Author:
Rimella, Lorenzo

Title:
High-dimensional hidden Markov models

methodology, computational issues, solutions and applications

High-dimensional Hidden Markov Models

Methodology, computational issues, solutions and applications

LORENZO RIMELLA

School of Mathematics
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the School of Mathematics.

OCTOBER 6, 2021

Word count: 67889

Abstract

The thesis presents the main definitions and concepts of hidden Markov models (HMMs), focusing on how they are built and the issues arising when scaling up to high-dimensional settings. Novel approximate algorithms to compute filtering and smoothing distributions are proposed, these improve the applicability of HMMs to large scale and overcome computational problems.

Firstly, an approximate forward-filtering and backward-smoothing algorithm for Factorial HMMs [Ghahramani and Jordan, 1997] is developed. The approximation involves discarding likelihood factors according to a notion of locality in the factor graph associated with the emission distribution. This allows the exponential-in-dimension cost of exact filtering and smoothing to be avoided. The approximation accuracy, measured in a local total variation norm, is proved to be "dimension-free" in the sense that as the overall dimension of the model increases the error bounds do not necessarily degrade. This method can be applied to data with known spatial or network structure, for instance in traffic models.

Secondly, an evolving in time Bayesian neural network called a Hidden Markov neural network is proposed. The weights of a feed-forward neural network are modelled with the hidden process of an HMM, whose observed process is given by the available data. An approximate filtering algorithm is used to learn a variational approximation to the evolving in time filtering distribution over the weights. Training is pursued through a sequential version of Bayes by Backprop [Blundell et al., 2015], which is enriched with a regularization technique called variational DropConnect.

Finally, the thesis produces also a new method for inference in stochastic epidemic models, which uses recursive multinomial approximations to filtering /smoothing distributions over unobserved variables and thus circumvent likelihood intractability. The method applies to a class of discrete-time finite-population compartmental models with partial, randomly under-reported or missing count observations. The algorithm is tested on real and synthetic data.

DEDICATION AND ACKNOWLEDGEMENTS

I sincerely thank my supervisor Prof. Nick Whiteley for guiding me through this journey by patiently teaching me the meaning of doing research and helping me to improve every day. I want also to thank him from a personal perspective, for being very comprehensive and making me love what I was doing.

I would like to thank: Prof. Chistophe Andrieu and Dr. Nikolas Kantas for agreeing on reviewing this thesis; Prof. Chistophe Andrieu, Dr. Matteo Fasiolo, Prof. Simon Wood, for reviewing my annual reports and for the useful feedback they gave me over the years; Prof. Francesco Mezzadri, for his support during the COVID-19 pandemic; all the people I met at the University of Bristol and at the Alan Turing Institute, for keeping me motivated and significantly contributed to my academic growth.

I want to personally thank: my housemates Ankur, Aurora, Jérôme, Jonny, Kate, for making me feel at home since the first day; the “coffee break team” and in particular Broncio, Christian, Fabio, João, Matteo, Rafa, Vlad, for all the time spent together; my friends Alex, Andrea, Flavio, Luca, Matteo, Simone, Steppo, Vincent, for keeping in touch no matter what and for all the amazing moments we have been through.

I thank my family for supporting me over the years and giving me the strength to overtake difficulties.

I cannot find the words to thank my girlfriend Raziel for being close in any decision I make, for making me happy and especially for all the love she is giving me.

Finally, I do not know how to thank my dear Stefano for all the amazing years of friendship, I hope that, wherever he is, he is still watching over me.

Author's declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: .... DATE: ....

Table of Contents

Page

Chapter 1

Introduction

Since early appearance in the literature [Baum and Petrie, 1966, Baum et al., 1970] and popularization in speech recognition [Rabiner, 1989], hidden Markov models (HMMs) have been used to solve a broad range of problems, from texture recognition [Bose and Kuo, 1994], to gene prediction [Stanke and Waack, 2003], and weather forecasting [Hughes et al., 1999]. An HMM is a statistical model which explains patterns in observed data through a hidden process. The unobserved process is a Markov chain, meaning that time step t depends on time step $t-1$ only. The main challenge in HMMs is to compute the filtering and the smoothing distributions, which are the probability of observing a state in the Markov chain at time t given the data up to time t and the probability of observing a state in the Markov chain at time t given all the data (marginal smoothing). These distributions can be then used to infer the parameters of the HMM (e.g. EM algorithm). The main purpose of this thesis has been to overcome the prohibitive computational cost of high-dimensional HMMs, by proposing approximate filtering and smoothing algorithms.

This chapter is organised as an introduction to the main contributions of this thesis, which are then developed in chapter 3, chapter 5 and chapter 6, with chapter 2 presenting the background and chapter 4 collecting the proofs of the results in chapter 3. Chapter 3, chapter 5 and chapter 6 are submitted/published and summarized in the following papers:

ch. 3-4: L. Rimella and N. Whiteley. Exploiting locality in high-dimensional factorial hidden Markov models. *arXiv preprint arXiv:1902.01639*, 2019;

ch. 5: L. Rimella and N. Whiteley. Dynamic Bayesian Neural Networks. *arXiv preprint arXiv:2004.06963*, 2020;

ch. 6: N. Whiteley and L. Rimella. Inference in stochastic epidemic models via multinomial approximations. In *International Conference on Artificial Intelligence and Statistics*, pages 1297–1305. PMLR, 2021.

The implementation of the corresponding algorithms is open-source and available on the Github page:

`https://github.com/LorenzoRimella`.

1.1 Exploiting locality in high-dimensional factorial hidden Markov models

As already mentioned, an HMM is a statistical model which explains patterns in observed data in terms of the evolution over time of a process that is not observed (latent process). Here, inference is pursued by the computation of conditional distributions over the latent process, which are called filtering and smoothing distributions, and generally obtained with a forward-filtering and a backward-smoothing step through the data. Under a finite state-space setting, i.e. the unobserved process takes values on a finite set, the cost of filtering-smoothing is cubic in the cardinality of the state-space. This raises computational issues when scaling up to high-dimensional state-spaces, for instance, if the state-space is the product form $\mathbb{X}^M$, with $\mathbb{X}$ finite set and $M \in \mathbb{N}$, the computational cost of filtering-smoothing is exponential in M.

Variational inference Blei et al. [2017] consistently reduces the computational cost by introducing an approximation on the posterior distribution over the states of the hidden process. Even though there are several strands of research into theoretical properties of variational methods for some classes of statistical models, as surveyed recently by Blei et al. [2017], such as convergence analysis for mixture models Wang and Titterington [2006], and consistency studies for stochastic block models Celisse et al. [2012], Bickel et al. [2013], little seems to be known about the quality of the approximation.

Boyen and Koller [1998, 1999] proposed and studied inference methods involving an approximation of posterior distributions by the product of their marginals. Particle filtering algorithms in the same vein appeared in Ng et al. [2002], Brandao et al. [2006], Besada-Portas et al. [2009]. The advantages of such approaches for finite state-space are the decomposition of a high-dimensional problem to multiple lower-dimensional ones. For instance, a probability distribution π over $\mathbb{X}^M$ can be decomposed in M probability distributions over $\mathbb{X}$ if independence is assumed across the components, i.e. $\pi = \bigotimes_{v=1}^{M} \pi^v$ with π^v marginal distribution of π over v, then π is not represented as a vector with length $\mathbf{card}(\mathbb{X})^M$, but as M vectors with length $\mathbf{card}(\mathbb{X})$, which is cheaper to store in memory. Moreover, if local dependencies are assumed in the HMM, approximating filtering-smoothing distributions with the product of their marginals can be even proved to be accurate [Rebeschini and Van Handel, 2015, Finke and Singh, 2017].

Chapter 3 analyses high-dimensional HMMs with local structures and proposes an approximate algorithm for filtering-smoothing, which can be proved to be accurate from a local total variation norm perspective, which is essentially a total variation norm over the marginals. Precisely, the algorithm applies to HMMs where the components of the latent process are independent, i.e. the HMM is a factorial hidden Markov model [Ghahramani and Jordan, 1997], and the data are generated from an emission distribution that factorises in such a way that each factor depends only on a subset of the latent variables. Under this setting, the filtering distributions are approximated by the product of their marginals, and each marginal is approximated by a localized version where the factors of the emission distribution are selected in a mathematically well-principled manner and some of them are thrown away. At this point, the resulting filtering approximations can plug into the backward-smoothing and approximate smoothing distributions can be inferred without any additional approximation. The proposed algorithm is called Graph Filter-Smoother.

The first advantage of Graph Filter-Smoother concerns the computational cost. Indeed, the algorithm depends linearly on the size of the state-space and exponentially on the local structures, whose meaning is going to be clarified in section 3.3 of chapter 3. This fact is of particular interest when the local dependencies remain unchanged while the overall size increases because in such a scenario scaling-up to high dimensions results in a cheap linear increase in the computational cost. Another pivotal point is the control over the quality of the approximation, indeed the proposed algorithm can shrink the error over the approximations by tuning its parameters. However, there is a trade-off between the quality of the approximation and computational cost, an higher quality of the approximation is associated with an increase in the computational cost of the algorithm. This is not necessarily a downside because it is found out experimentally that an equilibrium between quality and cost is possible and it leads to satisfactory results. Finally, it is verified empirically that the proposed approximation is suited to traffic flow predictions, and it could be extended to any application with straightforward spatial dependencies, such as other traffic models.

1.2 Dynamic Bayesian Neural Networks

When the data distribution evolves during training, most of the deep learning algorithms suffer the “catastrophic forgetting” [McCloskey and Cohen, 1989, Ratcliff, 1990], which is the phenomenon of not being able to integrate the new knowledge in the model without completely remove what was learned before. Continual learning, which is a branch of machine learning focusing on algorithms that are able to train on non-i.i.d. data, has mainly focused on overcoming catastrophic forgetting [Kirkpatrick et al., 2017, Nguyen et al., 2017, Ritter et al., 2018] without taking into account that sudden changes in the statistics of the data may be an intrinsic property of the generating process itself. There is then a need for techniques that are capable to forget and to evolve according to what the data requires. Also, creating neural networks with computational graphs connecting all the history of the data, such as in RNN/LSTM, usually ends up in over/underflow issues, e.g. exploding/vanishing gradients (i.e. the backpropagation algorithm output a long product of derivatives that can be either small or big), which motivates to model the underlying weights as time-varying and so build an evolving-in-time neural network.

Several attempts to create dynamic neural networks have been made in Bayesian filtering [Puskorius and Feldkamp, 1991, 1994, 2001, Shah et al., 1992, Feldkamp et al., 2003, Ollivier et al., 2018] and continual learning [Nguyen et al., 2017, Kurle et al., 2019], by proposing time-varying posterior distributions over the weights. However, the considered settings are generally limited to simple mean-field variational approximations [Kurle et al., 2019] and none of them has focused on modelling the weights as the latent process of an HMM.

Chapter 5 builds hidden Markov neural networks (HMNN), a novel hybrid between Bayesian neural networks and HMM, where the latent process of the HMM outlines the evolution in time of the weights of a neural network. On the one hand, the HMM prediction step allows to preserve knowledge from the previous steps and at the same time to forget useless information. On the other hand, the HMM correction step includes features from the new data granting both complete modelling during training and online learning if needed. The filtering distribution, which is an evolving-in-time posterior distribution over the weights, is not available in closed form, due to the complexity of the non-linear layers of the neural network. It is then computed approximatively through variational inference by minimizing Kullback-Leibler divergence criteria at each time step and by propagating forward the previous approximation. A sequential version of the well-known algorithm Bayes by Backprop [Blundell et al., 2015] is used to perform training, along with a reformulation of the reparameterization trick for variational approximations that are Gaussian mixtures.

HMNNs can adapt to the data and forget unnecessary information, making them particularly suited to non-stationary time series. At test time, HMNNs show good performances on static classification on MNIST, simple and complex concept drift, next frame forecasting in a video, and compared favourably against multiple baselines. Another appealing feature of the proposed algorithm is to regularize the neural network, i.e. avoid overfitting, through a variational approximation that is a product of Gaussian mixtures. Indeed, in Bayesian neural networks, the regularization (i.e. penalizing the complexity of the neural network) is generally induced by the prior choice and the variational approximation is chosen to be a product of Gaussians (mean-field variational approximation). Choosing a product of mixtures as variational approximation makes HMNNs also novel from a training perspective, indeed the reparameterization trick is modified to train on mixtures.

1.3 Inference in Stochastic Epidemic models via Multinomial Approximations

Compartmental models are used to predict multiple aspects of an epidemic, e.g. reproduction number, and so guide the governments' control measures to limit the spread of the disease [Brauer, 2008, O'Neill, 2010, Kucharski et al., 2020]. Due to the intractability of the likelihood function, statistical inference for these models is a major computational challenge [Bretó, 2018]. Indeed, likelihood computation involves summing over a prohibitively large number of configurations of latent variables, representing counts of subpopulations in disease states. For this reason, inference is generally approached with various form of stochastic simulations [Funk and King, 2020, Kypraios et al., 2017, McKinley et al., 2018, Brown et al., 2018, 2016, Lekone and Finkenstädt, 2006, Murray et al., 2018, Stocks, 2019, Fasiolo et al., 2016], which, however, have several downsides: a computational cost that depends on the number of simulations, which at the same time controls the inference quality; a computational cost that increases with the total population size, which is considerable in most of the epidemic applications; a large use of tuning parameters that are difficult or computationally expensive to choose.

A new inference approach for compartmental models is designed in chapter 6. The key idea is to reformulate a general compartmental model as an HMM and then propose an approximation for the filtering and the smoothing distributions. To build up the approximate procedure, the first step is to choose the approximation class. An intuitive choice is the class of multinomial distributions because it models the counting of individuals over compartments. After that approximate filtering and smoothing algorithms have to be defined to keep the approximation in the same class during the computations. The outcome is an algorithm providing a collection of multinomial approximations to the filtering and smoothing distributions where the number of trials is set to the total population size, while the events' probabilities, representing the probability of an individual being in a chosen compartment, are obtained through recursions, on the same flavour of the forward-backward algorithm.

The computational cost of the resulting procedure depends only on the compartmental size and not on the total population size. This is a significant improvement to the scalability, given that the compartmental size is generally several order of magnitude lower than the total population size. Moreover, in contrast to ODE models, the multinomial approximation can account for statistical variability in disease dynamics and allows approximate evaluation of the likelihood function for model parameters without any stochastic simulation or algorithm tuning parameters. In addition, the resulting marginal likelihood can be combined with MCMC or Expectation Maximization techniques for parameters estimation. Finally, from an experimental point of view, the algorithm shows to: recover the ground truth parameters in synthetic data scenarios and compares favourably with different baselines [Lekone and Finkenstädt, 2006, Brown et al., 2018, Chowell et al., 2004] and extend a method of Kucharski et al. [2020] for estimating the time-varying reproduction number of COVID-19 in Wuhan, China, from an ODE compartmental model to a stochastic model.

Chapter 2

Background

This chapter set the background that is necessary for the development of the research. Section 2.1 introduces the main notation, hidden Markov models, factorial hidden Markov models, the filtering-smoothing problem and parameter learning for hidden Markov models. Section 2.2 proposes alternative solutions to the filtering-smoothing problem, i.e. variational Bayes and bootstrap particle filter-smoother. Section 2.3 introduces the key concepts behind neural networks and Bayesian neural networks.

2.1 Hidden Markov Models

Throughout the thesis refer to $(\Omega, \mathscr{H}, \mathbb{P})$ as a background probability space, such that any random variable X has to be thought of as mapping from the measurable space $(\Omega, \mathscr{H})$, with its distribution given by the image of $\mathbb{P}$ under X. The dependence on the background probability space is dropped for simplicity. Further use the notation: $\delta_{\cdot}$ for the Dirac delta measure; $\mathbb{I}_A(\cdot)$ as the indicator function over the set A; $\mathbb{E}[\cdot]$ as the expected value and $\mathbb{E}[\cdot|\cdot]$ as the conditional expected value, moreover, if a probability measure is specified as a pedex of the expectation then the expectation is taken under the specified probability measure, for instance $\mathbb{E}_\nu[\cdot]$ is the expectation under ν; $a_{0:t}$ for the elements $a_0, \dots, a_t$ of the sequence $(a_t)_{t\geq 0}$.

2.1.1 Preliminaries

Let $(\mathbb{S}, \mathscr{S})$ be a measurable space, with $\mathbb{S}$ being a Polish space and $\mathscr{S}$ being a σ-algebra on $\mathbb{S}$. Moreover, let $(\mathbb{Y}, \mathscr{Y})$ be another measurable space, with $\mathbb{Y}$ being a Polish space and $\mathscr{Y}$ being a σ-algebra on $\mathbb{Y}$.

Definition 2.1. The bivariate discrete time stochastic process $(X_t, Y_t)_{t\geq 0}$, with $(X_t)_{t\geq 0}$ unobserved, on the state space $(\mathbb{S}\times\mathbb{Y}, \mathscr{S}\otimes\mathscr{Y})$ is called hidden Markov model if there exists transition kernels $P:\mathbb{S}\times\mathscr{S}\to[0,1]$ and $G:\mathbb{S}\times\mathscr{Y}\to[0,1]$ such that:

$$\mathbb{E}[f(X_{t+1}, Y_{t+1})|X_{0:t}, Y_{0:t}] = \int f(x,y)G(x,dy)P(X_t,dx), \quad \text{for any } t\geq 0$$

and a probability measure λ_0 on $\mathbb{S}$ such that:

$$\mathbb{E}[f(X_0,Y_0)] = \int f(x,y)G(x,dy)\lambda_0(dx),$$

for every bounded and measurable function $f:\mathbb{S}\times\mathbb{Y}\to\mathbb{R}$. Under this framework, λ_0 is named initial measure, P is called transition kernel and G is referred to as the emission kernel.

In the HMM literature, $(X_t)_{t\geq 0}$ is referred to as the hidden process with state-space $\mathbb{S}$, while $(Y_t)_{t\geq 0}$ is the observed process, or simply observations. The initial measure λ_0 is also called initial distribution and it represents the distribution of X_0, i.e. $X_0\sim\lambda_0$. At the same time, the transition kernel P gives the distribution of $X_{t+1}|X_t$, i.e. $X_{t+1}|X_t\sim P(X_t,\cdot)$ and the emission kernel G provides the distribution of $Y_t|X_t$, i.e. $Y_t|X_t\sim G(X_t,\cdot)$, which is often referred to as the emission distribution of the HMM.

The conditional independence relation in an HMM can be represented with a directed acyclic graph, which can be found in Figure 2.1. In this thesis there are frequent references to measure theory and the main results in chapter 3 requires a deeper formalization of HMMs. So in the same vein of Rebeschini and Van Handel [2015], van Handel [2008] one can propose a more formal definition of HMM.

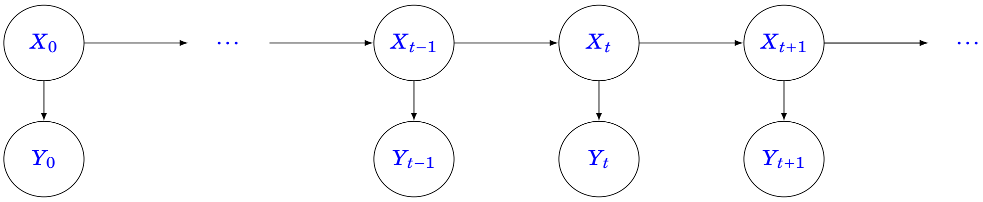


Figure 2.1: A directed acyclic graph representing the conditional independence structure of an HMM.

The finite dimensional distributions of the HMM $(X_t,Y_t)_{t\geq 0}$ are completely determined by the initial distribution λ_0, the transition kernel P and the emission kernel G, indeed for any bounded and measurable function f and for any $t\geq 0$:

$$\mathbb{E}[f(X_{0:t},Y_{0:t})] = \int f(x_{0:t},y_{0:t})G(x_t,dy_t)P(x_{t-1},dx_t)\ldots G(x_0,dy_0)\lambda_0(dx_0). \tag{2.1}$$

Further assume that there exists $p : \mathbb{S} \times \mathbb{S} \to \mathbb{R}_+$ and a measure ψ over $\mathbb{S}$ such that:

$$\int \mathbb{I}_A(z) P(x, dz) = \int \mathbb{I}_A(z) p(x,z) \psi(dz), \quad \text{for any } x \in \mathbb{S} \text{ and } A \in \mathscr{S}, \tag{2.2}$$

in this case p is called transition density and ψ is called reference measure. Moreover, assume non degenerate observations, meaning that there exists $g : \mathbb{S} \times \mathbb{Y} \to \mathbb{R}_+$ and a measure ϕ over $\mathbb{Y}$ such that:

$$\int \mathbb{I}_A(y) G(x, dy) = \int \mathbb{I}_A(y) g(x,y) \phi(dy), \quad \text{for any } x \in \mathbb{S} \text{ and } A \in \mathscr{Y}, \tag{2.3}$$

in such scenario g is referred to as the emission density and ϕ is called reference measure.

Under assumptions (2.2) and (2.3), (2.1) has a simpler form, i.e. for any bounded and measurable function f and for any $t \geq 0$:

$$\mathbb{E}[f(X_{0:t}, Y_{0:t})] = \int f(x_{0:t}, y_{0:t}) g(x_t, y_t) p(x_{t-1}, x_t) \dots g(x_0, y_0) \phi(dy_t) \psi(dx_t) \dots \phi(dy_0) \lambda_0(dx_0),$$

Factorial hidden Markov models In this thesis, there is a particular focus on a class of HMMs called factorial hidden Markov models (FHMMs) [Ghahramani and Jordan, 1997]. Start by defining the measurable space $(\mathbb{X}^V, \mathscr{X}^V)$ indexed by the finite set V, where $\mathbb{X}^V$ is the product space $\times_{v \in V} \mathbb{X}$ and $\mathscr{X}^V$ is the product σ-algebra $\bigotimes_{v \in V} \mathscr{X}$. Under this framework one can write $x = (x^v)_{v \in V}$ for any chosen $x \in \mathbb{X}^V$ and $A = (A^v)_{v \in V}$ for any chosen $A \in \mathscr{X}^V$. An FHMM is an HMM on the state space $(\mathbb{X}^V \times \mathbb{Y}, \mathscr{X}^V \times \mathscr{Y})$ where each component of the unobserved Markov chain evolves independently from the others, or equivalently for any $v \in V$, selected component of the Markov chain, X_t^v is conditionally independent of all other variables given X_{t-1}^v. A directed acyclic graph showing the conditional independence structure of an FHMM is shown in Figure 2.2.

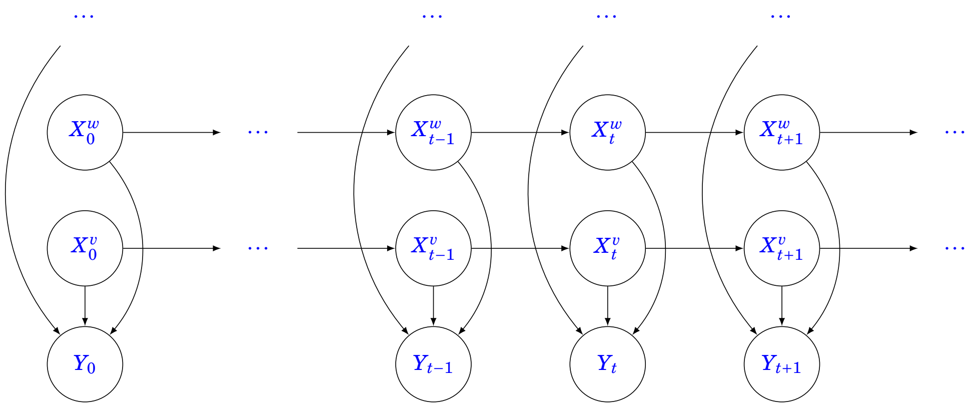


Figure 2.2: A directed acyclic graph representing the conditional independence structure of an FHMM.

Formally, an FHMM can be defined by assuming a product form for the transition kernel, or equally a factorization of the transition density.

Definition 2.2. The hidden Markov model $(X_t, Y_t)_{t\geq 0}$ with initial measure λ_0, transition kernel P and emission kernel G, is called factorial hidden Markov model (FHMM) if P is such that:

$$P(x,\cdot) = \bigotimes_{v\in V} P^v(x^v,\cdot) \quad \text{for any } x \in \mathbb{X}^V,$$

where $P^v : \mathbb{X} \times \mathscr{X} \to [0,1]$ is a transition kernel on $(\mathbb{X}, \mathscr{X})$ for any $v \in V$. Moreover, the product form of P is reflected as a factorization on the transition density p:

$$p(x,z) = \prod_{v\in V} p^v(x^v, z^v) \quad \forall x, z \in \mathbb{X}^V,$$

with $p^v : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_+$ for any $v \in V$.

For the rest of the thesis, an HMM $(X_t, Y_t)_{t\geq 0}$, or an FHMM, as in definition 2.1, or definition 2.2, is going to be identify through its initial distribution λ_0, transition density p and emission density g, so assumptions (2.2) and (2.3) are taken as granted.

Unless specified differently $(X_t, Y_t)_{t\geq 0}$ is an HMM with initial distribution λ_0, transition density p and emission density g.

2.1.2 Filtering and Smoothing distributions

Given a time horizon $T \in \mathbb{N}$ and an observation history $(y_t)_{t=0,\dots,T}$, i.e. the realization of the stochastic process $(Y_t)_{t=0,\dots,T}$, two conditional distributions can be defined for each time step $t \in \{0, \dots, T\}$:

- the conditional distribution of X_t given the realized observations $(y_1, \dots, y_t)$, i.e. the distribution of $X_t | Y_{0:t} = y_{0:t}$, called filtering distribution, and
- the conditional distribution of X_t given all the observations $(y_1, \dots, y_T)$, i.e. the distribution of $X_t | Y_{0:T} = y_{0:T}$, named smoothing distribution.

Both the filtering and the smoothing distribution at time $t \in \{0, \dots, T\}$ are probability measures on $\mathbb{S}$, and they are going to be denoted with π_t and $\pi_{t|T}$, respectively.

Filtering distribution Filtering can be conducted with a forward pass through the data, with a recursive application of two operations: “prediction” step and “correction” step.

Definition 2.3. Given a time horizon $T \in \mathbb{N}$ and an observation history $(y_t)_{t=0,\dots,T}$ the filtering distribution is defined recursively as the probability measure π_t:

$$\pi_0 := \lambda_0, \qquad \pi_t := \mathsf{F}_t \pi_{t-1}, \qquad \mathsf{F}_t := \mathsf{C}_t \mathsf{P}, \qquad t \in \{1, \dots, T\}, \tag{2.4}$$

where given the probability measure μ on $\mathbb{S}$ the prediction operator P and the correction operator C_t are defined as:

$$\mathsf{P}\mu(A) := \int \mathbb{1}_A(x) p(z,x) \mu(dz) \psi(dx), \qquad \mathsf{C}_t\mu(A) := \frac{\int \mathbb{1}_A(x) g(x, y_t) \mu(dx)}{\int g(x, y_t) \mu(dx)}, \qquad A \in \mathscr{S}. \tag{2.5}$$

As already anticipated, it is clear from the above definition that: P performs a transition from μ through the Markov transition kernel P, while C_t can be understood as applying a Bayes' update to the prior μ through the likelihood function $g(x, y_t)$.

Smoothing distribution Amongst various smoothing algorithms, this thesis focuses on the forward-filtering and backward-smoothing method presented by Kitagawa [1987], which involves a backward in time recursion performed after filtering.

Definition 2.4. Given a time horizon $T \in \mathbb{N}$, an observation history $(y_t)_{t=0,\dots,T}$ and the filtering distributions $(\pi_t)_{t=0,\dots,T}$, the smoothing distribution is defined recursively as the probability measure $\pi_{t|T}$:

$$\pi_{T|T} := \pi_T, \qquad \pi_{t|T} := \mathsf{R}_{\pi_t}\pi_{t+1|T}, \qquad t \in \{T-1, \dots, 0\}, \tag{2.6}$$

where given the probability measures μ, ν on $\mathbb{S}$ the operator $\mathsf{R}_\nu\mu$ is defined as:

$$\mathsf{R}_\nu\mu(A) := \int \frac{\int \mathbb{I}_A(x)p(x,z)\nu(dx)}{\int p(\tilde{x},z)\nu(d\tilde{x})}\mu(dz), \qquad A \in \mathscr{S}.$$

As explained at the beginning of the paragraph, at time t the operator R_{π_t} is including the information deriving from $y_{t+1}, \dots, y_T$ into the filtering distribution π_t through the application of the non-homogeneous Markov transition kernel $p(x,z)\pi_t(dx)/\int p(\tilde{x},z)\pi_t(d\tilde{x})$ to $\pi_{t+1|T}$. Hence, R_ν shows how to compute a reverse kernel to apply backward in time.

Note that the operator R_ν can be modified to output joint distributions, indeed given a time index t, a time horizon T, a probability measure μ on $\times_{\tilde{t}=t+1}^{T} \mathbb{S}$ and a probability measure ν on $\mathbb{S}$ then the operator:

$$\mathsf{R}_\nu^{\text{joint}}\mu(A) := \int \frac{\int \mathbb{I}_A(x,z)p(x,z)\nu(dx)}{\int p(\tilde{x},z)\nu(d\tilde{x})}\mu(dz), \qquad A \in \mathscr{S} \otimes \bigotimes_{\tilde{t}=t+1}^{T} \mathscr{S},$$

gives a probability measure on $\times_{\tilde{t}=t}^{T} \mathbb{S}$. A backward application of R_ν^{joint} can then provide the joint smoothing distribution $\pi_{0:T|T}$, i.e. the conditional distribution of $X_{0:T}$ given $Y_{0:T} = y_{0:T}$.

Unless specified differently, set a time horizon $T \in \mathbb{N}$ and an observation history $(y_t)_{t=0,\dots,T}$.

Curse of dimensionality and computational issues If the state-space $\mathbb{S}$ is continuous, solving recursions (2.4) and (2.6) is complicated or not even possible, unless some restrictive assumptions are taken, as for the Kalman filter [Welch et al., 1995]. Alternatively, sequential Monte Carlo methods (SMC) [Doucet et al., 2000], see subsection 2.2.2, provides samples from the filtering and smoothing distributions without need of a closed-form solution. Even though this is generally easy to implement when scaling up to high dimensions the curse of dimensionality becomes a considerable issue [Rebeschini and Van Handel, 2015], more details about the curse of dimensionality follows at the end of subsection 2.2.2. If the state-space $\mathbb{S}$ is discrete, all the integrals are going to be substituted with sums, the reference measure ψ becomes the counting measure, the transition density is a transition matrix and all the probability measures become simply probability vectors. Under this setting, recursions (2.4) and (2.6) form the well-known forward-backward algorithm, which computes filtering and smoothing distributions in close-form by solving some straightforward linear algebra operations at a cost of $\mathcal{O}(\mathbf{card}(\mathbb{S})^3)$. However, if the cardinality of $\mathbb{S}$ is big the whole procedure becomes unfeasible from a computational point of view.

2.1.3 Statistical inference

The next question is how to compute λ_0, p and g when unknown. This thesis considers a λ_0 with a density of parametric form and a p and a g with parametric forms, such us exponential families, whose parameters can be ideally inferred from the observations of the HMM. Let then $\theta^\star$ be the collection of parameters found: in the density of λ_0, in the transition density p and in the emission density g, and call the parameter space $(\Theta, \mathcal{T})$, with $\mathcal{T}$ σ-algebra on Θ. Note that λ_0, p and g are then dependent on $\theta^\star$, and it would be desirable referring to them with $\lambda_0^{\theta^\star}$, $p^{\theta^\star}$, $g^{\theta^\star}$, however to make the notation lighter the $\theta^\star$ apex is avoided.

Bayesian inference Assume that $\theta^\star$ has prior λ_θ with λ_θ probability measure over Θ. For any bounded and measurable function f one can write:

$$\begin{aligned}&\mathbb{E}[f(\theta^\star, X_{0:T}, Y_{0:T})]\\&\quad = \int f(\theta, x_{0:T}, y_{0:T}) g(x_T, y_T) p(x_{T-1}, x_T) \dots g(x_0, y_0) \phi(dy_T) \psi(dx_T) \dots \phi(dy_0) \lambda_0(dx_0) \lambda_\theta(d\theta),\end{aligned}$$

which can be used to estimate the posterior distribution over $\theta^\star$ and so extract an estimate for $\theta^\star$. For instance, given the prior λ_θ and the observation history $(y_t)_{t=0,\dots,T}$, an estimate for $\theta^\star$ can be given by $\hat{\theta}_T^B := \mathbb{E}[\theta^\star | Y_{0:T} = y_{0:T}]$. The posterior distribution of $\theta^\star$ given the data is needed to compute an estimate for $\theta^\star$, however this can be done through filtering as in (2.4) by expanding the hidden process of the HMM. Indeed, consider $(\tilde{X}_t, Y_t)_{t \ge 0}$ as the stochastic process $\tilde{X}_t = (\theta^\star, X_t)$ for any $t \ge 0$, then $(\tilde{X}_t, Y_t)_{t \ge 0}$ is still an HMM and all the procedures in subsection 2.1.1 apply.

Even though computing the posterior distribution over $\theta^\star$ is theoretically simple, in practice the computational issues and the curse of dimensionality mentioned in subsection 2.1.1 are still present. As an alternative, one can employ Markov chain Monte Carlo methods [Cappé et al., 2006] to sample from high-dimensional probability distributions, however, this is beyond the scope of this thesis.

Maximum likelihood and EM Consider now a frequentist scenario and so assume that $\theta^\star$ is not random. In this case, the hope is to find the maximum likelihood estimate $\hat{\theta}_T^{MLE}$ for a given observation history $(y_t)_{0:T}$ as:

$$\hat{\theta}_T^{MLE} := \arg\max_{\theta \in \Theta} L(\theta; y_{0:T}), \tag{2.7}$$

where:

$$L(\theta; y_{0:T}) := \int g(x_0, y_0)\lambda_0(dx_0) \prod_{t=1}^{T} \int g(x_t, y_t)p(x_{t-1}, x_t)\pi_{t-1}(dx_{t-1})\psi(dx_t),$$

with π_t the filtering distribution as in definition 2.3. However, it does not exist an algorithm that fulfils (2.7) when Θ is infinite, and even exploring a finite but large Θ might be computationally unfeasible. To this end, the expectation-maximization (EM) algorithm, also called Baum-Welch algorithm for discrete state-space HMMs, is widely used.

Precisely, let the joint density of $X_{0:T}, Y_{0:T}$ given the parameters θ be:

$$L(\theta; x_{0:T}, y_{0:T}) := g(x_0, y_0)\lambda_0(dx_0) \prod_{t=1}^{T} g(x_t, y_t)p(x_{t-1}, x_t), \quad \text{for any } x_t \in \mathbb{X}^V, y_t \in \mathbb{Y} \text{ with } t = 0, \ldots, T,$$

and let the function $Q_T(\theta, \theta')$ for the given observation history $(y_t)_{t=0,\ldots,T}$ be:

$$Q_T(\theta, \theta') := \mathbb{E}\left[\log(L(\theta; X_{0:T}, Y_{0:T}))|Y_{0:T} = y_{0:T}, \theta'\right],$$

where $\mathbb{E}[\cdot|\cdot, \theta']$ is the conditional expectation under the parameters θ'. Then the EM algorithm proceeds as follows:

E-step: compute the joint smoothing $\pi_{0:T|T}$ (this might be different depending on the form of Q_T) under the parameter θ';

M-step: set θ' to $\arg\max_{\theta \in \Theta} Q_T(\theta, \theta')$.

The EM algorithm recursively finds combinations of parameters θ that are associated with higher likelihood. Indeed, the expectation step consists of computing the joint smoothing distribution, which is then used in the maximization step to maximize a target function that guarantees the motion to regions with higher likelihood, see Lemma 2.1 [van Handel, 2008]. Note that to maximize Q_T, a close form solution for its gradient is needed, which is straightforward if λ_0 has a density that belongs to an exponential family and p, g belong to exponential families as well.

Lemma 2.1. *For a given observation history $(y_t)_{t=0,\ldots,T}$, if:*

$$\theta'' = \arg\max_{\theta \in \Theta} Q_T(\theta, \theta')$$

then $L(\theta''; y_{0:T}) \geq L(\theta'; y_{0:T})$.

Proof. The proof can be found in van Handel [2008] page 85 Lemma 6.10. ■

To conclude, remark that the EM algorithm still needs the joint smoothing distribution which requires recursion (2.4) and a more sophisticated version of (2.6), i.e. a recursion build on the operator R_V^{joint}, meaning that the computational cost and the curse of dimensionality still appear, as explained in subsection 2.1.1.

2.2 Alternatives to classic filtering and smoothing

As mentioned in subsection 2.1.1, filtering and smoothing distributions might be difficult to compute or computationally unfeasible. This section analyses some approximate approaches to the filtering and smoothing problem.

2.2.1 Variational inference

Variational inference [Blei et al., 2017] approximates the target posterior distribution with a parametric distribution picked up from a class of distributions by minimizing Kullback-Leibler divergence criteria. This significantly reduces the computational cost and provides approximate solutions for complicated posterior distributions.

Before describing the actual variational inference procedure it is essential to introduce the definition of Kullback-Leibler (**KL**) divergence. Given two probability measures μ and ν on $\mathbb{X}^V$ define the **KL**-divergence between μ and ν as:

$$\mathbf{KL}(\mu||\nu) := \int \log\left(\frac{d\mu}{d\nu}(x)\right)\frac{d\mu}{d\nu}(x)\nu(dx), \tag{2.8}$$

where $d\mu/d\nu$ is the Radon-Nikodym derivative of μ with respect to ν, with the convention that $0\log(0) = 0$. In order to have (2.8) well-defined μ needs to be absolutely continuous with respect to ν. A simpler version of (2.8) can be obtained by considering a common reference measure. Let $h(\cdot)$ and $q(\cdot)$ be the probability density (or mass) functions of the probability distributions μ and ν with respect to the reference measure ψ on $\mathbb{X}^V$, i.e.:

$$\int \mathbb{I}_A(x)\mu(dx) = \int \mathbb{I}_A(x)h(x)\psi(dx), \quad \text{and}$$
$$\int \mathbb{I}_A(x)\nu(dx) = \int \mathbb{I}_A(x)q(x)\psi(dx), \quad \text{for any } x \in \mathbb{X}^V \text{ and } A \in \mathcal{X}^V,$$

then the **KL** divergence can be reformulated as:

$$\mathbf{KL}(q||h) := \int \log\left(\frac{q(x)}{h(x)}\right)q(x)\psi(dx).$$

Now, given a target probability measure μ with probability density $h(\cdot)$ with respect to the reference measure ψ and the class of probability density functions $\mathcal{Q}$ with respect to the same reference measure, the variational approximation of $h(\cdot)$ is:

$$q^\star := \arg\min_{q\in\mathcal{Q}} \mathbf{KL}(q||h).$$

Variational inference for HMM and limitations Given an HMM with known initial distribution, transition density and emission density, the joint smoothing distribution $\pi_{0:T|T}$, i.e. the conditional distribution of $X_0,\dots,X_T$ given the observations $y_0,\dots,y_T$, can be approximated with variational inference at a low computational cost, for example in FHMMs the computational

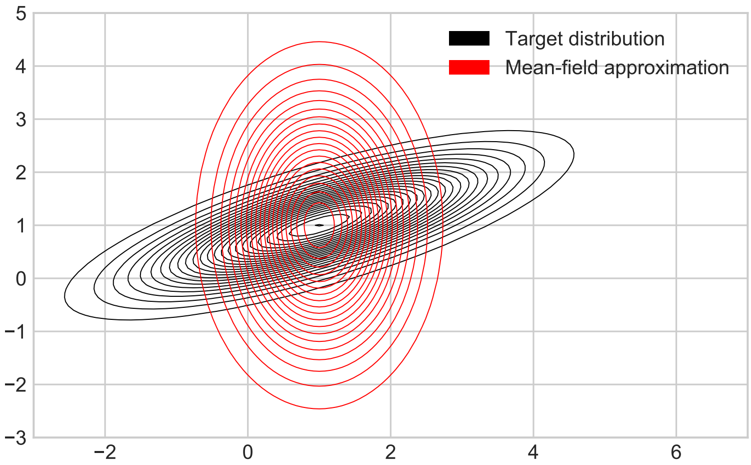


Figure 2.3: Example of variational inference where the target is a bivariate Gaussian, in black, and the variational approximation is picked from the class of bivariate Gaussian with diagonal covariance matrix (product of independent Gaussians), in red, also called mean-field approximation.

complexity of variational inference is a low-order polynomial in state dimension [Ghahramani and Jordan, 1997]. However, there are no theoretical guarantees on the quality of the variational approximation besides that it is minimizing the KL-divergence between the distributions, and at the same time, this approach is not online, meaning that given new data the whole optimization has to be rerun from scratch.

2.2.2 Particle Filter and Smoother

If $\mathbb{X}^V$ is continuous it is not usually possible to compute filtering and smoothing distributions in closed form, hence it is frequent to use Monte Carlo approximations. This section reviews straightforward particle filter and smoother for HMMs, more details can be found in Doucet et al. [2000], Andrieu et al. [2010].

Before starting, assume that the horizon T, the initial distribution λ_0, the transition density p and the emission density g are known. At the same time, assume that the probability measure λ_0 has density λ_0 (use the same notation). Moreover, assume also that it is possible to sample from λ_0 and from $p(x,\cdot)$ for any $x \in \mathbb{X}^V$.

Bootstrap particle filter As for recursion (2.4), the bootstrap particle filter performs a forward step through the observation history $(y_t)_{t=0,\ldots,T}$ and proposes a sequential importance sampling providing samples from the filtering distributions. Start by sampling N particles $(x_0^k)_{k=1,\ldots,N}$:

$$x_0^k \sim \lambda_0, \quad \text{for } k = 1,\ldots,N,$$

hence $(x_0^k)_{k=1,\dots,N}$ is a sample from the initial distribution. Then compute the weights $(w_0^k)_{k=1,\dots,N}$:

$$w_0^k = \frac{g(x_0^k, y_0)}{\sum_{\tilde{k}=1}^N g(x_0^{\tilde{k}}, y_0)}, \quad \text{for } k = 1, \dots, N$$

Notice that by the law of large numbers:

$$\pi_0 \approx \sum_{k=1}^N w_0^k \delta_{x_0^k} =: \hat{\pi}_0,$$

hence by resampling:

$$\hat{x}_0^k \sim \sum_{k=1}^N w_0^k \delta_{x_0^k}, \quad \text{for } k = 1, \dots, N,$$

surely $(\hat{x}_0^k)_{k=0,\dots,N}$ is a sample from the filtering distribution π_0, which conclude the initial step.

The procedure for a general step can be recursively formulated as follows. Start by sampling N particles $(x_t^k)_{k=1,\dots,N}$ from the kernel:

$$x_t^k \sim p(\hat{x}_t^k, \cdot), \quad \text{for } k = 1, \dots, N,$$

hence $(x_t^k)_{k=1,\dots,N}$ is a sample from the one step ahead predictive distribution. Then compute the weights $(w_t^k)_{k=1,\dots,N}$:

$$w_t^k = \frac{g(x_t^k, y_t)}{\sum_{\tilde{k}=1}^N g(x_t^{\tilde{k}}, y_t)}, \quad \text{for } k = 1, \dots, N.$$

Notice that by the law of large numbers:

$$\pi_t \approx \sum_{k=1}^N w_t^k \delta_{x_t^k} =: \hat{\pi}_t,$$

hence by resampling:

$$\hat{x}_t^k \sim \sum_{k=1}^N w_t^k \delta_{x_t^k}, \quad \text{for } k = 1, \dots, N,$$

surely $(\hat{x}_t^k)_{k=0,\dots,N}$ is a sample from the filtering distribution π_t. A detailed description of the steps can be found in algorithm 1, where $D(\cdot, (w^k)_{k=1,\dots,N})$ is the categorical distribution on $\{1, \dots, N\}$ with probabilities parameters $(w^k)_{k=1,\dots,N}$ subjected to $\sum_{k=1}^N w^k = 1$. The output $((x_t^k)_{k=1,\dots,N})_{t=0,\dots,T}$ of algorithm 1 is a collection of samples from the filtering distributions. Note that the bootstrap particle filter also output the set of ancestors $((A_t^k)_{k=1,\dots,N})_{t=1,\dots,T-1}$ and the final collection of weights $(w_T^k)_{k=1,\dots,N}$, these are used in the particle smoother.

