K. W. Halford and M. Brandt-Pearce. New-user identification in a CDMA system. *IEEE Transactions on Communications*, 46(1):144–155, jan 1998.

[56] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970.

[57] K. A. Heller and Z. Ghahramani. A nonparametric Bayesian approach to modeling overlapping clusters. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 2, pages 187–194, 2007.

[58] K. A. Heller, Y. W. Teh, and D. Görür. Infinite hierarchical hidden Markov models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 12, 2009.

[59] J. Hensman, N. D. Lawrence, and M. Rattray. Hierarchical Bayesian modelling of gene expression time series across irregularly sampled replicates and clusters. *BMC Bioinformatics*, 14:252, 2013.

[60] J. Herault and C. Jutten. Space or time adaptive signal processing by neural network models. In *AIP Conference Proceedings on Neural Networks for Computing*, pages 206–211, 1987.

[61] N. L. Hjort. Nonparametric Bayes estimators based on beta processes in models for life history data. *Annals of Statistics*, 18:1259–1294, 1990.

[62] C.-Y. Ho and C.-Y. Huang. Energy-saving massive access control and resource allocation schemes for M2M communications in OFDMA cellular networks. *IEEE Wireless Communications Letters*, 1(3):209–212, June 2012.

[63] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, May 2013.

[64] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.

[65] J. Hoydis, S. T. Brink, and M. Debbah. Massive MIMO in the UL/DL of cellular networks: How many antennas do we need? *IEEE Journal on Selected Areas in Communications*, 31(2):160–171, February 2013.

[66] P. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[67] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[68] A. Hyvärinen and U. Köter. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9:1483–1492, 1997.

[69] H. Ishwaran and M. Zarepour. Exact and approximate sum-representations for the Dirichlet process. *Canadian Journal of Statistics*, 30:269–283, 2002.

[70] S. Jain and R. Neal. A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13:158–182, 2000.

[71] M. J. Johnson and A. S. Willsky. Bayesian nonparametric hidden semi-Markov models. *Journal of Machine Learning Research*, 14:673–701, February 2013.

[72] M. I. Jordan, editor. *Learning in Graphical Models*. MIT Press, Cambridge, MA, USA, 1999.

[73] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, November 1999.

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

[75] M. Keralapura, M. Pourfathi, and B. Sirkeci-Mergen. Impact of contrast functions in Fast-ICA on twin ECG separation. *IAENG International Journal of Computer Science*, 38(1), 2011.

[76] H. Kim, M. Marwah, M. F. Arlitt, G. Lyon, and J. Han. Unsupervised disaggregation of low frequency power measurements. In *SDM*, pages 747–758, 2011.

[77] A. A. Kirilenko, A. S. Kyle, M. Samadi, and T. Tuzun. The flash crash: The impact of high frequency trading on an electronic market. *Social Science Research Network*, 2011.

[78] D. Knowles and Z. Ghahramani. Nonparametric Bayesian sparse factor models with application to gene expression modeling. *The Annals of Applied Statistics*, 5(2B):1534–1552, June 2011.

[79] J. Z. Kolter and M. J. Johnson. REDD: A public data set for energy disaggregation research. In *SustKDD Workshop on Data Mining Applications in Sustainability*, 2011.

[80] J. Kupiec. Robust part-of-speech tagging using a hidden Markov model. *Computer Speech & Language*, 6(3):225–242, 1992.

[81] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999.

[82] S.-Y. Lien, K.-C. Chen, and Y. Lin. Toward ubiquitous massive accesses in 3GPP machine-to-machine communications. *IEEE Communications Magazine*, 49(4):66–74, 2011.

[83] F. Lindsten, M. I. Jordan, and T. B. Schön. Particle Gibbs with ancestor sampling. *Journal of Machine Learning Research*, 15(1):2145–2184, 2014.

[84] F. Lindsten and T. B. Schön. Backward simulation methods for Monte Carlo statistical inference. *Foundations and Trends in Machine Learning*, 6(1):1–143, 2013.

[85] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang. An overview of massive MIMO: Benefits and challenges. *IEEE Journal of Selected Topics in Signal Processing*, 8(5):742–758, October 2014.

[86] S. N. Maceachern and P. Müller. Estimating mixture of Dirichlet process models. *Journal of Computational and Graphical Statistics*, 7(2):223–238, June 1998.