Remark that it is not always possible to sample from λ_0 and p. In that case, one can fix a proposal distribution q (which can also depend on the data), sample from this distribution and correct the weights according to this procedure.

Algorithm 1 Bootstrap particle filter

```
Require: $T; (y_t)_{t=0,\dots,T}; \lambda_0; p; g; N$
Sample $x_0^k \sim \lambda_0$ for $k = 1,\dots,N$
Compute $w_0^k = \frac{g(x_0^k, y_0)}{\sum_{\tilde{k}=1}^N g(x_0^{\tilde{k}}, y_0)}$ for $k = 1,\dots,N$
Resample $\hat{x}_0^k = x_0^{A_0^k}$ with $A_0^k \sim D\left(\cdot, \left(w_0^{\tilde{k}}\right)_{\tilde{k}=1,\dots,N}\right)$ for $k = 1,\dots,N$
for $t = 1,\dots,T$ do
    Sample $x_t^k \sim p(\hat{x}_{t-1}^k, \cdot)$ for $k = 1,\dots,N$
    Compute $w_t^k = \frac{g(x_t^k, y_t)}{\sum_{\tilde{k}=1}^N g(x_t^{\tilde{k}}, y_t)}$ for $k = 1,\dots,N$
    Resample $\tilde{x}_t^k = x_t^{A_{t-1}^k}$ with $A_t^k \sim D\left(\cdot, \left(w_t^{\tilde{k}}\right)_{\tilde{k}=1,\dots,N}\right)$ for $k = 1,\dots,N$
return $((A_t^k)_{k=1,\dots,N})_{t=0,\dots,T-1}, (w_T^k)_{k=1,\dots,N}$ and $((x_t^k)_{k=1,\dots,N})_{t=0,\dots,T}$
```

Particle smoother As for recursion (2.6), the particle smoother is performing a backward step through the data and provides samples from the joint smoothing distribution $\pi_{0:T|T}$. The key step is to be able to track back the parent node of each particle. Indeed, the particle smoother starts by sampling indexes from $D(\cdot, (w_T^k)_{k=1,\dots,N})$ and then it traces back the parent nodes up to the time step 0, to provide a sample from the joint smoothing [Doucet et al., 2000, Andrieu et al., 2010]. The full procedure is available in algorithm 2.

Algorithm 2 Particle smoother

```
Require: $((x_t^k)_{k=1,\dots,N})_{t=0,\dots,T}; (w_T^k)_{k=1,\dots,N}; ((A_t^k)_{k=1,\dots,N})_{t=1,\dots,T-1}, ; \tilde{N}$
Sample $P_T^i \sim D\left(\cdot, \left(w_T^i\right)_{i=1,\dots,N}\right)$ for each $i = 1,\dots,\tilde{N}$
Set $x_{T|T}^i = x_T^{P_T^i}$ for $i = 1,\dots,\tilde{N}$
for $t = T-1,\dots,0$ do
    Set $P_t^i = A_t^{P_{t+1}^i}$ and $x_{t|T}^i = x_t^{P_t^i}$ for $i = 1,\dots,\tilde{N}$
return $((x_{t|T}^i)_{i=1,\dots,\tilde{N}})_{t=0,\dots,T}$
```

It is worthed to mention that there exist particle smoother algorithms designed to overcome path degeneracy at a computational cost that grows quadratically with the number of particles (e.g FFBSa [Godsill et al., 2004] and FFBSm [Del Moral et al., 2010]), this becomes a considerable issue when a big number of particles is needed, see the following section.

Curse of dimensionality As already mentioned in subsection 2.1.1, algorithm 1 suffers the curse of dimensionality when scaling up to high dimension, meaning that the quality of the approximations $(\hat{\pi}_t)_{t\geq 0}$ deteriorates with the dimension **card**(V) of the underlying states-space $\mathbb{X}^V$ [Bengtsson et al., 2008, Rebeschini and Van Handel, 2015]. Precisely, the bootstrap particle filter can approximate the filtering distributions only if the number of particles N grows exponentially in the dimension **card**(V), more details can be found in Snyder et al. [2008], Bickel et al. [2008]. Tempering based methods [Marinari and Parisi, 1992] can tackle high-dimensional state-spaces in particle filters [Godsill and Clapp, 2001], but they are generally restricted to a low-dimensional observation space.

2.3 Neural networks

This section is a short overview of neural networks and Bayesian neural networks for supervised learning. Neural network (NN) is the most popular model in machine learning and it has shown outstanding performances in several fields. Supervised learning is a branch of machine learning where the data consists of an input x and an output y, with the aim of finding a map from the input onto the output. There are several other branches of machine learning where NNs are the state of art, however, this thesis focuses on the role of NNs in supervised learning only. In this section, use the notation w^{T} for the transpose of the matrix w.

2.3.1 Feedforward neural networks

Feedforward neural networks, or multilayer perceptrons, are complicated functions obtained by the composition of linear and non-linear maps [LeCun et al., 2015, Goodfellow et al., 2016].

A feed-forward neural network can be generally represented as

$$y^\star = f(x;w),$$

where x is the input, w are the parameters and $y^\star$ is the output of the NN (this is going to be compared with the actual output y). To be more precise, f is the result of a composition of multiple functions, and it takes the form of a chain where each element is a layer of the NN. The number of elements in the chain gives the depth of the NN, which is denoted by D. Hence:

$$f = f^{(D)} \circ \dots f^{(1)}. \tag{2.9}$$

Each element of the chain is obtained by the composition of a linear function, represented by a matrix, with a non-linear function, called activation [Leshno et al., 1993, Ramachandran et al., 2017]. The elements of the matrix representing the linear function are called weights of the NN and they are the parameters of the NN. Then for $i = 1,\dots,D$:

$$f^{(i)} = a^{(i)} \circ l^{(i)} \tag{2.10}$$

with $l^{(i)}$ linear function and $a^{(i)}$ elementwise application of the activation function. Precisely, for a given input x, let $h^{(i-1)}$ be the output of $f^{(i-1)} \circ \dots \circ f^{(1)}(x)$ then:

$$f^{(i)}\left(h^{(i-1)}\right) = a^{(i)}\left(\left(w^{(i)}\right)^{\mathrm{T}} h^{(i-1)}\right),$$

where $w^{(i)}$ is the matrix of weights associated to layer i. Note that under this notation $w = (w^{(1)},\dots,w^{(D)})$. Remark that some bias terms are generally present, but for the sake of simplicity

are not reported in the above formulation. After the application of the linear function, the length of the output is called width of the layer, with the constraint that the width of the final layer has to be the same as the dimension of the actual output y, i.e. the output of the NN $y^{\star}$ has to be a vector with the same length of y (remark that if y is categorical it is possible to switch to a one-hot encoding format). After the elementwise application of the activation function, each element of the output is called hidden unit or neuron of the NN.
The final step consists of relating the NN output $y^{\star}$ with the actual output y, this is done by the loss function ℓ, which is usually interpreted as a negative log-likelihood over (x, y) given the parameters w.

Given all the building blocks of an NN, the next question is how to learn the best combination of weights, or, from a machine learning perspective, how to train. The statistical intuition is that w are being selected as close as possible to the maximum likelihood estimator by minimizing the sum of the losses over the data, i.e. given the data $(x_i, y_i)_{i=1,\dots,n}$ the sum of the losses over the data is:

$$L\left((x_i, y_i)_{i=1,\dots,n}; w\right) := \sum_{i=1}^{n} \ell(y_i, f(x_i; w)).$$

Indeed, the loss in (x, y) has the interpretation of negative log-likelihood at (x, y), meaning that minimizing $L((x_i, y_i)_{i=1,\dots,n}; w)$ is equivalent to maximise the log-likelihood of the model. However, it is not possible to compute the minimum in closed form, due to the non-linear layers of the NN. Even though the form of the NN is complex, given a set of weights w computing the gradient of $L((x_i, y_i)_{i=1,\dots,n}; w)$ is simple because of the form of (2.9) and (2.10). Precisely, the gradient can be computed by backpropagation [Hecht-Nielsen, 1992], which is simply the chain rule for derivatives.
Once the gradient $\nabla_w L$ of $L((x_i, y_i)_{i=1,\dots,n}; w)$ is available, the weights can be updated according to any gradient descent technique (e.g. vanilla gradient descent, ADAM, etc.), for instance given the learning rate l a vanilla gradient descent step is:

$$w = w - l\nabla_w L.$$

The above procedure is then iterated until convergence, and the final combination of weights represent the trained neural network.

Remark that often computing the gradient on all the available data is too expensive, hence to reduce the computational cost, only a subset of the data is considered, this subset is called minibatch.

2.3.2 Bayesian neural networks

In a feedforward neural network, the weights are scalars and obtained through an optimization procedure. However, pointwise estimates of the weights do not provide any uncertainty quantification and might be a consistent limitation for the NN use. A Bayesian approach to NNs

[Graves, 2011, Kingma and Welling, 2013, Kingma et al., 2015, Blundell et al., 2015] can avoid overconfidence and so provide ways to access uncertainty. Under a Bayesian setting, the weights in the NN become random variables, and the aim is then to compute the conditional distribution of the weights given the data, i.e. the posterior distribution over the weights.

Consider an NN as in subsection 2.3.1, but with random weights:

$$Y^{\star} = f(x; W),$$

where x is the input, W is the random vector of weights and $Y^{\star}$ is the corresponding random output. Moreover, given the distribution p_0 over the weights' space, assume

$$W \sim p_0.$$

Then given the data $(x_i, y_i)_{i=1,\dots,n}$ the log-posterior over the weights $\log p(W|y_1,\dots,y_n)$ is given by:

$$\log p(W|y_1,\dots,y_n) = \text{const.} - L((x_i, y_i)_{i=1,\dots,n}; W) + \log p_0(W). \tag{2.11}$$

Unfortunately, the above posterior distribution cannot be computed easily, because it is not only high-dimensional (number of weights) and continuous (each weight takes value in $\mathbb{R}$), but it also does not have a closed-form solution for any non-trivial architecture of the NN.

One can apply variational inference, see section 2.2.1, to approximate the posterior distribution over the weights of the NN [Graves, 2011, Kingma and Welling, 2013, Kingma et al., 2015, Blundell et al., 2015]. Hence given the the data $(x_i, y_i)_{i=1,\dots,n}$, the posterior distribution over the weights $p(\cdot|y_1,\dots,y_n)$ and the variational class $\mathcal{Q}$ then $p(\cdot|y_1,\dots,y_n)$ can be approximated as:

$$q^{\star} := \arg\min_{q\in\mathcal{Q}} \mathbf{KL}(q||p(\cdot|y_1,\dots,y_n)) = \arg\min_{q\in\mathcal{Q}} \mathbb{E}_q \left[\log q(W) - \log p(W|y_1,\dots,y_n)\right]. \tag{2.12}$$

The above can be also reformulate as an evidence lower bound (ELBO) by applying the Bayes rule to $p(W|y_1,\dots,y_n)$ as in (2.11):

$$\mathbb{E}_q \left[\log q(W) - \log p(W|y_1,\dots,y_n)\right] = \mathbf{KL}(q||p_0) + \mathbb{E}_q \left[L((x_i, y_i)_{i=1,\dots,n}; W)\right] = -\mathbf{ELBO}(q). \tag{2.13}$$

The latest form is more convenient because it involves evaluating an expectation on known quantities: prior, NN output, variational approximation. The final step consists of solving (2.12)-(2.13) and so find the best variational approximation for the posterior distribution over the weights. This requires estimating the gradient of (2.13), which is not trivial and can be done by the reparameterization trick, more details can be found in [Graves, 2011, Kingma and Welling, 2013, Kingma et al., 2015, Blundell et al., 2015] and later in chapter 5.

CHAPTER 3

Exploiting locality in high-dimensional factorial hidden Markov models

This chapter presents a novel low-cost algorithm called the *Graph Filter-Smoother*, which computes approximate filtering and smoothing distributions for high-dimensional FHMMs with factorizable emission density. Sections 3.1, 3.2 introduce the background, the main problem and the notation. Section 3.3 explains how to overcome the prohibitive computational cost of filtering-smoothing algorithms by Graph Filter-Smoother, along with the main theoretical results. The proposed algorithm is indeed supported by mathematical proofs inspired by the work Rebeschini and Van Handel [2015], which are available in chapter 4. The chapter ends with some synthetic data experiments and a real-world application to traffic flow prediction, see section 3.4. As already mentioned, a significant part of this chapter has been submitted to publication and it is available in Rimella and Whiteley [2019].

3.1 Introduction and literature review

As explained in subsection 2.1.1 of chapter 2, the influential paper Ghahramani and Jordan [1997] introduced the class of Factorial hidden Markov models (FHMMs), in which the hidden Markov chain is a multivariate process, with a-priori independent coordinates. This structure provides a rich modelling framework to capture complex statistical patterns in data sequences and it is particularly suited to applications with straightforward local dependencies. Unfortunately, as briefly mentioned in subsection 2.1.2 of chapter 2, the computational cost of computing filtering and smoothing distributions is an issue to scalability of FHMMs. Indeed, the computational cost of the matrix-vector operations underlying the well known forward-backward algorithm, which computes conditional distributions over hidden states given data, and in turn the Baum-

Welch algorithm to perform maximum-likelihood estimation of static parameters, typically grows exponentially with the dimension of the underlying state-space.

Ghahramani and Jordan [1997] derived a variant of the forward-backward algorithm which achieves a degree of efficiency by exploiting the structure of FHMMs, but the exponential-in-dimension scaling of cost cannot be avoided. Ghahramani and Jordan [1997] also proposed two families of variational methods for FHMMs, which allow approximate solution of the smoothing problem: computing conditional distributions over hidden states given past and future data. This allows approximate maximum likelihood parameter estimation via an expectation-maximization (EM) algorithm. The attractive feature of these variational approximations is, in typical FHMMs, that their computational complexity is a low-order polynomial in state dimension. However, little seems to be known about their performance in the context of FHMMs from a theoretical point of view, other than the fact that by construction they minimize a Kullback-Liebler divergence criterion. Moreover, analysis specifically for FHMMs appears to be lacking, and, to the author's knowledge, there are currently no detailed mathematical studies of how variational approximation errors for FHMMs scale with dimension, data record length, model parameters etc.

This chapter focuses on a class of FHMMs whose emission distribution has a factorial structure and proposes approximate inference algorithms called the *Graph Filter* and *Graph Smoother*. In some ways, the Graph Filter and Smoother are similar in spirit to variational methods: they involve constructing approximate posterior distributions over hidden states which factorize across dimension. However, unlike variational methods, they do not involve minimization of a Kullback-Liebler divergence criterion and avoid the fixed-point iterations or other optimization procedures which variational methods typically involve. Instead, the Graph Filter and Smoother exploit the factorial structure of the emission distribution as expressed through a factor graph and perform approximation through *localization* – discarding likelihood factors in a principled manner with respect to graph distance. Another contrast between the variational methods of Ghahramani and Jordan [1997] and the Graph Filter and Smoother is that the latter share the recursive-in-time structure of the forward-backward algorithm. The forward pass, which conducts the task of *filtering*, can therefore be used for prediction in online settings. Variational methods for FHMMs work in a batch setting, suitable for offline data analysis.

Boyen and Koller [1998, 1999], proposed and studied inference methods in Dynamic Bayesian Networks which involve recursively approximating belief-state distributions by the product of their marginals and then propagating the result to the next time step. Particle filtering algorithms in the same vein appeared in Ng et al. [2002], Brandao et al. [2006] and Besada-Portas et al. [2009]. Ground-breaking theoretical work of Rebeschini and Van Handel [2015] proved that similar algorithms can be used to conduct particle filtering efficiently in high-dimensions, using techniques based on the Dobrushin Comparison theorem [see for instance Georgii, 2011]. Subsequently, Rebeschini and van Handel [2014] refined their analysis through generalized

Dobrushin comparison theorems. Finke and Singh [2017] extended these ideas from particle filtering to particle smoothing and studied dimension-independence of the asymptotic Monte Carlo variance. However, the algorithms proposed by Rebeschini and Van Handel [2015] and Finke and Singh [2017] do not apply to FHMMs, indeed they require observations that are conditionally independent across dimension given the signal.

3.1.1 Contributions

This chapter introduces Graph Filter-Smoother, which provides approximate filtering and smoothing distributions for high-dimensional discrete state-space FHMMs and avoids the exponential-in-dimension computational cost of the forward-backward algorithm. Remark that Graph Filter-Smoother does not use any particle filter /smoother, but it generates approximations of the filtering and smoothing distribution in finite FHMMs. The main advantages of using Graph Filter-Smoother are summarized in the following.

- It is composed by a simple forward recursion approximating each filtering distribution with a product of approximated marginal distributions, derived from a localized emission distribution, see subsection 3.3.1, and a backward recursion which does not require any further approximation.

- The computational cost of the proposed method does not depend exponentially on the overall dimension, it varies according to the form of the emission distribution and it can be controlled through some tuning parameters.

- The local total variation distance between the derived approximations and the filtering /smoothing distributions is bounded above. The bounds do not degrade with the dimension and they can be controlled with tuning parameters.

- Experimentally, it is significantly cheaper than running the low-cost filtering-smoothing algorithm for FHMMs proposed by Ghahramani and Jordan [1997].

- In a synthetic data scenario, it is shown to better recover the ground truth parameters compared to the variational methods proposed by Ghahramani and Jordan [1997].

- It can be used in applications with straightforward spatial structures. In particular, it is applied to underground traffic modelling where flows in-out stations are influenced by the surrounding lines, which are further assumed to be independent of each other.

3.2 Preliminaries

Consider an HMM with unobserved Markov chain $(X_t)_{t\in\{0,\dots,T\}}$, observed process $(Y_t)_{t\in\{1,\dots,T\}}$ and a time horizon of length $T \in \mathbb{N}$. The state-space of $(X_t)_{t\in\{0,\dots,T\}}$ is of product form, hence

$X_t = (X_t^v)_{v\in V} \in \mathbb{X}^V$, where $\mathbb{X}$ and V be finite sets, and write $L := \mathbf{card}(\mathbb{X})$ and $M := \mathbf{card}(V)$. Each $(Y_t)_{t\in\{1,\dots,T\}}$ is valued in a set $\mathbb{Y}$ which could be a discrete set, $\mathbb{R}^d$ or some subset thereof.

Use $\lambda_0(x)$ for the probability mass function of X_0 (initial distribution), $p(x,z)$ for the transition probability of $(X_t)_{t\in\{0,\dots,T\}}$ from x to z (transition kernel), and $g(x,y)$ for the conditional probability mass or density function of Y_t given X_t (emission distribution).

For any $U \subseteq V$ and $x = (x^v)_{v\in V} \in \mathbb{X}^V$ the shorthand $x^U = (x^v)_{v\in U}$ is used. Similarly, for any probability mass function μ on $\mathbb{X}^V$ its marginal associated with U is denoted by μ^U. When $\mathscr{K}$ is any partition of the set V, μ is said to factorize with respect to $\mathscr{K}$ if:

$$\mu(x) = \prod_{K\in\mathscr{K}} \mu^K(x^K), \quad x \in \mathbb{X}^V,$$

and in this situation the shorthand is used:

$$\mu = \bigotimes_{K\in\mathscr{K}} \mu^K.$$

The total variation distance between probability mass functions on $\mathbb{X}^V$, say μ and ν, is denoted by:

$$\|\mu - \nu\| := \sup_{A\in\sigma(\mathbb{X}^V)} |\mu(A) - \nu(A)|,$$

with the obvious overloading of notation $\mu(A) = \sum_{x\in A}\mu(x)$ and where $\sigma(\mathbb{X}^V)$ is the power set of $\mathbb{X}^V$. When working on the marginals of μ, ν it will be convenient to denote the local total variation (LTV) distance associated with $U \subseteq V$,

$$\|\mu - \nu\|_U := \sup_{A\in\sigma(\mathbb{X}^U)} |\mu^U(A) - \nu^U(A)|.$$

Further assume that $(X_t)_{t\in\{0,\dots,T\}}$, $(Y_t)_{t\in\{1,\dots,T\}}$ is an FHMM as in definition 2.2, and so:

$$p(x,z) = \prod_{v\in V} p^v(x^v, z^v), \quad x, z \in \mathbb{X}^V \tag{3.1}$$

where each $p^v(x^v, z^v)$ is a transition probability on $\mathbb{X}$.

Consider now a factorial structure of the likelihood function $x \mapsto g(x,y)$. Let F be a finite set and let $\mathscr{G} = (V, F, E)$ be a factor graph associated with $x \mapsto g(x,y)$, that is a bi-partite graph with vertex sets V, F and edge set E such that $g(x,y)$ can be written in terms of factors:

$$g(x,y) = \prod_{f\in F} g^f(x^{N(f)}, y), \quad x \in \mathbb{X}^V, y \in \mathbb{Y}, \tag{3.2}$$

where $N(\cdot)$ is the neighbourhood function,

$$N(w) := \{w' \in V \cup F : (w, w') \in E\}, \quad w \in V \cup F. \tag{3.3}$$

In applications each observation y will typically be multivariate and each likelihood factor $g^f(x^{N(f)}, y)$ may depend on y only through some subset of its constituent variates, but the details

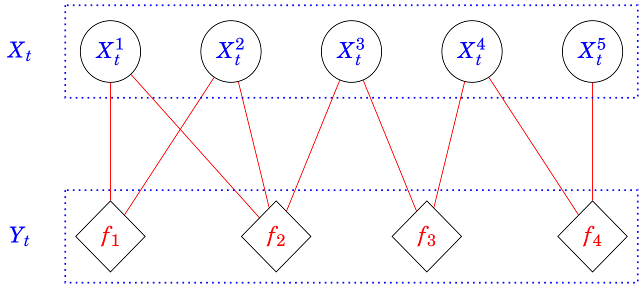


Figure 3.1: An example of factor graph for a fixed time step t of an FHMM where $V = \{1,2,3,4,5\}$ and $F = \{f_1, f_2, f_3, f_4\}$. So $g(x,y) = g^1((x^1,x^2),y)g^2((x^1,x^2,x^3),y)g^3((x^3,x^4),y)g^4((x^4,x^5),y)$

will be model specific, and they are not introduced at this stage. Remark that equations (3.1) and (3.2) are the mathematical formalization of "local structures" , indeed equation (3.1) says that each component of the hidden variable is evolving independently, while equation (3.2) stands that the emission distribution can be decomposed in pieces that can be built locally.

The filtering and smoothing distributions of the FHMM described in this section can be computed by the forward-backward algorithm given that the state-space is finite and discrete. The computational cost of doing so is $\mathcal{O}(TL^{2M})$ and it can be reduced to $\mathcal{O}(TML^{M+1})$ if the low-cost forward-backward algorithm by Ghahramani and Jordan [1997] is used. However, the exponential in the dimension term cannot be avoided, making the computation of the filtering and smoothing distributions unfeasible.

3.3 Approximate filtering and smoothing

To introduce the approximate filtering and smoothing techniques – called the Graph Filter and Graph Smoother – consider the filtering and smoothing recursions, (2.4) and (2.6), and fix any partition $\mathcal{K}$ of V. Suppose that one has already obtained an approximation to π_{t-1}, call it $\tilde{\pi}_{t-1}$, which factorizes with respect to $\mathcal{K}$. Then due to (3.1), $\mathsf{P}\tilde{\pi}_{t-1}$, also factorizes with respect to $\mathcal{K}$. However $\mathsf{C}_t\mathsf{P}\tilde{\pi}_{t-1}$ does not factorize with respect to $\mathcal{K}$ in general. Section 3.3.1 defines an approximation to the Bayes update operator C_t, denoted $\tilde{\mathsf{C}}_t^m$, where m is a parameter, such that $\tilde{\pi}_t := \tilde{\mathsf{C}}_t^m \mathsf{P}\tilde{\pi}_{t-1}$ does factorize with respect $\mathcal{K}$.

Once $(\tilde{\pi}_t)_{t\in\{0,\dots,T\}}$ have been computed in this manner, a sequence of approximate smoothing distributions $(\tilde{\pi}_{t|T})_{t\in\{0,\dots,T\}}$ will be obtained by setting $\tilde{\pi}_{T|T} := \tilde{\pi}_T$ and then $\tilde{\pi}_{t-1|T} := \mathsf{R}_{\tilde{\pi}_t}\tilde{\pi}_{t|T}$, for $t = T, T-1, \dots, 0$. Due to (3.1) and the fact that $(\tilde{\pi}_t)_{t\in\{0,\dots,T\}}$ each factorizes with respect to $\mathcal{K}$, it follows that $(\tilde{\pi}_{t|T})_{t\in\{0,\dots,T\}}$ also factorizes with respect to $\mathcal{K}$.

The key ingredient in all of this is finding a way to approximate the action of the Bayes update operator C_t in an accurate but computationally inexpensive manner. The next objective is to introduce the details of how to do so.

3.3.1 Approximate Bayes updates via localization and factorization

Let $d : (V \cup F)^2 \to \mathbb{R}_+$ be the graph distance on $\mathscr{G}$, meaning that $d(w,w')$ is the number of edges in a shortest path between w and w'. Augmenting the definition of the neighborhood function (3.3), define, for any $J \subseteq V$,

$$N_v^r(J) := \{v' \in V \text{ such that } \exists v \in J \text{ with } d(v,v') \leq 2r+2\},$$

$$N_f^r(J) := \{f \in F \text{ such that } \exists v \in J \text{ with } d(v,f) \leq 2r+1\}.$$

Then for a given probability mass function μ on $\mathbb{X}^V$, a partition of V denoted $\mathscr{K}$ and $m \geq 0$ define:

$$\tilde{\mathsf{C}}_t^{m,K}\mu(x^K) := \frac{\sum_{z\in\mathbb{X}^V : z^K = x^K} \prod_{f\in N_f^m(K)} g^f(z^{N(f)}, y_t)\mu(z)}{\sum_{z\in\mathbb{X}^V} \prod_{f\in N_f^m(K)} g^f(z^{N(f)}, y_t)\mu(z)}, \quad x^K \in \mathbb{X}^K, K \in \mathscr{K}, \tag{3.4}$$

$$\tilde{\mathsf{C}}_t^{m}\mu := \bigotimes_{K\in\mathscr{K}} \tilde{\mathsf{C}}_t^{m,K}\mu. \tag{3.5}$$

Note the dependence of $\tilde{\mathsf{C}}_t^m$ on $\mathscr{K}$ is not shown in the notation.

To see the motivation for (3.4)-(3.5), observe from the definition of the Bayes update operator (2.5) and factorial likelihood function (3.2) that the marginal distribution of $\mathsf{C}_t\mu$ associated with some $K \in \mathscr{K}$ is given by:

$$(\mathsf{C}_t\mu)^K(x^K) := \frac{\sum_{z\in\mathbb{X}^V : z^K = x^K} \prod_{f\in F} g^f(z^{N(f)}, y_t)\mu(z)}{\sum_{z\in\mathbb{X}^V} \prod_{f\in F} g^f(z^{N(f)}, y_t)\mu(z)}, \quad x^K \in \mathbb{X}^K. \tag{3.6}$$

The definition (3.4)-(3.5) thus embodies two ideas: *localization*, in that $\tilde{\mathsf{C}}_t^{m,K}\mu$ is an approximation to the exact marginal $(\mathsf{C}_t\mu)^K$ obtained by replacing the likelihood function $\prod_{f\in F} g^f(z^{N(f)}, y_t)$ in (3.6) by the "local-to-K" product $\prod_{f\in N_f^m(K)} g^f(z^{N(f)}, y_t)$; and *factorization*, in that $\tilde{\mathsf{C}}_t^m\mu$ factorizes with respect to $\mathscr{K}$ by construction. Figure 3.2 illustrates sub-graphs of $\mathscr{G}$ associated with each of the neighborhoods $N_f^m(K)$, $K \in \mathscr{K}$.

It is important to note that the factorization idea alone is not enough: computing the marginal distribution $(\mathsf{C}_t\mu)^K$ has a cost which is exponential in M in general for likelihoods of the form (3.2), even when μ factorizes with respect to $\mathscr{K}$. So taking $\bigotimes_{K\in\mathscr{K}}(\mathsf{C}_t\mu)^K$ as an approximation to $\mathsf{C}_t\mu$ would offer no computational advantage. This distinguishes the setup in this chapter from the one in [Rebeschini and Van Handel, 2015, Finke and Singh, 2017], discussed in section 3.1, and it is the reason motivating the introduction of localization through the parameter m.

More detailed consideration of the complexity of computing $\tilde{\mathsf{C}}_t^m\mu$ is given after proposition 3.1, which quantifies the approximation error associated with $\tilde{\mathsf{C}}_t^m$ and is one of the building blocks in the overall analysis of the approximate filtering and smoothing method.

In order to state proposition 3.1 some further definitions are needed. Firstly, introduce the

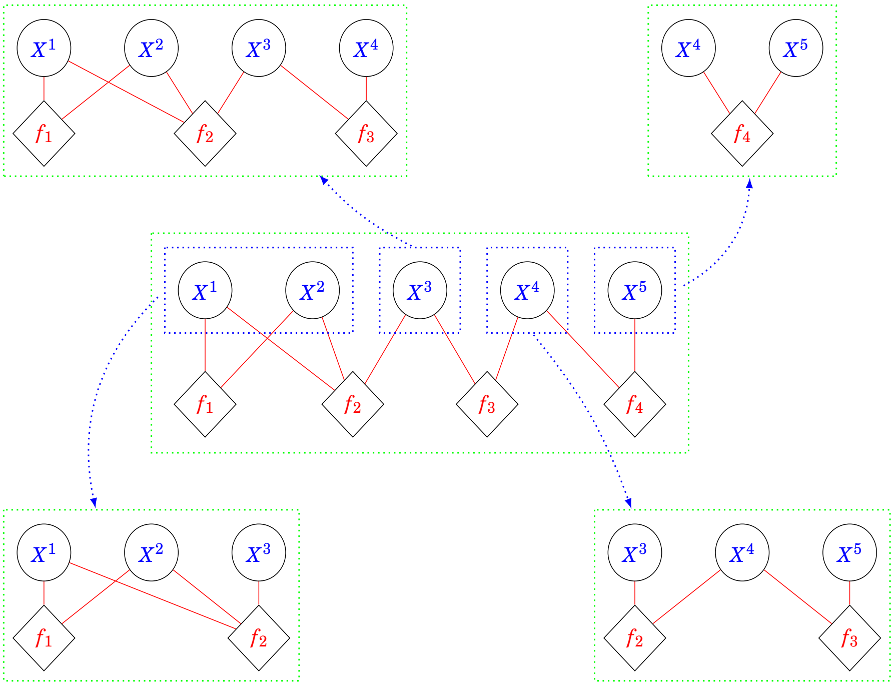


Figure 3.2: An example of sub-graphs associated with the neighbourhoods $N_f^m(K)$, $K \in \mathscr{K}$, when $V = \{1,2,3,4,5\}$, $F = \{f_1, f_2, f_3, f_4\}$, $\mathscr{K} = \{\{1,2\},\{3\},\{4\},\{5\}\}$ and $m = 0$.

following attributes of the factor graph $\mathscr{G}$.

$$d(J,J') := \min_{w\in J}\min_{w'\in J'} d(w,w'), \quad J,J' \subseteq V \cup F, \tag{3.7}$$

$$n_K := \frac{1}{2}\max_{v\in V} d(K,v), \quad K \in \mathscr{K},$$

$$n := \max_{K\in\mathscr{K}} n_K,$$

$$\Upsilon := \max_{v\in V} \mathbf{card}(N(v)),$$

$$\Upsilon^{(2)} := \max_{v\in V} \mathbf{card}(N_v^0(v)),$$

$$\tilde{\Upsilon} := \max_{v,v'\in V} \mathbf{card}(N(v)\cap N(v')). \tag{3.8}$$

Note the dependence of n on $\mathscr{K}$ is not shown in the notation. The interpretation of the above attributes of the factor graph is simple:

- $d(J,J')$ is the minimal distance between the elements of two sets in the factor graph;
- n_K is the maximal distance between elements in K and outside K;
- n is the maximal n_K associated to the partition $\mathscr{K}$;
- Υ is the maximal number of factors in F connected to a component in V;
- $\Upsilon^{(2)}$ is the maximal number of components in V which are distant 2 from a component in V;

- $\tilde{\Upsilon}$ is the maximal number of common factors in F connected to two components in V;

Secondly, given a probability mass function μ on $\mathbb{X}^V$ and a random variable $X \sim \mu$, denote by μ_x^v the conditional distribution of X^v given $\{X^{V\setminus v} = x^{V\setminus v}\}$, and define

$$C^{\mu}_{v,v'} := \frac{1}{2} \sup_{x,z\in\mathbb{X}^V : x^{V\setminus v'} = z^{V\setminus v'}} \|\mu_x^v - \mu_z^v\|, \quad v,v' \in V,$$
$$\mathrm{Corr}(\mu,\beta) := \max_{v\in V} \sum_{v'\in V} e^{\beta d(v,v')} C^{\mu}_{v,v'},$$

where $\beta > 0$ is a given constant.

Proposition 3.1. *Fix any partition $\mathscr{K}$ of V and any $t \in \{1,\dots,T\}$. Suppose there exists $\kappa \in (0,1)$ such that:*

$$\kappa \le g^f\left(x^{N(f)}, y_t\right) \le \frac{1}{\kappa}, \quad \forall x \in \mathbb{X}^V, f \in F. \tag{3.9}$$

Assume that for a given probability mass function μ on $\mathbb{X}^V$ there exists $\beta > 0$ such that:

$$2\kappa^{-2\Upsilon}\mathrm{Corr}(\mu,\beta) + e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right) \le \frac{1}{2}. \tag{3.10}$$

Then for any $K \in \mathscr{K}$, $J \subseteq K$ and $m \in \{0,\dots,n\}$,

$$\left\|\mathsf{C}_t\mu - \tilde{\mathsf{C}}_t^m\mu\right\|_J \le 4e^{-\beta}\left(1-\kappa^{b(m,\mathscr{K})}\right)\mathbf{card}(J)e^{-\beta m}, \tag{3.11}$$

where $b(m,\mathscr{K}) := 2\max_{K\in\mathscr{K}} \max_{v\notin N_v^{m-1}(K)}\{\mathbf{card}(N(v))\}$, with the convention that the maximum over an empty set is zero.

The proof of proposition 3.1 is given in section 4.2 of chapter 4. The term $\mathrm{Corr}(\mu,\beta)$ quantifies the strength of dependence across the coordinates of $X = (X^v)_{v\in V} \sim \mu$. A probability distribution μ satisfies the decay of correlation property if it exists a β such that $\mathrm{Corr}(\mu,\beta)$ is bounded above [Rebeschini and Van Handel, 2015]. The hypothesis (3.10) places a combined constraint on this dependence, the constant κ which in (3.9) controls the oscillation of the likelihood function factors $g^f(x^{N(f)}, y_t)$, and the graph attributes Υ, $\Upsilon^{(2)}$ and $\tilde{\Upsilon}$. In particular, if Υ, $\Upsilon^{(2)}$ and $\tilde{\Upsilon}$ are particularly small then the hypothesis (3.10) is likely to be satisfied, meaning that there is a preference on factor graphs with a few number of connections (see the meaning of Υ, $\Upsilon^{(2)}$ and $\tilde{\Upsilon}$ after equation (3.8)), where the local structures are more evident. This preference on sparse factor graphs is also reflected in the bound, indeed $b(m,\mathscr{K})$ is small if the elements of V have few connections in the factor graph.

Turning to the bound (3.11), one can examine its dependence on the partition $\mathscr{K}$ and the parameter m. The quantity $b(m,\mathscr{K})$ is non-increasing with m. In practice, $b(m,\mathscr{K})$ will often be decreasing with m, and is always zero when $m = n$ since then $N_v^{m-1}(K)$ is V. Also $b(m,\mathscr{K})$ will often decrease as $\mathscr{K}$ becomes coarser and in the extreme case of the trivial partition $\mathscr{K} = \{V\}$, the constant $b(m,\mathscr{K})$ is always zero, because $N_v^{m-1}(K)$ is again V, and so $v \notin V$ is equivalent to

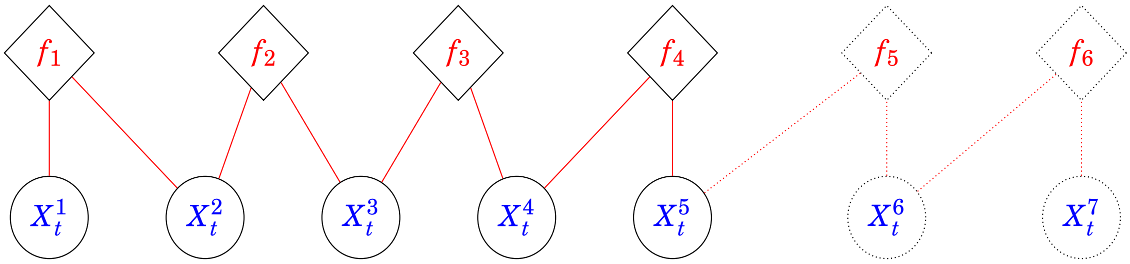


Figure 3.3: Solid lines indicate a chain factor graph with 4 likelihood factors and $V = \{1,2,3,4,5\}$, hence $M = 5$. Dashed lines indicate extension to $M = 6,7$ by adding f_5 and $X^{(6)}$ then f_6 and $X^{(7)}$.

$v \in \emptyset$. Combined with the $e^{-\beta m}$ term, this means $\left\|\mathsf{C}_t\mu - \tilde{\mathsf{C}}_t^m\mu\right\|_J$ can be made small by choosing the partition $\mathcal{K}$ to be suitably coarse and m to be suitably large.

It is important to note that $\mathrm{Corr}(\mu,\beta)$, κ and Υ appearing in the hypotheses (3.9) and (3.10), and the quantities on the right hand side of (3.11) do not necessarily have any dependence on M, where recall that $M := \mathrm{card}(V)$. For instance, when $\mu = \otimes_{v\in V}\mu^v$ then $\mathrm{Corr}(\mu,\beta) = 0$ and one can easily construct families of FHMMs of increasing dimension in which κ, Υ, $\Upsilon^{(2)}$, $\tilde{\Upsilon}$, $b(m,\mathcal{K})$ are independent of M: consider the simple case where the factor graph is a chain as shown in figure 3.3, and the dimension of the model is increased by adding f_5 and $X^{(6)}$ then f_6 and $X^{(7)}$ as shown by the dashed lines. In this situation, for any $v \in V$ the cardinality of $N(v)$ and $N_v^0(v)$ remain unchanged as the dimension of the model increases.

Algorithm 3 Approximate Bayes update

```
Require: $\mathcal{K}, \{N_f^m(K)\}_{K\in\mathcal{K}}, \{N_v^m(K)\}_{K\in\mathcal{K}}, \{\mu^K\}_{K\in\mathcal{K}}, \{g^f(\cdot,y)\}_{f\in F}$
for $K \in \mathcal{K}$ do
    $\hat{K} \leftarrow \{K' \in \mathcal{K} : K' \cap N_v^m(K) \neq \emptyset\}$
    for $x \in \mathbb{X}^{\hat{K}}$ do
        $\hat{\mu}(x) \leftarrow \prod_{K'\in\hat{K}} \mu^{K'}(x^{K'})$
        for $f \in N_f^m(K)$ do
            $\hat{\mu}(x) \leftarrow \hat{\mu}(x) \cdot g^f\left(x^{N(f)}, y\right)$
    Normalize $\hat{\mu}$ to a probability mass function on $\mathbb{X}^{\hat{K}}$
    Marginalize out components $\hat{K} \setminus K$: $\tilde{\mu}^K \leftarrow \hat{\mu}^K$
return $(\tilde{\mu}^K)_{K\in\mathcal{K}}$
```

Algorithm 3 shows the steps involved in computing $\tilde{\mathsf{C}}_t^m\mu$ in the case that μ factorizes with respect to $\mathcal{K}$. To simplify considerations of the computational cost of algorithm 3, suppose that for each $K \in \mathcal{K}$ there exists a collection of elements in $\mathcal{K}$ that is a partition of $N_v^m(K)$. This is a typical feature of regular graphs such as lattices. In this case, the complexity of algorithm 3 is readily found to be:

$$\mathcal{O}\left(\mathbf{card}(\mathcal{K})\max_{K\in\mathcal{K}}\mathbf{card}\left(N_f^m(K)\right)L^{\max_{K\in\mathcal{K}}\mathbf{card}(N_v^m(K))}\right).$$

The elements of the computational cost are derived as follows:

- $\mathbf{card}(\mathscr{K})$ is obtained because each correction is repeated on all the elements of the partition;
- $\max_{K\in\mathscr{K}} \mathbf{card}\left(N_f^m(K)\right)$ is derived from including all the desired factors in the computation;
- $L^{\max_{K\in\mathscr{K}} \mathbf{card}(N_v^m(K))}$ is the cost of including all the desired factors in the approximate correction.

Crucially the exponent of L in this cost, which is proportional to $\max_{K\in\mathscr{K}} \mathbf{card}(N_v^m(K))$, does not necessarily grow with the overall dimension M, see again the chain example in figure 3.3. This suggests that the overall cost of filtering and smoothing method built around the approximate Bayes update operator $\tilde{\mathsf{C}}_t^m$ may avoid the exponential-in-M factor in the cost $\mathscr{O}(TML^{M+1})$ of exact filtering and smoothing for FHMMs.

3.3.2 Graph Filter

As an approximation to the operator F_t introduced in subsection 2.1.2 define:

$$\tilde{\mathsf{F}}_t^m := \tilde{\mathsf{C}}_t^m \mathsf{P},$$

where the dependence of $\tilde{\mathsf{F}}_t^m$ on $\mathscr{K}$, inherited from $\tilde{\mathsf{C}}_t^m$, is not shown in the notation. The approximate filtering distributions are then defined by the recursion:

$$\tilde{\pi}_0 := \lambda_0, \qquad \tilde{\pi}_t := \tilde{\mathsf{F}}_t^m \tilde{\pi}_{t-1}, \quad t \in \{1,\dots,T\} \tag{3.12}$$

theorem 3.1 builds from proposition 3.1 and quantifies the approximation error associated with $(\tilde{\pi}_t)_{t\in\{0,\dots,T\}}$. Further to (3.7)-(3.8), in order to state theorem 3.1, some additional quantities and definitions must be introduce. For $J \subseteq V$,

$$\begin{aligned} \widetilde{J} &:= \{v \in J : \forall f \in N(v), N(f) \subseteq J\}, \\ \partial J &:= J \setminus \widetilde{J}, \\ \partial N(J) &:= \{f \in N(J) : N(f) \cap V \setminus J \neq \emptyset\}. \end{aligned} \tag{3.13}$$

Moreover, given a probability mass function μ on $\mathbb{X}^V$ and the transition kernel P on $\mathbb{X}^V$ and two random variables such that $X \sim \mu$ and $Z|X \sim p(X,\cdot)$, denote by $\mu^v_{x,z}$ the conditional distribution of X^v given $\{X^{V\setminus v} = X^{V\setminus v}, Z = z\}$, and define

$$\begin{aligned} \tilde{C}^{\mu}_{v,v'} &:= \frac{1}{2} \sup_{z\in\mathbb{X}^V} \sup_{\substack{x,\hat{x}\in\mathbb{X}^V: \\ x^{V\setminus v'} = \hat{x}^{V\setminus v'}}} \left\| \mu^v_{x,z} - \mu^v_{\hat{x},z} \right\|, \quad v,v' \in V \\ \widetilde{\mathrm{Corr}}(\mu,\beta) &:= \max_{v\in V} \sum_{v'\in V} e^{\beta d(v,v')} C^{\mu}_{v,v'}, \end{aligned}$$

where $\beta > 0$.

Remark 3.1. *If the components of X are independent, i.e. $\mu = \otimes_{v\in V} \mu^v$, then $\tilde{C}^{\mu}_{v,v'} = 0$ for any $v \neq v'$ and so $\widetilde{\mathrm{Corr}}(\mu,\beta) = 0$ for any $\beta > 0$.*

Theorem 3.1. *Fix any collection of observations* $\{y_1,\dots,y_T\}$ *and any partition* $\mathscr{K}$ *of* V. *There exists a region* $\mathscr{R}_0 \subseteq (0,1)^3$ *depending only on* $\tilde{\Upsilon},\Upsilon$ *and* $\Upsilon^{(2)}$, *such that if, for given* $(\epsilon_-,\epsilon_+,\kappa) \in \mathscr{R}_0$,

$$\epsilon_- \le p^v(x^v,z^v) \le \epsilon_+, \quad \text{and} \quad \kappa \le g^f\left(x^{N(f)},y_t\right) \le \frac{1}{\kappa},$$

for all $x,z \in \mathbb{X}^V, v \in V, f \in F, t \in \{1,\dots,T\}$, *then for* $\beta > 0$ *small enough depending only on* $\tilde{\Upsilon},\Upsilon,\Upsilon^{(2)},\epsilon_-,\epsilon_+,$ κ *for any* λ_0 *satisfying:*

$$\widetilde{\mathrm{Corr}}(\lambda_0,\beta) \le 2e^{-\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right) + 2e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)$$

and for any $K \in \mathscr{K}$, $J \subseteq K$ *and* $m \in \{0,\dots,n\}$*:*

$$\|\pi_t - \tilde{\pi}_t\|_J \le \alpha_1(\beta)\left(1-\kappa^{a(\mathscr{K})}\right)\mathbf{card}(J) + \gamma_1(\beta)\left(1-\kappa^{b(\mathscr{K},m)}\right)\mathbf{card}(J)e^{-\beta m}, \quad \forall t \in \{1,\dots,T\},$$

where $\pi_t,\tilde{\pi}_t$ *are given by* (2.4) *and* (3.12) *with initial condition* λ_0; $\alpha_1(\beta),\gamma_1(\beta)$ *are constants depending only on* β, *and*

$$a(\mathscr{K}) := 2\max_{K\in\mathscr{K}}\max_{v\in\partial K}\mathbf{card}(N(v)\cap\partial N(K)),$$
$$b(m,\mathscr{K}) := 2\max_{K\in\mathscr{K}}\max_{v\notin N_v^{m-1}(K)}\mathbf{card}(N(v)),$$

with the convention that the maximum over an empty set is zero.

The proof of theorem 3.1 is in section 4.2 of chapter 4. Explicit expressions for $\mathscr{R}_0$, β, $\alpha_1(\beta)$ and $\gamma_1(\beta)$ are given in the proof of the theorem and its supporting results, see (4.9) for $\mathscr{R}_0$ and β, and (4.10) for $\alpha_1(\beta)$ and $\gamma_1(\beta)$. Note that the assumption on $\widetilde{\mathrm{Corr}}(\lambda_0,\beta)$ plays the same role of (3.10) in proposition 3.1 and ensures that the initial local dependencies are not too strong.

The second term on the right hand side of the bound on $\|\pi_t - \tilde{\pi}_t\|_J$ given in theorem 3.1 is, up to a numerical constant, equal to the upper bound obtained in proposition 3.1, see discussion there of its dependence on m and $\mathscr{K}$. The first term on the right hand side of the bound on $\|\pi_t - \tilde{\pi}_t\|_J$ depends on the neighborhood structure of the factor graph $\mathscr{G}$. Loosely speaking, the constant $a(\mathscr{K})$ is small when the graph is sparsely connected and the partition is coarse, and in the extreme case of the trivial partition $\mathscr{K} = \{V\}$, the constant $a(\mathscr{K})$ is zero because, in the notation of (3.13), $\tilde{V} = V$ and so ∂V is empty. The quantities in the hypotheses and bound of theorem 3.1 exhibit the same dimension-free qualities as discussed after proposition 3.1.

Implementation of the approximate filtering method is shown in algorithm 4, which is referred to from now on as the Graph Filter. Noting that the complexity of computing $\tilde{\pi}^K \leftarrow (\mathsf{P}\mu)^K$, which is $\mathscr{O}\left(TL^{2\max_{K\in\mathscr{K}}\mathbf{card}(K)}\right)$, is dominated by the cost of algorithm 3, with an additional 2 at the exponent of L, the overall complexity of algorithm 4 is then:

$$\mathscr{O}\left(T\mathbf{card}(\mathscr{K})\max_{K\in\mathscr{K}}\mathbf{card}\left(N_f^m(K)\right)L^{2\max_{K\in\mathscr{K}}\mathbf{card}(N_v^m(K))}\right).$$

The elements of the computational cost are derived as follows:

Algorithm 4 Graph Filter

```
Require: $\mathcal{K}, \{N_f^m(K)\}_{K\in\mathcal{K}}, \{N_v^m(K)\}_{K\in\mathcal{K}}, \{\lambda_0^K\}_{K\in\mathcal{K}}, \{p^v(\cdot,\cdot)\}_{v\in V}, \{g^f(\cdot,\cdot)\}_{f\in F}, \{y_t\}_{t=\{1,\dots,T\}}$
for $K \in \mathcal{K}$ do
    $\tilde{\pi}_0^K \leftarrow \lambda_0^K$
    Compute $p^K(\cdot,\cdot) := \prod_{v\in K} p^v(\cdot,\cdot)$
for $t \in \{1,\dots,T\}$ do
    for $K \in \mathcal{K}$ do
        for $z^K \in \mathbb{X}^K$ do
            $\hat{\pi}^K(z^K) \leftarrow \sum_{x^K\in\mathbb{X}^K} p^K(x^K,z^K)\cdot\tilde{\pi}_{t-1}^K(x^K)$
    $(\tilde{\pi}_t^K)_{K\in\mathcal{K}} \leftarrow$ algorithm 3$\left(\mathcal{K}, \{N_f^m(K)\}_{K\in\mathcal{K}}, \{N_v^m(K)\}_{K\in\mathcal{K}}, \{\hat{\pi}^K\}_{K\in\mathcal{K}}, \{g^f(\cdot,y)\}_{f\in F}\right)$
return $\{(\tilde{\pi}_t^K)_{K\in\mathcal{K}}\}_{t=\{0,\dots,T\}}$
```

- T is derived from cycling over time;
- $\mathbf{card}(\mathcal{K})$ is obtained because each prediction and correction is repeated on all the elements of the partition;
- $\max_{K\in\mathcal{K}} \mathbf{card}\left(N_f^m(K)\right)$ is derived from algorithm 3;
- $L^{2\max_{K\in\mathcal{K}}\mathbf{card}(K)}$ is the cost of the matrix-vector operation when computing prediction over all $v \in K$ and $K \in \mathcal{K}$ which is then upper bounded by the cost of algorithm 3, given that $L^{\max_{K\in\mathcal{K}}\mathbf{card}(K)} \leq L^{\max_{K\in\mathcal{K}}\mathbf{card}(N_v^m(K))}$.

For the chain graph example in figure 3.3 with $\mathcal{K} = \{\{1\},\{2\},\dots\}$, the complexity is:

$$\mathcal{O}\left(T\,\mathbf{card}(\mathcal{K})\min\{2(m+1), M-1\}L^{2\min\{2(m+1)+1,M\}}\right).$$

Thus the exponent of L has the dimension-free property of becoming independent of M depending on m when M is large enough.

3.3.3 Graph Smoother

The approximate smoothing distributions are defined by simply substituting the approximate filtering distributions into recursion (2.6):

$$\tilde{\pi}_{T|T} := \tilde{\pi}_T, \qquad \tilde{\pi}_{t|T} := \mathsf{R}_{\tilde{\pi}_t}\tilde{\pi}_{t+1|T}, \qquad t \in \{T-1,\dots,0\}. \tag{3.14}$$

Theorem 3.2. *Fix any collection of observations* $\{y_1,\dots,y_T\}$ *and any partition* $\mathcal{K}$ *of* V. *There exists a region* $\tilde{\mathscr{R}}_0 \subseteq (0,1)^3$ *depending only on* $\tilde{\Upsilon}, \Upsilon$ *and* $\Upsilon^{(2)}$, *such that if, for given* $(\epsilon_-,\epsilon_+,\kappa) \in \tilde{\mathscr{R}}_0$,

$$\epsilon_- \leq p^v(x^v,z^v) \leq \epsilon_+, \quad \textit{and} \quad \kappa \leq g^f\left(x^{N(f)},y_t\right) \leq \frac{1}{\kappa},$$

Algorithm 5 Graph Smoother

```
Require: $\mathscr{K}, (p^v(\cdot,\cdot))_{v\in V}, (\tilde{\pi}_t^K)_{K\in\mathscr{K}, t=\{0,\dots,T\}}$
for $K \in \mathscr{K}$ do
    $\tilde{\pi}_{T|T}^K \leftarrow \tilde{\pi}_T^K$
    Compute $p^K(\cdot,\cdot) := \prod_{v\in K} p^v(\cdot,\cdot)$
for $t \in \{T-1,\dots,0\}$ do
    for $K \in \mathscr{K}$ do
        for $z^K \in \mathbb{X}^K$ do
            for $x^K \in \mathbb{X}^K$ do
                $\overleftarrow{p}^K(z^K, x^K) \leftarrow p^K(x^K, z^K)\tilde{\pi}_t^K(x^K)$
            Normalize $\overleftarrow{p}^K(z^K,\cdot)$ to a probability mass function on $\mathbb{X}^K$
        for $x^K \in \mathbb{X}^K$ do
            $\tilde{\pi}_{t|T}^K(x^K) \leftarrow \sum_{z^K\in\mathbb{X}^K} \overleftarrow{p}^K(z^K, x^K)\tilde{\pi}_{t+1|T}^K(z^K)$
return $((\tilde{\pi}_{t|T}^K)_{K\in\mathscr{K}})_{t=\{1,\dots,T\}}$
```

for all $x, z \in \mathbb{X}^V, v \in V, f \in F, t \in \{1,\dots,T\}$, then for $\beta > \log(2)$ small enough depending only on $\tilde{\Upsilon}, \Upsilon, \Upsilon^{(2)}, \epsilon_-, \epsilon_+$, κ and for any λ_0 satisfying satisfying:

$$\widetilde{\mathrm{Corr}}(\lambda_0, \beta) \le 2e^{-\beta}\left(1 - \frac{\epsilon_-}{\epsilon_+}\right) + 2e^{2\beta}\Upsilon^{(2)}\left(1 - \kappa^{2\tilde{\Upsilon}}\right),$$

and for $K \in \mathscr{K}$, $J \subseteq K$ and $m \in \{0,\dots,n\}$:

$$\left\|\tilde{\pi}_{t|T} - \pi_{t|T}\right\|_J \le \alpha_2(\beta,\epsilon_-,\epsilon_+)\left(1 - \kappa^{a(\mathscr{K})}\right)\mathbf{card}(J) + \gamma_2(\beta,\epsilon_-,\epsilon_+)\left(1 - \kappa^{b(\mathscr{K},m)}\right)\mathbf{card}(J)e^{-\beta m},$$

where $\pi_{t|T}, \tilde{\pi}_{t|T}$ are given by (2.6) *and* (3.14) *with initial condition λ_0, $\alpha_2(\beta,\epsilon_-,\epsilon_+)$ and $\gamma_2(\beta,\epsilon_-,\epsilon_+)$ are constants depending on ϵ_-, ϵ_+, β and*

$$a(\mathscr{K}) := 2\max_{K\in\mathscr{K}}\max_{v\in\partial K}\mathbf{card}(N(v)\cap\partial N(K)),$$
$$b(m,\mathscr{K}) := 2\max_{K\in\mathscr{K}}\max_{v\notin N_v^{m-1}(K)}\mathbf{card}(N(v)),$$

with the convention that the maximum over an empty set is zero.

The proof of theorem 3.2 is in section 4.3 of chapter 4. The only difference between the bound in this theorem and that in theorem 3.1 are the constants $\alpha_2(\beta,\epsilon_-,\epsilon_+)$ and $\gamma_2(\beta,\epsilon_-,\epsilon_+)$, explicit expressions can be deduced from the proof. The assumption on the initial distribution λ_0 is the same as in theorem 3.1.

The approximate smoothing method, shown in algorithm 5 has complexity

$$\mathcal{O}\left(T\mathbf{card}(\mathscr{K})L^{3\max_{K\in\mathscr{K}}\mathbf{card}(K)}\right),$$

where:

- T is derived from cycling over time;

- $\mathbf{card}(\mathscr{K})$ is derived from repeating the operation on all the elements of the partition;
- $L^{3\max_{K\in\mathscr{K}}\mathbf{card}(K)}$ is obtained from the computation of the reversed kernel and its application to the previous smoothing approximation.

Assuming $3\max_{K\in\mathscr{K}}\mathbf{card}(K)$ is smaller than $2\max_{K\in\mathscr{K}}\mathbf{card}(N_v^m(K))$, which is typically the case in practice, the overall complexity of algorithm 4 combined with algorithm 5 is:

$$\mathcal{O}\left(T\mathbf{card}(\mathscr{K})\max_{K\in\mathscr{K}}\mathbf{card}\left(N_f^m(K)\right)L^{2\max_{K\in\mathscr{K}}\mathbf{card}(N_v^m(K))}\right).$$

Discussion on m and $\mathscr{K}$ As already mentioned, the proposed algorithms are built upon two layers of approximation:

- factorization: the target distribution is approximated with a product of marginals;
- localization: each marginal is approximated with a localized version, which excludes “far away” factors.

The two layers are guided by two quantities: the partition $\mathscr{K}$ and the parameter m. m is associated to localization, instructing how many factors $f \in F$ to include in each approximation of the marginals. Obviously, a bigger m guarantees a better approximation, but it also implies a higher computational cost. $\mathscr{K}$ refers to factorization and determines the structure of the marginals to be computed. Again, a coarser $\mathscr{K}$ gives a better approximation, but it is more computationally intensive.

Choosing m and $\mathscr{K}$ is not straightforward, and several aspects must be considered. In the following we give a list of suggestions for the choices of m and $\mathscr{K}$ to contain the computational cost of the algorithms.

1. m and $\mathscr{K}$ should be chosen such that in any of the marginals approximation only a subset of F is used, otherwise, if all the factors are included, running the Graph Filter-Smoother is as expensive as the forward-backward algorithm.
2. $\mathscr{K}$ should look at the structure of the factor graph and try to include the least factors. For instance, in figure 3.2 the partition is chosen to be $\mathscr{K} = \{\{1,2\},\{3\},\{4\},\{5\}\}$, but another valid partition is $\mathscr{K}^\star = \{\{1,3\},\{2\},\{4\},\{5\}\}$. However, $\mathscr{K}$ is better than $\mathscr{K}^\star$ because for the same $m = 0$ the element $\{1,2\}$ in $\mathscr{K}$ requires less factors than the element $\{1,3\}$ in $\mathscr{K}^\star$ (similar argument for $\{3\}$ in $\mathscr{K}$ and $\{2\}$ in $\mathscr{K}^\star$).
3. m should be selected to avoid factors with high degree. Indeed, including too many “hub” factors could blow up the computational cost.

Remark that the above suggestions are focused on controlling the computational cost and they do not take into account the approximation quality. This latter aspect is difficult to evaluate and it is generally suggested to try multiple combinations of m and $\mathscr{K}$.

3.4 Numerical results

This section treats the experimental part. Subsection 3.4.1 describes a class of FHMM's with conditionally Gaussian observations used as a running example in Ghahramani and Jordan [1997] and which is used in the numerical experiments run in this thesis. The purpose of the first set of experiments, in section 3.4.1, is to illustrate the practical implications of the previous theoretical results, assessing the performance of the Graph Filter and Smoother methods against exact filtering and smoothing, both in terms of accuracy and computational speed. In subsection 3.4.1 the performance of EM algorithms for parameter estimation built around the Graph Smoother is compared with the variational approximations presented in Ghahramani and Jordan [1997]. Subsection 3.4.2 outlines a model of traffic flow on the London Underground and illustrate parameter estimation and prediction using the Graph Filter and Smoother.

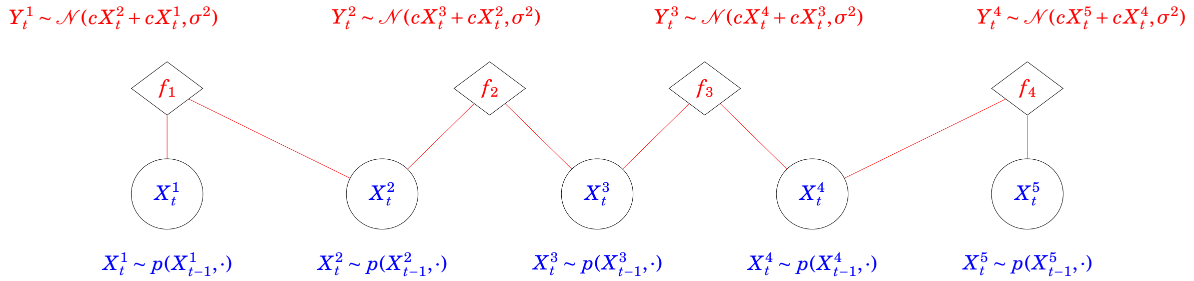
The experiments were run on the University of Bristol's BlueCrystal High-Performance Computing machine. Precisely, they used either one or two standard compute nodes each with 2 x 2.6GHz 8-core Intel E5-2670 (SandyBridge) chips and 4GB of RAM per core.

3.4.1 Synthetic data

With $\mathbb{X}$ a finite subset of $\mathbb{Z}$, $V = \{1,\ldots,M\}$ and $\mathbb{Y} = \mathbb{R}^{d_y}$, consider the Gaussian emission model from Ghahramani and Jordan [1997]:

$$g(x,y) = |\Sigma|^{-\frac{1}{2}}(2\pi)^{-\frac{d_y}{2}} \exp\left\{-\frac{1}{2}[y-a(x)]^T \Sigma^{-1}[y-a(x)]\right\},$$

where $a(x)$ is a vector whose entries may depend on x.



$Y_t^1 \sim \mathcal{N}(cX_t^2 + cX_t^1, \sigma^2)$
$Y_t^2 \sim \mathcal{N}(cX_t^3 + cX_t^2, \sigma^2)$
$Y_t^3 \sim \mathcal{N}(cX_t^4 + cX_t^3, \sigma^2)$
$Y_t^4 \sim \mathcal{N}(cX_t^5 + cX_t^4, \sigma^2)$
f_1
f_2
f_3
f_4
X_t^1
X_t^2
X_t^3
X_t^4
X_t^5
$X_t^1 \sim p(X_{t-1}^1, \cdot)$
$X_t^2 \sim p(X_{t-1}^2, \cdot)$
$X_t^3 \sim p(X_{t-1}^3, \cdot)$
$X_t^4 \sim p(X_{t-1}^4, \cdot)$
$X_t^5 \sim p(X_{t-1}^5, \cdot)$

Figure 3.4: Factor graph for the model in section 3.4.1 the case $F = \{f_1, f_2, f_3, f_4\}$ and $V = \{1,2,3,4,5\}$.

The case $d_y = M-1$ is considered along with the specific forms of Σ and $a(x)$:

$$\Sigma = \sigma^2 I \quad \text{and} \quad a(x) = \left(a^f(x)\right)_{f\in\{1\ldots M-1\}} \quad \text{with} \quad a^f(x) := c\left(x^f + x^{f+1}\right),$$

where $c > 0$ is a constant. Under these assumptions, $N(f) = \{f, f+1\}$ and

$$g^f(x^{N(f)}, y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{[y^f - c(x^f + x^{f+1})]^2}{2\sigma^2} \right\}.$$

The corresponding factor graph $\mathscr{G}$ is a chain, as already seen in figure 3.3, and it is illustrated in figure 3.4. The transition probabilities and initial probability mass function are assumed to have identical components across V,

$$\lambda_0(x) = \prod_{v \in V} \lambda_0^v(x^v), x \in \mathbb{X}^V \text{ and } \lambda_0^v = \hat{\mu}_0, \quad \forall v \in V;$$
$$p(x,z) = \prod_{v \in V} p^v(x^v, z^v), x, z \in \mathbb{X}^V \text{ and } p^v = \hat{p}, \quad \forall v \in V.$$

Throughout the experiments the partition $\mathscr{K}$ is:

$$\mathscr{K} = \{\{1\}, \dots, \{M\}\}.$$

Accuracy and speed performance for filtering and smoothing $\mathbb{X} = \{0,1\}$ and three data sets of length $T = 500$ are simulated from the model with parameters:

$$\hat{\mu}_0(x^v) = 1,\ x^v = 1, \forall v \in V, \quad \{\hat{p}(x^v, z^v)\}_{x^v, z^v \in \mathbb{X}} = \begin{pmatrix} 0.6 & 0.4 \\ 0.2 & 0.8 \end{pmatrix}, \quad c = 1, \quad \sigma^2 = 1. \tag{3.15}$$

First consider the execution time of the approximate filtering and smoothing method, i.e., the combination of algorithm 4 and algorithm 5, as a function of the parameters m and M. Figure 3.5 shows execution time as m and M vary. The execution time of exact filtering and smoothing using the algorithm of Ghahramani and Jordan [1997], henceforth "GJ" , is included for reference.

It is apparent from the top row of plots that with m fixed, the execution time of the Graph Filter and Smoother initially increases super-linearly with M up to some point which depends on m, and from then on it is linear in M. This is most visually evident for the large values of m and is consistent with the complexity of the combined Graph Filter and Smoother method discussed in subsection 3.3.3, which for the model considered here is:

$$\mathscr{O}\left(TM \min\{2(m+1), M-1\} L^{2\min\{2(m+1)+1, M\}} \right). \tag{3.16}$$

By contrast, the execution time of GJ increases exponentially with M, making its implementation extremely expensive in high-dimensional cases.

When M is fixed, it is clear from the bottom row of plots in figure 3.5 that the execution time of the Graph Filter and Smoother is super-linear in m up to some point which depends on M, and then is constant in m. Again this is consistent with (3.16). The phenomenon of the cost becoming constant in m arises because as m grows, eventually all factors are included in the products in $\tilde{\mathsf{C}}_t^{m,K}$, see (3.4).

The next set of experiments examine the accuracy. Recall two important characteristics of the bound of theorem 3.2: the bound does not depend on the overall dimension, M, and decays

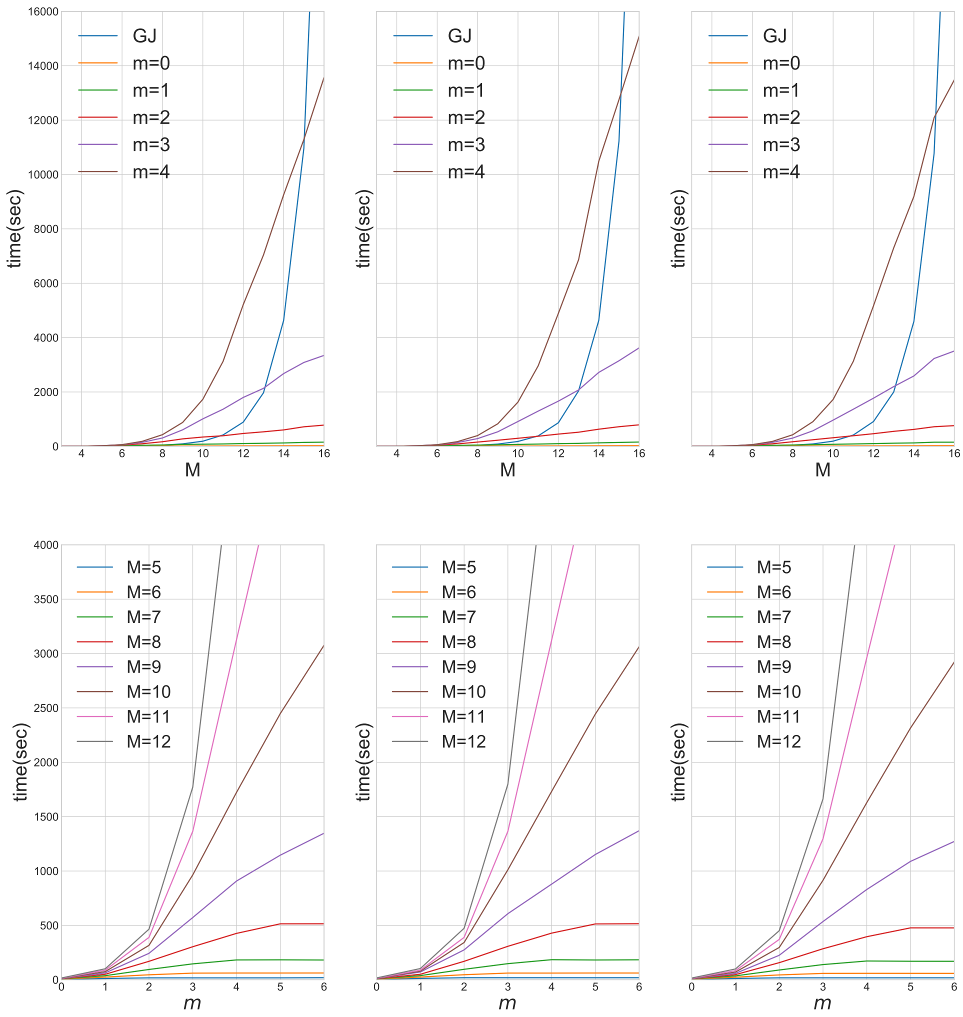


Figure 3.5: Execution time for the combined filtering and smoothing algorithms as a function of m and M. Each vertical pair of plots corresponds to one of three simulated data sets. GJ is the exact filtering and smoothing algorithm of Ghahramani and Jordan [1997].

exponentially with m. The region $\tilde{\mathscr{R}}_0$ in theorem 3.2 is non-empty, but for the specific parameter settings in (3.15) there does not exist $(\epsilon_-, \epsilon_+, \kappa) \in \tilde{\mathscr{R}}_0$ such that the assumptions of the theorem on p^v and g^f hold. Thus technically theorem 3.2 does not hold in this example. However, figure 3.6 and figure 3.7 encouragingly show that the LTV between the exact and approximate smoothing distributions exhibits the characteristics of not depending on the overall dimension, M, and decaying exponentially with m.

Comparison to variational inference within EM for parameter estimation Here the accuracy of parameter estimation using the Graph Smoother within an approximate EM algorithm is illustrated. The performance is compared with the approximate EM approach of

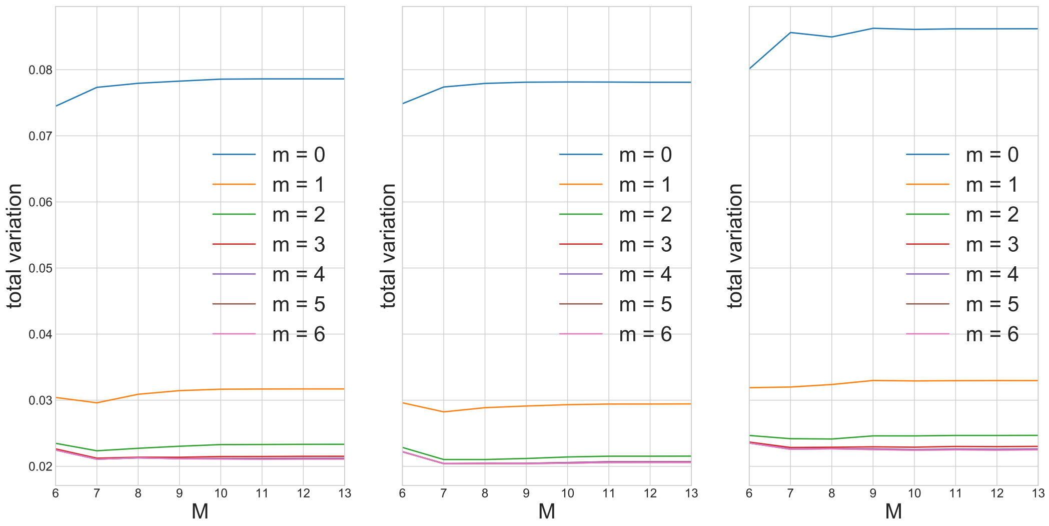


Figure 3.6: The LTV distance between the approximate and exact marginal smoothing distributions averaged over both the components $X_t^i, \dots, X_t^{i+4}$, with $i = 1, \dots, 8$, and the $T = 500$ time steps. With m fixed the average LTV is constant in M for M large enough. The three plots correspond to the three simulated data sets.

Ghahramani and Jordan [1997] in which variational approximations to the smoothing distributions are employed. For background on EM see Dempster et al. [1977] and Ghahramani and Jordan [1997] (section 3.1).

Ghahramani and Jordan [1997] (sections 3.4 and 3.5) describes two families of variational distributions for FHMM which can be used to compute the E-step in EM approximately: a "fully-factorized" scheme in which the variational distribution is chosen to statistically decouple all state variables, $(X_t^v)_{v \in V}$, $t = 0, \dots, T$, in the HMM, and a "structured" approximation, in which the variational distribution is Markovian in time but statistically decouples state-variables across V. The former is going to be called *completely decoupled*, while the latter is going to be named *spatially decoupled*.

The computational cost of computing the approximate smoothing distributions using either the completely decoupled or spatially decoupled schemes is:

$$\mathcal{O}(ITL^2M^2(M-1)^4), \tag{3.17}$$

where I is the number of iterations of the fixed-point equations needed to find the variational approximation. $I = 20$ was sufficient for convergence in the considered experiments, indeed Ghahramani and Jordan [1997] (page 254) suggest $2-10$ iterations is typically sufficient. Recall that for M large enough, (3.16) is exponential in m, but linear in M, while (3.17) scales no faster than M^6. Whether or not the variational approximations can be computed more quickly than the Graph Smoother is dependent on the model in question. The experiments did not show a

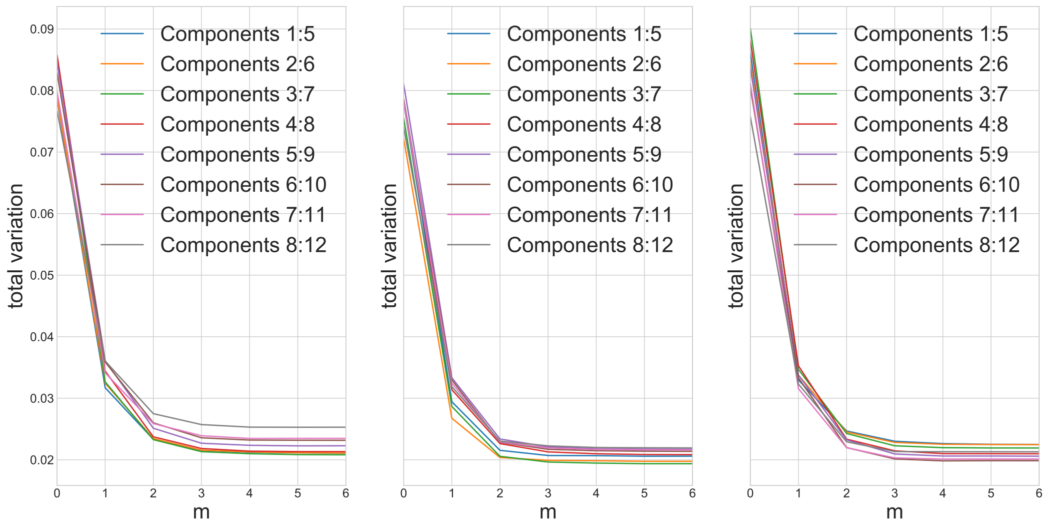


Figure 3.7: The LTV distance between the approximate and exact marginal smoothing distributions of certain components of $X_t^1, \ldots, X_t^M$ and the $T = 500$ time steps. The LTV decreases exponentially with m. The three plots correspond to the three simulated data sets.

substantial difference in speed.

Details of the EM updates using the Graph Smoother are given in the appendix A. The only difference between these updates and those using the variational approximations is in the E-step, where the expectation is simply taken with respect to the corresponding approximate smoothing distribution.

The model described in subsection 3.4.1 is considered, with $\mathbb{X} = \{0,1\}$ and $T = 200$, and generated a data set with true parameter values:

$$\hat{\mu}_0(x^v) = 1,\ x^v = 1, \forall v \in V, \quad \{\hat{p}(x^v, z^v)\}_{x^v, z^v \in \mathbb{X}} = \begin{pmatrix} 0.6 & 0.4 \\ 0.2 & 0.8 \end{pmatrix}, \quad c = 2 \quad \text{and} \quad \sigma^2 = 4.$$

The EM algorithms based on the Graph Smoother and the fully and spatially decoupled variational approximations were run for 20 different EM initializations. Tables 3.1 and 3.2 show the mean of the final estimates over the different initial conditions for $M = 3$ and $M = 10$, respectively. Graphical results are presented in figures 3.8 and 3.10 for $M = 3$ and figures 3.9 and 3.11 for $M = 10$.

In figure 3.8 the EM algorithms associated with the Graph Smoother converge to points closer to the true parameter values than those using the variational approximations. Using $m = 1$ rather than $m = 0$ in the former yields a slight increase in the accuracy. In figure 3.10 the results using the Graph Smoother are not more accurate in all cases, but the completely decoupled variational method generally performs badly. In figure 3.9 the results for the Graph Smoother are again more accurate. In figure 3.11 it is notable that the estimates of the transition probabilities are a little more accurate with $m = 0$ rather than $m = 1$ but substantially more accurate than with either of the variational schemes. Again the completely decoupled variational approximation performs poorly. These observations are numerically evident also from tables 3.1 and 3.2.

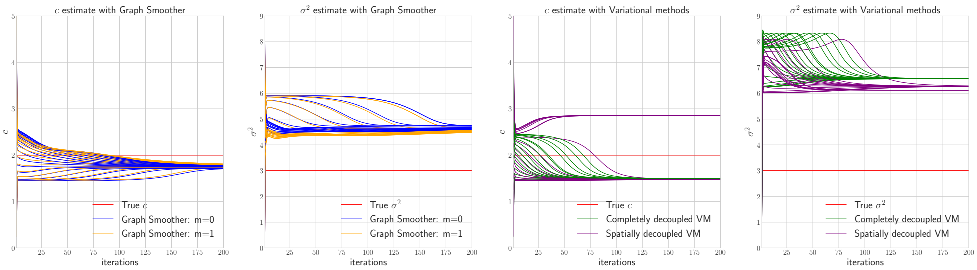


Figure 3.8: $M = 3$. Estimation of c and σ^2 using approximate EM based on the Graph Smoother and the completely and spatially decoupled variational approximations by Ghahramani and Jordan [1997]. Horizontal axes correspond to EM iterations. 20 different EM initializations shown for each algorithm setting.

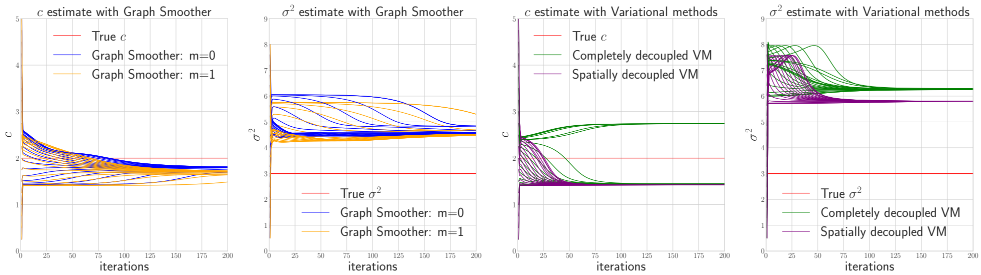


Figure 3.9: $M = 10$. Estimation of c and σ^2 using approximate EM based on the Graph Smoother and the completely and spatially decoupled variational approximations by Ghahramani and Jordan [1997]. Horizontal axes correspond to EM iterations. 20 different EM initializations shown for each algorithm setting.

Method	$\lambda_0(0), \lambda_0(1)$	c	σ^2	$p(0,0), p(0,1), p(1,0), p(1,1)$	$\|p\|_F$
True values	0.000, 1.000	2.000	4.000	0.600, 0.400, 0.200, 0.800	1.095
Graph Smoother $m = 0$	0.137, 0.863	1.753	4.642	0.072, 0.928, 0.518, 0.482	1.171
Graph Smoother $m = 1$	0.001, 0.999	1.779	4.542	0.075, 0.925, 0.549, 0.451	1.171
Completely decoupled Variational Bayes	0.384, 0.616	1.498	6.562	0.075, 0.925, 0.084, 0.916	1.306
Spatially decoupled Variational Bayes	0.393, 0.607	1.960	6.223	0.362, 0.638, 0.385, 0.615	1.366

Table 3.1: Parameters estimates for the case $M = 3$ with Graph Filter-Smoother and variational Bayes at the end of the EM algorithm. The estimates are found by taking the mean over the different initial conditions. $\|p\|_F$ stands for the Frobenius norm of the transition matrix p.

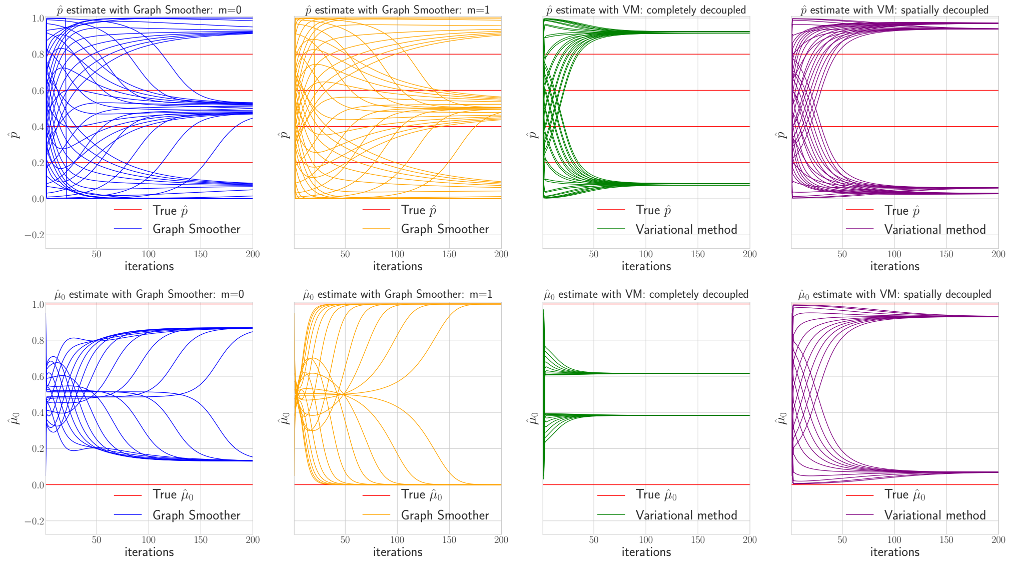


Figure 3.10: $M = 3$. Estimation of $\hat{\mu}_0$ and $\hat{p}$ using approximate EM based on the Graph Smoother and the completely and spatially decoupled variational approximations by Ghahramani and Jordan [1997]. Horizontal axes correspond to EM iterations. 20 different EM initializations are shown for each algorithm setting. Traces corresponding to the 2 elements of the initial distribution $\hat{\mu}_0$ and 4 elements of the transition matrix $\hat{p}$ are superimposed on each plot.

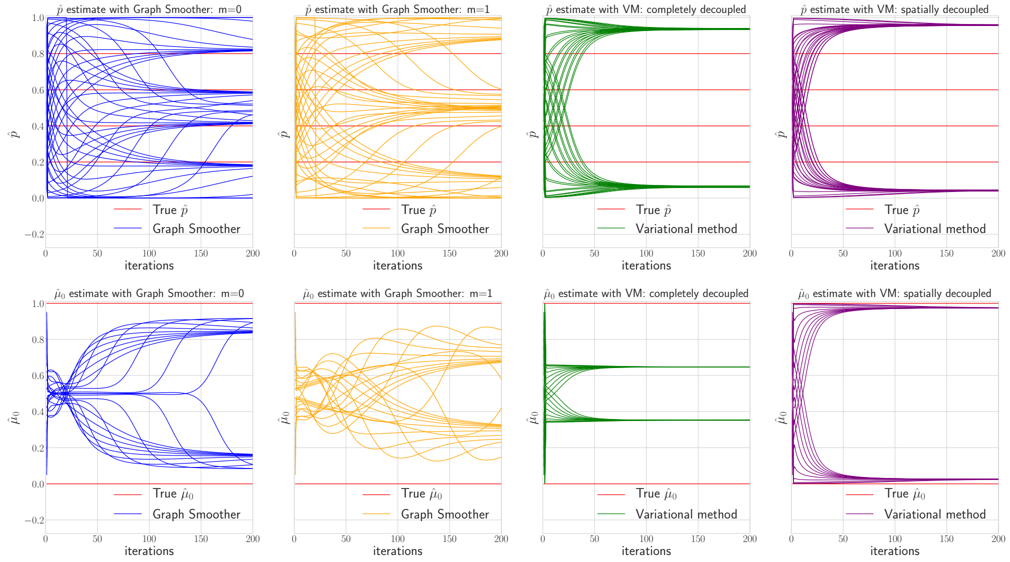


Figure 3.11: $M = 10$. Estimation of $\hat{\mu}_0$ and $\hat{p}$ using approximate EM based on the Graph Smoother and the completely and spatially decoupled variational approximations by Ghahramani and Jordan [1997]. Horizontal axes correspond to EM iterations. 20 different EM initializations are shown for each algorithm setting. Traces corresponding to the 2 elements of the initial distribution $\hat{\mu}_0$ and 4 elements of the transition matrix $\hat{p}$ are superimposed on each plot.

Method	$\lambda_0(0), \lambda_0(1)$	c	σ^2	$p(0,0), p(0,1), p(1,0), p(1,1)$	$\|p\|_F$
True values	0.000, 1.000	2.000	4.000	0.600, 0.400, 0.200, 0.800	1.095
Graph Smoother $m = 0$	0.152, 0.848	1.786	4.617	0.570, 0.430, 0.150, 0.850	1.121
Graph Smoother $m = 1$	0.320, 0.678	1.691	4.566	0.466, 0.534, 0.099, 0.901	1.158
Completely decoupled Variational Bayes	0.469, 0.531	1.833	6.260	0.343, 0.657, 0.336, 0.664	1.347
Spatially decoupled Variational Bayes	0.026, 0.678	1.423	5.804	0.044, 0.956, 0.040, 0.960	1.356

Table 3.2: Parameters estimates for the case $M = 10$ with Graph Filter-Smoother and variational Bayes at the end of the EM algorithm. The estimates are found by taking the mean over the different initial conditions. $\|p\|_F$ stands for the Frobenius norm of the transition matrix p.

3.4.2 Analyzing traffic flows on the London Underground

This section analyses the passenger inflow-outflow on the London Underground from three perspectives: one step ahead prediction; missing data imputation with multi-step-ahead prediction; multi-step-ahead prediction under disruptions and improvements (building new lines).

Transport For London, the operator of the London Underground, has made publicly available "tap" data, consisting of a 5% sample of all Oyster card journeys in a week during November 2009 [Transport for London, 2018]. The data consist of the locations and times of entry to and exit from the transport network for each trip.

Similar Transport for London data have been analyzed by Silva et al. [2015], who developed models of numbers of trips between pairs of stations in the Underground network to quantify the effects of shocks such as line and station closures and to predict traffic volumes. The modelling approach described by Silva et al. [2015](Supporting Information) is very sophisticated, including several components such as regression of the numbers of passengers entering stations onto time, a cascade of nonparametric binomial models for the numbers of passengers inside the transport system who entered at each station and a Bayesian probabilistic flow model. One of many attractive features of this approach is that it avoids the computational cost of network tomography models for traffic data [Guimera et al., 2005, Colizza et al., 2006, Newman et al., 2011] which is prohibitive in the context of large transport systems due to the exponential growth of problem size in the number of network links.

Similar computational difficulties are encountered with some dynamic Bayesian network models of flow on transport networks. For example, Hofleitner et al. [2012] proposes a dynamic mixture model for travel times where the mixture component represents a time-varying congestion state associated with each link in the transport network. In principle, inference in this model can be performed using a particle filter, but as noted by Woodard et al. [2017], due to high-dimensionality the cost of doing so accurately (with respect to Monte Carlo error) is very demanding.

It is not the objective of this thesis to conduct as detailed modelling exercise as in these works, but rather to establish a proof of principle that the Graph Filter-Smoother is naturally suited to the topological structure of transport networks and shows promise for traffic prediction using even a very simple FHMM. This leaves a potential for a deeper investigation of traffic modelling using FHMM's in future work.