[87] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I. V. Bajic. AMPds: A public dataset for load disaggregation and eco-feedback research. In *Pro-*

*ceedings of the 2013 IEEE Electrical Power and Energy Conference (EPEC)*, 2013.

[88] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, June 1953.

[89] J. Míguez and L. Castedo. Semiblind maximum-likelihood demodulation for CDMA systems. *IEEE Transactions on Vehicular Technology*, 51(4):775–781, jul 2002.

[90] J. W. Miller and M. T. Harrison. A simple example of Dirichlet process mixture inconsistency for the number of components. In *Advances in Neural Information Processing Systems*, pages 199–206, 2013.

[91] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[92] R. Nag, K. Wong, and F. Fallside. Script recognition using hidden Markov models. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86.*, volume 11, pages 2071–2074, apr 1986.

[93] R. Neal. Bayesian mixture modeling by Monte Carlo simulation. Technical report, 1991.

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

[95] R. Neal. Slice sampling. *Annals of Statistics*, 31:705–767, 2000.

[96] B. Neenan and J. Robinson. Residential electricity use feedback: A research synthesis and economic framework. Technical report, Electric Power Research Institute, 2009.

[97] P. Orbanz and Y. W. Teh. Bayesian nonparametric models. In *Encyclopedia of Machine Learning*. Springer, 2010.

[98] P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, June 1994.

[99] K. Palla, D. A. Knowles, and Z. Ghahramani. A reversible infinite HMM using normalised random measures. In *International Conference on Machine Learning*, June 2014.

[100] F. Perez-Cruz, S. Van Vaerenbergh, J. J. Murillo-Fuentes, M. Lázaro-Gredilla, and I. Santamaría. Gaussian processes for nonlinear signal processing. *IEEE Signal Processing Magazine*, 30(4):40–50, July 2013.

[101] J. Pitman. *Combinatorial stochastic processes*, volume 1875 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 2006. Lectures from the 32nd Summer School on Probability Theory held in Saint-Flour, July 7–24, 2002, With a foreword by Jean Picard.

[102] J. Pitman and M. Yor. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2):855–900, April 1997.

[103] L. Rabiner and B. Juang. An introduction to hidden Markov models. *ASSP Magazine, IEEE*, 3(1):4–16, January 1986.

[104] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[105] R. Raheli, A. Polydoros, and Ching-Kae Tzou. Per-survivor processing: a general approach to MLSE in uncertain environments. *IEEE Transactions on Communications*, 43(234):354–364, feb/mar/apr 1995.

[106] C. E. Rasmussen. The infinite Gaussian mixture model. In *Advances in Neural Information Processing Systems 12*, pages 554–560, 2000.

[107] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[108] S. Richardson and P. J. Green. On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society, Series B (Methodological)*, 59(4):731–792, 1997.

[109] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[110] C. P. Robert, T. Rydén, and D. M. Titterington. Bayesian inference in hidden Markov models through reversible jump Markov chain Monte Carlo. *Journal of the Royal Statistical Society, Series B*, 62:57–75, 2000.

[111] F. J. R. Ruiz, I. Valera, C. Blanco, and F. Perez-Cruz. Bayesian nonparametric modeling of suicide attempts. *Advances in Neural Information Processing Systems*, 25:1862–1870, 2012.

[112] F. J. R. Ruiz, I. Valera, C. Blanco, and F. Perez-Cruz. Bayesian nonparametric comorbidity analysis of psychiatric disorders. *Journal of Machine Learning Research*, 15:1215–1247, April 2014.

[113] F. J. R. Ruiz, I. Valera, L. Svensson, and F. Perez-Cruz. Infinite factorial finite state machine for blind multiuser channel estimation. *In preparation for IEEE Transactions on Cognitive Communications and Networking*, 2015.

[114] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

[115] S. L. Scott. Bayesian methods for hidden Markov models: recursive computing in the 21st century. *Journal of the American Statistical Association*, 97(457):337–351, 2002.

[116] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[117] Y. W. Teh, D. Görür, and Z. Ghahramani. Stick-breaking construction for the Indian buffet process. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 11, 2007.