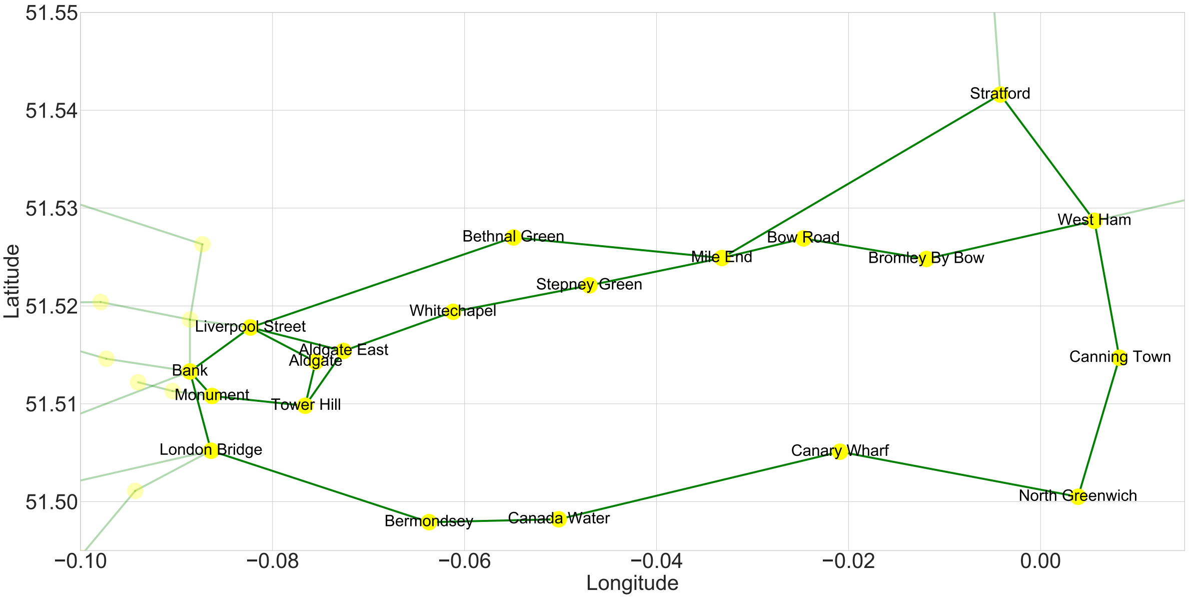


Figure 3.12: Locations in longitude and latitude of 20 stations from the the Central line and the Jubilee line of the London underground. Stations are represented as yellow nodes, while train lines are green edges.

3.4.2.1 Dataset and model

Consider 20 stations on a central portion of the Central line and the Jubilee line. Stations' names and geographical locations are shown in figure 3.12. The dataset consists of the inflow and the outflow of passengers from Monday to Friday per station every 10 minutes from 00:00 am to 00:00 am of the next day. The data are split into training given by Monday, Tuesday and Wednesday and test consisting of Thursday and Friday.

Given that inflow and outflow are modelled, consider two factors per each station, one for the inflow, the other for the outflow. Precisely, per each $f \in \{1,\dots,20\}$, denote with $y_t^{f,\text{in}}$ the inflow at time t in station f and with $y_t^{f,\text{out}}$ the outflow at time t in station f. Consider then the lines as the hidden unit of an HMM. Model each line as bidirectional and use the notation $x_t^{i,j}$ for $i,j \in \{1,\dots,20\}$ and $i \neq j$ to the state of the line connecting station i with station j at time t with the train moving from i to j. In this case, the index set V is the collection of all bidirectional lines in the tube's network. The nodes of the factor graph are then: the factors associated to the stations' inflow and outflow (the number of nodes in figure 3.12 multiply by two); the bidirectional lines (the number of edges in figure 3.12 multiply by two). To build the edges of the factor graph consider the relation between stations and lines. If at time t the inflow in station i is substantial then lots of passengers are going to leave station i and fill the lines $x_t^{i,\cdot}$. Similarly, if the outflow in j at time t is large then lots of people are coming from the lines $x_t^{\cdot,j}$. For these reasons for $i,j \in \{1,\dots,20\}$, with i,j connected through a line in the tube's network, the line $x_t^{i,j}$ is connected with $y_t^{i,in}$ and $y_t^{j,out}$. The described procedure builds the factor graph $\mathcal{G}$ and the related neighbourhood function $N(\cdot)$.

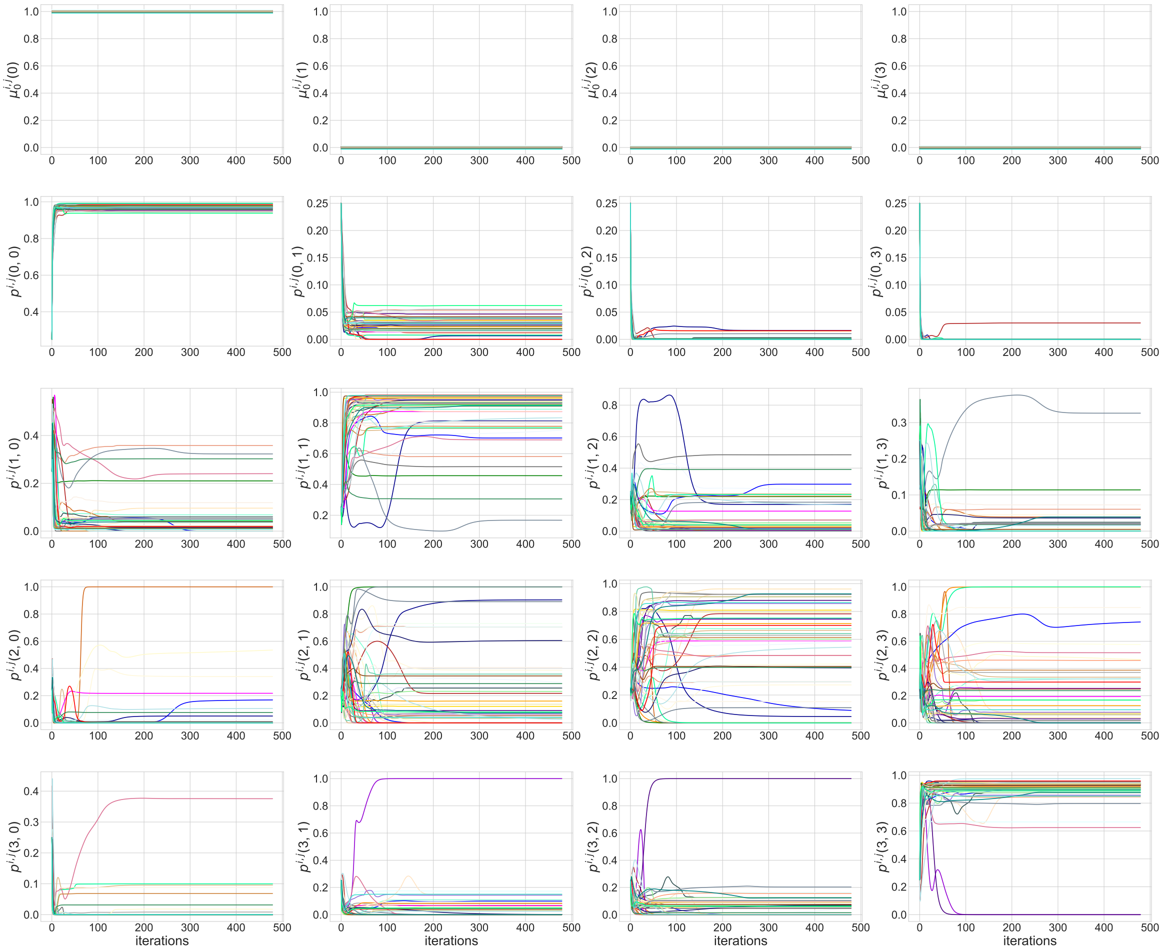


Figure 3.13: EM estimates using the Graph Filter-Smoother algorithm, see appendix A, of the initial distribution and transition matrix per each line. Each coloured line corresponds to a different line and direction in the tube's network.

The state-space of each line is $\mathbb{X} = \{0,1,2,3\}$, which can be interpreted as different levels of busyness in the line. Although the study is restricted to only 20 out of over 300 stations in the full London Underground network, exact filtering and smoothing would be computationally unfeasible since $M = 48$ (the number of lines multiplied by the number of directions), $L = 4$ and hence the cardinality of the overall state-space $\mathbb{X}^V$ is 4^{48}.

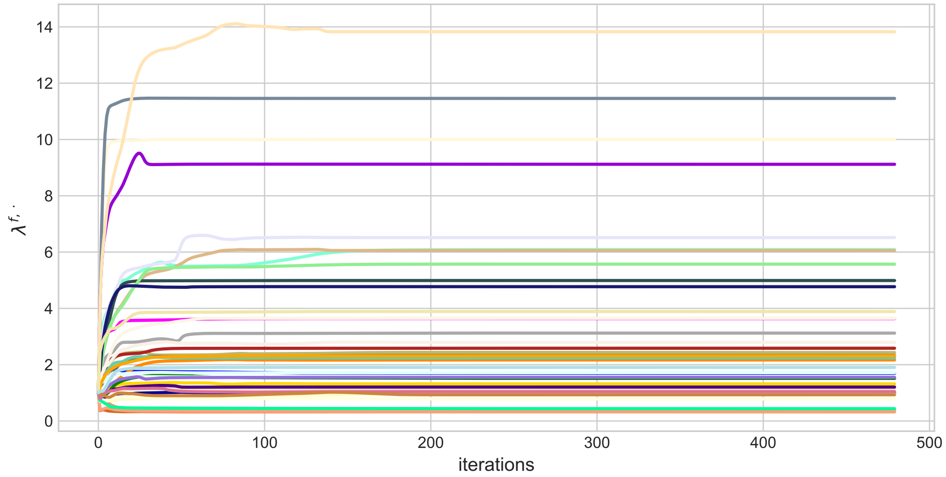


Figure 3.14: EM estimates using the Graph Filter-Smoother algorithm, of $\lambda^{f,\text{in}}$ and $\lambda^{f,\text{out}}$. Each coloured line correspond to a different station flow (inflow or outflow) in the tube's network.

The emission distribution is given by:

$$g(x,y) = \prod_{f=1}^{20} g^{f,\text{in}}\left(x^{f,N(f)}, y^{f,\text{in}}\right) g^{f,\text{out}}\left(x^{N(f),f}, y^{f,\text{out}}\right), \quad x \in \mathbb{X}^V,$$

where $g^{f,\text{in}}\left(x^{f,N(f)}, y^{f,\text{in}}\right)$ and $g^{f,\text{out}}\left(x^{N(f),f}, y^{f,\text{out}}\right)$ are Poisson distributions:

$$g^{f,\text{in}}(x^{f,N(f)}, y^{f,\text{in}}) = \frac{\left(\lambda^{f,\text{in}} \sum_{j\in N(f)} x^{f,j}\right)^{y^{f,\text{in}}}}{(y^{f,\text{in}})!} e^{-(\lambda^{f,\text{in}} \sum_{j\in N(f)} x^{f,j})}$$

$$g^{f,\text{out}}(x^{N(f),f}, y^{f,\text{out}}) = \frac{\left(\lambda^{f,\text{out}} \sum_{i\in N(f)} x^{i,f}\right)^{y^{f,\text{out}}}}{(y^{f,\text{out}})!} e^{-(\lambda^{f,\text{out}} \sum_{i\in N(f)} x^{i,f})}$$

where $x_t^{f,N(f)} = (x_t^{f,j})_{j\in N(f)}$ and $x_t^{N(f),f} = (x_t^{i,f})_{i\in N(f)}$. The λ parameter can be interpreted as a flow intensity which varies across stations and inflow-outflow. Similarly, consider an initial distribution and a transition matrix per each line, i.e.:

$$\mu_0(x) = \prod_{i=1}^{20} \prod_{j\in N(i)} \mu_0^{i,j}\left(x^{i,j}\right), \quad x \in \mathbb{X}^V$$

$$p(x,z) = \prod_{i=1}^{20} \prod_{j\in N(i)} p^{i,j}\left(x^{i,j}, z^{i,j}\right), \quad x,z \in \mathbb{X}^V.$$

The total number of parameters is then 1000: 40 from the flow intensity, 768 from the transition matrices, 192 from the initial distribution.

The static parameters are learnt through an EM algorithm, more details are available in appendix A. The EM algorithm uses the Graph Filter-Smoother with $m = 0$ and $\mathscr{K}$ as in subsection 3.4.1 (partition of singleton over V). The algorithm was run on the training set with random initial conditions on the learnable parameters. The estimates are shown in figure 3.13 and figure 3.14. The estimates are found to be robust for different initial conditions.

The estimate of the initial distribution sets all the lines to the state 0, meaning that all the lines are quiet in the early morning of Monday (the interval from 00:00 am to 00:10 am).

The interpretation of the transition kernel is more difficult. Generally, the estimates can be clustered in three groups, which are heuristically called: “stable” , “quiet” , “busy” . Remark that the names of the groups do not have any quantitative meaning and they are used to refer to estimates with commonalities. Firstly, a line belonging to the stable group has a transition kernel that prefers keeping the state fixed, i.e. the stochastic matrix has high probabilities on the diagonal. Most of the lines are found in this group. Secondly, the quiet cluster has lines with transition kernel's estimates where state $\{3\}$ is difficult to reach, this means that given the state $\{2\}$ at time t it is very likely to go backward to the state $\{1\}$ or stay in $\{2\}$. These estimates appear in lines connecting low-flow stations and precisely in lines between: hubs and low-flow stations (e.g Liverpool Street inflow with Aldgate outflow); high-flow stations and low-flow stations (e.g Canary Wharf inflow with North Greenwich outflow); low-flow stations and low-flow stations (e.g. Stepney Green inflow with Whitechapel outflow). Finally, a line in the busy group can easily reach states $\{2,3\}$. For instance, if a line in the busy group reaches state $\{1\}$ at time t, it moves to state $\{2\}$ with high probability and it keeps moving forward to state $\{3\}$ in the next time step. These estimates are typical for lines connecting high-flow stations with high-flow stations, e.g. Liverpool Street inflow with Bank outflow.

The interpretation of figure 3.14 is straightforward, indeed $\lambda^{f,\cdot}$ scales the intensity of the flow, hence bigger estimates of $\lambda^{f,\cdot}$ are referred to stations with higher flows.

Flow prediction without missing data Prediction over the test data is performed through the posterior predictive of Graph Filter-Smoother, i.e. per each time step $t \geq 0$ a posterior predictive sample y_{t+1} is obtained as:

$$x_t \sim \tilde{\pi}_{t|t}, \quad x_{t+1} \sim p(x_t, \cdot), \quad y_{t+1} \sim g(x_{t+1}, \cdot), \tag{3.18}$$

where g, p are computed through the EM algorithm run on the training set (Monday, Tuesday, Wednesday) and $\tilde{\pi}_{\cdot|\cdot}$ is obtained by running recursively the Graph Filter on the test data (Thursday, Friday). Note that (3.18) provides a sample from the posterior predictive, the mean of this sample is then used as prediction for the test set. The first column of figure 3.15 shows the mean and the 0.95 credible interval of posterior predictive samples for the data from Thursday to Friday. For ease of presentation consider the plots on four stations, plots on all the other stations are available in appendix A section A.2.2. The results indicate that the model is able to track the peaks of the inflow and outflow that occur during the morning and afternoon rush hours, which vary in magnitude from station to station. Moreover, credible intervals show satisfying coverage of the true data.

The proposed method is compared with an LSTM trained on one-step-ahead prediction over the inflow-outflow, more details about the architecture and training are available in appendix A, section A.2.1. The LSTM takes as input inflow-outflow data over all the stations at time t and

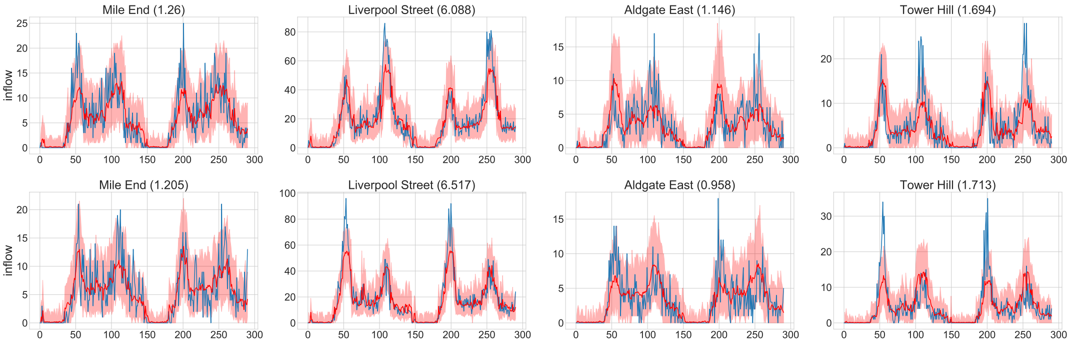


Figure 3.15: One step-ahead posterior predictive mean (solid red line) and 0.95 credible intervals (red bands) using the Graph Filter-Smoother on four stations. Blue solid lines stand for the observed data from Thursday to Friday. The first row shows the inflow, the second row shows the outflow. The name of the station is reported at the top of each plot, along with the estimate of the corresponding $\lambda^{f,\cdot}$.

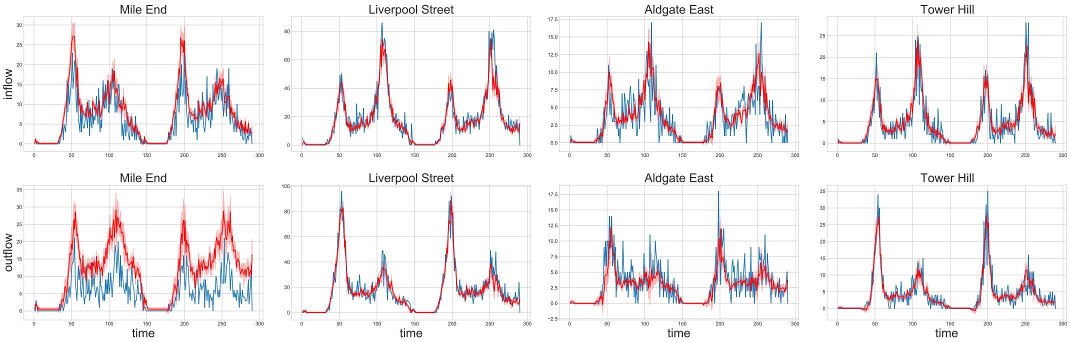


Figure 3.16: One step-ahead prediction with the LSTM. Per time step, a sample of size 100 is built over different training of the LSTM where solid red lines show the mean and red bands show the region between the 0.025 and the 0.975 quantiles. Blue solid lines stand for the observed data from Thursday to Friday. The first row reports the inflow, the second row reports the outflow. The name of the station is written at the top of each plot.

output predictions for time $t+1$. 100 LSTMs (different initial conditions and seeds) are trained on Monday, Tuesday and Wednesday. As for the Graph Filter-Smoother method, testing is performed on Thursday and Friday. First column of figure 3.16 shows LSTM's predictions on four stations only, more details are available in the appendix A, section A.2.2. Figure 3.16 shows that different trainings of the LSTM lead to similar performances with bands that are narrower than figure 3.15.

Table 3.3 reports RMSEs for the posterior predictive mean of Graph Filter-Smoother and the LSTM. The mean and standard deviation of the RMSE for the posterior predictive mean of Graph Filter-Smoother are computed over 100 samples of the posterior predictive mean. The mean and standard deviation of the RMSE for LSTM are computed over 100 LSTMs optimization (i.e. different initial conditions and seeds, which is a common procedure in machine learning).

The proposed algorithm has performances that are comparable with the LSTM even though the LSTM is explicitly trained to minimize the RMSE on the one-step-ahead prediction.

Method	No missing	Missing without peak	Missing with peak
Graph Filter-Smoother	4.796 ± 0.011	5.399 ± 0.014	5.493 ± 0.019
LSTM	4.639 ± 0.138	7.427 ± 2.080	6.178 ± 1.425

Table 3.3: RMSE comparison between the posterior predictive mean of Graph Filter-Smoother and LSTM. “No missing” refers to the performance on the full test set. “Missing without peak” refers to the test set performance when data from $t = 130$ (around 9 pm on Thursday) to $t = 170$ (around 4 am on Friday) are missing. “Missing with peak” refers to the test set performance when data from $t = 230$ (around 2 pm on Friday) to $t = 270$ (around 9 pm on Friday) are missing.

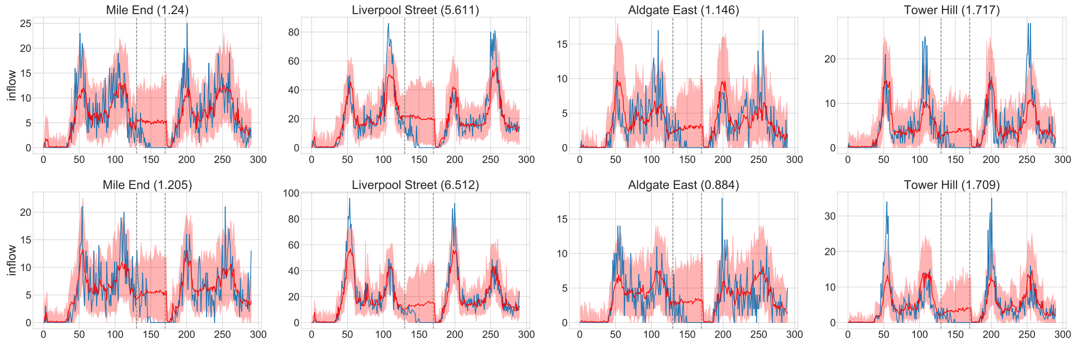


Figure 3.17: Multi-step-ahead posterior predictive mean (solid red line) and 0.95 credible intervals (red bands) using the Graph Filter-Smoother on four stations with missing data in a quiet period (without peak). Blue solid lines stand for the observed data from Thursday to Friday (the missing data are included). Grey dashed lines show the start and the end of the missing data window. Stations’ names are reported at the top of each plot.

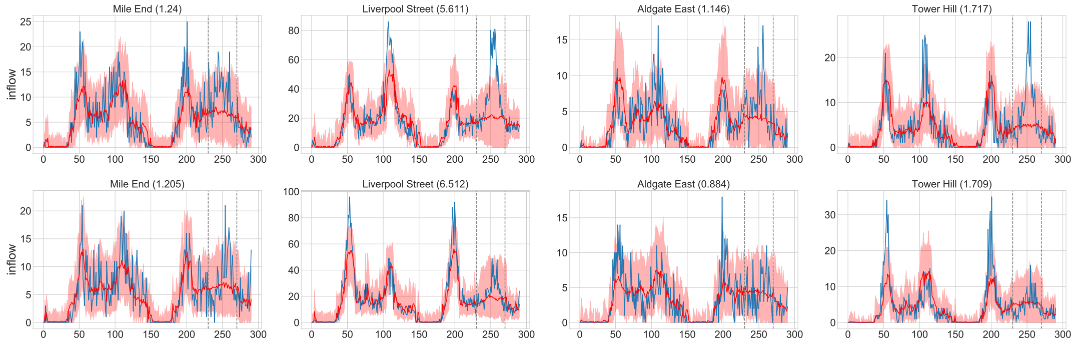


Figure 3.18: Multi-step-ahead posterior predictive mean (solid red line) and 0.95 credible intervals (red bands) using the Graph Filter-Smoother on four stations with missing data in a busy period (with peak). Blue solid lines stand for the observed data from Thursday to Friday (the missing data are included). Grey dashed lines show the start and the end of the missing data window. Stations’ names are reported at the top of each plot.

Flow prediction with missing data Suppose now that data cannot be collected in the period from t to $t+h$, for instance, because the sensors in the stations are broken and people come in and out without tapping their Oyster cards.

Graph Filter can automatically impute missing data by applying the "prediction" operator without the "correction" operator. Precisely, multi-step-ahead posterior predictive from t to $t+h$ can be pursued as follows:

$$\begin{aligned}
&x_t \sim \tilde{\pi}_{t|t},\\
&x_{t+1} \sim p(x_t,\cdot), \quad \tilde{y}_{t+1} \sim g(x_{t+1},\cdot),\\
&x_{t+2} \sim p(x_{t+1},\cdot), \quad \tilde{y}_{t+1} \sim g(x_{t+2},\cdot),\\
&\dots\\
&x_{t+h} \sim p(x_{t+h-1},\cdot), \quad \tilde{y}_{t+h} \sim g(x_{t+h},\cdot).
\end{aligned}$$

The sample $\tilde{y}_{t+1},\dots,\tilde{y}_{t+h}$ provides a way to impute the missing data $y_{t+1},\dots,y_{t+h}$.

Similarly, LSTM can do multi-step-ahead prediction by using the output on the current time step as input for the next time step. Remark that the considered LSTM is trained exclusively for one-step-ahead prediction (i.e. LSTM maps y_t onto y_{t+1}). As an alternative, one could train LSTM on multi-step-ahead predictions (i.e. LSTM maps y_t onto $y_{t+1},\dots,y_{t+h}$), which requires to know the missing data window in advance. However, that is generally not plausible, e.g. missing data in traffic applications are caused by broken sensors and the time they are out of order cannot be predicted in advance.

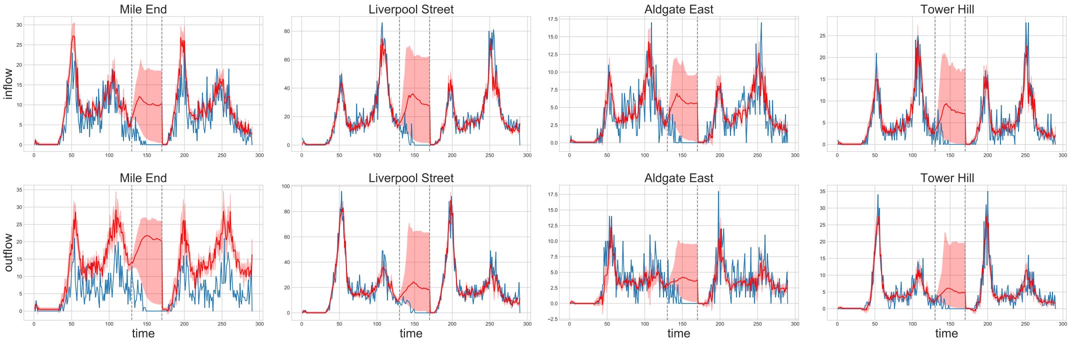


Figure 3.19: Multi-step-ahead prediction with the LSTM on four stations with missing data in a quiet period (without peak). Per time step, a sample of size 100 is built over different training of the LSTM where solid red lines show the mean and red bands show the region between the 0.025 and the 0.975 quantiles. Blue solid lines stand for the observed data from Thursday to Friday. Grey dashed lines show the start and the end of the missing data window. Stations' names are reported at the top of each plot.

RMSE performances between LSTM and the proposed method in a missing data scenario are compared in the second and the third column of Table 3.3. Two cases are distinguished: missing data in a quiet period (second column) and missing data in a busy period (third column). LSTM

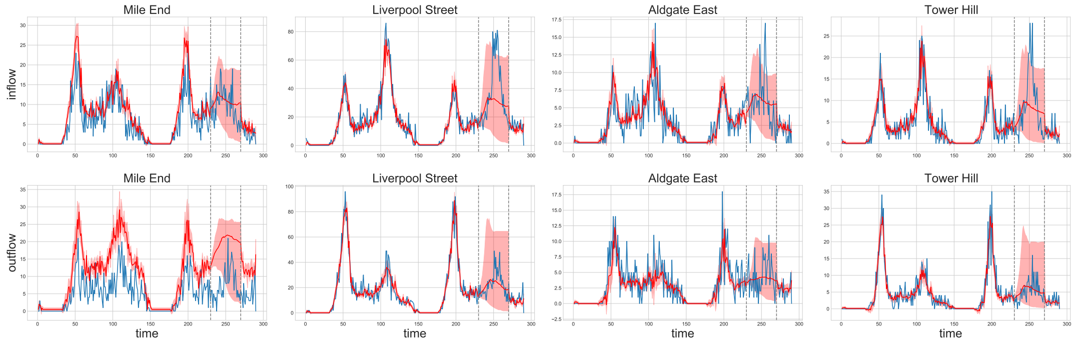


Figure 3.20: Multi-step-ahead prediction with the LSTM on four stations with missing data in a busy period (with peak). Per time step, a sample of size 100 is built over different training of the LSTM where solid red lines show the mean and red bands show the region between the 0.025 and the 0.975 quantiles. Blue solid lines stand for the observed data from Thursday to Friday. Grey dashed lines show the start and the end of the missing data window. Stations' names are reported at the top of each plot.

has evident robustness issues, the RMSE varies consistently depending on the initial condition and on the training (high standard deviation). On the contrary, the proposed method is more stable, with a standard deviation that is 100 times lower than LSTM and an RMSE that is significantly lower in mean. This appears in the experiments for missing data in both busy and quiet periods. Graphical illustrations on selected stations can be found in figures 3.17, 3.18, 3.19, 3.20, more figures can be found in appendix A.

Flow prediction under disruptions and improvements The section concludes with a multi-step-ahead prediction when the factor graph changes over time. Indeed, improvements (new lines are built) and /or disruptions (some lines are closed) may occur in the underground, which can be modelled by adding and /or removing lines from the tube's network.

LSTM is not able to adapt to the new structure of the network and it keeps predicting as nothing has changed. On the contrary, the Graph Filter-Smoother can modify the factor graph (remove and /or add elements of V) and study the effects that this has on future inflow-outflow.

In the case of improvements at time t, a line connecting i and j is added. To include this line in future predictions the extra line has to be added to the factor graph and two additional hidden units, referring to the two directions of the added line, are created. Then each new hidden unit is associated with a prior at time $t-1$ and an estimate of the transition matrix. The former is done by setting to zero with probability one the new line because the line does not exist at $t-1$. The latter is estimated as the mean element by element of the transition matrices of the other lines (an alternative approach could be to consider a mean over the lines connected to the stations i, j only).

In the case of disruption at time t, a line connecting i and j is closed (removed). To exclude the line from future prediction the two hidden units associated with the closed line can be simply removed from the factor graph. After that, the removed hidden unit is not considered in the recursion.

Figures 3.21 compare graphically how the flows react when removing a line connecting Mile End and North Greenwich and when adding a line between Bank and London Bridge. Generally, removing a line in the proposed model implies a decrease in the inflow-outflow of the connected stations, while adding a line causes an increase in the inflow-outflow of the connected stations. Note that competition among stations as in Silva et al. [2015] is not implemented, hence it should be expected a change only on the surrounding stations. More complicated spatial dependencies can be considered in the Poisson emission, but these studies are left to future works.

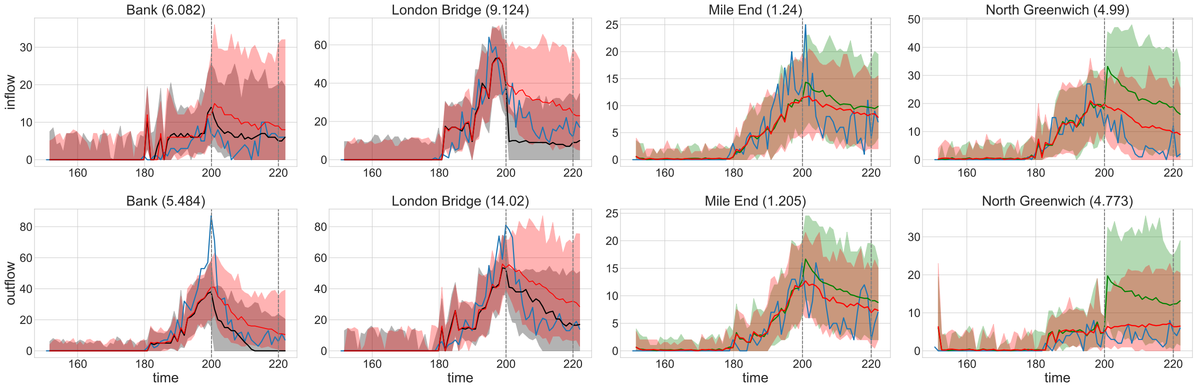


Figure 3.21: Predicted flows for Bank and London Bridge when removing a line (first and second columns), and for Mile End and North Greenwich when adding a line (third and fourth columns). Solid lines are posterior predictive means. Bands are 0.95 credible intervals. In red: Graph Filter-Smoother without any changes. In black: Graph Filter-Smoother when removing a line. In green: Graph Filter-Smoother when adding a line. Blue solid lines are the observed data without any changes. Stations names and $\lambda^{f,\cdot}$ estimates (without changes) are reported at the top.

Summary There are several advantages of using Graph Filter-Smoother when modelling traffic flows.

- As it has been described in the paragraph “Dataset and model” , all the parameters of the proposed model can be easily interpreted. This compares with LSTM whose weights do not have any interpretation and it must be considered as a black-box.

- The proposed method can quantify uncertainty. In particular, the experiments showed that 0.95 credible intervals cover most of the data. On the contrary, LSTM does not have any way to quantify uncertainty and it is overconfident in its predictions.

- Experimentally, Graph Filter-Smoother is able to impute missing data efficiently without drastic changes in the prediction performance (RMSE in table 3.3). Conversely, LSTM has robustness issues with different training (different initial conditions and gradient steps) leading to very different predictions.

- The proposed method is flexible and it can adapt to sudden changes in the structure of the factor graph without retraining from scratch. The latest part of the experiments proposed a way to modify the Graph Filter-Smoother to adapt to changes in the tube's network.

Chapter 4

Exploiting locality in high-dimensional factorial hidden Markov models-Theoretical proofs

This chapter consists of three sections. In section 4.1, some of the key definitions are gathered together for the reader's convenience and some further objects and lemmas needed for the main proofs are introduced. Section 4.2 and section 4.3 are dedicated to proving respectively theorems 3.1 and 3.2. Upon first reading, one may skip straight to section 4.2, where an outline of the main steps in the proof of theorem 3.1 is given. As already mentioned, a significant part of this chapter has been submitted to publication and it is available in the appendix of Rimella and Whiteley [2019].

Before commencing, a comment on generality and notation. In the previous sections a state-space $\mathbb{X}$, which is a set of finite cardinality, and probability mass function on $\mathbb{X}^U$ for some $U \subseteq V$ are considered. In order to make notation visually compact, throughout the current section expectations with respect to such mass functions are written as measure theoretic integrals, e.g. $\int \mathbb{1}_A(x)\mu(dx)$. Here it is to be understood that $\mu(dx) = \mu(x)\psi(dx)$ where ψ denotes counting measure, and hence $\int \mathbb{1}_A(x)\mu(dx) \equiv \mu(A) \equiv \sum_{x \in A} \mu(x)$. This integral notation reflects the fact that many of the results from this thesis do not rely on $\mathbb{X}$ being a set of finite cardinality and could be replaced by a Polish spaces as for the definitions in section 2.1 of chapter 2.

4.1 Definitions and Preliminary results

The following definitions associated with the factor graph $\mathscr{G}$ are needed (some definitions are repeated from (3.7)-(3.8) and reported here for the convenience of the reader):

$$d(J,J') := \min_{e\in J}\min_{e'\in J'}\{d(e,e')\}, \quad J,J' \subseteq V\times F,$$
$$N(J) := \{f \in F : \exists v \in J \text{ with } d(v,f) \leq 1\}, \quad J \subseteq V,$$

$$
\begin{aligned}
N^2(J) &:= \{v' \in V : \exists v \in J \text{ with } d(v',v) \leq 2\}, \quad J \subseteq V,\\
N_v^r(J) &:= \{v' \in V \text{ such that } \exists v \in J \text{ with } d(v,v') \leq 2r+2\}, \quad J \subseteq V,\\
N_f^r(J) &:= \{f \in F \text{ such that } \exists v \in J \text{ with } d(v,f) \leq 2r+1\}, \quad J \subseteq V,\\
n_J &:= \frac{1}{2}\max_{v\in V} d(J,v),\\
\Upsilon &:= \max_{v\in V}\{\mathbf{card}(N(v))\},\\
\Upsilon^{(2)} &:= \max_{v\in V}\{\mathbf{card}(N_v^0(v))\},\\
\tilde{\Upsilon} &:= \max_{v,v'\in V}\{\mathbf{card}(N(v)\cap N(v'))\},\\
\widetilde{J} &:= \{v \in J : \forall f \in N(v), N(f) \subseteq J\}, \quad J \subseteq V,\\
\partial J &:= J \setminus \widetilde{J}, \quad J \subseteq V.
\end{aligned}
$$

Remark that $N_v^0(J) = N^2(J)$ and the following inclusion relation holds:

$$J \subseteq N^2(J) = N_v^0(J) \subseteq N_v^1(J) \subseteq \cdots \subseteq N_v^m(J), \quad m \geq 1,$$

and similarly on the set F:

$$N(J) = N_f^0(J) \subseteq N_f^1(J) \subseteq \cdots \subseteq N_f^m(J), \quad m \geq 1.$$

Let $\mathbb{S}$ be a product of Polish spaces, i.e.: $\mathbb{S} = \bigotimes_{k\in I}\mathbb{S}_k$, where I is a finite index set.

Definition 4.1. Given two probability distribution μ,ν on $\mathbb{S}$, the total variation distance (TV) and the local total variation distance (LTV) are:

- $\|\mu-\nu\| := \sup_{A\in\sigma(\mathbb{S})} |\mu(A)-\nu(A)|$.
- $\|\mu-\nu\|_J := \sup_{A\in\sigma(\bigotimes_{k\in J}\mathbb{S}_k)} |\mu^J(A)-\nu^J(A)|, \quad J \subset I.$

Definition 4.2. Let μ be a probability distribution on $\mathbb{S}$ and let $X \sim \mu$. The conditional distribution over the component $i \in I$ is defined as:

$$\mu_x^i(A) := \mathbb{P}\left(X^i \in A | X^{I\setminus i} = x^{I\setminus i}\right), \quad x \in \mathbb{S}.$$

Definition 4.3. Let μ be a probability distribution on $\mathbb{X}^V$ with $X \sim \mu$ and let $P(x,dz) := p(x,z)\psi(dz)$ with $x,z \in \mathbb{X}^V$ be the transition kernel of the considered FHMM with $Z|X \sim P(X,\cdot)$, then for $v \in V$:

$$\mu^v_{x,z}(A) := \mathbb{P}\left(X^v \in A | X^{V\setminus v} = x^{V\setminus v}, Z = z\right), \quad x,z \in \mathbb{X}^V.$$

Lemma 4.1. *Fix any collection of observations $\{y_1,\dots,y_T\}$ and consider a probability distribution μ on $\mathbb{X}^V$. Given the optimal correction operator as in* (2.5)*, the conditional distribution of $\mathsf{C}_t\mu$ over the component $v \in V$ is given by:*

$$(\mathsf{C}_t\mu)^v_x(A) = \frac{\int \mathbb{1}_A(x^v)\prod_{f\in N(v)} g^f(x^{N(f)},y_t)\mu^v_x(dx^v)}{\int \prod_{f\in N(v)} g^f(x^{N(f)},y_t)\mu^v_x(dx^v)}, \quad x \in \mathbb{X}^V.$$

Similarly, the one step forward conditional distribution over the component $v \in V$ is:

$$(\mathsf{C}_t\mu)^v_{x,z}(A) = \frac{\int \mathbb{1}_A(x^v)\prod_{f\in N(v)} g^f(x^{N(f)},y_t)p^v(x^v,z^v)\mu^v_x(dx^v)}{\int \prod_{f\in N(v)} g^f(x^{N(f)},y_t)p^v(x^v,z^v)\mu^v_x(dx^v)}, \quad x,z \in \mathbb{X}^V.$$

Proof. The proof is a trivial consequence of the form of the correction operator as in (2.5) and Definition 4.2. The procedure is the following:

$$\begin{aligned}
(\mathsf{C}_t\mu)^v_x(A) &= \frac{(\mathsf{C}_t\mu)(A \times x^{V\setminus v})}{(\mathsf{C}_t\mu)(\mathbb{X} \times x^{V\setminus v})}\\
&= \frac{\int \mathbb{1}_{A\times x^{V\setminus v}}(\tilde{x}^v \times \tilde{x}^{V\setminus v})\prod_{f\in F\setminus N(v)} g^f(\tilde{x}^{N(f)},y_t)\prod_{f\in N(v)} g^f(\tilde{x}^{N(f)},y_t)\mu(d\tilde{x})}{\int \mathbb{1}_{\mathbb{X}\times x^{V\setminus v}}(\tilde{x}^v \times \tilde{x}^{V\setminus v})\prod_{f\in F\setminus N(v)} g^f(\tilde{x}^{N(f)},y_t)\prod_{f\in N(v)} g^f(\tilde{x}^{N(f)},y_t)\mu(d\tilde{x})}\\
&= \frac{\prod_{f\in F\setminus N(v)} g^f(x^{N(f)},y_t)}{\prod_{f\in F\setminus N(v)} g^f(x^{N(f)},y_t)}\frac{\int \mathbb{1}_A(\tilde{x}^v)\mathbb{1}_{x^{V\setminus v}}(\tilde{x}^{V\setminus v})\prod_{f\in N(v)} g^f(\tilde{x}^{N(f)},y_t)\mu(d\tilde{x})}{\int \mathbb{1}_{\mathbb{X}}(\tilde{x}^v)\mathbb{1}_{x^{V\setminus v}}(\tilde{x}^{V\setminus v})\prod_{f\in N(v)} g^f(\tilde{x}^{N(f)},y_t)\mu(d\tilde{x})}\\
&= \frac{\int \mathbb{1}_A(\tilde{x}^v)\mathbb{1}_{x^{V\setminus v}}(\tilde{x}^{V\setminus v})\prod_{f\in N(v)} g^f(\tilde{x}^{N(f)},y_t)\frac{\mu(d\tilde{x})}{\int \mathbb{1}_{\mathbb{X}\times x^{V\setminus v}}(\tilde{x}^v\times\tilde{x}^{V\setminus v})\mu(d\tilde{x})}}{\int \mathbb{1}_{\mathbb{X}}(\tilde{x}^v)\mathbb{1}_{x^{V\setminus v}}(\tilde{x}^{V\setminus v})\prod_{f\in N(v)} g^f(\tilde{x}^{N(f)},y_t)\frac{\mu(d\tilde{x})}{\int \mathbb{1}_{\mathbb{X}\times x^{V\setminus v}}(\tilde{x}^v\times\tilde{x}^{V\setminus v})\mu(d\tilde{x})}}\\
&= \frac{\int \mathbb{1}_A(\tilde{x}^v)\prod_{f\in N(v)} g^f(\tilde{x}^{N(f)},y_t)\mu^v_x(d\tilde{x}^v)}{\int \prod_{f\in N(v)} g^f(\tilde{x}^{N(f)},y_t)\mu^v_x(d\tilde{x}^v)}.
\end{aligned}$$

The form of $(\mathsf{C}_t\mu)^v_{x,z}(A)$ follow the same procedure with the addition of the kernel $p^v(x^v,z^v)$ at the end, where the isolation of the component v is a consequence of the factorization of the transition kernel.

■

Definition 4.4. Fix any collection of observations $\{y_1,\dots,y_T\}$ and consider a probability distribution μ on $\mathbb{X}^V$. For $K \in \mathcal{K}$ and $m = \{0,\dots,n\}$ define:

$$(\tilde{\mathsf{C}}^{m,K}_t\mu)(A) := \frac{\int \mathbb{1}_A(x^K)\prod_{f\in N^m_f(K)} g^f(x^{N(f)},y_t)\mu(dx)}{\int \prod_{f\in N^m_f(K)} g^f(x^{N(f)},y_t)\mu(dx)},$$

and define:

$$\tilde{\mathsf{C}}^m_t\mu := \bigotimes_{K\in\mathcal{K}} \tilde{\mathsf{C}}^{m,K}_t\mu. \tag{4.1}$$

Given that the approximated correction operator is applied to probability distributions that factorize over the partition $\mathscr{K}$, the quantity of interest is $\mu = \bigotimes_{K \in \mathscr{K}} \mu^K$ hence:

$$(\tilde{\mathsf{C}}_t^{m,K}\mu)(A) = \frac{\int \mathbb{1}_A(x^K) \int \prod_{f \in N_f^m(K)} g^f(x_K^{N(f)}, y_t) \mu^{V \setminus K}(d\tilde{x}) \mu^K(dx^K)}{\int \prod_{f \in N_f^m(K)} g^f(x_K^{N(f)}, y_t) \mu^{V \setminus K}(d\tilde{x}) \mu^K(dx^K)},$$

meaning that $\tilde{\mathsf{C}}_t^m \mu$ can be written as:

$$(\tilde{\mathsf{C}}_t^m \mu)(A) = \frac{\int \mathbb{1}_A(x) \prod_{K \in \mathscr{K}} \int \prod_{f \in N_f^m(K)} g^f(x_K^{N(f)}, y_t) \mu^{V \setminus K}(d\tilde{x}) \mu(dx)}{\int \prod_{K \in \mathscr{K}} \int \prod_{f \in N_f^m(K)} g^f(x_K^{N(f)}, y_t) \mu^{V \setminus K}(d\tilde{x}) \mu(dx)},$$

where x_K is a collection of auxiliary variables:

$$x_K^v := \begin{cases} x^v & \text{if } v \in K \\ \tilde{x}^v & \text{if } v \notin K \end{cases} \quad \text{with } \tilde{x} \in \mathbb{X}^{V \setminus K} \text{ and } x \in \mathbb{X}^V.$$

The above definition is used to distinguish the components that are integrated out (with a tilde) from the ones that are not (without a tilde).

Lemma 4.2. *Fix any collection of observations $\{y_1, \ldots, y_T\}$ and let μ be a probability distribution on $\mathbb{X}^V$ such that $\mu = \bigotimes_{K' \in \mathscr{K}} \mu^{K'}$. Given the new correction operator as in* (4.1) *the conditional distribution of $\tilde{\mathsf{C}}_t^m \mu$ over the component $v \in K$ with $K \in \mathscr{K}$ is given by:*

$$(\tilde{\mathsf{C}}_t^m \mu)_x^v(A) = \frac{\int \mathbb{1}_A(x^v) \int \prod_{f \in N_f^m(K)} g^f(x_K^{N(f)}, y_t) \mu^{V \setminus K}(d\tilde{x}) \mu_x^v(dx^v)}{\int \prod_{f \in N_f^m(K)} g^f(x_K^{N(f)}, y_t) \mu^{V \setminus K}(d\tilde{x}) \mu_x^v(dx^v)}, \quad x \in \mathbb{X}^{\mathbb{V}}.$$

Similarly, the one step forward conditional distribution over the component $v \in K$ with $K \in \mathscr{K}$ is given by:

$$(\tilde{\mathsf{C}}_t^m \mu)_{x,z}^v(A) = \frac{\int \mathbb{1}_A(x^v) \int \prod_{f \in N_f^m(K)} g^f(x_K^{N(f)}, y_t) \mu^{V \setminus K}(d\tilde{x}) p^v(x^v, z^v) \mu_x^v(dx^v)}{\int \prod_{f \in N_f^m(K)} g^f(x_K^{N(f)}, y_t) \mu^{V \setminus K}(d\tilde{x}) p^v(x^v, z^v) \mu_x^v(dx^v)}, \quad x, z \in \mathbb{X}^{\mathbb{V}}.$$

Proof. The proof follows from Definition 4.2, Definition 4.4 and the form of the operator when μ factorizes. The form of these conditional distributions can be obtained with the same procedure as in the proof of lemma 4.1. Consider $v \in K$:

$$\begin{aligned}
(\tilde{\mathsf{C}}_t^m \mu)_x^v(A) &= \frac{(\tilde{\mathsf{C}}_t^m \mu)(A \times x^{V\setminus v})}{(\tilde{\mathsf{C}}_t^m \mu)(\mathbb{X} \times x^{V\setminus v})} \\
&= \frac{\int \mathbb{1}_{A \times x^{V\setminus v}}(\hat{x}) \prod_{K' \in \mathcal{K}} \int \prod_{f \in N_f^m(K')} g^f(\hat{x}_{K'}^{N(f)}, y_t) \mu^{V\setminus K'}(d\tilde{x}) \mu(d\hat{x})}{\int \mathbb{1}_{\mathbb{X} \times x^{V\setminus v}}(\hat{x}) \prod_{K' \in \mathcal{K}} \int \prod_{f \in N_f^m(K')} g^f(\hat{x}_{K'}^{N(f)}, y_t) \mu^{V\setminus K'}(d\tilde{x}) \mu(d\hat{x})} \\
&= \frac{\int \mathbb{1}_{A \times x^{V\setminus v}}(\hat{x}) \int \prod_{f \in N_f^m(K)} g^f(\hat{x}_K^{N(f)}, y_t) \mu^{V\setminus K}(d\tilde{x}) \mu(d\hat{x})}{\int \mathbb{1}_{\mathbb{X} \times x^{V\setminus v}}(\hat{x}) \int \prod_{f \in N_f^m(K)} g^f(\hat{x}_K^{N(f)}, y_t) \mu^{V\setminus K}(d\tilde{x}) \mu(d\hat{x})} \\
&= \frac{\int \mathbb{1}_{A \times x^{V\setminus v}}(\hat{x}) \int \prod_{f \in N_f^m(K)} g^f(\hat{x}_K^{N(f)}, y_t) \mu^{V\setminus K}(d\tilde{x}) \frac{\mu(d\hat{x})}{\int \mathbb{1}_{\mathbb{X} \times x^{V\setminus v}}(\hat{x}^v \times \hat{x}^{V\setminus v}) \mu(d\hat{x})}}{\int \mathbb{1}_{\mathbb{X} \times x^{V\setminus v}}(\hat{x}) \int \prod_{f \in N_f^m(K)} g^f(\hat{x}_K^{N(f)}, y_t) \mu^{V\setminus K}(d\tilde{x}) \frac{\mu(d\hat{x})}{\int \mathbb{1}_{\mathbb{X} \times x^{V\setminus v}}(\hat{x}^v \times \hat{x}^{V\setminus v}) \mu(d\hat{x})}} \\
&= \frac{\int \mathbb{1}_A(x) \int \prod_{f \in N_f^m(K)} g^f(x_K^{N(f)}, y_t) \mu^{V\setminus K}(d\tilde{x}) \mu_x^v(dx)}{\int \prod_{f \in N_f^m(K)} g^f(x_K^{N(f)}, y_t) \mu^{V\setminus K}(d\tilde{x}) \mu_x^v(dx)}.
\end{aligned}$$

Moreover given that for $v \in K$ and $\mu = \bigotimes_{K' \in \mathcal{K}} \mu^{K'}$ then $\mu_x^v = (\mu^K)_x^v$ then $(\tilde{\mathsf{C}}_t^m \mu)_x^v(A) = (\tilde{\mathsf{C}}_t^{m,K} \mu)_x^v(A)$, which can be easily checked by substituting μ_x^v with $(\mu^K)_x^v$ in the previous computations. ■

Lemma 4.3. *Let μ, μ' and ν, ν' be probability distributions on $\mathbb{S}$. Assume that there exists $\epsilon \in (0,1)$ such that:*

$$\nu(A) \geq \epsilon \mu(A) \quad \textit{and} \quad \nu'(A) \geq \epsilon \mu'(A).$$

Then:

$$\left\| \nu - \nu' \right\| \leq 2(1-\epsilon) + \epsilon \left\| \mu - \mu' \right\|.$$

Proof. The proof is available in lemma 4.1, pag.32 of Rebeschini and Van Handel [2015]. ■

Lemma 4.4. *Let μ, ν be probability distributions on $\mathbb{S}$ and Λ a bounded-strictly positive measurable function on the same space. Consider:*

$$\mu_\Lambda(A) := \frac{\int \mathbb{1}_A(x) \Lambda(x) \mu(dx)}{\int \Lambda(x) \mu(dx)}, \qquad \nu_\Lambda(A) := \frac{\int \mathbb{1}_A(x) \Lambda(x) \nu(dx)}{\int \Lambda(x) \nu(dx)}.$$

Then:

$$\left\| \mu_\Lambda - \nu_\Lambda \right\| \leq 2 \frac{\sup_x \Lambda(x)}{\inf_x \Lambda(x)} \left\| \mu - \nu \right\|.$$

Proof. The complete proof is available in lemma 4.2, pag.32 of Rebeschini and Van Handel [2015]. ■

Theorem 4.1 (Dobrushin Comparison theorem)**.** *Let μ, ν be probability distributions on $\mathbb{S}$. For $i, j \in I$ define the quantities:*

$$C_{i,j} = \frac{1}{2} \sup_{\substack{x, \hat{x} \in \mathbb{S} \\ x^{I\setminus j} = \hat{x}^{I\setminus j}}} \left\| \mu_x^i - \mu_{\hat{x}}^i \right\| \quad \textit{and} \quad b_j = \sup_{x \in \mathbb{S}} \left\| \mu_x^j - \nu_x^j \right\|. \tag{4.2}$$

Assume that:

$$\max_{i\in I}\sum_{j\in I} C_{i,j} < 1 \quad \textit{(Dobrushin condition)},$$

then the matrix $D := \sum_{t\geq 0} C^t$, *where* C^0 *is the identity matrix, converges and for an arbitrary* $J \subseteq I$ *it holds:*

$$\|\mu - \nu\|_J \leq \sum_{i\in J}\sum_{j\in I} D_{i,j} b_j.$$

Proof. See for example theorem 8.20 of Georgii [2011]. ■

Lemma 4.5. *Let* I *be a finite index set with* $m(\cdot,\cdot)$ *a pseudometric on it. Let* C *be a nonnegative matrix with rows and columns indexed by* I*. Assume that there exists* $\lambda \in (0,1)$ *such that:*

$$\max_{i\in I}\sum_{j\in I} e^{m(i,j)} C_{i,j} \leq \lambda.$$

Then the matrix $D := \sum_{t\geq 0} C^t$ *satisfies:*

$$\max_{i\in I}\sum_{j\in I} e^{m(i,j)} D_{i,j} \leq \frac{1}{1-\lambda}.$$

Moreover:

$$\sum_{j\in J} D_{i,j} \leq \frac{e^{-m(i,J)}}{1-\lambda}.$$

Proof. The proof is available in Rebeschini and Van Handel [2015] (lemma 4.3, pag.33). Remark that a pseudometric $m(\cdot,\cdot)$ on I is a metric on I where it is allowed that $m(i,j) = 0$ even if $i \neq j$. ■

4.2 Graph Filter

The proof of theorem 3.1 consists of three main steps:

- establish an LTV bound between filtering distributions with different initial conditions, under a decay of correlation assumption for the initial distributions. This is the subject of proposition 4.1 in subsection 4.2.1;
- control the approximation error between the Bayes update operator C_t and $\tilde{\mathsf{C}}_t^m$, under decay of correlation assumptions, this is the subject of the propositions in subsection 4.2.2;
- prove that the required decay of correlation assumptions hold uniformly in time, this is the subject of subsection 4.2.3.

These steps are then brought together to complete the proof of theorem 3.1 in subsection 4.2.4.

4.2.1 Stability of the filter with respect to initial distributions

Definition 4.5. Given $v, v' \in V$ and μ probability distribution on $\mathbb{X}^V$ define the quantity:

$$C^{\mu}_{v,v'} := \frac{1}{2} \sup_{x,\hat{x}\in\mathbb{X}^V : x^{V\setminus v'} = \hat{x}^{V\setminus v'}} \left\| \mu^v_x - \mu^v_{\hat{x}} \right\|.$$

Then for a fixed $\beta > 0$:

$$\mathrm{Corr}(\mu,\beta) := \max_{v\in V} \sum_{v'\in V} e^{\beta d(v,v')} C^{\mu}_{v,v'}.$$

Definition 4.6. Given $v, v' \in V$ and μ probability distribution on $\mathbb{X}^V$ define the quantity:

$$\tilde{C}^{\mu}_{v,v'} := \frac{1}{2} \sup_{z\in\mathbb{X}^V} \sup_{\substack{x,\hat{x}\in\mathbb{X}^V : \\ x^{V\setminus v'} = \hat{x}^{V\setminus v'}}} \left\| \mu^v_{x,z} - \mu^v_{\hat{x},z} \right\|.$$

Then for a fixed $\beta > 0$:

$$\widetilde{\mathrm{Corr}}(\mu,\beta) := \max_{v\in V} \sum_{v'\in V} e^{\beta d(v,v')} C^{\mu}_{v,v'}.$$

Proposition 4.1. *Fix any collection of observations $\{y_1, \ldots, y_T\}$ and any partition $\mathscr{K}$ of V. Suppose that there exists $(\epsilon_-, \epsilon_+) \in (0,1)^2$ and $\kappa \in (0,1)$ such that:*

$$\epsilon_- \le p^v(x^v, z^v) \le \epsilon_+ \quad \mathit{and} \quad \kappa \le g^f(x^{N(f)}, y_t) \le \frac{1}{\kappa},$$

for all $x \in \mathbb{X}^V, z \in \mathbb{X}^V, v \in V, f \in F, t \in \{1, \ldots, T\}$. Let μ, ν two probability distributions on $\mathbb{X}^V$, and assume there exists $\beta > 0$ such that:

$$\widetilde{\mathrm{Corr}}(\mu,\beta) + 2e^{\beta}\left(1 - \frac{\epsilon_-}{\epsilon_+}\right) + e^{2\beta}\Upsilon^{(2)}\left(1 - \kappa^{2\tilde{\Upsilon}}\right) \le \frac{1}{2},$$

then for all $t \in \{0, \ldots, T\}, K \in \mathscr{K}$ and $J \subseteq K$:

$$\left\| \mathsf{F}_T \ldots \mathsf{F}_{t+1}\mu - \mathsf{F}_T \ldots \mathsf{F}_{t+1}\nu \right\|_J \le 2e^{-\beta(T-t)} \sum_{v\in J} \max_{v'\in V} \left\{ e^{-\beta d(v,v')} \sup_{x,z\in\mathbb{X}^V} \left\| \mu^{v'}_{x,z} - \nu^{v'}_{x,z} \right\| \right\}.$$

Proof. Define the probability distributions:

$$\rho(A) \propto \int \mathbb{1}_A(x_{0:T}) \left[\prod_{k=1}^{T} p(x_{k-1}, x_k) g(x_k, y_k) \right] \bigotimes_{k=1}^{T} \psi(dx_k)\mu(dx_0),$$

$$\tilde{\rho}(A) \propto \int \mathbb{1}_A(x_{0:T}) \left[\prod_{k=1}^{T} p(x_{k-1}, x_k) g(x_k, y_k) \right] \bigotimes_{k=1}^{T} \psi(dx_k)\nu(dx_0).$$

It can be observed that:

$$\left\| \rho - \tilde{\rho} \right\|_{(T,J)} = \left\| \mathsf{F}_T \ldots \mathsf{F}_1\mu - \mathsf{F}_T \ldots \mathsf{F}_1\nu \right\|_J,$$

and the proof proceeds by applying the Dobrushin theorem (theorem 4.1) to $\rho, \tilde{\rho}$ where the index set is given by $I = \bigcup_{t=0}^{T}(t, V)$ and the subset is (T, J).

The first step is to bound $C_{i,j}$ for all the possible combination of $i,j \in I$, as in (4.2) of theorem 4.1, i.e.:

$$C_{i,j} = \frac{1}{2} \sup_{\substack{x,\hat{x}\in\mathbb{X}^I: \\ x^{I\setminus j}=\hat{x}^{I\setminus j}}} \left\| \rho_x^i - \rho_{\hat{x}}^i \right\|.$$

In the following passages consider $x = (x_0, \dots, x_T)$, where $x_t \in \mathbb{X}^V$.

- Consider $i = (0,v)$ and $v \in V$ then:

$$\begin{aligned}
\rho_x^{(0,v)}(A) &= \frac{\int \mathbb{1}_A(\tilde{x}_0^v)\mathbb{1}_{\{x_0^{V\setminus v},x_{1:T}\}}(\tilde{x}_0^{V\setminus v},\tilde{x}_{1:T}) \prod_{k=1}^{T} p(\tilde{x}_{k-1},\tilde{x}_k)g(\tilde{x}_k,y_k) \bigotimes_{k=1}^{T} \psi(d\tilde{x}_k)\mu(d\tilde{x}_0)}{\int \mathbb{1}_{\mathbb{X}}(\tilde{x}_0^v)\mathbb{1}_{\{x_0^{V\setminus v},x_{1:T}\}}(\tilde{x}_0^{V\setminus v},\tilde{x}_{1:T}) \prod_{k=1}^{T} p(\tilde{x}_{k-1},\tilde{x}_k)g(\tilde{x}_k,y_k) \bigotimes_{k=1}^{T} \psi(d\tilde{x}_k)\mu(d\tilde{x}_0)} \\
&= \frac{g(x_1,y_1)\prod_{k=2}^{T} p(x_{k-1},x_k)g(x_k,y_k)}{g(x_1,y_1)\prod_{k=2}^{T} p(x_{k-1},x_k)g(x_k,y_k)} \frac{\int \mathbb{1}_A(\tilde{x}_0^v)\mathbb{1}_{\{x_0^{V\setminus v},x_1\}}(\tilde{x}_0^{V\setminus v},\tilde{x}_1)p(\tilde{x}_0,\tilde{x}_1)\mu(d\tilde{x}_0)}{\int \mathbb{1}_{\mathbb{X}}(\tilde{x}_0^v)\mathbb{1}_{\{x_0^{V\setminus v},x_1\}}(\tilde{x}_0^{V\setminus v},\tilde{x}_1)p(\tilde{x}_0,\tilde{x}_1)\mu(d\tilde{x}_0)} \\
&= \frac{\int \mathbb{1}_A(\tilde{x}_0^v)p^v(\tilde{x}_0^v,x_1^v)\mu_{x_0}^v(d\tilde{x}_0^v)}{\int p^v(\tilde{x}_0^v,x_1^v)\mu_{x_0}^v(d\tilde{x}_0^v)} = \mu_{x_0,x_1}^v(A),
\end{aligned}$$

  where the last passage follows from the factorization of the kernel and the definition of μ_x^v. The next step is to distinguish the different cases in which ρ_x^i can differ from $\rho_{\tilde{x}}^i$, where $x^{I\setminus j} = \tilde{x}^{I\setminus j}$.

  - If $j = (0,v')$ and $v' \in V$ then: $\quad C_{i,j} \leq \tilde{C}_{v,v'}^{\mu}$.

  - If $j = (1,v')$ and $v' \in V$ then: $\quad C_{i,j} \leq \begin{cases} \left(1 - \frac{\epsilon_-}{\epsilon_+}\right) & v' = v \\ 0 & v' \neq v \end{cases},$
    where the result follows from lemma 4.3, obtained by a majorization of the kernel part.

  - If $j = (k,v')$ with $k > 1$ and $v' \in V$ then: $\quad C_{i,j} = 0$,
    because in ρ_x^i there is no dependence on x_t with $t > 1$.

- Consider $i = (t,v)$ with $0 < t < T$ and $v \in V$ define:

$$x_t^{N(f)\setminus v} := (\tilde{x}_t^v, x_t^{N(f)\setminus v}), \quad \tilde{x}_{0:T\setminus t} := (\tilde{x}_{0:t-1}, \tilde{x}_{t+1:T}) \quad \text{and} \quad x_{0:T\setminus t} := (x_{0:t-1}, x_{t+1:T})$$

then:

$$
\begin{aligned}
\rho_x^{(t,v)}(A) &= \frac{\int \mathbb{1}_A(\tilde{x}_t^v)\mathbb{1}_{\{x_t^{V\setminus v},x_{0:T\setminus t}\}}(\tilde{x}_t^{V\setminus v},\tilde{x}_{0:T\setminus t}) \prod_{k=1}^{T} p(\tilde{x}_{k-1},\tilde{x}_k)g(\tilde{x}_k,y_k) \bigotimes_{k=1}^{T} \psi(d\tilde{x}_k)\mu(d\tilde{x}_0)}{\int \mathbb{1}_{\mathbb{X}}(\tilde{x}_t^v)\mathbb{1}_{\{x_t^{V\setminus v},x_{0:T\setminus t}\}}(\tilde{x}_t^{V\setminus v},\tilde{x}_{0:T\setminus t}) \prod_{k=1}^{T} p(\tilde{x}_{k-1},\tilde{x}_k)g(\tilde{x}_k,y_k) \bigotimes_{k=1}^{T} \psi(d\tilde{x}_k)\mu(d\tilde{x}_0)} \\
&= \frac{\prod_{k\neq t} g(x_k,y_k) \prod_{k\neq t,t+1} p(x_{k-1},x_k)}{\prod_{k\neq t} g(x_k,y_k) \prod_{k\neq t,t+1} p(x_{k-1},x_k)} \\
&\quad \frac{\int \mathbb{1}_A(\tilde{x}_t^v)\mathbb{1}_{\{x_t^{V\setminus v},x_{0:T\setminus t}\}}(\tilde{x}_t^{V\setminus v},\tilde{x}_{0:T\setminus t})p(\tilde{x}_{t-1},\tilde{x}_t)p(\tilde{x}_t,\tilde{x}_{t+1})g(\tilde{x}_t,y_t) \bigotimes_{k=1}^{T} \psi(d\tilde{x}_k)\mu(d\tilde{x}_0)}{\int \mathbb{1}_{\mathbb{X}}(\tilde{x}_t^v)\mathbb{1}_{\{x_t^{V\setminus v},x_{0:T\setminus t}\}}(\tilde{x}_t^{V\setminus v},\tilde{x}_{0:T\setminus t})p(\tilde{x}_{t-1},\tilde{x}_t)p(\tilde{x}_t,\tilde{x}_{t+1})g(\tilde{x}_t,y_t) \bigotimes_{k=1}^{T} \psi(d\tilde{x}_k)\mu(d\tilde{x}_0)} \\
&= \frac{\int \mathbb{1}_A(\tilde{x}_t^v)p^v(x_{t-1}^v,\tilde{x}_t^v)p^v(\tilde{x}_t^v,x_{t+1}^v)\prod_{f\in N(v)} g^f(x_t^{N(f)\setminus v},y_t)\psi^v(d\tilde{x}_t^v)}{\int p^v(x_{t-1}^v,\tilde{x}_t^v)p^v(\tilde{x}_t^v,x_{t+1}^v)\prod_{f\in N(v)} g^f(x_t^{N(f)\setminus v},y_t)\psi^v(d\tilde{x}_t^v)},
\end{aligned}
$$

where the last passage follows from the factorization of the kernel and the factorial representation of the observation density. The next stage is to distinguish the different cases in which ρ_x^i can differ from $\rho_{\tilde{x}}^i$, where $x^{I\setminus j} = \tilde{x}^{I\setminus j}$.

- If $j = (k,v')$ with $k \leq t-2$ and $v' \in V$ then: $C_{i,j} = 0$.

- If $j = (t-1,v')$ and $v' \in V$ then $C_{i,j} \leq \begin{cases} \left(1-\frac{\epsilon_-}{\epsilon_+}\right) & v' = v \\ 0 & v' \neq v \end{cases}$,
  where the result follows from lemma 4.3, obtained by a majorization of the kernel part.

- If $j = (t,v')$ and $v' \in V$ then $C_{i,j} \leq \begin{cases} \left(1-\kappa^{2\mathbf{card}(N(v)\cap N(v'))}\right) & v' \in N^2(v)\setminus v \\ 0 & \text{otherwise} \end{cases}$,
  where the result follows from lemma 4.3, obtained by a majorization of the observation density part. Recall that the only factors that contain v are the ones in $N(v)$ so the components that are connected to these factors are the ones in $N^2(v)$.

- If $j = (t+1,v')$ and $v' \in V$ then $C_{i,j} \leq \begin{cases} \left(1-\frac{\epsilon_-}{\epsilon_+}\right) & v' = v \\ 0 & v' \neq v \end{cases}$,
  where the result follows from lemma 4.3, obtained by a majorization of the kernel part.

- If $j = (k,v')$ with $k \geq t+2$ and $v' \in V$ then: $C_{i,j} = 0$.

- Consider $i=(T,v)$ and $v \in V$ then:

$$\begin{aligned}
\rho_x^{(T,v)}(A) &= \frac{\int \mathbb{1}_A(\tilde{x}_T^v)\mathbb{1}_{\{x_T^{V\setminus v},x_{0:T-1}\}}(\tilde{x}_T^{V\setminus v},\tilde{x}_{0:T-1}) \prod_{k=1}^{T} p(\tilde{x}_{k-1},\tilde{x}_k)g(\tilde{x}_k,y_k) \bigotimes_{k=1}^{T} \psi(d\tilde{x}_k)\mu(d\tilde{x}_0)}{\int \mathbb{1}_{\mathbb{X}}(\tilde{x}_T^v)\mathbb{1}_{\{x_T^{V\setminus v},x_{0:T-1}\}}(\tilde{x}_T^{V\setminus v},\tilde{x}_{0:T-1}) \prod_{k=1}^{T} p(\tilde{x}_{k-1},\tilde{x}_k)g(\tilde{x}_k,y_k) \bigotimes_{k=1}^{T} \psi(d\tilde{x}_k)\mu(d\tilde{x}_0)} \\
&= \frac{\prod_{k=1}^{T-1} g(x_k,y_k) \prod_{k=0}^{T-1} p(x_{k-1},x_k)}{\prod_{k=1}^{T-1} g(x_k,y_k) \prod_{k=0}^{T-1} p(x_{k-1},x_k)} \\
&\quad \frac{\int\int \mathbb{1}_A(\tilde{x}_T^v)\mathbb{1}_{\{x_T^{V\setminus v},x_{0:T-1}\}}(\tilde{x}_T^{V\setminus v},\tilde{x}_{0:T-1})p(\tilde{x}_{T-1},\tilde{x}_T)g(\tilde{x}_T,y_T) \bigotimes_{k=1}^{T} \psi(d\tilde{x}_k)\mu(d\tilde{x}_0)}{\int\int \mathbb{1}_{\mathbb{X}}(\tilde{x}_T^v)\mathbb{1}_{\{x_T^{V\setminus v},x_{0:T-1}\}}(\tilde{x}_T^{V\setminus v},\tilde{x}_{0:T-1})p(\tilde{x}_{T-1},\tilde{x}_T)g(\tilde{x}_T,y_T) \bigotimes_{k=1}^{T} \psi(d\tilde{x}_k)\mu(d\tilde{x}_0)} \\
&= \frac{\int \mathbb{1}_A(\tilde{x}_T^v)p(x_{T-1}^v,\tilde{x}_T^v)\prod_{f\in N(v)} g^f(x_T^{N(f)\setminus v},y_T)\psi^v(dx_T^v)}{\int p(x_{T-1}^v,\tilde{x}_T^v)\prod_{f\in N(v)} g^f(x_T^{N(f)\setminus v},y_T)\psi^v(d\tilde{x}_T^v)}.
\end{aligned}$$

  - If $j=(k,v')$ with $k \le T-2$ and $v' \in V$ then: $C_{i,j}=0$.
  - If $j=(T-1,v')$ and $v' \in V$ then: $C_{i,j} \le \begin{cases} \left(1-\frac{\epsilon_-}{\epsilon_+}\right) & v'=v \\ 0 & v' \neq v \end{cases}$,
    where the result follows from lemma 4.3, obtained by a majorization of the kernel part.
  - If $j=(T,v')$ and $v' \in V$ then: $C_{i,j} \le \begin{cases} \left(1-\kappa^{2\mathbf{card}(N(v)\cap N(v'))}\right) & v' \in N^2(v)\setminus v \\ 0 & \text{otherwise} \end{cases}$,
    where the result follows from lemma 4.3, obtained by a majorization of the observation density part.

Given the previous results, for any $v \in K$:

$$\sum_{j\in I} e^{m(i,j)}C_{i,j} \le \begin{cases} \sum_{v'\in V} e^{\beta d(v,v')}\tilde{C}^{\mu}_{v,v'} + e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right) & i=(0,v) \\ 2e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right) + \sum_{v'\in N^2(v)} (1-\kappa^{2\mathbf{card}(N(v)\cap N(v'))})e^{\beta d(v,v')} & i=(t,v) \\ e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right) + \sum_{v'\in N^2(v)} (1-\kappa^{2\mathbf{card}(N(v)\cap N(v'))})e^{\beta d(v,v')} & i=(T,v) \end{cases}$$

where $m(i,j)=\beta|k-k'|+\beta d(v,v')$ for $i=(k,v)$ and $j=(k',v')$ with $k,k' \in \{0,\dots,T\}$ and $v,v' \in V$ is the pseudometric of interest on the index set I. But then by combining the above calculation with the assumption:

$$\begin{aligned}
\max_{i\in I}\sum_{j\in I} C_{i,j} &\le \max_{i\in I}\sum_{j\in I} e^{m(i,j)}C_{i,j} \\
&\le \widetilde{\mathrm{Corr}}(\mu,\beta) + 2e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right) + \left(1-\kappa^{2\tilde{\Upsilon}}\right)\max_{v\in V}\sum_{v'\in N^2(v)} e^{\beta d(v,v')} \\
&= \widetilde{\mathrm{Corr}}(\mu,\beta) + 2e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right) + e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right) \le \frac{1}{2}.
\end{aligned}$$

Given that $\sum_{j\in I} C_{i,j} \le \sum_{j\in I} e^{m(i,j)} C_{i,j}$ then the Dobrushin theorem (theorem 4.1) can be applied, meaning that:

$$\left\|\mathsf{F}_T \dots \mathsf{F}_1 \mu - \mathsf{F}_T \dots \mathsf{F}_1 \nu\right\|_J = \left\|\rho - \tilde{\rho}\right\|_{(T,J)} \le \sum_{v\in J}\sum_{j\in I} D_{(T,v),j} b_j.$$

The second step is to control the quantities b_j, as in (4.2) of theorem 4.1:

$$b_j = \sup_{x\in\mathbb{X}^I} \left\|\rho_x^j - \tilde{\rho}_x^j\right\|.$$

Remark that the form of ρ_x^j is already known from the study on $C_{i,j}$, hence $\tilde{\rho}_x^j$ is the only thing that has to be computed.

- If $j = (0, v')$ and $v' \in V$ then:

  $$\tilde{\rho}_x^j(A) = \frac{\int \mathbb{I}_A(x_0^{v'}) p^{v'}(x_0^{v'}, x_1^{v'}) \nu_{x_0}^{v'}(dx_0^{v'})}{\int p^{v'}(x_0^{v'}, x_1^{v'}) \nu_{x_0}^{v'}(dx_0^{v'})} = \nu_{x_0,x_1}^{v}(A),$$

  where the procedure is exactly the same as for $\tilde{\rho}_x^{(0,v')}(A)$ with ν rather than μ, hence:

  $$b_j = \sup_{x_0,x_1\in\mathbb{X}^V} \left\|\mu_{x_0,x_1}^{v'} - \nu_{x_0,x_1}^{v'}\right\|.$$

- If $j = (k', v')$ with $k' \ge 1$ and $v' \in V$ then:

  $$\rho_x^j(A) = \tilde{\rho}_x^j(A),$$

  because the difference is only on the initial distribution which disappear as consequence of the Markov property, hence:

  $$b_j = 0.$$

Moreover, given that $\max_{i\in I} \sum\limits_{j\in I} e^{m(i,j)} C_{i,j} \le \frac{1}{2}$ then lemma 4.5 can be applied and so:

$$\max_{i\in I} \sum_{j\in J} e^{m(i,J)} D_{i,j} \le 2.$$

By joining step one and step two it follows that:

$$\begin{aligned}
&\left\|\mathsf{F}_T \dots \mathsf{F}_1 \mu - \mathsf{F}_T \dots \mathsf{F}_1 \nu\right\|_J \\
&\quad \le \sum_{v\in J}\sum_{j\in I} D_{(T,v),j} b_j \\
&\quad \le \sum_{v\in J}\sum_{v'\in V} D_{(T,v),(0,v')} b_{(0,v')} \\
&\quad \le \sum_{v\in J}\sum_{v'\in V} e^{\beta|T|+\beta d(v,v')} D_{(T,v),(0,v')} e^{-\beta|T|-\beta d(v,v')} \sup_{x_0,x_1\in\mathbb{X}^V} \left\|\mu_{x_0,x_1}^{v'} - \nu_{x_0,x_1}\right\| \\
&\quad \le 2e^{-\beta T} \sum_{v\in J} \max_{v'\in V} \left\{ e^{-\beta d(v,v')} \sup_{x_0,x_1\in\mathbb{X}^V} \left\|\mu_{x_0,x_1}^{v'} - \nu_{x_0,x_1}\right\| \right\}.
\end{aligned}$$

Given that the above bound depends only on how many times the operator F_t is applied then:

$$\left\|\mathsf{F}_T \dots \mathsf{F}_{t+1}\mu - \mathsf{F}_T \dots \mathsf{F}_{t+1}\nu\right\|_J \leq 2e^{-\beta(T-t)} \sum_{v \in J} \max_{v' \in V} \left\{ e^{-\beta d(v,v')} \sup_{x,z \in \mathbb{X}^V} \left\|\mu^{v'}_{x,z} - \nu^{v'}_{x,z}\right\| \right\}.$$

■

4.2.2 Control on the approximation error

The next objective is to bound the approximation errors:

$$\sup_{x,z \in \mathbb{X}^V} \left\|(\tilde{\mathsf{F}}^m_t \tilde{\pi}_{t-1})^v_{x,z} - (\mathsf{F}_t \tilde{\pi}_{t-1})^v_{x,z}\right\|, \quad t < T,$$
$$\left\|\tilde{\mathsf{F}}^m_T \tilde{\pi}_{t-1} - \mathsf{F}_t \tilde{\pi}_{t-1}\right\|_J, \quad t = T,$$

or equivalently:

$$\sup_{x,z \in \mathbb{X}^V} \left\|(\tilde{\mathsf{C}}^m_t \mu)^v_{x,z} - (\mathsf{C}_t \mu)^v_{x,z}\right\|, \quad t < T,$$
$$\left\|\tilde{\mathsf{C}}^m_t \mu - \mathsf{C}_t \mu\right\|_J, \quad t = T,$$

where $\mu = \mathsf{P}\tilde{\pi}_{t-1}$.

Proposition 4.2. *(case: $t < T$) Fix any collection of observations $\{y_1, \dots, y_T\}$ and any partition $\mathscr{K}$ of V. Suppose that there exist $\kappa \in (0,1)$ such that:*

$$\kappa \leq g^f\left(x^{N(f)}, y_t\right) \leq \frac{1}{\kappa},$$

for all $x \in \mathbb{X}^V, f \in F, t \in \{1, \dots, T\}$ and $p^v(x^v, z^v) > 0$ for all $x, z \in \mathbb{X}^V, v \in V$. Let μ be a probability distribution on $\mathbb{X}^V$ such that $\mu = \bigotimes_{K \in \mathscr{K}} \mu^K$ and assume that there exists $\beta > 0$ such that:

$$2\kappa^{-2\Upsilon}\mathrm{Corr}(\mu, \beta) + 2e^{2\beta}\Upsilon^{(2)}\left(1 - \kappa^{2\tilde{\Upsilon}}\right) \leq \frac{1}{2}.$$

Then for a fixed $t \in \{1, \dots, T-1\}$, $K \in \mathscr{K}$, $v \in K$ and $m \in \{0, \dots, n\}$:

$$\sup_{x,z \in \mathbb{X}^V} \left\|(\tilde{\mathsf{C}}^m_t \mu)^v_{x,z} - (\mathsf{C}_t \mu)^v_{x,z}\right\| \leq 2\left(1 - \kappa^{a(\mathscr{K})}\right) + 4e^{-\beta m}\left(1 - \kappa^{b(\mathscr{K}, m)}\right),$$

where

$$a(\mathscr{K}) := 2 \max_{K \in \mathscr{K}} \max_{v \in \partial K} \mathbf{card}(N(v) \cap \partial N(K)),$$
$$b(m, \mathscr{K}) := 2 \max_{K \in \mathscr{K}} \max_{v \notin N^{m-1}_v(K)} \mathbf{card}(N(v)).$$

Proof. Remark that if $\mu = \bigotimes_{K' \in \mathscr{K}} \mu^{K'}$ and $v \in K$ with $K \in \mathscr{K}$ then from lemma 4.2:

$$\begin{aligned}(\tilde{\mathsf{C}}^m_t \mu)^v_{x,z}(A) &= \frac{\int \mathbb{1}_A(x^v) \int \prod_{f \in N^m_f(K)} g^f(x^{N(f)}_K, y_t) \mu^{V \setminus K}(d\tilde{x}) p^v(x^v, z^v) \mu^v_x(dx^v)}{\int \prod_{f \in N^m_f(K)} g^f(x^{N(f)}_K, y_t) \mu^{V \setminus K}(d\tilde{x}) p^v(x^v, z^v) \mu^v_x(dx^v)} \\ &= (\tilde{\mathsf{C}}^{m,K}_t \mu)^v_{x,z}(A),\end{aligned}$$

where $x, z \in \mathbb{X}^V$.

Given $v \in K$ with $K \in \mathcal{K}$, it must be noticed that two cases can be distinguished: v connected with factors that are connected only with elements inside K (using the thesis' notation $v \in \tilde{K}$) and its complement (exists a factor connected with v that is connected with elements outside K).

Consider the case $v \in \widetilde{K}$, then $N(v)$ are factors that depend only on components in K, so for $x, z \in \mathbb{X}^V$:

$$\begin{aligned}
&(\tilde{\mathsf{C}}_t^{m,K}\mu)_{x,z}^v(A)\\
&= \frac{\int \mathbb{1}_A(x^v)\prod_{f\in N(v)} g^f(x^{N(f)},y_t)\int \prod_{f\in N_f^m(K)\setminus N(v)} g^f(x_K^{N(f)},y_t)\mu^{V\setminus K}(d\tilde{x})p^v(x^v,z^v)\mu_x^v(dx^v)}{\int \prod_{f\in N(v)} g^f(x^{N(f)},y_t)\int \prod_{f\in N_f^m(K)\setminus N(v)} g^f(x_K^{N(f)},y_t)\mu^{V\setminus K}(d\tilde{x})p^v(x^v,z^v)\mu_x^v(dx^v)}\\
&= \frac{\int \prod_{f\in N_f^m(K)\setminus N(v)} g^f(x_K^{N(f)},y_t)\mu^{V\setminus K}(d\tilde{x})}{\int \prod_{f\in N_f^m(K)\setminus N(v)} g^f(x_K^{N(f)},y_t)\mu^{V\setminus K}(d\tilde{x})}\frac{\int \mathbb{1}_A(x^v)\prod_{f\in N(v)} g^f(x^{N(f)},y_t)p^v(x^v,z^v)\mu_x^v(dx^v)}{\int \prod_{f\in N(v)} g^f(x^{N(f)},y_t)p^v(x^v,z^v)\mu_x^v(dx^v)}\\
&= \frac{\int \mathbb{1}_A(x^v)\prod_{f\in N(v)} g^f(x^{N(f)},y_t)p^v(x^v,z^v)\mu_x^v(dx^v)}{\int \prod_{f\in N(v)} g^f(x^{N(f)},y_t)p^v(x^v,z^v)\mu_x^v(dx^v)} = (\mathsf{C}_t\mu)_{x,z}^v(A),
\end{aligned}$$

where the last passage follows because all the dependencies on v inside the previous integral are removed. From the above procedure it has been proved that $(\tilde{\mathsf{C}}_t^{m,K}\mu)_{x,z}^v(A) = (\mathsf{C}_t\mu)_{x,z}^v(A)$ for $v \in \widetilde{K}$ and so:

$$\sup_{x,z\in\mathbb{X}^V} \left\| (\tilde{\mathsf{C}}_t^m\mu)_{x,z}^v - (\mathsf{C}_t\mu)_{x,z}^v \right\| = 0,$$

which proves the statement for $v \in \tilde{K}$.

Consider now the case $v \in \partial K$, using the triangular inequality:

$$\left\| (\tilde{\mathsf{C}}_t^m\mu)_{x,z}^v - (\mathsf{C}_t\mu)_{x,z}^v \right\| \leq \left\| (\tilde{\mathsf{C}}_t^m\mu)_{x,z}^v - (\tilde{\mathsf{C}}_t^n\mu)_{x,z}^v \right\| + \left\| (\tilde{\mathsf{C}}_t^n\mu)_{x,z}^v - (\mathsf{C}_t\mu)_{x,z}^v \right\|,$$

where $n := \max_{K\in\mathcal{K}} n_K$.