[118] Y. W. Teh and M. I. Jordan. Hierarchical Bayesian nonparametric models with applications. In N. Hjort, C. Holmes, P. Müller, and S. Walker, editors, *Bayesian Nonparametrics: Principles and Practice*. Cambridge University Press, 2010.

[119] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[120] R. Thibaux and M. I. Jordan. Hierarchical beta processes and the Indian buffet process. In *11th Conference on Artificial Intelligence and Statistics*, 2007.

[121] M. Titsias. The infinite gamma-Poisson feature model. In *Advances in Neural Information Processing Systems*, volume 19, 2007.

[122] M. K. Titsias and C. Yau. Hamming ball auxiliary sampling for factorial hidden Markov models. In *Advances in Neural Information Processing Systems 27*, 2014.

[123] C.-Y. Tu, C.-Y. Ho, and C.-Y. Huang. Energy-efficient algorithms and evaluations for massive access management in cellular based machine to machine communications. In *Proceedings of 2011 IEEE Vehicular Technology Conference*, pages 1–5, September 2011.

[124] W. Turin. Hidden Markov modeling of flat fading channels. *IEEE Journal on Selected Areas in Communications*, 16(9):1809–1817, December 1998.

[125] I. Valera, F. J. R. Ruiz, and F. Perez-Cruz. Infinite factorial unbounded hidden Markov model for blind multiuser channel estimation. In *4th International Workshop on Cognitive Information Processing*, May 2014.

[126] I. Valera, F. J. R. Ruiz, and F. Perez-Cruz. Infinite factorial unbounded-state hidden Markov model. *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[127] I. Valera, F. J. R. Ruiz, L. Svensson, and F. Perez-Cruz. A Bayesian non-parametric approach for blind multiuser channel estimation. In *European Signal Processing Conference*, 2015.

[128] I. Valera, F. J. R. Ruiz, L. Svensson, and F. Perez-Cruz. Infinite factorial dynamical model. In *Advances in Neural Information Processing Systems (Submitted)*, 2015.

[129] J. Van Gael, Y. Saatci, Y. W. Teh, and Z. Ghahramani. Beam sampling for the infinite hidden Markov model. In *Proceedings of the International Conference on Machine Learning*, volume 25, 2008.

[130] J. Van Gael, Y. W. Teh, and Z. Ghahramani. The infinite factorial hidden Markov model. In *Advances in Neural Information Processing Systems*, volume 21, 2009.

[131] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, USA, 1995.

[132] M. A. Vázquez and J. Míguez. Maximum-likelihood sequence detection in time- and frequency-selective MIMO channels with unknown order. *IEEE Transactions on Vehicular Technology*, 58(1):499–504, jan 2009.

[133] M. A. Vázquez and J. Míguez. A per-survivor processing receiver for MIMO transmission systems with one unknown channel order per output. *IEEE Transactions on Vehicular Technology*, 60(9):4415–4426, nov 2011.

[134] M. A. Vázquez and J. Míguez. User activity tracking in DS-CDMA systems. *IEEE Transactions on Vehicular Technology*, 62(7):3188–3203, 2013.

[135] J. Via, I. Santamaria, and J. Perez. Effective channel order estimation based

on combined identification/equalization. *IEEE Transactions on Signal Processing*, 54(9):3518–3526, sept 2006.

[136] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, January 2008.

[137] J. Wang. *Handbook of Finite State Based Models and Applications*. Chapman & Hall/CRC, 1st edition, 2012.

[138] L. Wasserman. *All of Nonparametric Statistics (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[139] N. Whiteley, C. Andrieu, and A. Doucet. Efficient Bayesian inference for switching state-space models using particle Markov chain Monte Carlo methods. Technical report, Bristol Statistics Research Report 10:04, 2010.

[140] S. Williamson, C. Wang, K. A. Heller, and D. M. Blei. The IBP compound Dirichlet process and its application to focused topic modeling. *Proceedings of the 27th Annual International Conference on Machine Learning*, 2010.

[141] W.-C. Wu and K.-C. Chen. Identification of active users in synchronous CDMA multiuser detection. *IEEE Journal on Selected Areas in Communications*, 16(9):1723–1735, dec 1998.

[142] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, December 2003.

[143] H. Zhu and G. B. Giannakis. Exploiting sparse user activity in multiuser detection. *IEEE Transactions on Communications*, 59(2):454–465, February 2011.