Firstly, $\left\| (\tilde{\mathsf{C}}_t^n\mu)_{x,z}^v - (\mathsf{C}_t\mu)_{x,z}^v \right\|$ can be controlled by rewriting $(\tilde{\mathsf{C}}_t^n\mu)_{x,z}^v$ as an integration of $(\mathsf{C}_t\mu)_{x,z}^v$. Indeed, given that $N_f^n(K) = F$, it is possible to rearrange $(\tilde{\mathsf{C}}_t^n\mu)_{x,z}^v$ as follows:

$$\begin{aligned}
(\tilde{\mathsf{C}}_t^n\mu)_{x,z}^v(A) &= \frac{\int \mathbb{1}_A(x^v)\int \prod_{f\in F} g^f(x_K^{N(f)},y_t)\mu^{V\setminus K}(d\tilde{x})p^v(x^v,z^v)\mu_x^v(dx^v)}{\int \prod_{f\in F} g^f(x_K^{N(f)},y_t)\mu^{V\setminus K}(d\tilde{x})p^v(x^v,z^v)\mu_x^v(dx^v)}\\
&= \frac{\int \mathbb{1}_A(x^v)g(x_K,y_t)p^v(x^v,z^v)\mu_x^v(dx^v)\mu^{V\setminus K}(d\tilde{x})}{\int g(x_K,y_t)p^v(x^v,z^v)\mu_x^v(dx^v)\mu^{V\setminus K}(d\tilde{x})}\\
&= \frac{\int\int \mathbb{1}_A(x^v)g(x_K,y_t)p^v(x^v,z^v)\mu_x^v(dx^v)\frac{\int g(x_K,y_t)p^v(x^v,z^v)\mu_x^v(dx^v)}{\int g(x_K,y_t)p^v(x^v,z^v)\mu_x^v(dx^v)}\mu^{V\setminus K}(d\tilde{x})}{\int g(x_K,y_t)p^v(x^v,z^v)\mu_x^v(dx^v)\mu^{V\setminus K}(d\tilde{x})}\\
&= \int \frac{(\mathsf{C}_t\mu)_{x_K,z}^v(A)\int g(x_K,y_t)p^v(x^v,z^v)\mu_x^v(dx^v)}{\int g(x_K,y_t)p^v(x^v,z^v)\mu_x^v(dx^v)\mu^{V\setminus K}(d\tilde{x})}\mu^{V\setminus K}(d\tilde{x}),
\end{aligned}$$

where $x,z \in \mathbb{X}^V$. But then:

$$\begin{aligned}
&|(\tilde{\mathsf{C}}_t^n \mu)^v_{x,z}(A) - (\mathsf{C}_t\mu)^v_{x,z}(A)| \\
&\quad = |\int \frac{[(\mathsf{C}_t\mu)^v_{x_K,z}(A) - (\mathsf{C}_t\mu)^v_{x,z}(A)] \int g(x_K, y_t) p^v(x^v, z^v) \mu^v_x(dx^v)}{\int g(x_K, y_t) p^v(x^v, z^v) \mu^v_x(dx^v) \mu^{V\setminus K}(d\tilde{x})} \mu^{V\setminus K}(d\tilde{x})| \\
&\quad \leq \sup_{\substack{\hat{x}, x \in \mathbb{X}^{V\setminus v}: \\ \hat{x}^K = x^K}} |(\mathsf{C}_t\mu)^v_{\hat{x},z}(A) - (\mathsf{C}_t\mu)^v_{x,z}(A)|.
\end{aligned}$$

But given that A is arbitrary it follows:

$$\left\|(\tilde{\mathsf{C}}_t^n \mu)^v_{x,z} - (\mathsf{C}_t\mu)^v_{x,z}\right\| \leq \sup_{\substack{\hat{x}, x \in \mathbb{X}^{V\setminus v}: \\ \hat{x}^K = x^K}} \left\|(\mathsf{C}_t\mu)^v_{\hat{x},z}(A) - (\mathsf{C}_t\mu)^v_{x,z}(A)\right\|.$$

Note that the supremum is constrained on $\hat{x}^K = x^K$ meaning that the factors in $N(v)$ calling elements outside K can be removed:

$$\begin{aligned}
(\mathsf{C}_t\mu)^v_{\hat{x},z}(A) &\geq \frac{\int \mathbb{I}_A(\hat{x}^v) \prod_{f \in N(v) \cap \partial N(K)} g^f(\hat{x}^{N(f)}, y_t) \prod_{f \in N(v) \setminus \partial N(K)} g^f(\hat{x}^{N(f)}, y_t) p^v(\hat{x}^v, z^v) \mu^v_{\hat{x}}(d\hat{x}^v)}{\int \prod_{f \in N(v) \cap \partial N(K)} g^f(\hat{x}^{N(f)}, y_t) \prod_{f \in N(v) \setminus \partial N(K)} g^f(\hat{x}^{N(f)}, y_t) p^v(\hat{x}^v, z^v) \mu^v_{\hat{x}}(d\hat{x}^v)} \\
&\geq \kappa^{2\mathbf{card}(N(v) \cap \partial N(K))} \frac{\int \mathbb{I}_A(\hat{x}^v) \prod_{f \in N(v) \setminus \partial N(K)} g^f(\hat{x}^{N(f)}, y_t) p^v(\hat{x}^v, z^v) \mu^v_{\hat{x}}(d\hat{x}^v)}{\int \prod_{f \in N(v) \setminus \partial N(K)} g^f(\hat{x}^{N(f)}, y_t) p^v(\hat{x}^v, z^v) \mu^v_{\hat{x}}(d\hat{x}^v)},
\end{aligned}$$

where the majorization follows from the assumption on the kernel density. The procedure can be repeated for $(\mathsf{C}_t\mu)^v_{x,z}$ and the same inequality is obtained, because all the differences between x and $\hat{x}$ are outside K. Hence by applying lemma 4.3:

$$\begin{aligned}
\left\|(\tilde{\mathsf{C}}_t^n \mu)^v_{x,z} - (\mathsf{C}_t\mu)^v_{x,z}\right\| &\leq 2\left(1 - \kappa^{2\mathbf{card}(N(v) \cap \partial N(K))}\right) \leq 2\ \left(1 - \kappa^{2 \max_{K \in \mathcal{K}} \max_{v \in \partial K} \mathbf{card}(N(v) \cap \partial N(K))}\right) \\
&\leq 2\left(1 - \kappa^{a(\mathcal{K})}\right).
\end{aligned}$$

Secondly, the control $\left\|(\tilde{\mathsf{C}}_t^m \mu)^v_{x,z} - (\tilde{\mathsf{C}}_t^n \mu)^v_{x,z}\right\|$ must be done, to do so, the Dobrushin theorem can be used. Define the probability distributions:

$$\rho(A) := \frac{\int \mathbb{I}_A(x_K^{V\setminus K}, x^K) g(x_K, y_t) p^v(x^v, z^v) \mu^{V\setminus K}(d{x_K}^{V\setminus K}) \mu^K(dx^K)}{\int g(x_K, y_t) p^v(x^v, z^v) \mu^{V\setminus K}(d{x_K}^{V\setminus K}) \mu^K(dx^K)},$$

$$\tilde{\rho}(A) := \frac{\int \mathbb{I}_A(x_K^{V\setminus K}, x^K) \prod_{f \in N_f^m(K)} g(x_K^{N(f)}, y_t) p^v(x^v, z^v) \mu^{V\setminus K}(d{x_K}^{V\setminus K}) \mu^K(dx^K)}{\int \prod_{f \in N_f^m(K)} g(x_K^{N(f)}, y_t) p^v(x^v, z^v) \mu^{V\setminus K}(d{x_K}^{V\setminus K}) \mu^K(dx^K)},$$

where $x,z \in \mathbb{X}^V$. It can be observed that by construction:

$$\left\|\rho - \tilde{\rho}\right\|_{(1,v)} = \left\|(\tilde{\mathsf{C}}_t^m \mu)^v_{x,z} - (\tilde{\mathsf{C}}_t^n \mu)^v_{x,z}\right\|,$$

meaning that the Dobrushin theorem can be applied to $\rho, \tilde{\rho}$ where the index set is $I = (0, V \setminus K) \cup (1, K)$ and the subset is $(1, v)$. Remark that in this case the first number has not a meaning of time, but they are just some indexes that distinguish the spaces.

The first step is to bound $C_{i,j}$ for all the possible combination of $i,j \in I$, as in (4.2) of theorem 4.1.

- Consider $i=(0,b)$ and $b \in V \setminus K$ with $b \in K'$ then:

$$\begin{aligned}
\rho^{(0,b)}_{x^{V\setminus K},x^K}(A) &= \frac{\int \mathbb{1}_A(\tilde{x}^b)\mathbb{1}_{\{x^{V\setminus K\setminus b},x^K\}}(\tilde{x}^{V\setminus K\setminus b},\tilde{x}^K)g(\tilde{x},y_t)p^v(\tilde{x}^v,z^v)\mu^{V\setminus K}(d\tilde{x}^{V\setminus K})\mu^K(d\tilde{x}^K)}{\int \mathbb{1}_A(\tilde{x}^b)\mathbb{1}_{\{x^{V\setminus K\setminus b},x^K\}}(\tilde{x}^{V\setminus K\setminus b},\tilde{x}^K)g(\tilde{x},y_t)p^v(\tilde{x}^v,z^v)\mu^{V\setminus K}(d\tilde{x}^{V\setminus K})\mu^K(d\tilde{x}^K)} \\
&= \frac{\prod_{f\in F\setminus N(b)} g(x^{N(b)},y_t)}{\prod_{f\in F\setminus N(b)} g(x^{N(b)},y_t)} \\
&\quad \frac{\int \mathbb{1}_A(\tilde{x}^b)\mathbb{1}_{\{x^{V\setminus K\setminus b},x^K\}}(\tilde{x}^{V\setminus K\setminus b},\tilde{x}^K)\prod\limits_{f\in N(b)} g(\tilde{x}^{N(b)},y_t)p^v(\tilde{x}^v,z^v)\mu^{V\setminus K}(d\tilde{x}^{V\setminus K})}{\int \mathbb{1}_{\mathbb{X}}(\tilde{x}^b)\mathbb{1}_{\{x^{V\setminus K\setminus b},x^K\}}(\tilde{x}^{V\setminus K\setminus b},\tilde{x}^K)\prod\limits_{f\in N(b)} g(\tilde{x}^{N(b)},y_t)p^v(\tilde{x}^v,z^v)\mu^{V\setminus K}(d\tilde{x}^{V\setminus K})} \\
&= \frac{\int \mathbb{1}_A(x^b)\prod_{f\in N(b)} g^f(x^{N(f)},y_t)\mu^b_x(dx^b)}{\int \prod_{f\in N(b)} g^f(x^{N(f)},y_t)\mu^b_x(dx^b)} \\
&= \frac{\int \mathbb{1}_A(x^b)\prod_{f\in N(b)} g^f(x^{N(f)},y_t)(\mu^{K'})^b_x(dx^b)}{\int \prod_{f\in N(b)} g^f(x^{N(f)},y_t)(\mu^{K'})^b_x(dx^b)}.
\end{aligned}$$

  - If $j=(0,b')$ and $b' \in V \setminus K$ then there are two cases:
    - if $b' \in N^2(b)$ then by lemma 4.3:

$$C_{i,j} \le \left(1-\kappa^{2\mathbf{card}(N(b)\cap N(b'))}\right)+\kappa^{2\mathbf{card}(N(b)\cap N(b'))}C^{\mu}_{b,b'};$$

    - if $b' \notin N^2(b)$ then $\frac{\int \mathbb{1}_A(x)\Lambda(x)\nu(dx)}{\int \Lambda(x)\nu(dx)}$ and by lemma 4.4:

$$C_{i,j} \le 2\kappa^{-2\mathbf{card}(N(b))}C^{\mu}_{b,b'},$$

    where the bound follows from a multiple application of the kernel assumption $\kappa^{\mathbf{card}(N(b))} \le \prod_{f\in N(b)} g^f(x^{N(f)},y_t) \le \kappa^{-\mathbf{card}(N(b))}$.

  - If $j=(1,b')$ and $b' \in K$ then from lemma 4.3:

$$C_{i,j} \le \begin{cases} (1-\kappa^{2\mathbf{card}(N(b)\cap N(b'))}) & \text{if } b' \in N^2(b) \\ 0 & \text{otherwise} \end{cases},$$

  given that $K \neq K'$, because $b \in V \setminus K$, then for sure the only difference is in the factors because the conditional distribution K' depends only on elements inside K'.

- Consider $i=(1,b)$ and $b \in K$ then:

$$\begin{aligned}
\rho^{(1,b)}_x(A) &= \frac{\int \mathbb{1}_A(\tilde{x}^b)\mathbb{1}_{\{x^{V\setminus K},x^{K\setminus b}\}}(\tilde{x}^{V\setminus K\setminus b},\tilde{x}^K)g(\tilde{x},y_t)p^v(\tilde{x}^v,z^v)\mu^{V\setminus K}(d\tilde{x}^{V\setminus K})\mu^K(d\tilde{x}^K)}{\int \mathbb{1}_A(\tilde{x}^b)\mathbb{1}_{\{x^{V\setminus K},x^{K\setminus b}\}}(\tilde{x}^{V\setminus K\setminus b},\tilde{x}^K)g(\tilde{x},y_t)p^v(\tilde{x}^v,z^v)\mu^{V\setminus K}(d\tilde{x}^{V\setminus K})\mu^K(d\tilde{x}^K)} \\
&= \frac{\int \mathbb{1}_A(x^b)\prod_{f\in N(b)} g^f(x^{N(f)},y_t)[p^v(\tilde{x}^v,z^v)]^{\mathbb{1}_b(v)}\mu^b_x(dx^b)}{\int \prod_{f\in N(b)} g^f(x^{N(f)},y_t)[p^v(\tilde{x}^v,z^v)]^{\mathbb{1}_b(v)}\mu^b_x(dx^b)} \\
&= \frac{\int \mathbb{1}_A(x^b)\prod_{f\in N(b)} g^f(x^{N(f)},y_t)[p^v(\tilde{x}^v,z^v)]^{\mathbb{1}_b(v)}(\mu^K)^b_x(dx^b)}{\int \prod_{f\in N(b)} g^f(x^{N(f)},y_t)[p^v(\tilde{x}^v,z^v)]^{\mathbb{1}_b(v)}(\mu^K)^b_x(dx^b)},
\end{aligned}$$

where the procedure is the same as in $i=(0,b)$. Remark $[p^v(\tilde{x}^v,z^v)]^{\mathbb{1}_b(v)}$ can be not considered because if $b=v$ then that variable is integrated out.

- If $j=(0,b')$ and $b'\in V\setminus K$ then from lemma 4.3:

$$C_{i,j}\leq\begin{cases}(1-\kappa^{2\mathbf{card}(N(b)\cap N(b'))}) & b'\in N^2(b)\\ 0 & \text{otherwise}\end{cases},$$

  again the difference can be only on the factors because b' is outside K.

- If $j=(1,b')$ and $b'\in K$ then there are two cases:
  - if $b'\in N^2(b)$ then by lemma 4.3:

$$C_{i,j}\leq\left(1-\kappa^{2\mathbf{card}(N(b)\cap N(b'))}\right)+\kappa^{2\mathbf{card}(N(b)\cap N(b'))}C^{\mu}_{b,b'};$$

  - if $b'\notin N^2(b)$ then by lemma 4.4:

$$C_{i,j}\leq 2\kappa^{-2\mathbf{card}(N(b))}C^{\mu}_{b,b'}.$$

But then:

$$\begin{aligned}\max_{i\in I}\sum_{j\in I}C_{i,j}\leq\max_{i\in I}\sum_{j\in I}e^{m(i,j)}C_{i,j}\leq\max_{b\in V}&\left\{\sum_{b'\in N^2(b)}e^{m((0,b),(0,b'))}\left[\left(1-\kappa^{2\mathbf{card}(N(b)\cap N(b'))}\right)\right.\right.\\ &\left.+\kappa^{2\mathbf{card}(N(b)\cap N(b'))}C^{\mu}_{b,b'}\right]+\sum_{b'\notin N^2(b)}e^{m((0,b),(0,b'))}2\kappa^{-2\mathbf{card}(N(b))}C^{\mu}_{b,b'}\\ &\left.+\sum_{b'\in N^2(b)}e^{m((0,b),(1,b'))}\left(1-\kappa^{2\mathbf{card}(N(b)\cap N(b'))}\right)\right\}\\ &\leq 2e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)+\max_{b\in V}\left\{\sum_{b'\in N^2(b)}2\kappa^{-2\mathbf{card}(N(b))}e^{\beta d(b,b')}C^{\mu}_{b,b'}\right.\\ &\left.+\sum_{b'\notin N^2(b)}2\kappa^{-2\mathbf{card}(N(b))}e^{\beta d(b,b')}C^{\mu}_{b,b'}\right\}\\ &\leq 2e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)+2\kappa^{-2\Upsilon}\max_{b\in V}\sum_{b'\in V}e^{\beta d(b,b')}C^{\mu}_{b,b'}\\ &\leq 2\kappa^{-2\Upsilon}\operatorname{Corr}(\mu,\beta)+2e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)\leq\frac{1}{2},\end{aligned}$$

where $m(i,j)=\beta d(v,v')$ for $i=(k,v)$ and $j=(k,v')$ with $v,v'\in V$ and $k,k'\in\{0,1\}$ is the pseudometric of interest on the index set. Hence the Dobrushin theorem applies:

$$\left\|(\tilde{\mathsf{C}}^m_t\mu)^v_{x,z}-(\tilde{\mathsf{C}}^n_t\mu)^v_{x,z}\right\|=\left\|\rho-\tilde{\rho}\right\|_{(1,v)}\leq\sum_{j\in I}D_{(1,v),j}b_j.$$

The second step is to control the quantities b_j, as in (4.2) of theorem 4.1. Recall that if $b'\in N^{m-1}_v(K)$ then $N(b')\subset N^m_f(K)$. Given the form of the conditional distribution of $\tilde{\rho}$ is the same of the conditional distribution of ρ, with a restricted number of factors, then $\tilde{\rho}$ can be analysed at first and then the resulting form can be extended to ρ.

- If $j=(0,b')$ and $b' \in V \setminus K$, then:

$$
\begin{aligned}
&\tilde{\rho}_x^{(0,b')}(A)\\
&= \frac{\int \mathbb{1}_A(\tilde{x}^{b'})\mathbb{1}_{\{x^{V\setminus K\setminus b'},x^K\}}(\tilde{x}^{V\setminus K\setminus b'},\tilde{x}^K)\prod_{f\in N_f^m(K)} g^f(\tilde{x}^{N(f)},y_t)p^v(\tilde{x}^v,z^v)\mu^{V\setminus K}(d\tilde{x}^{V\setminus K})\mu^K(d\tilde{x}^K)}{\int \mathbb{1}_A(\tilde{x}^{b'})\mathbb{1}_{\{x^{V\setminus K\setminus b'},x^K\}}(\tilde{x}^{V\setminus K\setminus b'},\tilde{x}^K)\prod_{f\in N_f^m(K)} g^f(\tilde{x}^{N(f)},y_t)p^v(\tilde{x}^v,z^v)\mu^{V\setminus K}(d\tilde{x}^{V\setminus K})\mu^K(d\tilde{x}^K)}\\
&= \frac{\prod_{f\in N_f^m(K)\setminus N(b')} g(\tilde{x}^{N(b')},y_t)}{\prod_{f\in N_f^m(K)\setminus N(b')} g(\tilde{x}^{N(b')},y_t)}\\
&\quad \frac{\int \mathbb{1}_A(\tilde{x}^{b'})\mathbb{1}_{\{x^{V\setminus K\setminus b'},x^K\}}(\tilde{x}^{V\setminus K\setminus b'},\tilde{x}^K)\prod_{f\in N_f^m(K)\cap N(b')} g(\tilde{x}^{N(b')},y_t)p^v(\tilde{x}^v,z^v)\mu^{V\setminus K}(d\tilde{x}^{V\setminus K})}{\int \mathbb{1}_{\mathbb{X}}(\tilde{x}^{b'})\mathbb{1}_{\{x^{V\setminus K\setminus b'},x^K\}}(\tilde{x}^{V\setminus K\setminus b'},\tilde{x}^K)\prod_{f\in N_f^m(K)\cap N(b')} g(\tilde{x}^{N(b')},y_t)p^v(\tilde{x}^v,z^v)\mu^{V\setminus K}(d\tilde{x}^{V\setminus K})}\\
&= \frac{\int \mathbb{1}_A(\tilde{x}^{b'})\prod_{f\in N_f^m(K)\cap N(b')} g^f(x_K^{N(f)},y_t)\mu_x^{b'}(d\tilde{x}^{b'})}{\int \prod_{f\in N_f^m(K)\cap N(b')} g^f(x_K^{N(f)},y_t)\mu_x^{b'}(d\tilde{x}^{b'})}.
\end{aligned}
$$

Remark that $N(N_v^{m-1}(K)) = N_f^m(K)$ so if $b' \in N_v^{m-1}(K)$ then $N(b') \subseteq N_f^m(K)$, hence by lemma 4.3:

$$
b_j \le \begin{cases} 2(1-\kappa^{2\mathbf{card}(N(b'))}) & b' \notin N_v^{m-1}(K) \\ 0 & \text{otherwise} \end{cases},
$$

where the result follows from the majorization of the observation density in $\rho_x^{(0,b')}$ in the worst case scenario when $N_f^m(K)\cap N(b') = \emptyset$.

- If $j=(1,b')$ and $b' \in K$ then:

$$
\begin{aligned}
&\tilde{\rho}_x^{(1,b')}(A)\\
&= \frac{\int \mathbb{1}_A(\tilde{x}^{b'})\mathbb{1}_{\{x^{V\setminus K},x^{K\setminus b'}\}}(\tilde{x}^{V\setminus K},\tilde{x}^{K\setminus b'})\prod_{f\in N_f^m(K)} g^f(\tilde{x}^{N(f)},y_t)p^v(\tilde{x}^v,z^v)\mu^{V\setminus K}(d\tilde{x}^{V\setminus K})\mu^K(d\tilde{x}^K)}{\int \mathbb{1}_A(\tilde{x}^{b'})\mathbb{1}_{\{x^{V\setminus K},x^{K\setminus b'}\}}(\tilde{x}^{V\setminus K},\tilde{x}^{K\setminus b'})\prod_{f\in N_f^m(K)} g^f(\tilde{x}^{N(f)},y_t)p^v(\tilde{x}^v,z^v)\mu^{V\setminus K}(d\tilde{x}^{V\setminus K})\mu^K(d\tilde{x}^K)}\\
&= \frac{\int \mathbb{1}_A(x^{b'})\prod_{f\in N_f^m(K)\cap N(b')} g^f(x^{N(f)},y_t)\mu_x^{b'}(dx^{b'})}{\int \prod_{f\in N_f^m(K)\cap N(b')} g^f(x^{N(f)},y_t)\mu_x^{b'}(dx^{b'})},
\end{aligned}
$$

where the procedure is the same as in $i=(0,b')$. Given that $b' \in K$ then surely $b' \in N_v^{m-1}(K)$ hence:

$$
b_j = 0.
$$

By putting all together and by applying lemma 4.5:

$$\begin{aligned}\left\|(\tilde{\mathsf{C}}_t^m\mu)_{x,z}^v-(\tilde{\mathsf{C}}_t^n\mu)_{x,z}^v\right\| &\leq \sum_{j\in I} D_{(1,v),j}b_j \leq 2\left(1-\kappa^{2\max_{K\in\mathscr{K}}\{\max_{v\notin N_v^{m-1}(K)}\mathbf{card}(N(v))\}}\right)\\ &\quad \sum_{b'\notin N_v^{m-1}(K)} e^{\beta d(v,b')}e^{-\beta d(v,b')}D_{(1,v),(1,b')}\\ &\leq 4\left(1-\kappa^{b(\mathscr{K},m)}\right)e^{-\beta d(v,V\setminus N_v^{m-1}(K))}\\ &\leq 4e^{-\beta m}\left(1-\kappa^{b(\mathscr{K},m)}\right),\end{aligned}$$

where the last passage follows from $v\in J\subseteq K$ and so:

$$d(v,V\setminus N_v^{m-1}(K))\geq d(K,V\setminus N_v^{m-1}(K))\geq m,$$

indeed the minimum distance between $v\in J$ and $V\setminus N_v^{m-1}(K)$ is bigger than the minimum distance between K and $V\setminus N_v^{m-1}(K)$ given that $J\subseteq K$. At the same time all the element that are far $m-1$ from K (the ones in $N_v^{m-1}(K)$) are removed meaning that the minimum distance is surely m. It can be concluded that:

$$\left\|(\tilde{\mathsf{C}}_t^m\mu)_{x,z}^v-(\mathsf{C}_t\mu)_{x,z}^v\right\|\leq 2(1-\kappa^{a(\mathscr{K})})+4e^{-\beta m}\left(1-\kappa^{b(\mathscr{K},m)}\right).$$

■

Proposition 4.3. *Fix any collection of observations $\{y_1,\ldots,y_T\}$ and any partition $\mathscr{K}$ on the set V. Suppose that there exist $\kappa\in(0,1)$ such that:*

$$\kappa\leq g^f\left(x^{N(f)},y_t\right)\leq\frac{1}{\kappa},$$

for all $x\in\mathbb{X}^V, f\in F, t\in\{1,\ldots,T\}$. Let μ be a probability distribution on $\mathbb{X}^V$ such that $\mu=\bigotimes_{K\in\mathscr{K}}\mu^K$ and assume that there exists $\beta>0$ such that:

$$2\kappa^{-2\Upsilon}\mathrm{Corr}(\mu,\beta)+2e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)\leq\frac{1}{2}.$$

Then for a fixed $t\in\{1,\ldots,T-1\}$, $K\in\mathscr{K}$, $v\in K$ and $m\in\{0,\ldots,n\}$:

$$\sup_{x\in\mathbb{X}^V}\left\|(\tilde{\mathsf{C}}_t^{m,K}\mu)_x^v-(\mathsf{C}_t\mu)_x^v\right\|\leq 2\left(1-\kappa^{a(\mathscr{K})}\right)+4e^{-\beta m}\left(1-\kappa^{b(\mathscr{K},m)}\right),$$

where

$$a(\mathscr{K}):=2\max_{K\in\mathscr{K}}\max_{v\in\partial K}\mathbf{card}(N(v)\cap\partial N(K)),$$

$$b(m,\mathscr{K}):=2\max_{K\in\mathscr{K}}\max_{v\notin N_v^{m-1}(K)}\mathbf{card}(N(v)).$$

Proof. The proof follows the same procedure of proposition 4.2, where the kernel term $p^v(x^v,z^v)$ has been removed. ■

Consider now the case $t = T$, here marginal distributions over a set J are considered, this differs from the previous analysis which focused on conditional distributions. This alternative scenario simplifies the bound compared to the one in proposition 4.2, indeed the dependencies from the components outside K, with $K \in \mathscr{K}$ and $J \subseteq K$, are removed. Moreover, this case can be decoupled from the FHMM scenario, indeed it is a general Bayes update. Note that proposition 4.4 is equivalent to prove proposition 3.1, given that T is completely arbitrary.

Proposition 4.4. *(case: $t = T$) Fix any observation y_T and any partition $\mathscr{K}$ on the set V. Suppose there exists $\kappa \in (0,1)$ such that:*

$$\kappa \leq g^f(x^{N(f)}, y_T) \leq \frac{1}{\kappa},$$

for all $x \in \mathbb{X}^V, f \in F$. Let μ be a probability distribution on $\mathbb{X}^V$ and assume that there exists $\beta > 0$ such that:

$$2\kappa^{-2\Upsilon}\mathrm{Corr}(\mu,\beta) + e^{2\beta}\Upsilon^{(2)}\left(1 - \kappa^{2\tilde{\Upsilon}}\right) \leq \frac{1}{2}.$$

Then for a fixed $v \in K \in \mathscr{K}$, $J \subseteq K$ and $m \in \{0,\dots,n\}$:

$$\left\|\mathsf{C}_T\mu - \tilde{\mathsf{C}}_T^m\mu\right\|_J \leq 4\left(1 - \kappa^{b(m,\mathscr{K})}\right)\mathbf{card}(J)e^{-\beta m},$$

where $b(m,\mathscr{K}) := 2\max_{K \in \mathscr{K}}\max_{v \notin N_v^{m-1}(K)}\mathbf{card}(N(v))$.

Proof. Given that $J \subseteq K$ then:

$$\left\|\mathsf{C}_T\mu - \tilde{\mathsf{C}}_T^m\mu\right\|_J = \left\|(\mathsf{C}_T\mu)^K - \tilde{\mathsf{C}}_T^{m,K}\mu\right\|_J,$$

which is obvious because the comparison is on marginals on J that is equivalent to marginalize first on K and then marginalize again on J. Remark that:

$$(\tilde{\mathsf{C}}_T^{m,K}\mu)(A) = \frac{\int \mathbb{1}_A(x^K)\prod_{f \in N_f^m(K)} g^f(x^{N(f)}, y_T)\mu(dx)}{\int \prod_{f \in N_f^m(K)} g^f(x^{N(f)}, y_T)\mu(dx)},$$

$$(\mathsf{C}_T\mu)^K(A) = \frac{\int \mathbb{1}_A(x^K) g(x, y_T)\mu(dx)}{\int g(x, y_t)\mu(dx)}.$$

Define the probability distributions:

$$\rho(A) := \frac{\int \mathbb{1}_A(x) g(x, y_T)\mu(dx)}{\int g(x, y_t)\mu(dx)},$$

$$\tilde{\rho}(A) := \frac{\int \mathbb{1}_A(x)\prod_{f \in N_f^m(K)} g^f(x^{N(f)}, y_T)\mu(dx)}{\int \prod_{f \in N_f^m(K)} g^f(x^{N(f)}, y_T)\mu(dx)}.$$

It can be observed that by definition:

$$\left\|\rho - \tilde{\rho}\right\|_J = \left\|(\mathsf{C}_T\mu)^K - \tilde{\mathsf{C}}_T^{m,K}\mu\right\|_J,$$

meaning that the Dobrushin theorem can be applied to $\rho, \tilde{\rho}$ where the index set is $I = V$.

The first step is to bound $C_{i,j}$ for all the possible combination of $i,j \in I$, as in (4.2) of theorem 4.1.

- Consider $i = v$ and $v \in V$ then:

$$\begin{aligned}\rho_x^v(A) &= \frac{\int \mathbb{1}_A(\tilde{x}^v)\mathbb{1}_{x^{V\setminus v}}(\tilde{x}^{V\setminus v})\prod_{f\in N(v)} g^f(\tilde{x}^{N(f)},y_T)\prod_{f\in F\setminus N(v)} g^f(\tilde{x}^{N(f)},y_T)\mu(d\tilde{x})}{\int \mathbb{1}_{\mathbb{X}}(\tilde{x}^v)\mathbb{1}_{x^{V\setminus v}}(\tilde{x}^{V\setminus v})\prod_{f\in N(v)} g^f(\tilde{x}^{N(f)},y_T)\prod_{f\in F\setminus N(v)} g^f(\tilde{x}^{N(f)},y_T)\mu(d\tilde{x})}\\ &= \frac{\prod_{f\in F\setminus N(v)} g^f(x^{N(f)},y_T)}{\prod_{f\in F\setminus N(v)} g^f(x^{N(f)},y_T)}\frac{\int \mathbb{1}_A(\tilde{x}^v)\mathbb{1}_{x^{V\setminus v}}(\tilde{x}^{V\setminus v})\prod_{f\in N(v)} g^f(\tilde{x}^{N(f)},y_T)\mu(d\tilde{x})}{\int \mathbb{1}_{\mathbb{X}}(\tilde{x}^v)\mathbb{1}_{x^{V\setminus v}}(\tilde{x}^{V\setminus v})\prod_{f\in N(v)} g^f(\tilde{x}^{N(f)},y_T)\mu(d\tilde{x})}\\ &= \frac{\int \mathbb{1}_A(\tilde{x}^v)\prod_{f\in N(v)} g^f(\tilde{x}^{N(f)},y_T)\mu_x^v(d\tilde{x}^v)}{\int \prod_{f\in N(v)} g^f(\tilde{x}^{N(f)},y_T)\mu_x^v(d\tilde{x}^v)}.\end{aligned}$$

  - If $j = v'$ and $v' \in V$ then two cases can be distinguished:
    - if $v' \in N^2(b)$ then by lemma 4.3:

$$C_{i,j} \le \left(1-\kappa^{2\mathbf{card}(N(v)\cap N(v'))}\right)+\kappa^{2\mathbf{card}(N(v)\cap N(v'))}C^{\mu}_{v,v'}.$$

    - if $v' \notin N^2(v)$ then by lemma 4.4:

$$C_{i,j} \le 2\kappa^{-2\mathbf{card}(N(v))}C^{\mu}_{v,v'}.$$

But then by assumption:

$$\begin{aligned}\max_{i\in I}\sum_{j\in I} C_{i,j} &\le \max_{v\in V}\left\{\sum_{v'\in N^2(v)} e^{m(v,v')}\left[\left(1-\kappa^{2\mathbf{card}(N(v)\cap N(v'))}\right)\right.\right.\\ &\qquad \left.\left.+\kappa^{2\mathbf{card}(N(v)\cap N(v'))}C^{\mu}_{v,v'}\right]+\sum_{v'\notin N^2(v)} e^{m(v,v')}2\kappa^{-2\mathbf{card}(N(v))}C^{\mu}_{v,v'}\right\}\\ &\le 2\kappa^{-2\Upsilon}\mathrm{Corr}(\mu,\beta)+e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)\le\frac{1}{2}\end{aligned}$$

where $m(i,j) = \beta d(v,v')$ is the pseudometric of the index set. Hence the Dobrushin theorem applies:

$$\left\|(\mathsf{C}_T\mu)^K-\tilde{\mathsf{C}}_T^{m,K}\mu\right\|_J = \|\rho-\tilde{\rho}\|_J \le \sum_{i\in J}\sum_{j\in V} D_{i,j}b_j.$$

The second step is to control the quantities b_j, as in (4.2) of theorem 4.1.

- If $j = v'$ and $v' \in V$:

$$\begin{aligned}
&\tilde{\rho}_x^{v'}(A)\\
&= \frac{\int \mathbb{1}_A(\tilde{x}^{v'})\mathbb{1}_{x^{V\setminus v'}}(\tilde{x}^{V\setminus v'}) \prod_{f\in N_f^m(K)\cap N(v')} g^f(\tilde{x}^{N(f)}, y_T) \prod_{f\in N_f^m(K)\setminus N(v')} g^f(\tilde{x}^{N(f)}, y_T)\mu(d\tilde{x})}{\int \mathbb{1}_{\mathbb{X}}(\tilde{x}^{v'})\mathbb{1}_{x^{V\setminus v'}}(\tilde{x}^{V\setminus v'}) \prod_{f\in N_f^m(K)\cap N(v')} g^f(\tilde{x}^{N(f)}, y_T) \prod_{f\in N_f^m(K)\setminus N(v')} g^f(\tilde{x}^{N(f)}, y_T)\mu(d\tilde{x})}\\
&= \frac{\prod_{f\in N_f^m(K)\setminus N(v')} g^f(x^{N(f)}, y_T)}{\prod_{f\in N_f^m(K)\setminus N(v')} g^f(x^{N(f)}, y_T)}\\
&\quad\cdot \frac{\int \mathbb{1}_A(\tilde{x}^{v'})\mathbb{1}_{x^{V\setminus v'}}(\tilde{x}^{V\setminus v'}) \prod_{f\in N_f^m(K)\cap N(v')} g^f(\tilde{x}^{N(f)}, y_T)\mu(d\tilde{x})}{\int \mathbb{1}_{\mathbb{X}}(\tilde{x}^{v'})\mathbb{1}_{x^{V\setminus v'}}(\tilde{x}^{V\setminus v'}) \prod_{f\in N_f^m(K)\cap N(v')} g^f(\tilde{x}^{N(f)}, y_T)\mu(d\tilde{x})}\\
&= \frac{\int \mathbb{1}_A(x^{v}) \prod_{f\in N_f^m(K)\cap N(v')} g^f(x^{N(f)}, y_T)\mu_x^v(dx^{v'})}{\int \prod_{f\in N_f^m(K)\cap N(v')} g^f(x^{N(f)}, y_T)\mu_x^v(dx^{v'})}.
\end{aligned}$$

Remark that $v' \in N_v^{m-1}(K)$ implies $N(v') \subseteq N_f^m(K)$, hence:

$$b_j = \begin{cases} 2(1-\kappa^{2\mathbf{card}(N(v'))}) & v' \in N_v^{m-1}(K) \\ 0 & \text{otherwise} \end{cases},$$

where the bound follows from an application of lemma 4.4.

By putting all together and by using lemma 4.5:

$$\begin{aligned}
\left\|(\mathsf{C}_T\mu)^K - \tilde{\mathsf{C}}_T^{m,K}\mu\right\|_J &\leq \sum_{i\in J}\sum_{j\in V} D_{i,j}b_j \leq 2\left(1-\kappa^{2\max_{K\in\mathcal{K}}\max_{v\notin N_v^{m-1}(K)}\mathbf{card}(N(v))}\right)\sum_{v\in J}\sum_{v'\notin N_v^{m-1}(K)} D_{v,v'}\\
&\leq 4\left(1-\kappa^{2\max_{K\in\mathcal{K}}\max_{v\notin N_v^{m-1}(K)}\mathbf{card}(N(v))}\right)\sum_{v\in J} e^{-\beta d(v,V\setminus N_v^{m-1}(K))}\\
&\leq 4\left(1-\kappa^{b(m,\mathcal{K})}\right)\mathbf{card}(J)e^{-\beta m},
\end{aligned}$$

where the last part follows from the same observation on the distance as in proposition 4.2.

■

Proof. **of Proposition 3.1** An application of proposition 4.4, since the time step T is arbitrary. ■

4.2.3 Decay of correlation

The next step is to prove that the following decay of correlation conditions hold uniformly in t:

$$\widetilde{\mathrm{Corr}}(\tilde{\mathsf{F}}_t^m\tilde{\pi}_{t-1},\beta) + 2e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right) + e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right) \leq \frac{1}{2},$$

$$2\kappa^{-2\Upsilon}\mathrm{Corr}(\mathsf{P}\tilde{\pi}_{t-1},\beta) + 2e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right) \leq \frac{1}{2}.$$

Proposition 4.5. *Suppose there exists $(\epsilon_-, \epsilon_+) \in (0,1)$ such that:*

$$\epsilon_- \le p^v(x^v, z^v) \le \epsilon_+,$$

for all $x, z \in \mathbb{X}^V, v \in V$. Given a probability distribution μ on $\mathbb{X}^V$ assume that there exists $\beta > 0$ such that:

$$\widetilde{\mathrm{Corr}}(\mu, \beta) + e^{\beta}\left(1 - \frac{\epsilon_-}{\epsilon_+}\right) \le \frac{1}{2},$$

then:

$$\mathrm{Corr}(\mathsf{P}\mu, \beta) \le 2\left(1 - \frac{\epsilon_-}{\epsilon_+}\right)e^{-\beta}.$$

Proof. Recall that:

$$(\mathsf{P}\mu)_z^v = \frac{\int \mathbb{I}_A(z^v) p(x,z)\mu(dx)\psi^v(dz^v)}{\int p(x,z)\mu(dx)\psi^v(dz^v)},$$

where ψ is the counting measure. Recall also that:

$$\mathrm{Corr}(\mathsf{P}\mu, \beta) = \max_{v \in V} \sum_{v' \in V} e^{\beta d(v,v')} C_{v,v'}^{\mathsf{P}\mu},$$

where:

$$C_{v,v'}^{\mathsf{P}\mu} = \frac{1}{2} \sup_{\substack{z,\tilde{z} \in \mathbb{X}^V: \\ z^{V \setminus v'} = \hat{z}^{V \setminus v'}}} \left\| (\mathsf{P}\mu)_z^v - (\mathsf{P}\mu)_{\hat{z}}^v \right\|.$$

The strategy is hence to firstly control:

$$\left\| (\mathsf{P}\mu)_z^v - (\mathsf{P}\mu)_{\hat{z}}^v \right\|, \quad \text{with } z^{V \setminus v'} = \hat{z}^{V \setminus v'}$$

and then sum over all v' to find a bound for $\mathrm{Corr}(\mathsf{P}\mu, \beta)$.

Define the probability distributions:

$$\rho(A) := \frac{\int \mathbb{I}_A(x, z^v) p(x,z)\mu(dx)\psi^v(dz^v)}{\int p(x,z)\mu(dx)\psi^v(dz^v)},$$

$$\tilde{\rho}(A) := \frac{\int \mathbb{I}_A(x, \hat{z}^v) p(x,\hat{z})\mu(dx)\psi^v(d\hat{z}^v)}{\int p(x,\hat{z})\mu(dx)\psi^v(d\hat{z}^v)},$$

where $z, \hat{z} \in \mathbb{X}^V$ such that $z^{V \setminus v'} = \hat{z}^{V \setminus v'}$. It can be observed that by definition:

$$\left\| \rho - \tilde{\rho} \right\|_{(1,v)} = \left\| (\mathsf{P}\mu)_z^v - (\mathsf{P}\mu)_{\hat{z}}^v \right\|,$$

meaning that the Dobrushin theorem can be applied to $\rho, \tilde{\rho}$ where the set of index is $I = (0, V) \cup (1, v)$.

The first step is to bound $C_{i,j}$ for all the possible combinations of $i,j \in I$, as in (4.2) of theorem 4.1.

- Consider $i = (0,b)$ and $b \in V$ then:

$$\rho_{x,z}^{(0,b)}(A) = \frac{\int \mathbb{I}_A(\tilde{x}^b)\mathbb{I}_{x^{V\setminus b},\tilde{z}^v}(\tilde{x}^{V\setminus b}, z^v)p(\tilde{x},\tilde{z})\mu(d\tilde{x})\psi^v(d\tilde{z}^v)}{\int \mathbb{I}_{\mathbb{X}}(\tilde{x}^b)\mathbb{I}_{x^{V\setminus b},\tilde{z}^v}(\tilde{x}^{V\setminus b}, z^v)p(\tilde{x},\tilde{z})\mu(d\tilde{x})\psi^v(d\tilde{z}^v)}$$
$$= \frac{\int \mathbb{I}_A(\tilde{x}^b)p^b(\tilde{x}^b, z^b)\mu_x^b(d\tilde{x}^b)}{\int p^b(\tilde{x}^b, z^b)\mu_x^b(d\tilde{x}^b)}.$$

  - If $j = (0,b')$ and $b' \in V$ then:

$$C_{i,j} \leq \tilde{C}^{\mu}_{b,b'}.$$

  - If $j = (1,v)$ then by lemma 4.3:

$$C_{i,j} \leq \begin{cases} \left(1 - \frac{\epsilon_-}{\epsilon_+}\right) & b = v \\ 0 & \text{otherwise} \end{cases},$$

    where the kernel part is upper bounded.

- Consider $i = (1,v)$ then:

$$\rho_{x,z}^{(1,v)}(A) = \frac{\int \mathbb{I}_A(\tilde{z}^v)\mathbb{I}_x(\tilde{x})p(\tilde{x},\tilde{z})\mu(d\tilde{x})\psi^v(d\tilde{z}^v)}{\int \mathbb{I}_{\mathbb{X}}(\tilde{z}^v)\mathbb{I}_x(\tilde{x})p(\tilde{x},\tilde{z})\mu(d\tilde{x})\psi^v(d\tilde{z}^v)}$$
$$= \frac{\int \mathbb{I}_A(\tilde{z}^v)p^v(x^v,\tilde{z}^v)\psi^v(d\tilde{z}^v)}{\int p^v(x^v,\tilde{z}^v)\psi^v(d\tilde{z}^v)}.$$

  - If $j = (0,b')$ and $b' \in V$ then by lemma 4.3:

$$C_{i,j} \leq \begin{cases} \left(1 - \frac{\epsilon_-}{\epsilon_+}\right) & b' = v \\ 0 & \text{otherwise} \end{cases},$$

    where the kernel part is upper bounded.

  - If $j = (1,v)$ then:

$$C_{i,j} = 0.$$

    because the component v is integrated out.

But then:

$$\max_{i\in I} \sum_{j\in I} e^{m(i,j)} C_{i,j} \leq \widetilde{\mathrm{Corr}}(\mu,\beta) + e^{\beta}\left(1 - \frac{\epsilon_-}{\epsilon_+}\right) \leq \frac{1}{2},$$

where $m(i,j) = \beta|k-k'| - \beta d(v,v')$ for $i = (k,v)$ and $j = (k',v')$ with $k,k' \in \{0,1\}$ and $v,v' \in V$ is the pseudometric of interest. Hence the Dobrushin theorem applies:

$$\|(\mathsf{P}\mu)_x^v - (\mathsf{P}\mu)_z^v\| = \|\rho - \tilde{\rho}\|_{(0,v)} \leq \sum_{i\in J}\sum_{j\in V} D_{i,j} b_j.$$

The second step is to control the quantities b_j, as in (4.2) of theorem 4.1. Remark that the conditional distributions of $\tilde{\rho}$ are the same of ρ with $\hat{z}$ instead of z.

- Consider $j = (0, b')$ and $b' \in V$ then:

$$\tilde{\rho}_{x,\hat{z}}^{(0,b')}(A) = \frac{\int \mathbb{1}_A(\tilde{x}^{b'}) p^{b'}(\tilde{x}^{b'}, \hat{z}^{b'}) \mu_x^{b'}(d\tilde{x}^{b'})}{\int p^{b'}(\tilde{x}^{b'}, \hat{z}^{b'}) \mu_x^{b'}(d\tilde{x}^{b'})}.$$

  Then by lemma 4.3:

$$b_j \le \begin{cases} 2\left(1 - \frac{\epsilon_-}{\epsilon_+}\right) & b' = v' \\ 0 & \text{otherwise} \end{cases},$$

  because $\hat{z}^{V \setminus v'} = z^{V \setminus v'}$.

- Consider $j = (1, v)$ then:

$$\tilde{\rho}_{x,\hat{z}}^{(1,v)}(A) = \frac{\int \mathbb{1}_A(\tilde{z}^{v}) p^{v}(x^{v}, \tilde{z}^{v}) \psi^{v}(d\tilde{z}^{v})}{\int p^{v}(x^{v}, \tilde{z}^{v}) \psi^{v}(d\tilde{z}^{v})}.$$

  Then:

$$b_j = 0,$$

  because the variable z is integrated out.

By putting all together:

$$\|(\mathsf{P}\mu)_z^v - (\mathsf{P}\mu)_{\hat{z}}^v\| \le \sum_{j \in V} D_{(1,v),j} b_j \le 2\left(1 - \frac{\epsilon_-}{\epsilon_+}\right) D_{(1,v),(0,v')}.$$

Hence:

$$\begin{aligned} \mathrm{Corr}(\mathsf{P}\mu, \beta) &= \max_{v \in V} \sum_{v' \in V} e^{d(v,v')} C_{v,v'}^{\mathsf{P}\mu} \le \left(1 - \frac{\epsilon_-}{\epsilon_+}\right) e^{-\beta} \max_{v \in V} \sum_{v' \in V} e^{\beta d(v,v') + \beta} D_{(1,v),(0,v')} \\ &\le 2\left(1 - \frac{\epsilon_-}{\epsilon_+}\right) e^{-\beta}. \end{aligned}$$

■

Proposition 4.6. *Fix any collection of observations* $\{y_1, \dots, y_T\}$ *and any partition* $\mathcal{K}$ *on the set* V. *Suppose that there exists* $(\epsilon_-, \epsilon_+) \in (0,1)$ *and* $\kappa \in (0,1)$ *such that:*

$$\epsilon_- \le p^v(x^v, z^v) \le \epsilon_+ \quad \text{and} \quad \kappa \le g^f(x^{N(f)}, y) \le \frac{1}{\kappa},$$

for all $x, z \in \mathbb{X}^V, v \in V, f \in F, t \in \{1, \dots, T\}$. *Let* μ *be a probability distribution on* $\mathbb{X}^V$ *and assume that there exists* $\beta > 0$ *such that:*

$$\widetilde{\mathrm{Corr}}(\mu, \beta) + 2e^{\beta}\left(1 - \frac{\epsilon_-}{\epsilon_+}\right) + e^{2\beta}\Upsilon^{(2)}\left(1 - \kappa^{2\tilde{\Upsilon}}\right) \le \frac{1}{2}.$$

Then for any $t \in \{1, \dots, T\}$ *and* $m \in \{0, \dots, n\}$*:*

$$\widetilde{\mathrm{Corr}}(\tilde{\mathsf{F}}_t^m \mu, \beta) \le 2e^{-\beta}\left(1 - \frac{\epsilon_-}{\epsilon_+}\right) + 2e^{2\beta}\Upsilon^{(2)}\left(1 - \kappa^{2\tilde{\Upsilon}}\right).$$

Proof. Recall that for $x, z \in \mathbb{X}^V$:

$$
\begin{aligned}
&(\tilde{\mathsf{F}}_t^m \mu)_{x,z}^v(A) \\
&= \frac{\int \mathbb{1}_A(\tilde{x}^v) \int \prod_{f \in N_f^m(K)} g^f(x_K^{N(f)}, y_t) \prod_{w \in N_v^m(K)} p^w(x_0^w, x_K^w) \mu(dx_0) \psi^{V \setminus K}(\tilde{x}) p^v(\tilde{x}^v, z^v) \psi^v(d\tilde{x}^v)}{\int \prod_{f \in N_f^m(K)} g^f(x_K^{N(f)}, y_t) \prod_{w \in N_v^m(K)} p^w(x_0^w, x_K^w) \mu(dx_0) \psi^{V \setminus K}(\tilde{x}) p^v(\tilde{x}^v, z^v) \psi^v(d\tilde{x}^v)},
\end{aligned}
$$

and:

$$
\widetilde{\mathrm{Corr}}(\tilde{\mathsf{F}}_t^m \mu, \beta) = \max_{v \in V} \sum_{v' \in V} e^{\beta d(v,v')} \tilde{C}_{v,v'}^{\tilde{\mathsf{F}}_t^m \mu},
$$

where $\tilde{C}_{v,v'}^{\tilde{\mathsf{F}}_t^m \mu} = 1/2 \sup_{z \in \mathbb{X}^V} \sup_{x, \hat{x} \in \mathbb{X}^V : x^{V \setminus v'} = \hat{x}^{V \setminus v'}} \left\| (\tilde{\mathsf{F}}_t^m \mu)_{x,z}^v - (\tilde{\mathsf{F}}_t^m \mu)_{\hat{x},z}^v \right\|$. The idea is again to firstly control each term of the $\widetilde{\mathrm{Corr}}$ and then sum on them.

Consider the probability distributions ρ and $\tilde{\rho}$:

$$
\begin{aligned}
&\rho(A) \\
&:= \frac{\int \mathbb{1}_A(x_0, x^{V \setminus K \cup v}) \prod_{f \in N_f^m(K)} g^f(x^{N(f)}, y_t) \prod_{w \in N_v^m(K)} p^w(x_0^w, x^w) p^v(x^v, z^v) \mu(dx_0) \psi^{V \setminus K \cup v}(x^{V \setminus K \cup v})}{\int \prod_{f \in N_f^m(K)} g^f(x^{N(f)}, y_t) \prod_{w \in N_v^m(K)} p^w(x_0^w, x^w) p^v(x^v, z^v) \mu(dx_0) \psi^{V \setminus K \cup v}(x^{V \setminus K \cup v})},
\end{aligned}
$$

$$
\begin{aligned}
&\tilde{\rho}(A) \\
&:= \frac{\int \mathbb{1}_A(x_0, \hat{x}^{V \setminus K \cup v}) \prod_{f \in N_f^m(K)} g^f(\hat{x}^{N(f)}, y_t) \prod_{w \in N_v^m(K)} p^w(x_0^w, \hat{x}^w) p^v(\hat{x}^v, z^v) \mu(dx_0) \psi^{V \setminus K \cup v}(\hat{x}^{V \setminus K \cup v})}{\int \prod_{f \in N_f^m(K)} g^f(\hat{x}^{N(f)}, y_t) \prod_{w \in N_v^m(K)} p^w(x_0^w, \hat{x}^w) p^v(\hat{x}^v, z^v) \mu(dx_0) \psi^{V \setminus K \cup v}(\hat{x}^{V \setminus K \cup v})},
\end{aligned}
$$

where $x, \hat{x} \in \mathbb{X}^V$, such that $\hat{x}^{V \setminus v'} = x^{V \setminus v'}$. It can be observed that by definition:

$$
\left\| \rho - \tilde{\rho} \right\|_{(1,v)} = \left\| (\tilde{\mathsf{F}}_t^m \mu)_{x,z}^v - (\tilde{\mathsf{F}}_t^m \mu)_{\hat{x},z}^v \right\|,
$$

meaning that the Dobrushin theorem can be applied to $\rho, \tilde{\rho}$ where the complete set of index $I = (0, V) \cup (1, V \setminus K \cup v)$.

The first step is to bound $C_{i,j}$ for all the possible combinations of $i, j \in I$, as in (4.2) of theorem 4.1.

- Consider $i = (0, b)$ and $b \in V$ then:

$$
\begin{aligned}
\rho_{x_0,x}^{(0,b)}(A) &= \frac{\prod_{f \in N_f^m(K)} g^f(x^{N(f)}, y_t) p^v(x^v, z^v)}{\prod_{f \in N_f^m(K)} g^f(x^{N(f)}, y_t) p^v(x^v, z^v)} \frac{\int \mathbb{1}_A(x_0^b) \prod_{w \in N_v^m(K)} p^w(x_0^w, x^w) \mu(dx_0)}{\int \prod_{w \in N_v^m(K)} p^w(x_0^w, x^w) \mu(dx_0)} \\
&= \frac{\int \mathbb{1}_A(x_0^b) p^b(x_0^b, x^b) \mu_{x_0}^b(dx_0^b)}{\int p^b(x_0^b, x^b) \mu_{x_0}^b(dx_0^b)} = \mu_{x_0,x}^b(A).
\end{aligned}
$$

  - If $j = (0, b')$ and $b' \in V$ then: $\quad C_{i,j} \le \tilde{C}^{\mu}_{b,b'}$.
  - If $j = (1, b')$ and $b' \in V \setminus K \cup v$ then by lemma 4.3: $\quad C_{i,j} \le \begin{cases} \left(1 - \frac{\epsilon_-}{\epsilon_+}\right) & b' = b \\ 0 & \text{otherwise} \end{cases}$, where the kernel part is upper bounded.

- Consider $i = (1, b)$ and $b \in V \setminus K \cup v$ then:

$$\begin{aligned}
&\rho^{(1,b)}_{x_0,x}(A) \\
&= \frac{\prod_{f \in N^m_f(K) \setminus N(b)} g^f(x^{N(f)}, y_t)}{\prod_{f \in N^m_f(K) \setminus N(b)} g^f(x^{N(f)}, y_t)} \\
&\quad \frac{\int \mathbb{1}_A(x^b) \prod_{f \in N^m_f(K) \cap N(b)} g^f(x^{N(f)}, y_t) p^b(x^b_0, x^b) [p^v(x^v, z^v)]^{\mathbb{1}_b(v)} \psi^b(dx^b)}{\int \prod_{f \in N^m_f(K) \cap N(b)} g^f(x^{N(f)}, y_t) p^b(x^b_0, x^b) [p^v(x^v, z^v)]^{\mathbb{1}_b(v)} \psi^b(dx^b)} \\
&= \frac{\int \mathbb{1}_A(x^b) \prod_{f \in N^m_f(K) \cap N(b)} g^f(x^{N(f)}, y_t) p^b(x^b_0, x^b) [p^v(x^v, z^v)]^{\mathbb{1}_b(v)} \psi^b(dx^b)}{\int \prod_{f \in N^m_f(K) \cap N(b)} g^f(x^{N(f)}, y_t) p^b(x^b_0, x^b) [p^v(x^v, z^v)]^{\mathbb{1}_b(v)} \psi^b(dx^b)}
\end{aligned}$$

  - If $j = (0, b')$ and $b' \in V$ then by lemma 4.3: $\quad C_{i,j} \le \begin{cases} \left(1 - \frac{\epsilon_-}{\epsilon_+}\right) & b' = b \\ 0 & \text{otherwise} \end{cases}$, where the kernel part is upper bounded.
  - If $j = (1, b')$ and $b' \in V \setminus K \cup v$ then by lemma 4.3:

$$C_{i,j} \le \begin{cases} (1 - \kappa^{\mathbf{2card}(N(b) \cap N(b'))}) & b' \in N^m_v(K) \cap N^2(b) \\ 0 & \text{otherwise} \end{cases},$$

where the observation density part is upper bounded.

But then:

$$\begin{aligned}
\max_{i \in I} \sum_{j \in I} e^{m(i,j)} C_{i,j} &\le \widetilde{\mathrm{Corr}}(\mu, \beta) + e^{\beta}\left(1 - \frac{\epsilon_-}{\epsilon_+}\right) + \max_{b \in V} \sum_{b' \in N^2(b)} e^{\beta d(b,b')} \left(1 - \kappa^{\mathbf{2card}(N(b) \cap N(b'))}\right) \\
&\le \widetilde{\mathrm{Corr}}(\mu, \beta) + 2e^{\beta}\left(1 - \frac{\epsilon_-}{\epsilon_+}\right) + \Upsilon^{(2)} e^{2\beta}\left(1 - \kappa^{2\tilde{\Upsilon}}\right) \le \frac{1}{2},
\end{aligned}$$

where $m(i,j) = \beta|k - k'| + \beta d(v, v')$ with $i = (k, v)$ and $j = (k', v')$ for $k, k' \in \{0, 1\}$ and $v, v' \in V$ is the pseudometric of interest. Remark also that $2e^{\beta} > e^{\beta}$. Then the Dobrushin theorem can be applied:

$$\left\| (\tilde{\mathsf{F}}^m_t \mu)^v_{x,z} - (\tilde{\mathsf{F}}^m_t \mu)^v_{\hat{x},z} \right\| = \|\rho - \tilde{\rho}\|_{(1,v)} \le \sum_{j \in I} D_{(1,v),j} b_j.$$

The second step is to control b_j, as in (4.2) of theorem 4.1.

- If $j = (0, b')$ and $b' \in V$ then:

$$\tilde{\rho}^{(0,b')}_{x_0,\hat{x}}(A) = \frac{\int \mathbb{1}_A(x^{b'}_0) p^{b'}(x^{b'}_0, \hat{x}^{b'}) \mu^{b'}_{x_0}(dx^{b'}_0)}{\int p^{b'}(x^{b'}_0, \hat{x}^{b'}) \mu^{b'}_{x_0}(dx^{b'}_0)} = \mu^{b'}_{x_0,\hat{x}}(A),$$

where the computations are the same as in $\rho^{(0,b)}_{x_0,x}$. Hence by lemma 4.3:

$$b_j \leq \begin{cases} 2\left(1-\frac{\epsilon_-}{\epsilon_+}\right) & b'=v' \\ 0 & \text{otherwise} \end{cases},$$

because x_1 and $\tilde{x}_1$ differ only on v'.

- If $j=(1,b')$ and $b' \in V \setminus K \cup v$ then:

$$\tilde{\rho}^{(1,b')}_{x_0,\hat{x}}(A) = \frac{\int \mathbb{I}_A(\hat{x}^{b'}) \prod_{f \in N^m_f(K) \cap N(b')} g^f(\hat{x}^{N(f)}, y_t) p^{b'}(x_0^{b'}, \hat{x}^{b'})[p^v(\hat{x}^v, z^v)]^{\mathbb{I}_{b'}(v)} \psi^{b'}(d\hat{x}^{b'})}{\int \prod_{f \in N^m_f(K) \cap N(b')} g^f(\hat{x}^{N(f)}, y_t) p^{b'}(x_0^{b'}, \hat{x}^{b'})[p^v(\hat{x}^v, z^v)]^{\mathbb{I}_{b'}(v)} \psi^{b'}(d\hat{x}^{b'})}.$$

Hence by lemma 4.3:

$$b_j \leq \begin{cases} 2(1-\kappa^{2\mathbf{card}(N(b')\cap N(v'))}) & b' \in N^2(v') \cap N^m_v(K) \setminus v \\ 0 & \text{otherwise} \end{cases},$$

where the case $b'=v$ is still zero because the only difference is on v'.

By joining step one and step two it follows:

$$\begin{aligned} \left\|\rho - \tilde{\rho}\right\|_{(1,v)} &\leq D_{(1,v),(0,v')} b_{(0,v')} + \sum_{b' \in N^2(v')} D_{(1,v),(1,b')} b_{(1,b')} \\ &\leq 2\left(1-\frac{\epsilon_-}{\epsilon_+}\right) D_{(1,v),(0,v')} + 2\left(1-\kappa^{2\tilde{\Upsilon}}\right) \sum_{b' \in N^2(v')} D_{(1,v),(1,b')}. \end{aligned}$$

Remark that if $v' \in V$ and $b' \in N^2(v')$ then obviously $b' \in V$ and $v' \in N^2(b')$. Moreover, by the triangular inequality $d(v,v') \leq d(v,b')+d(v',b')$. Then by summing over V and by applying lemma 4.5:

$$\begin{aligned} \sum_{v' \in V} e^{\beta d(v,v')} \tilde{C}^{\tilde{\mathsf{F}}^m_t \mu}_{v,v'} &\leq \left(1-\frac{\epsilon_-}{\epsilon_+}\right) \sum_{v' \in V} e^{\beta d(v,v')} D_{(1,v),(0,v')} + \left(1-\kappa^{2\tilde{\Upsilon}}\right) \sum_{v' \in V} e^{\beta d(v,v')} \sum_{b' \in N^2(v')} D_{(1,v),(1,b')} \\ &\leq \left(1-\frac{\epsilon_-}{\epsilon_+}\right) \sum_{v' \in V} e^{\beta d(v,v')} D_{(1,v),(0,v')} \\ &\quad + \left(1-\kappa^{2\tilde{\Upsilon}}\right) \sum_{v' \in V} \sum_{b' \in N^2(v')} e^{\beta d(v,b')+d(b',v')} D_{(1,v),(1,b')} \\ &\leq 2\left(1-\frac{\epsilon_-}{\epsilon_+}\right) e^{-\beta} + \left(1-\kappa^{2\tilde{\Upsilon}}\right) e^{2\beta} \sum_{b' \in V} \sum_{v' \in N^2(b')} e^{\beta d(v,b')} D_{(1,v),(1,b')} \\ &\leq 2\left(1-\frac{\epsilon_-}{\epsilon_+}\right) e^{-\beta} + 2\Upsilon^{(2)} \left(1-\kappa^{2\tilde{\Upsilon}}\right) e^{2\beta}. \end{aligned}$$

Given that the bound does not depend on v the thesis follows from the definition of $\widetilde{\mathrm{Corr}}(\tilde{\mathsf{F}}^m_t \mu, \beta)$.

■

Corollary 4.1. *Fix any collection of observations $\{y_1,\dots,y_T\}$ and any partition $\mathscr{K}$ on the set V. There exists a region $\mathscr{R}_0 \subseteq (0,1)^3$ depending only on $\tilde{\Upsilon}$, Υ and $\Upsilon^{(2)}$, such that if, for given $(\epsilon_-,\epsilon_+,\kappa)\in\mathscr{R}_0$,*

$$\epsilon_- \leq p^v(x^v,z^v) \leq \epsilon_+ \quad \text{and} \quad \kappa \leq g^f(x^{N(f)},y_t) \leq \frac{1}{\kappa},$$

for all $x,z\in\mathbb{X}^V, f\in F, v\in V, t\in\{1,\dots,T\}$, then for $\beta>0$ small enough depending only on $\tilde{\Upsilon},\Upsilon$, $\Upsilon^{(2)}$, ϵ_-,ϵ_+, κ, for any λ_0 satisfying the decay of correlation property:

$$\widetilde{\operatorname{Corr}}(\lambda_0,\beta) \leq 2e^{-\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+2e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right) \tag{4.3}$$

and for any $t\in\{1,\dots,T\}$ and $m\in\{0,\dots,n\}$:

$$\widetilde{\operatorname{Corr}}(\tilde{\mathsf{F}}_t^m\tilde{\pi}_{t-1},\beta)+2e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)\leq\frac{1}{2} \tag{4.4}$$

and

$$2\kappa^{-2\Upsilon}\operatorname{Corr}(\mathsf{P}\tilde{\pi}_{t-1},\beta)+2e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)\leq\frac{1}{2}, \tag{4.5}$$

where $\tilde{\pi}_{t-1}$ is the approximated filtering distribution obtained through (3.12).

Proof. The proof is inductive in t. To initialize the induction, let $t=1$. The aim is to identify ranges of values for $\beta,\epsilon_-,\epsilon_+$ and κ such that:

$$\widetilde{\operatorname{Corr}}(\tilde{\mathsf{F}}_1^m\lambda_0,\beta)+2e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)\leq\frac{1}{2} \tag{4.6}$$

and

$$2\kappa^{-2\Upsilon}\operatorname{Corr}(\mathsf{P}\lambda_0,\beta)+e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)\leq\frac{1}{2}. \tag{4.7}$$

Start by analysing (4.6). Note that there is a bound for $\widetilde{\operatorname{Corr}}(\lambda_0,\beta)$ given by (4.3). So for any $\beta>0$, $(\epsilon_-,\epsilon_+)\in(0,1)^2$ and $\kappa\in(0,1)$ there is the upper bound:

$$\begin{aligned}
&\widetilde{\operatorname{Corr}}(\lambda_0,\beta)+2e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)\\
&\leq 2e^{-\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+2e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)+2e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)\\
&\leq 2\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+\left[3\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)+2\left(1-\frac{\epsilon_-}{\epsilon_+}\right)\right]e^{2\beta},
\end{aligned}$$

In order to apply proposition 4.6 and obtain (4.6), the next step is to derive constraints on β, ϵ_-,ϵ_+ and κ such that:

$$2\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+\left[3\Upsilon^{(2)}(1-\kappa^{2\tilde{\Upsilon}})+2\left(1-\frac{\epsilon_-}{\epsilon_+}\right)\right]e^{2\beta}\leq\frac{1}{2}. \tag{4.8}$$

This holds for β such that:

$$\beta \le \frac{1}{2}\log\left\{\frac{1-4\left(1-\frac{\epsilon_-}{\epsilon_+}\right)}{6\Upsilon^{(2)}(1-\kappa^{2\tilde{\Upsilon}})+4\left(1-\frac{\epsilon_-}{\epsilon_+}\right)}\right\} = \beta_0^1.$$

and to guarantee $\beta > 0$ when $(\epsilon_-, \epsilon_+, \kappa) \in (0,1)^3$, i.e. the logarithm being positive, further impose:

$$1-4\left(1-\frac{\epsilon_-}{\epsilon_+}\right) > 6\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)+4\left(1-\frac{\epsilon_-}{\epsilon_+}\right).$$

Informed by these considerations define the region:

$$\mathscr{R}_0^1 := \{(\epsilon_-,\epsilon_+,\kappa)\in(0,1)^3 : 8\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+6\Upsilon^{(2)}(1-\kappa^{2\tilde{\Upsilon}})<1\}.$$

Hence by choosing $(\epsilon_-,\epsilon_+,\kappa)\in\mathscr{R}_0^1$ and $\beta<\beta_0^1$, the inequality (4.8) holds as required and so, noting $\tilde{\pi}_0 = \lambda_0$, proposition 4.6 can be applied to give:

$$\begin{aligned}
&\widetilde{\mathrm{Corr}}(\tilde{\mathsf{F}}_1^m\tilde{\pi}_0,\beta)+2e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)\\
&\le 2e^{-\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+2e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)+2e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)\\
&\le 2\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+\left[3\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)+2\left(1-\frac{\epsilon_-}{\epsilon_+}\right)\right]e^{2\beta},
\end{aligned}$$

which has been already proved to be less than 1/2 for $\beta\le\beta_0^1$ and $(\epsilon_-,\epsilon_+,\kappa)\in\mathscr{R}_0^1$.

Turning to (4.7), first note the following upper bound:

$$\begin{aligned}
&\widetilde{\mathrm{Corr}}(\lambda_0,\beta)+e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right)\\
&\le 2e^{-\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+2e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)+e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right),
\end{aligned}$$

and with the previous choice of $\beta, \epsilon_-, \epsilon_+$ and κ it can be noticed that:

$$\begin{aligned}
\widetilde{\mathrm{Corr}}(\delta_x,\beta)+e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right) &\le 2e^{-\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+2e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)+e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right)\\
&\le 2\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+\left[3\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)+2\left(1-\frac{\epsilon_-}{\epsilon_+}\right)\right]e^{2\beta}\le\frac{1}{2}.
\end{aligned}$$

Hence the assumption of proposition 4.5 holds without any additional restrictions, meaning that:

$$2\kappa^{-2\Upsilon}\mathrm{Corr}(\mathsf{P}\tilde{\pi}_0,\beta)\le 4\kappa^{-2\Upsilon}e^{-\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right).$$

The next step is to identify constraints on β, ϵ_-,ϵ_+ and κ in order to guarantee the second of the following two inequalities:

$$2\kappa^{-2\Upsilon}\mathrm{Corr}(\mathsf{P}\tilde{\pi}_0,\beta)+2e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)\le 4\kappa^{-2\Upsilon}e^{-\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+2e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)\le\frac{1}{2}.$$

To do so, impose:

$$\beta \le \frac{1}{2}\log\left(\frac{\kappa^{2\Upsilon} - 8\left(1-\frac{\epsilon_-}{\epsilon_+}\right)}{4\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)\kappa^{2\Upsilon}}\right) = \beta_0^2,$$

and again for positivity of the logarithm:

$$\kappa^{2\Upsilon} - 8\left(1-\frac{\epsilon_-}{\epsilon_+}\right) > 4\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)\kappa^{2\Upsilon}.$$

This leads to define:

$$\mathscr{R}_0^2 := \left\{(\epsilon_-,\epsilon_+,\kappa) \in (0,1)^3 : \kappa^{2\Upsilon} - 4\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)\kappa^{2\Upsilon} > 8\left(1-\frac{\epsilon_-}{\epsilon_+}\right)\right\},$$

and hence by choosing $\beta \le \beta_0^2$ and $(\epsilon_-,\epsilon_+,\kappa) \in \mathscr{R}_0^2$:

$$2\kappa^{-2\Upsilon}\mathrm{Corr}(\mathsf{P}\tilde{\pi}_0,\beta) + 2e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right) \le \frac{1}{2}.$$

It has been proved that with:

$$\beta_0 = \min\{\beta_0^1,\beta_0^2\} \quad \text{and} \quad (\epsilon_-,\epsilon_+,\kappa) \in \mathscr{R}_0 := \mathscr{R}_0^1 \cap \mathscr{R}_0^2, \tag{4.9}$$

both (4.4) and (4.5) hold for $t = 1$.

Suppose now that (4.4) holds for t. Then since $\tilde{\mathsf{F}}^m_{t+1}\tilde{\pi}_t = \tilde{\pi}_{t+1}$, proposition 4.6 can be applied for $t+1$ and so:

$$\widetilde{\mathrm{Corr}}(\tilde{\mathsf{F}}^m_{t+1}\tilde{\pi}_t,\beta) + 2e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right) + e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right) \le$$
$$2\left(1-\frac{\epsilon_-}{\epsilon_+}\right) + \left[3\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right) + 2\left(1-\frac{\epsilon_-}{\epsilon_+}\right)\right]e^{2\beta},$$

which has been already proved to be less than 1/2 for $\beta \le \beta_0$ and $(\epsilon_-,\epsilon_+,\kappa) \in \mathscr{R}_0$ – see (4.8).

Given that (4.4) holds for t then:

$$\widetilde{\mathrm{Corr}}(\tilde{\pi}_t,\beta) + e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right) \le \frac{1}{2},$$

proposition 4.5 applies and so for the previous choices of $\beta,\epsilon_-,\epsilon_+,\kappa$:

$$2e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right) + 2\kappa^{-2\Upsilon}\mathrm{Corr}(\mathsf{P}\tilde{\pi}_t,\beta) \le 2e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right) + 4\kappa^{-2\Upsilon}e^{-\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right) \le \frac{1}{2},$$

which completes the treatment of (4.5).

Hence the induction is complete and for $\beta \le \beta_0$, $(\epsilon_-,\epsilon_+,\kappa) \in \mathscr{R}_0$ both (4.4) and (4.5) hold for all t.
■

4.2.4 Proof of theorem 3.1

Proof. For notational convenience the proof is stated for $\pi_T - \tilde{\pi}_T$, but since T is arbitrary this is also not a restriction.

The quantity $\pi_T - \tilde{\pi}_T$ can be expressed as a telescopic sum, indeed:

$$\pi_T - \tilde{\pi}_T = \mathsf{F}_T \dots \mathsf{F}_1 \delta_x - \tilde{\mathsf{F}}_T^m \dots \tilde{\mathsf{F}}_1^m \delta_x = \sum_{t=1}^{T} (\mathsf{F}_T \dots \mathsf{F}_{t+1} \mathsf{F}_t \tilde{\pi}_{t-1} - \mathsf{F}_T \dots \mathsf{F}_{t+1} \tilde{\mathsf{F}}_t^m \tilde{\pi}_{t-1}),$$

hence given $J \subseteq K \in \mathscr{K}$, by the triangular inequality:

$$\|\pi_T - \tilde{\pi}_T\|_J \leq \sum_{t=1}^{T-1} \left\| \mathsf{F}_T \dots \mathsf{F}_{t+1} \mathsf{F}_t \tilde{\pi}_{t-1} - \mathsf{F}_T \dots \mathsf{F}_{t+1} \tilde{\mathsf{F}}_t^m \tilde{\pi}_{t-1} \right\|_J + \left\| \mathsf{F}_T \tilde{\pi}_{T-1} - \tilde{\mathsf{F}}_T^m \tilde{\pi}_{T-1} \right\|_J .$$

If $\epsilon_-, \epsilon_+, \kappa$ and β are chosen according to corollary 4.1 then proposition 4.1 can be applied for $t \in \{1, \dots T-1\}$:

$$\begin{aligned} \|\pi_T - \tilde{\pi}_T\|_J \leq & \sum_{t=1}^{T-1} 2e^{-\beta(T-t)} \sum_{v \in J} \max_{v' \in V} \left\{ e^{-\beta d(v,v')} \sup_{x \in \mathbb{X}^V, z^v \in \mathbb{X}} \left\| (\mathsf{F}_t \tilde{\pi}_{t-1})_{x_0, z^v}^{v'} - (\tilde{\mathsf{F}}_t^m \tilde{\pi}_{t-1})_{x_0, z^v}^{v'} \right\| \right\} \\ & + \left\| \mathsf{F}_T \tilde{\pi}_{T-1} - \tilde{\mathsf{F}}_T^m \tilde{\pi}_{T-1} \right\|_J . \end{aligned}$$

But given that also proposition 4.2 and proposition 4.4 can be applied then by considering $v' \in K'$:

$$\sup_{x_1 \in \mathbb{X}^V, z^{v'} \in \mathbb{X}} \left\| (\mathsf{F}_t \tilde{\pi}_{t-1})_{x_1, z}^{v'} - (\tilde{\mathsf{F}}_t^m \tilde{\pi}_{t-1})_{x_1, z}^{v'} \right\| \leq 2 \left(1 - \kappa^{a(\mathscr{K})}\right) + 4e^{-\beta m} \left(1 - \kappa^{b(\mathscr{K}, m)}\right),$$

and

$$\left\| \mathsf{F}_T \tilde{\pi}_{T-1} - \tilde{\mathsf{F}}_T^m \tilde{\pi}_{T-1} \right\|_J \leq 4e^{-\beta m} \left(1 - \kappa^{b(m, \mathscr{K})}\right) \mathbf{card}(J).$$

Hence putting everything together:

$$\begin{aligned} & \|\pi_T - \tilde{\pi}_T\|_J \\ & \leq \left[2 \left(1 - \kappa^{a(\mathscr{K})}\right) + 4e^{-\beta m} \left(1 - \kappa^{b(\mathscr{K}, m)}\right) \right] \\ & \cdot \sum_{t=1}^{T-1} 2e^{-\beta(T-t)} \sum_{v \in J} 1 + 4e^{-\beta m} \left(1 - \kappa^{b(m, \mathscr{K})}\right) \mathbf{card}(J) \\ & \leq \left[2 \left(1 - \kappa^{a(\mathscr{K})}\right) + 4e^{-\beta m} \left(1 - \kappa^{b(\mathscr{K}, m)}\right) \right] \frac{2}{(e^{\beta} - 1)} \mathbf{card}(J) \\ & \quad + 4e^{-\beta m} \left(1 - \kappa^{b(m, \mathscr{K})}\right) \mathbf{card}(J) \\ & = \alpha_1(\beta) \left(1 - \kappa^{a(\mathscr{K})}\right) \mathbf{card}(J) + \gamma_1(\beta) \left(1 - \kappa^{b(\mathscr{K}, m)}\right) \mathbf{card}(J) e^{-\beta m}. \qquad (4.10) \end{aligned}$$

■

4.3 Graph Smoother

As for the filtering distribution, the proof of theorem 3.2 follows by breaking down the problem. Consider the difference between the optimal smoothing and the approximate one:

$$\begin{aligned}\mathsf{R}_{\tilde{\pi}_t}\tilde{\pi}_{t+1|T} - \mathsf{R}_{\pi_t}\pi_{t+1|T} =& \mathsf{R}_{\tilde{\pi}_t}\dots\mathsf{R}_{\tilde{\pi}_{T-1}}\tilde{\pi}_T - \mathsf{R}_{\tilde{\pi}_t}\dots\mathsf{R}_{\tilde{\pi}_{T-1}}\pi_T\\ &+\sum_{s=0}^{T-t-1}\mathsf{R}_{\tilde{\pi}_t}\dots\mathsf{R}_{\tilde{\pi}_{t+s-1}}\mathsf{R}_{\tilde{\pi}_{t+s}}\pi_{t+s+1|T} - \mathsf{R}_{\tilde{\pi}_t}\dots\mathsf{R}_{\tilde{\pi}_{t+s-1}}\mathsf{R}_{\pi_{t+s}}\pi_{t+s+1|T},\end{aligned}$$

hence the overall proof can be split in three steps:

1. control the part outside the sum: approximate smoothing stability (section 4.3.1);
2. control the part inside the sum: approximate smoothing stability and smoothing error control (section 4.3.2).

Remark that for the proof of theorem 3.2 corollary 4.1 is needed, so condition 4.3 must hold.

The following definition is needed.

Definition 4.7. Let μ, ν be probability distributions on $\mathbb{X}^V$, let $Z \sim \mu$ and $X|Z \sim \overleftarrow{P}_\nu(Z,\cdot)$, where $\overleftarrow{P}_\nu(z,\cdot) := p(x,z)\nu(dx)/\int p(x,z)\nu(dx)$. Then define:

$$(\overset{\nu}{\overleftarrow{\mu}})^v_{z,x}(A) := \mathbb{P}(Z^v \in A | Z^{V\setminus v} = z, X = x).$$

4.3.1 Approximate smoothing stability

It must be proved that an application of the approximate smoothing operator to $\tilde{\pi}_T$ is not too different from the same applied on π_T.

Proposition 4.7. *Fix any collection of observations $\{y_1,\dots,y_T\}$ and any partition $\mathscr{K}$ on the set V. There exists a region $\mathscr{R}_0 \subseteq (0,1)^3$, as in corollary 4.1, depending only on $\tilde{\Upsilon}, \Upsilon$ and $\Upsilon^{(2)}$, such that if, for given $(\epsilon_-,\epsilon_+,\kappa) \in \mathscr{R}_0$,*

$$\epsilon_- \le p^v(x^v,z^v) \le \epsilon_+ \quad \textit{and} \quad \kappa \le g^f(x^{N(f)},y_t) \le \frac{1}{\kappa},$$

for all $x,z \in \mathbb{X}^V, f \in F, v \in V, t \in \{1,\dots,T\}$, then for $\beta > 0$ small enough depending only on $\tilde{\Upsilon}, \Upsilon$, $\Upsilon^{(2)}$, ϵ_-,ϵ_+, κ, for any $t \in \{0,\dots,T-1\}, K \in \mathscr{K}$ and $J \subseteq K$ and $m \in \{0,\dots,n\}$:

$$\begin{aligned}&\left\|\mathsf{R}_{\tilde{\pi}_t}\dots\mathsf{R}_{\tilde{\pi}_{T-1}}\tilde{\pi}_T - \mathsf{R}_{\tilde{\pi}_t}\dots\mathsf{R}_{\tilde{\pi}_{T-1}}\pi_T\right\|_J\\ &\qquad\le 2e^{-\beta(T-t)}\sum_{v\in J}\max_{v'\in V}\left\{e^{-\beta d(v,v')}\sup_{x_{T-1},x_T\in\mathbb{X}^V}\left\|(\overset{\tilde{\pi}_{T-1}}{\overleftarrow{\tilde{\pi}_T}})^{v'}_{x_T,x_{T-1}} - (\overset{\tilde{\pi}_{T-1}}{\overleftarrow{\pi_T}})^{v'}_{x_T,x_{T-1}}\right\|\right\}.\end{aligned}$$

Proof. Denote with $\overleftarrow{\tilde{p}}_t(\cdot,\cdot)$ the reverse kernel density with reference distribution the approximated filtering distribution, i.e.:

$$\overleftarrow{\tilde{p}}_t(z,x) := \frac{p(x,z)}{\int p(\hat{x},z)\tilde{\pi}_t(d\hat{x})}.$$

Then:

$$\mathsf{R}_{\tilde{\pi}_t}\dots\mathsf{R}_{\tilde{\pi}_{T-1}}\tilde{\pi}_T(A)=\int \mathbb{1}_A(x_t)\overleftarrow{\tilde{p}}_t(x_{t+1},x_t)\tilde{\pi}_t(dx_t)\dots\overleftarrow{\tilde{p}}_{T-1}(x_T,x_{T-1})\tilde{\pi}_{T-1}(dx_{T-1})\tilde{\pi}_T(dx_T),$$

$$\mathsf{R}_{\tilde{\pi}_t}\dots\mathsf{R}_{\tilde{\pi}_{T-1}}\pi_T(A)=\int \mathbb{1}_A(x_t)\overleftarrow{\tilde{p}}_t(x_{t+1},x_t)\tilde{\pi}_t(dx_t)\dots\overleftarrow{\tilde{p}}_{T-1}(x_T,x_{T-1})\tilde{\pi}_{T-1}(dx_{T-1})\pi_T(dx_T).$$

Consider the following probability distributions:

$$\rho(A):=\int \mathbb{1}_A(x_t,\dots,x_T)\overleftarrow{\tilde{p}}_t(x_{t+1},x_t)\tilde{\pi}_t(dx_t)\dots\overleftarrow{\tilde{p}}_{T-1}(x_T,x_{T-1})\tilde{\pi}_{T-1}(dx_{T-1})\pi_T(dx_T),$$

$$\tilde{\rho}(A):=\int \mathbb{1}_A(x_t,\dots,x_T)\overleftarrow{\tilde{p}}_t(x_{t+1},x_t)\tilde{\pi}_t(dx_t)\dots\overleftarrow{\tilde{p}}_{T-1}(x_T,x_{T-1})\tilde{\pi}_{T-1}(dx_{T-1})\tilde{\pi}_T(dx_T),$$

the quantity of interest can be reformulated in terms of LTV on $\rho,\tilde{\rho}$:

$$\left\|\mathsf{R}_{\tilde{\pi}_t}\dots\mathsf{R}_{\tilde{\pi}_{T-1}}\tilde{\pi}_T-\mathsf{R}_{\tilde{\pi}_t}\dots\mathsf{R}_{\tilde{\pi}_{T-1}}\pi_T\right\|_J=\left\|\rho-\tilde{\rho}\right\|_{(t,J)}.$$

So, it is enough to find a bound for $\left\|\rho-\tilde{\rho}\right\|_{(t,J)}$ to guarantee the proof of the statement. The Dobrushin theorem can be used on the distributions $\rho,\tilde{\rho}$ where the index set is $I=\bigcup_{k=t}^{T}(k,V)$ and the subset is (t,J).

The first step is to bound $C_{i,j}$ for all the possible combination of $i,j\in I$, as in (4.2) of theorem 4.1. The notation $x=(x_t,\dots,x_T)$ is used, where $x_k\in\mathbb{X}^V$ for $k=t,\dots,T$ and $x\setminus x_k^v:=(x_t,\dots,x_k^{V\setminus v},\dots,x_T)$ (and the same with a tilde).

- Consider $i=(t,v)$ and $v\in V$ then:

  $$\begin{aligned}&\tilde{\rho}_x^{(t,v)}(A)\\&=\frac{\int\mathbb{1}_A(\tilde{x}_t^v)\mathbb{1}_{x\setminus x_t^v}(\tilde{x}\setminus\tilde{x}_t^v)\overleftarrow{\tilde{p}}_t(\tilde{x}_{t+1},\tilde{x}_t)\tilde{\pi}_t(d\tilde{x}_t)\dots\overleftarrow{\tilde{p}}_{T-1}(\tilde{x}_T,\tilde{x}_{T-1})\tilde{\pi}_{T-1}(d\tilde{x}_{T-1})\tilde{\pi}_T(d\tilde{x}_T)}{\int\mathbb{1}_{x\setminus x_t^v}(\tilde{x}\setminus\tilde{x}_t^v)\overleftarrow{\tilde{p}}_t(\tilde{x}_{t+1},\tilde{x}_t)\tilde{\pi}_t(d\tilde{x}_t)\dots\overleftarrow{\tilde{p}}_{T-1}(\tilde{x}_T,\tilde{x}_{T-1})\tilde{\pi}_{T-1}(d\tilde{x}_{T-1})\tilde{\pi}_T(d\tilde{x}_T)}\\&=\frac{\int\mathbb{1}_A(x_t^v)\overleftarrow{\tilde{p}}_t(x_{t+1},x_t)(\tilde{\pi}_t)_{x_t}^v(dx_t^v)}{\int\overleftarrow{\tilde{p}}_t(x_{t+1},x_t)(\tilde{\pi}_t)_{x_t}^v(dx_t^v)}=\frac{\int\mathbb{1}_A(x_t^v)p^v(x_t^v,x_{t+1}^v)(\tilde{\pi}_t)_{x_t}^v(dx_t^v)}{\int p^v(x_t^v,x_{t+1}^v)(\tilde{\pi}_t)_{x_t}^v(dx_t^v)},\end{aligned}$$

  where the last passage follow from the independence of the numerator of the reverse kernel from x_t. Now the different cases in which ρ_x^i can differ from $\rho_{\tilde{x}}^i$, where $x^{I\setminus j}=\tilde{x}^{I\setminus j}$, must be distinguished.

  - If $j=(t,v')$ and $v'\in V$ then: $C_{i,j}\le\tilde{C}_{v,v'}^{\tilde{\pi}_t}$.
  - If $j=(t+1,v')$ and $v'\in V$ then: $C_{i,j}\le\begin{cases}\left(1-\frac{\epsilon_-}{\epsilon_+}\right) & v'=v\\ 0 & v'\neq v\end{cases}$,
    where the result follows from lemma 4.3 is obtained by a majorization of the kernel part.
  - If $j=(k,v')$ with $k>t+1$ and $v'\in V$ then: $C_{i,j}=0$,
    because in ρ_x^i there is no dependence on x_k with $k>t+1$.

- Consider $i=(k,v)$ with $t+1<k<T$ and $v\in K$ with $K\in\mathcal{K}$ then:

$$
\begin{aligned}
&\tilde{\rho}_x^{(k,v)}(A)\\
&=\frac{\int \mathbb{1}_A(\tilde{x}_k^v)\mathbb{1}_{x\setminus x_k^v}(\tilde{x}\setminus\tilde{x}_k^v)\overleftarrow{\tilde{p}}_t(\tilde{x}_{t+1},\tilde{x}_t)\tilde{\pi}_t(d\tilde{x}_t)\dots\overleftarrow{\tilde{p}}_{T-1}(\tilde{x}_T,\tilde{x}_{T-1})\tilde{\pi}_{T-1}(d\tilde{x}_{T-1})\tilde{\pi}_T(d\tilde{x}_T)}{\int \mathbb{1}_{x\setminus x_k^v}(\tilde{x}\setminus\tilde{x}_k^v)\overleftarrow{\tilde{p}}_t(\tilde{x}_{t+1},\tilde{x}_t)\tilde{\pi}_t(d\tilde{x}_t)\dots\overleftarrow{\tilde{p}}_{T-1}(\tilde{x}_T,\tilde{x}_{T-1})\tilde{\pi}_{T-1}(d\tilde{x}_{T-1})\tilde{\pi}_T(d\tilde{x}_T)}\\
&=\frac{\int \mathbb{1}_A(x_k^v)\overleftarrow{\tilde{p}}_{k-1}(x_k,x_{k-1})\overleftarrow{\tilde{p}}_k(x_{k+1},x_k)(\tilde{\pi}_k)_{x_k}^v(dx_k)}{\int \overleftarrow{\tilde{p}}_{k-1}(x_k,x_{k-1})\overleftarrow{\tilde{p}}_k(x_{k+1},x_k)(\tilde{\pi}_k)_{x_k}^v(dx_k)}\\
&=\frac{\int \mathbb{1}_A(x_k^v)\overleftarrow{\tilde{p}}_{k-1}(x_k,x_{k-1})p^v(x_k^v,x_{k+1}^v)(\tilde{\pi}_k)_{x_k}^v(dx_k)}{\int \overleftarrow{\tilde{p}}_{k-1}(x_k,x_{k-1})p^v(x_k^v,x_{k+1}^v)(\tilde{\pi}_k)_{x_k}^v(dx_k)},
\end{aligned}
$$

where the last passage follows from the definition of the denominator of $\overleftarrow{\tilde{p}}_k(x_{k+1},x_k)$ that is independent from x_k^v given that x_k is integrated out. At this point the computations on the numerator are carried out and similar calculations follow on the denominator when $A=\mathbb{X}$. Firstly the definition of $(\tilde{\pi}_k)_{x_k}^v$ can be expanded:

$$
\begin{aligned}
&\int \mathbb{1}_A(x^v)\overleftarrow{\tilde{p}}_{k-1}(x,x_{k-1})p^v(x^v,x_{k+1}^v)\\
&\qquad\frac{\prod_{f\in N_f^m(K)} g^f(x_K^{N(f)},y_t)\int\prod_{w\in N_v^m(K)} p^w(x_0^w,x_K^w)\tilde{\pi}_{k-1}(dx_0)\psi^{V\setminus K}(\tilde{x})}{\int\prod_{f\in N_f^m(K)} g^f(x_K^{N(f)},y_t)\int\prod_{w\in N_v^m(K)} p^w(x_0^w,x_K^w)\tilde{\pi}_{k-1}(dx_0)\psi^{V\setminus K}(\tilde{x})\psi^v(dx^v)}\psi^v(dx^v)\\
&=\int \mathbb{1}_A(x^v)\frac{\prod_{w\in V} p^w(x_{k-1}^w,x^w)}{\prod_{K'\in\mathcal{K}}\int\prod_{w\in K'} p^w(x_{k-1}^w,x^w)\tilde{\pi}_{k-1}^{K'}(dx_{k-1}^{K'})}p^v(x^v,x_{k+1}^v)\\
&\qquad\frac{\prod_{f\in N_f^m(K)} g^f(x_K^{N(f)},y_t)\int\prod_{w\in N_v^m(K)} p^w(x_0^w,x_K^w)\tilde{\pi}_{k-1}(dx_0)\psi^{V\setminus K}(\tilde{x})}{\int\prod_{f\in N_f^m(K)} g^f(x_K^{N(f)},y_t)\int\prod_{w\in N_v^m(K)} p^w(x_0^w,x_K^w)\tilde{\pi}_{k-1}(dx_0)\psi^{V\setminus K}(\tilde{x})\psi^v(dx^v)}\psi^v(dx^v)\\
&=\int \mathbb{1}_A(x^v)\frac{p^v(x_{k-1}^v,x^v)}{\int\prod_{w\in K} p^w(x_{k-1}^w,x^w)\tilde{\pi}_{k-1}^{K}(dx_{k-1}^{K})}p^v(x^v,x_{k+1}^v)\\
&\qquad\frac{\prod_{f\in N_f^m(K)} g^f(x_K^{N(f)},y_t)\int\prod_{w\in N_v^m(K)} p^w(x_0^w,x_K^w)\tilde{\pi}_{k-1}(dx_0)\psi^{V\setminus K}(\tilde{x})}{\int\prod_{f\in N_f^m(K)} g^f(x_K^{N(f)},y_t)\int\prod_{w\in N_v^m(K)} p^w(x_0^w,x_K^w)\tilde{\pi}_{k-1}(dx_0)\psi^{V\setminus K}(\tilde{x})\psi^v(dx^v)}\psi^v(dx^v)\\
&=\int \mathbb{1}_A(x^v)p^v(x_{k-1}^v,x^v)p^v(x^v,x_{k+1}^v)\\
&\qquad\frac{\prod_{f\in N_f^m(K)} g^f(x_K^{N(f)},y_t)\int\prod_{w\in N_v^m(K)\setminus K} p^w(x_0^w,x_K^w)\tilde{\pi}_{k-1}^{V\setminus K}(dx_0^{V\setminus K})\psi^{V\setminus K}(\tilde{x})}{\int\prod_{f\in N_f^m(K)} g^f(x_K^{N(f)},y_t)\int\prod_{w\in N_v^m(K)} p^w(x_0^w,x_K^w)\tilde{\pi}_{k-1}(dx_0)\psi^{V\setminus K}(\tilde{x})\psi^v(dx^v)}\psi^v(dx^v),
\end{aligned}
$$

where the factorization of $\tilde{\pi}_{k-1}$ and the factorization of the kernel have been used. Given that the same holds for the denominator the expression can be simplified a bit more. Define:

$$\begin{aligned} N_k :=& \int \mathbb{1}_A(x^v) p^v(x^v_{k-1}, x^v) p^v(x^v, x^v_{k+1}) \\ & \int \prod_{f \in N^m_f(K)} g^f(x^{N(f)}_K, y_t) \prod_{w \in N^m_v(K) \setminus K} p^w(x^w_0, x^w_K) \tilde{\pi}^{V \setminus K}_{k-1}(dx^{V \setminus K}_0) \psi^{V \setminus K}(\tilde{x}) \psi^v(dx^v), \end{aligned}$$

$$\begin{aligned} D_k :=& \int p^v(x^v_{k-1}, x^v) p^v(x^v, x^v_{k+1}) \\ & \int \prod_{f \in N^m_f(K)} g^f(x^{N(f)}_K, y_t) \prod_{w \in N^m_v(K) \setminus K} p^w(x^w_0, x^w_K) \tilde{\pi}^{V \setminus K}_{k-1}(dx^{V \setminus K}_0) \psi^{V \setminus K}(\tilde{x}) \psi^v(dx^v). \end{aligned}$$

Then:

$$\tilde{\rho}^{(k,v)}_x(A) = \frac{N_k}{D_k}.$$

- If $j = (k', v')$ with $k' \leq k-2$ and $v' \in V$ then: $\quad C_{i,j} = 0$.
- If $j = (k-1, v')$ and $v' \in V$ then $\quad C_{i,j} \leq \begin{cases} \left(1 - \frac{\epsilon_-}{\epsilon_+}\right) & v' = v \\ 0 & v' \neq v \end{cases},$ where the result follows from lemma 4.3, obtained by a majorization of the kernel part.
- If $j = (k, v')$ and $v' \in V$ then $\quad C_{i,j} \leq \begin{cases} \left(1 - \kappa^{2\mathbf{card}(N(v) \cap N(v'))}\right) & v' \in N^2(v) \setminus v \\ 0 & \text{otherwise} \end{cases},$ where the result follows from lemma 4.3, obtained by a majorization of the observation density part. Recall that the only factors that contain v are the one in $N(v)$ so the components that are connected to these factors are the one in $N^2(v)$.
- If $j = (k+1, v')$ and $v' \in V$ then $\quad C_{i,j} \leq \begin{cases} \left(1 - \frac{\epsilon_-}{\epsilon_+}\right) & v' = v \\ 0 & v' \neq v \end{cases},$ where the result follows from lemma 4.3, obtained by a majorization of the kernel part.
- If $j = (k', v')$ with $k' \geq k+2$ and $v' \in V$ then: $\quad C_{i,j} = 0$.

- Consider $i = (T, v)$ and $v \in V$ then:

$$\begin{aligned} &\tilde{\rho}^{(T,v)}_x(A) \\ &= \frac{\int \mathbb{1}_A(\tilde{x}^v_T) \mathbb{1}_{x \setminus x^v_T}(\tilde{x} \setminus \tilde{x}^v_T) \overleftarrow{\tilde{p}}_t(\tilde{x}_{t+1}, \tilde{x}_t) \tilde{\pi}_t(d\tilde{x}_t) \dots \overleftarrow{\tilde{p}}_{T-1}(\tilde{x}_T, \tilde{x}_{T-1}) \tilde{\pi}_{T-1}(d\tilde{x}_{T-1}) \tilde{\pi}_T(d\tilde{x}_T)}{\int \mathbb{1}_{x \setminus x^v_T}(\tilde{x} \setminus \tilde{x}^v_T) \overleftarrow{\tilde{p}}_t(\tilde{x}_{t+1}, \tilde{x}_t) \tilde{\pi}_t(d\tilde{x}_t) \dots \overleftarrow{\tilde{p}}_{T-1}(\tilde{x}_T, \tilde{x}_{T-1}) \tilde{\pi}_{T-1}(d\tilde{x}_{T-1}) \tilde{\pi}_T(d\tilde{x}_T)} \\ &= \frac{\int \mathbb{1}_A(x^v_T) \overleftarrow{\tilde{p}}_{T-1}(x_T, x_{T-1}) (\tilde{\pi}_T)^v_{x_T}(dx^v_T)}{\int \overleftarrow{\tilde{p}}_{T-1}(x_T, x_{T-1}) (\tilde{\pi}_T)^v_{x_T}(dx^v_T)} = \frac{N_T}{D_T}, \end{aligned}$$

where:

$$N_T := \int \mathbb{1}_A(x_T^v) p^v(x_{T-1}^v, x_T^v) \int \prod_{f \in N_f^m(K)} g^f(x_T^{N(f)}, y_T)$$
$$\prod_{w \in N_v^m(K) \setminus K} p^w(x_{T-1}^w, x_T^w) \tilde{\pi}_{T-1}^{V \setminus K}(dx_{T-1}^{V \setminus K}) \psi^{V \setminus K}(dx_T^{V \setminus K}) \psi^v(dx_T^v),$$

$$D_T := \int p^v(x_{T-1}^v, x_T^v) \int \prod_{f \in N_f^m(K)} g^f(x_T^{N(f)}, y_T)$$
$$\prod_{w \in N_v^m(K) \setminus K} p^w(x_{T-1}^w, x_T^w) \tilde{\pi}_{T-1}^{V \setminus K}(dx_{T-1}^{V \setminus K}) \psi^{V \setminus K}(dx_T^{V \setminus K}) \psi^v(dx_T^v),$$

which follows from similar passages as in $i = (k, v)$.

- If $j = (k', v')$ with $k' \leq T-2$ and $v' \in V$ then: $\quad C_{i,j} = 0$.
- If $j = (T-1, v')$ and $v' \in V$ then $\quad C_{i,j} \leq \begin{cases} \left(1 - \frac{\epsilon_-}{\epsilon_+}\right) & v' = v \\ 0 & v' \neq v \end{cases}$,
  where the result follows from lemma 4.3, obtained by a majorization of the kernel part.
- If $j = (T, v')$ and $v' \in V$ then $\quad C_{i,j} \leq \begin{cases} \left(1 - \kappa^{2\mathbf{card}(N(v) \cap N(v'))}\right) & v' \in N^2(v) \setminus v \\ 0 & \text{otherwise} \end{cases}$,
  where the result follows from lemma 4.3, obtained by a majorization of the observation density part. Recall that the only factors that contain v are the one in $N(v)$ so the components that are connected to these factors are the one in $N^2(v)$.

Given the previous results, for any $v \in V$:

$$\sum_{j \in I} e^{m(i,j)} C_{i,j} \leq \begin{cases} \sum_{v' \in V} e^{\beta d(v,v')} \tilde{C}_{v,v'}^{\tilde{\pi}_t} + e^{\beta}\left(1 - \frac{\epsilon_-}{\epsilon_+}\right) & i = (0, v) \\ 2e^{\beta}\left(1 - \frac{\epsilon_-}{\epsilon_+}\right) + \sum_{v' \in N^2(v)} \left(1 - \kappa^{2\mathbf{card}(N(v) \cap N(v'))}\right) e^{\beta d(v,v')} & i = (k, v), \\ e^{\beta}\left(1 - \frac{\epsilon_-}{\epsilon_+}\right) + \sum_{v' \in N^2(v)} \left(1 - \kappa^{2\mathbf{card}(N(v) \cap N(v'))}\right) e^{\beta d(v,v')} & i = (T, v) \end{cases}$$

where $m(i,j) = \beta|k - k'| + \beta d(v, v')$ for $i = (k, v)$ and $j = (k', v')$ with $k, k' \in \{t, \ldots, T\}$ and $v, v' \in V$ is the pseudometric of interest on the index set I. But then given the region $\mathscr{R}_0$ is considered:

$$\max_{i \in I} \sum_{j \in I} C_{i,j} \leq \widetilde{\mathrm{Corr}}(\tilde{\pi}_t, \beta) + 2e^{\beta}\left(1 - \frac{\epsilon_-}{\epsilon_+}\right) + e^{2\beta} \Upsilon^{(2)}\left(1 - \kappa^{2\tilde{\Upsilon}}\right) \leq \frac{1}{2}.$$

Given that $\sum_{j \in I} C_{i,j} \leq \sum_{j \in I} e^{m(i,j)} C_{i,j}$ then the Dobrushin theorem (theorem 4.1) can be applied, meaning that:

$$\left\| \mathsf{R}_{\tilde{\pi}_t} \ldots \mathsf{R}_{\tilde{\pi}_{T-1}} \tilde{\pi}_T - \mathsf{R}_{\tilde{\pi}_t} \ldots \mathsf{R}_{\tilde{\pi}_{T-1}} \pi_T \right\|_J = \left\| \rho - \tilde{\rho} \right\|_{(t,J)} \leq \sum_{v \in J} \sum_{j \in I} D_{(t,v),j} b_j.$$

The second step is to control the quantities b_j, as in (4.2) of theorem 4.1:

$$b_j = \sup_{x \in \mathbb{X}^I} \left\| \rho_x^j - \tilde{\rho}_x^j \right\|.$$

Remark that the form of $\tilde{\rho}_x^i$ is already known from the study on $C_{i,j}$, hence it is enough to compute ρ_x^i and then compare it.

- If $j = (k, v')$ with $k < T$ and $v' \in V$ then:

  $$\rho_x^j(A) = \tilde{\rho}_x^j(A),$$

  because the difference is only on the final integral (on π_T and $\tilde{\pi}_T$) which disappear as consequence of the Markov property derived from the reversed kernel, hence:

  $$b_j = 0.$$

- If $j = (T, v')$ and $v' \in V$ then:

  $$\begin{aligned}
  &\rho_x^{(T,v')}(A) \\
  &= \frac{\int \mathbb{1}_A(\tilde{x}_T^{v'}) \mathbb{1}_{x \setminus x_T^{v'}}(\tilde{x} \setminus \tilde{x}_T^{v'}) \overleftarrow{\tilde{p}}_t(\tilde{x}_{t+1}, \tilde{x}_t) \tilde{\pi}_t(d\tilde{x}_t) \ldots \overleftarrow{\tilde{p}}_{T-1}(\tilde{x}_T, \tilde{x}_{T-1}) \tilde{\pi}_{T-1}(d\tilde{x}_{T-1}) \pi_T(d\tilde{x}_T)}{\int \mathbb{1}_{x \setminus x_T^{v'}}(\tilde{x} \setminus \tilde{x}_T^{v'}) \overleftarrow{\tilde{p}}_t(\tilde{x}_{t+1}, \tilde{x}_t) \tilde{\pi}_t(d\tilde{x}_t) \ldots \overleftarrow{\tilde{p}}_{T-1}(\tilde{x}_T, \tilde{x}_{T-1}) \tilde{\pi}_{T-1}(d\tilde{x}_{T-1}) \pi_T(d\tilde{x}_T)} \\
  &= \frac{\int \mathbb{1}_A(x_T^{v'}) \overleftarrow{\tilde{p}}_{T-1}(x_T, x_{T-1}) (\pi_T)_{x_T}^{v'}(dx_T^{v'})}{\int \overleftarrow{\tilde{p}}_{T-1}(x_T, x_{T-1}) (\pi_T)_{x_T}^{v'}(dx_T^{v'})}.
  \end{aligned}$$

Moreover, given that $\max_{i \in I} \sum\limits_{j \in I} e^{m(i,j)} C_{i,j} \leq \frac{1}{2}$ then lemma 4.5 can be applied and so:

$$\max_{i \in I} \sum_{j \in J} e^{m(i,J)} D_{i,j} \leq 2.$$

By joining step one and step two it follows that:

$$\begin{aligned}
&\left\| \mathsf{R}_{\tilde{\pi}_t} \ldots \mathsf{R}_{\tilde{\pi}_{T-1}} \tilde{\pi}_T - \mathsf{R}_{\tilde{\pi}_t} \ldots \mathsf{R}_{\tilde{\pi}_{T-1}} \pi_T \right\|_J \\
&\quad \leq \sum_{v \in J} \sum_{j \in I} D_{(t,v),j} b_j \quad \leq \sum_{v \in J} \sum_{v' \in V} D_{(t,v),(T,v')} b_{(T,v')} \\
&\quad \leq \sum_{v \in J} \sum_{v' \in V} e^{\beta |T-t| + \beta d(v,v')} D_{(t,v),(T,v')} e^{-\beta |T-t| - \beta d(v,v')} \\
&\qquad\qquad \sup_{x_{T-1}, x_T \in \mathbb{X}^V} \left\| \left(\overset{\tilde{\pi}_{T-1}}{\overleftarrow{\tilde{\pi}_T}} \right)_{x_T, x_{T-1}}^{v'} - \left(\overset{\tilde{\pi}_{T-1}}{\overleftarrow{\pi_T}} \right)_{x_T, x_{T-1}}^{v'} \right\| \\
&\quad \leq 2 e^{-\beta(T-t)} \sum_{v \in J} \max_{v' \in V} \left\{ e^{-\beta d(v,v')} \sup_{x_{T-1}, x_T \in \mathbb{X}^V} \left\| \left(\overset{\tilde{\pi}_{T-1}}{\overleftarrow{\tilde{\pi}_T}} \right)_{x_T, x_{T-1}}^{v'} - \left(\overset{\tilde{\pi}_{T-1}}{\overleftarrow{\pi_T}} \right)_{x_T, x_{T-1}}^{v'} \right\| \right\}.
\end{aligned}$$

■

Proposition 4.8. *Suppose there exist $(\epsilon_-, \epsilon_+, \kappa) \in (0,1)^3$, such that:*

$$\epsilon_- \leq p^v(x^v, z^v) \leq \epsilon_+,$$

for all $x, z \in \mathbb{X}^V, v \in V$. Then for any $v \in V$ and for any $x_0, x_1 \in \mathbb{X}^V$ and $m \in \{0, \ldots, n\}$:

$$\left\| (\overset{\tilde\pi_{T-1}}{\overleftarrow{\pi_T}})^v_{x_1,x_0} - (\overset{\tilde\pi_{T-1}}{\overleftarrow{\tilde\pi_T}})^v_{x_1,x_0} \right\| \leq 2 \left(\frac{\epsilon_+}{\epsilon_-} \right)^2 \sup_{x_1 \in \mathbb{X}^V} \left\| (\pi_T)^v_{x_1} - (\tilde\pi_T)^v_{x_1} \right\|.$$

where $\tilde\pi_t$ is the approximated filtering distribution obtained through recursion 3.12 and π_t is the approximated filtering distribution obtained through recursion 2.4.

Proof. Denote with $\overleftarrow{\tilde p}_t(\cdot,\cdot)$ the reverse kernel with reference distribution the approximated filtering distribution, i.e.:

$$\overleftarrow{\tilde p}_t(z, x) := \frac{p(x,z)}{\int p(\hat x, z) \tilde\pi_t(d\hat x)}.$$

Consider the probability distributions:

$$(\overset{\tilde\pi_{T-1}}{\overleftarrow{\pi_T}})^v_{x_1,x_0}(A) = \frac{\int \mathbb{1}_A(x_1^v) \overleftarrow{\tilde p}_{T-1}(x_1, x_0)(\pi_T)^v_{x_1}(dx_1^v)}{\int \overleftarrow{\tilde p}_{T-1}(x_1, x_0)(\pi_T)^v_{x_1}(dx_1^v)},$$

$$(\overset{\tilde\pi_{T-1}}{\overleftarrow{\tilde\pi_T}})^v_{x_1,x_0}(A) = \frac{\int \mathbb{1}_A(x_1^v) \overleftarrow{\tilde p}_{T-1}(x_1, x_0)(\tilde\pi_T)^v_{x_1}(dx_1^v)}{\int \overleftarrow{\tilde p}_{T-1}(x_1, x_0)(\tilde\pi_T)^v_{x_1}(dx_1^v)}.$$

It can be observed that the form $\int \mathbb{1}_A(x)\Lambda(x)\nu(dx) / \int \Lambda(x)\nu(dx)$ allows to apply lemma 4.4. Hence:

$$\left\| (\overset{\tilde\pi_{T-1}}{\overleftarrow{\pi_T}})^v_{x_1,x_0} - (\overset{\tilde\pi_{T-1}}{\overleftarrow{\tilde\pi_T}})^v_{x_1,x_0} \right\| \leq 2 \left(\frac{\epsilon_+}{\epsilon_-} \right)^2 \sup_{x_1 \in \mathbb{X}^V} \left\| (\pi_t)^v_{x_1} - (\tilde\pi_t)^v_{x_1} \right\|. \tag{4.11}$$

Remark that in this case the function Λ in lemma 4.4 is $\overleftarrow{\tilde p}_{t-1}(\tilde x_1, x_0)\mathbb{1}_{x_1^{V\setminus v}}(\tilde x_1^{V\setminus v})$ hence the result follows from the following observations:

- $\overleftarrow{\tilde p}_{t-1}(\tilde x_1, x_0)\mathbb{1}_{x_1^{V\setminus v}}(\tilde x_1^{V\setminus v}) \leq \left(\frac{\epsilon_+}{\epsilon_-}\right) \frac{\prod_{v' \in V\setminus v} p^{v'}(x_0^{v'}, x_1^{v'})}{\int \prod_{v' \in V\setminus v} p^{v'}(\hat x_0^{v'}, x_1^{v'}) \tilde\pi_{t-1}(d\hat x_0)}$;

- $\overleftarrow{\tilde p}_{t-1}(\tilde x_1, x_0)\mathbb{1}_{x_1^{V\setminus v}}(\tilde x_1^{V\setminus v}) \geq \left(\frac{\epsilon_-}{\epsilon_+}\right) \frac{\prod_{v' \in V\setminus v} p^{v'}(x_0^{v'}, x_1^{v'})}{\int \prod_{v' \in V\setminus v} p^{v'}(\hat x_0^{v'}, x_1^{v'}) \tilde\pi_{t-1}(d\hat x_0)}$;

where the ratio is constant in x_1^v and the inequality holds also for the sup and the inf. ■

4.3.2 Approximate smoothing stability and smoothing error control

Here it must be proved that an initial application of the optimal smoothing operator followed by a sequential application of the approximate smoothing operator to a probability distribution μ is not too different from an initial application of the approximate smoothing operator followed by a sequential application of the approximate smoothing operator to the same probability distribution μ.

Proposition 4.9. *Fix any collection of observations $\{y_1,\dots,y_T\}$ and any partition $\mathcal{K}$ on the set V. There exists a region $\mathscr{R}_0 \subseteq (0,1)^3$, as in corollary 4.1, depending only on $\tilde{\Upsilon}, \Upsilon$ and $\Upsilon^{(2)}$, such that if, for given $(\epsilon_-,\epsilon_+,\kappa) \in \mathscr{R}_0$,*

$$\epsilon_- \le p^v(x^v,z^v) \le \epsilon_+ \quad \textit{and} \quad \kappa \le g^f(x^{N(f)},y_t) \le \frac{1}{\kappa}, \quad \forall x,z \in \mathbb{X}^V, f \in F, v \in V, t \in \{1,\dots,T\},$$

then for $\beta > 0$ small enough depending only on $\tilde{\Upsilon}, \Upsilon, \Upsilon^{(2)}$, $\epsilon_-, \epsilon_+, \kappa$, for any $t \in \{0,\dots,T-1\}$, $m \in \{0,\dots,n\}$ and for any $s = \{0,\dots,T-t+1\}$:

$$\begin{aligned}\big\| \mathsf{R}_{\tilde{\pi}_t} \dots \mathsf{R}_{\tilde{\pi}_{t+s-1}} \mathsf{R}_{\tilde{\pi}_{t+s}} \mu - \mathsf{R}_{\tilde{\pi}_t} \dots \mathsf{R}_{\tilde{\pi}_{t+s-1}} \mathsf{R}_{\pi_{t+s}} \mu \big\|_J \\ \le 2e^{-\beta s} \sum_{v \in J} \max_{v' \in V} \left\{ e^{-\beta d(v,v')} \left\| (\tilde{\pi}_{t+s})^{v'}_{x_{t+s},z} - (\pi_{t+s})^{v'}_{x_{t+s},z} \right\| \right\}.\end{aligned}$$

Proof. Denote with $\overleftarrow{p}_t(\cdot,\cdot)$ the reverse kernel with reference distribution the optimal filtering distribution and with $\overleftarrow{\tilde{p}}_t(\cdot,\cdot)$ the reverse kernel with reference distribution the approximated filtering distribution, i.e.:

$$\overleftarrow{p}_t(z,x) := \frac{p(x,z)}{\int p(\hat{x},z)\pi_t(d\hat{x})} \quad \text{and} \quad \overleftarrow{\tilde{p}}_t(z,x) := \frac{p(x,z)}{\int p(\hat{x},z)\tilde{\pi}_t(d\hat{x})}$$

Then:

$$\begin{aligned}&\mathsf{R}_{\tilde{\pi}_t} \dots \mathsf{R}_{\tilde{\pi}_{t+s}} \mu(A) \\ &= \int \mathbb{1}_A(x_t) \overleftarrow{\tilde{p}}_t(x_{t+1},x_t)\tilde{\pi}_t(dx_t) \dots \overleftarrow{\tilde{p}}_{t+s}(x_{t+s+1},x_{t+s})\tilde{\pi}_{t+s}(dx_{t+s})\mu(dx_{t+s+1}),\end{aligned}$$

$$\begin{aligned}&\mathsf{R}_{\tilde{\pi}_t} \dots \mathsf{R}_{\pi_{t+s}} \mu(A) \\ &= \int \mathbb{1}_A(x_t) \overleftarrow{\tilde{p}}_t(x_{t+1},x_t)\tilde{\pi}_t(dx_t) \dots \overleftarrow{p}_{t+s}(x_{t+s+1},x_{t+s})\pi_{t+s}(dx_{t+s})\mu(dx_{t+s+1}).\end{aligned}$$

Define the probability distributions:

$$\rho_z(A) := \int \mathbb{1}_A(x_t,\dots,x_{t+s}) \overleftarrow{\tilde{p}}_t(x_{t+1},x_t)\tilde{\pi}_t(dx_t) \dots \overleftarrow{p}_{t+s}(z,x_{t+s})\pi_{t+s}(dx_{t+s}),$$

$$\tilde{\rho}_z(A) := \int \mathbb{1}_A(x_t,\dots,x_{t+s}) \overleftarrow{\tilde{p}}_t(x_{t+1},x_t)\tilde{\pi}_t(dx_t) \dots \overleftarrow{\tilde{p}}_{t+s}(z,x_{t+s})\tilde{\pi}_{t+s}(dx_{t+s}).$$

The quantity of interest can be reformulated as follows:

$$
\begin{aligned}
&\left\|\mathsf{R}_{\tilde\pi_t}\dots\mathsf{R}_{\tilde\pi_{t+s-1}}\mathsf{R}_{\tilde\pi_{t+s}}\mu-\mathsf{R}_{\tilde\pi_t}\dots\mathsf{R}_{\tilde\pi_{t+s-1}}\mathsf{R}_{\pi_{t+s}}\mu\right\|_J\\
&=\sup_{A\in\sigma(\mathbb{X}^J))}\left|\int\mathbb{1}_A(x_t)\overleftarrow{\tilde p}_t(x_{t+1},x_t)\tilde\pi_t(dx_t)\dots\overleftarrow{\tilde p}_{t+s}(z,x_{t+s})\tilde\pi_{t+s}(dx_{t+s})\mu(dz)\right.\\
&\quad\left.-\int\mathbb{1}_A(x_t)\overleftarrow{\tilde p}_t(x_{t+1},x_t)\tilde\pi_t(dx_t)\dots\overleftarrow{p}_{t+s}(z,x_{t+s})\pi_{t+s}(dx_{t+s})\mu(dz)\right|\\
&=\sup_{A\in\sigma(\mathbb{X}^J))}\left|\int\left[\int\mathbb{1}_A(x_t)\overleftarrow{\tilde p}_t(x_{t+1},x_t)\tilde\pi_t(dx_t)\dots\overleftarrow{\tilde p}_{t+s}(z,x_{t+s})\tilde\pi_{t+s}(dx_{t+s})\right.\right.\\
&\quad\left.\left.-\int\mathbb{1}_A(x_t)\overleftarrow{\tilde p}_t(x_{t+1},x_t)\tilde\pi_t(dx_t)\dots\overleftarrow{p}_{t+s}(z,x_{t+s})\pi_{t+s}(dx_{t+s})\right]\mu(dz)\right|\\
&\le\sup_{A\in\sigma(\mathbb{X}^J))}\int\left|\left[\int\mathbb{1}_A(x_t)\overleftarrow{\tilde p}_t(x_{t+1},x_t)\tilde\pi_t(dx_t)\dots\overleftarrow{\tilde p}_{t+s}(z,x_{t+s})\tilde\pi_{t+s}(dx_{t+s})\right.\right.\\
&\quad\left.\left.-\int\mathbb{1}_A(x_t)\overleftarrow{\tilde p}_t(x_{t+1},x_t)\tilde\pi_t(dx_t)\dots\overleftarrow{p}_{t+s}(z,x_{t+s})\pi_{t+s}(dx_{t+s})\right]\right|\mu(dz)\\
&\le\sup_{z\in\mathbb{X}^V}\left\{\sup_{A\in\sigma(\mathbb{X}^J))}\left|\left[\int\mathbb{1}_A(x_t)\overleftarrow{\tilde p}_t(x_{t+1},x_t)\tilde\pi_t(dx_t)\dots\overleftarrow{\tilde p}_{t+s}(z,x_{t+s})\tilde\pi_{t+s}(dx_{t+s})\right.\right.\right.\\
&\quad\left.\left.\left.-\int\mathbb{1}_A(x_t)\overleftarrow{\tilde p}_t(x_{t+1},x_t)\tilde\pi_t(dx_t)\dots\overleftarrow{p}_{t+s}(z,x_{t+s})\pi_{t+s}(dx_{t+s})\right]\right|\right\}\\
&=\sup_{z\in\mathbb{X}^V}\left\|\rho_z-\tilde\rho_z\right\|_{(t,J)}.
\end{aligned}
$$

Hence it is enough to find a bound for $\left\|\rho_z-\tilde\rho_z\right\|_{(t,J)}$ to guarantee the proof of the statement. The Dobrushin theorem can be used on the distributions $\rho_z,\tilde\rho_z$ where the index set is $I=\bigcup_{k=t}^{t+s}(k,V)$ and the subset is (t,J).

The first step is to bound $C_{i,j}$ for all the possible combination of $i,j\in I$, as in (4.2) of theorem 4.1. In the following passages the notation $x=(x_t,\dots,x_{t+s})$ is considered, where $x_k\in\mathbb{X}^V$ for $k=t,\dots,t+s$ and $x\setminus x_k^v:=(x_t,\dots,x_k^{V\setminus v},\dots,x_{t+s})$ (and the same with a tilde).

- Consider $i=(t,v)$ and $v\in V$ then:

$$
\begin{aligned}
(\tilde\rho_z)_x^{(t,v)}(A)&=\frac{\int\mathbb{1}_A(\tilde x_t^v)\mathbb{1}_{x\setminus x_t^v}(\tilde x\setminus\tilde x_t^v)\overleftarrow{\tilde p}_t(\tilde x_{t+1},\tilde x_t)\tilde\pi_t(d\tilde x_t)\dots\overleftarrow{\tilde p}_{t+s}(z,\tilde x_{t+s})\tilde\pi_{t+s}(d\tilde x_{t+s})}{\int\mathbb{1}_{x\setminus x_t^v}(\tilde x\setminus\tilde x_t^v)\overleftarrow{\tilde p}_t(\tilde x_{t+1},\tilde x_t)\tilde\pi_t(d\tilde x_t)\dots\overleftarrow{\tilde p}_{t+s}(z,\tilde x_{t+s})\tilde\pi_{t+s}(d\tilde x_{t+s})}\\
&=\frac{\int\mathbb{1}_A(x_t^v)\overleftarrow{\tilde p}_t(x_{t+1},x_t)(\tilde\pi_t)_{x_t}^v(dx_t^v)}{\int\overleftarrow{\tilde p}_t(x_{t+1},x_t)(\tilde\pi_t)_{x_t}^v(dx_t^v)}=\frac{\int\mathbb{1}_A(x_t^v)p^v(x_t^v,x_{t+1}^v)(\tilde\pi_t)_{x_t}^v(dx_t^v)}{\int p^v(x_t^v,x_{t+1}^v)(\tilde\pi_t)_{x_t}^v(dx_t^v)},
\end{aligned}
$$

  where the last passage follow from the independence of the numerator of the reverse kernel from x_t. Now the different cases in which ρ_x^i can differ from $\rho_{\tilde x}^i$, where $x^{I\setminus j}=\tilde x^{I\setminus j}$, must be distinguished.

  - If $j=(t,v')$ and $v'\in V$ then: $\quad C_{i,j}\le\tilde C_{v,v'}^{\tilde\pi_t}$.

- If $j=(t+1,v')$ and $v'\in V$ then: $\quad C_{i,j}\leq \begin{cases} \left(1-\frac{\epsilon_-}{\epsilon_+}\right) & v'=v \\ 0 & v'\neq v \end{cases}$,
  where the result follows from lemma 4.3, obtained by a majorization of the kernel part.
- If $j=(k,v')$ with $k>t+1$ and $v'\in V$ then: $\quad C_{i,j}=0$,
  which is obvious given that in ρ^i_x there is no dependence on x_k with $k>t+1$.

- Consider $i=(k,v)$ with $t+1<k\leq t+s$ and $v\in K\subseteq V$ then:

$$\begin{aligned}
&(\tilde{\rho}_z)^{(k,v)}_x(A)\\
&=\frac{\int \mathbb{1}_A(\tilde{x}^v_k)\mathbb{1}_{x\setminus x^v_k}(\tilde{x}\setminus\tilde{x}^v_k)\overleftarrow{\tilde{p}}_t(\tilde{x}_{t+1},\tilde{x}_t)\tilde{\pi}_t(d\tilde{x}_t)\dots\overleftarrow{\tilde{p}}_{t+s}(z,\tilde{x}_{t+s})\tilde{\pi}_{t+s}(d\tilde{x}_{t+s})}{\int \mathbb{1}_{x\setminus x^v_k}(\tilde{x}\setminus\tilde{x}^v_k)\overleftarrow{\tilde{p}}_t(\tilde{x}_{t+1},\tilde{x}_t)\tilde{\pi}_t(d\tilde{x}_t)\dots\overleftarrow{\tilde{p}}_{t+s}(z,\tilde{x}_{t+s})\tilde{\pi}_{t+s}(d\tilde{x}_{t+s})}\\
&=\frac{\int \mathbb{1}_A(x^v_k)\overleftarrow{\tilde{p}}_{k-1}(x_k,x_{k-1})\overleftarrow{\tilde{p}}_k(x_{k+1},x_k)(\tilde{\pi}_k)^v_{x_k}(dx_k)}{\int \overleftarrow{\tilde{p}}_{k-1}(x_k,x_{k-1})\overleftarrow{\tilde{p}}_k(x_{k+1},x_k)(\tilde{\pi}_k)^v_{x_k}(dx_k)}\\
&=\frac{\int \mathbb{1}_A(\tilde{x}^v_k)\overleftarrow{\tilde{p}}_{k-1}(x_k,x_{k-1})p^v(x^v_k,x^v_{k+1})(\tilde{\pi}_k)^v_{x_k}(dx_k)}{\int \overleftarrow{\tilde{p}}_{k-1}(x_k,x_{k-1})p^v(x^v_k,x^v_{k+1})(\tilde{\pi}_k)^v_{x_k}(dx_k)}=\frac{N_k}{D_k},
\end{aligned}$$

where $x_{t+s+1}=z$ and everything follows the same procedure explained in the proof of proposition 4.7, in particular:

$$\begin{aligned}
N_k &:= \int \mathbb{1}_A(x^v)p^v(x^v_{k-1},x^v)p^v(x^v,x^v_{k+1})\\
&\quad \int \prod_{f\in N^m_f(K)} g^f(x^{N(f)}_K,y_t)\prod_{w\in N^m_v(K)\setminus K} p^w(x^w_0,x^w_K)\tilde{\pi}^{V\setminus K}_{k-1}(dx^{V\setminus K}_0)\psi^{V\setminus K}(\tilde{x})\psi^v(dx^v),
\end{aligned}$$

$$\begin{aligned}
D_k &:= \int p^v(x^v_{k-1},x^v)p^v(x^v,x^v_{k+1})\\
&\quad \int \prod_{f\in N^m_f(K)} g^f(x^{N(f)}_K,y_t)\prod_{w\in N^m_v(K)\setminus K} p^w(x^w_0,x^w_K)\tilde{\pi}^{V\setminus K}_{k-1}(dx^{V\setminus K}_0)\psi^{V\setminus K}(\tilde{x})\psi^v(dx^v).
\end{aligned}$$

  - If $j=(k',v')$ with $k'\leq k-2$ and $v'\in V$ then: $\quad C_{i,j}=0$.
  - If $j=(k-1,v')$ and $v'\in V$ then $\quad C_{i,j}\leq \begin{cases} \left(1-\frac{\epsilon_-}{\epsilon_+}\right) & v'=v \\ 0 & v'\neq v \end{cases}$,
    where the result follows from lemma 4.3, obtained by a majorization of the kernel part.
  - If $j=(k,v')$ and $v'\in V$ then $\quad C_{i,j}\leq \begin{cases} \left(1-\kappa^{\mathbf{2card}(N(v)\cap N(v'))}\right) & v'\in N^2(v)\setminus v \\ 0 & \text{otherwise} \end{cases}$,
    where the result follows from lemma 4.3, obtained by a majorization of the observation density part. Recall that the only factors that contain v are the one in $N(v)$ so the components that are connected to these factors are the one in $N^2(v)$.

- If $j=(k+1,v')$ and $v'\in V$ then $\quad C_{i,j}\leq\begin{cases}\left(1-\frac{\epsilon_-}{\epsilon_+}\right) & v'=v\\ 0 & v'\neq v\end{cases}$,

  where the result follows from lemma 4.3, obtained by a majorization of the kernel part.
- If $j=(k',v')$ with $k'\geq k+2$ and $v'\in V$ then: $\quad C_{i,j}=0$.

Given the previous results, for any $v\in V$ and $t+1<k<t+s$:

$$\sum_{j\in I}e^{m(i,j)}C_{i,j}\leq\begin{cases}\sum\limits_{v'\in V}e^{\beta d(v,v')}\tilde{C}^{\tilde{\pi}_t}_{v,v'}+e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right) & i=(0,v)\\ 2e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+\sum\limits_{v'\in N^2(v)}(1-\kappa^{2\mathbf{card}(N(v)\cap N(v'))})e^{\beta d(v,v')} & i=(k,v),\end{cases}$$

where $m(i,j)=\beta|k-k'|+\beta d(v,v')$ for $i=(k,v)$ and $j=(k',v')$ with $k,k'\in\{t,\dots,t+s\}$ and $v,v'\in V$ is the pseudometric of interest on the index set I. But then by combining the above calculation with the corollary 4.1:

$$\max_{i\in I}\sum_{j\in I}C_{i,j}\leq\widetilde{\mathrm{Corr}}(\tilde{\pi}_t,\beta)+2e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right)+e^{2\beta}\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)\leq\frac{1}{2}.$$

Given that $\sum_{j\in I}C_{i,j}\leq\sum_{j\in I}e^{m(i,j)}C_{i,j}$ then the Dobrushin theorem (theorem 4.1) can be applied, meaning that:

$$\left\|\mathsf{R}_{\tilde{\pi}_t}\dots\mathsf{R}_{\tilde{\pi}_{t+s-1}}\mathsf{R}_{\tilde{\pi}_{t+s}}\mu-\mathsf{R}_{\tilde{\pi}_t}\dots\mathsf{R}_{\tilde{\pi}_{t+s-1}}\mathsf{R}_{\pi_{t+s}}\mu\right\|_J=\sup_{z\in\mathbb{X}^V}\left\|\rho_z-\tilde{\rho}_z\right\|_{(t,J)}\leq\sum_{v\in J}\sum_{j\in I}D_{(t,v),j}b_j.$$

The second step is to control the quantities b_j, as in (4.2) of theorem 4.1:

$$b_j=\sup_{x\in\mathbb{X}^I}\left\|(\rho_z)^j_x-(\tilde{\rho}_z)^j_x\right\|.$$

Remark that the form of $(\tilde{\rho}_z)^i_x$ is already known from the study on $C_{i,j}$, hence it is enough to compute $(\rho_z)^i_x$ and then compare it.

- If $j=(k,v')$ with $k<t+s$ and $v'\in V$ then:

  $$(\rho_z)^j_x(A)=(\tilde{\rho}_z)^j_x(A),$$

  because the difference is only on the final kernel which disappear as consequence of the Markov property derived from the reversed kernel, hence:

  $$b_j=0.$$

- If $j = (t+s, v')$ and $v' \in V$ then:

$$\begin{aligned}
\rho_x^{(t+s,v')}(A) &= \frac{\int \mathbb{1}_A(\tilde{x}_{t+s}^{v'})\mathbb{1}_{x \setminus x_{t+s}^{v'}}(\tilde{x} \setminus \tilde{x}_{t+s}^{v'})\overleftarrow{\tilde{p}}_t(\tilde{x}_{t+1}, \tilde{x}_t)\tilde{\pi}_t(d\tilde{x}_t)\dots \overleftarrow{p}_{t+s}(z, \tilde{x}_{t+s})\pi_{t+s}(d\tilde{x}_{t+s})}{\int \mathbb{1}_{x \setminus x_{t+s}^{v'}}(\tilde{x} \setminus \tilde{x}_{t+s}^{v'})\overleftarrow{\tilde{p}}_t(\tilde{x}_{t+1}, \tilde{x}_t)\tilde{\pi}_t(d\tilde{x}_t)\dots \overleftarrow{p}_{t+s}(z, \tilde{x}_{t+s})\pi_{t+s}(d\tilde{x}_{t+s})} \\
&= \frac{\int \mathbb{1}_A(x_{t+s}^{v'})\overleftarrow{\tilde{p}}_{t+s-1}(x_{t+s}, x_{t+s-1})\overleftarrow{p}_{t+s}(z, x_{t+s})(\pi_{t+s})_{x_{t+s}}^{v'}(dx_{t+s}^{v'})}{\int \overleftarrow{\tilde{p}}_{t+s-1}(x_{t+s}, x_{t+s-1})\overleftarrow{p}_{t+s}(z, x_{t+s})(\pi_{t+s})_{x_{t+s}}^{v'}(dx_{t+s}^{v'})} \\
&= \frac{\int \mathbb{1}_A(x_{t+s}^{v'})\overleftarrow{\tilde{p}}_{t+s-1}(x_{t+s}, x_{t+s-1})p^v(x_{t+s}^v, z^v)(\pi_{t+s})_{x_{t+s}}^{v'}(dx_{t+s}^{v'})}{\int \overleftarrow{\tilde{p}}_{t+s-1}(x_{t+s}, x_{t+s-1})p^v(x_{t+s}^v, z^v)(\pi_{t+s})_{x_{t+s}}^{v'}(dx_{t+s}^{v'})} \\
&= \frac{\int \mathbb{1}_A(x_{t+s}^{v'})\overleftarrow{\tilde{p}}_{t+s-1}(x_{t+s}, x_{t+s-1})(\pi_{t+s})_{x_{t+s},z}^{v'}(dx_{t+s}^{v'})}{\int \overleftarrow{\tilde{p}}_{t+s-1}(x_{t+s}, x_{t+s-1})(\pi_{t+s})_{x_{t+s},z}^{v'}(dx_{t+s}^{v'})},
\end{aligned}$$

where the last few passages follow from: the factorization of the transition kernel $p(x,z)$; the fact that z is a constant and at the denominator of the reversed kernel the dependent variable x_{t+s} is integrated out; the fact that the numerator and the denominator can be rewritten as integrals with respect to the one step forward conditional distribution. Now from the same procedure as in(4.11) in proposition 4.8:

$$b_j \leq 2\left(\frac{\epsilon_+}{\epsilon_-}\right)^2 \sup_{x_{t+s}, z \in \mathbb{X}^V} \left\| (\tilde{\pi}_{t+s})_{x_{t+s},z}^{v'} - (\pi_{t+s})_{x_{t+s},z}^{v'} \right\|.$$

Moreover, given that $\max_{i \in I} \sum\limits_{j \in I} e^{m(i,j)} C_{i,j} \leq \frac{1}{2}$ then lemma 4.5 can be applied and so:

$$\max_{i \in I} \sum_{j \in J} e^{m(i,J)} D_{i,j} \leq 2.$$

By joining step one and step two it follows that:

$$\begin{aligned}
\left\| \mathsf{R}_{\tilde{\pi}_t} \dots \mathsf{R}_{\tilde{\pi}_{t+s-1}} \mathsf{R}_{\tilde{\pi}_{t+s}} \mu - \mathsf{R}_{\tilde{\pi}_t} \dots \mathsf{R}_{\tilde{\pi}_{t+s-1}} \mathsf{R}_{\pi_{t+s}} \mu \right\|_J &= \sup_{z \in \mathbb{X}^V} \left\| \rho_z - \tilde{\rho}_z \right\|_{(t,J)} \\
\leq \sup_{z \in \mathbb{X}^V} \sum_{v \in J} \sum_{j \in I} D_{(t,v),j} b_j \leq \sum_{v \in J} \sum_{v' \in V} D_{(t,v),(t+s,v')} b_{(t+s,v')} \\
\leq \sup_{z \in \mathbb{X}^V} \sum_{v \in J} \sum_{v' \in V} e^{\beta|t+s-t| + \beta d(v,v')} D_{(t,v),(t+s,v')} e^{-\beta|t+s-t| - \beta d(v,v')} \\
2\left(\frac{\epsilon_+}{\epsilon_-}\right)^2 \sup_{x_{t+s}, z \in \mathbb{X}^V} \left\| (\tilde{\pi}_{t+s})_{x_{t+s},z}^{v'} - (\pi_{t+s})_{x_{t+s},z}^{v'} \right\| \\
\leq 4\left(\frac{\epsilon_+}{\epsilon_-}\right)^2 e^{-\beta s} \sum_{v \in J} \max_{v' \in V} \left\{ e^{-\beta d(v,v')} \left\| (\tilde{\pi}_{t+s})_{x_{t+s},z}^{v'} - (\pi_{t+s})_{x_{t+s},z}^{v'} \right\| \right\} \\
= 4\left(\frac{\epsilon_+}{\epsilon_-}\right)^2 e^{-\beta s} \sum_{v \in J} \max_{v' \in V} \left\{ e^{-\beta d(v,v')} \left\| (\tilde{\pi}_{t+s})_{x_{t+s},z}^{v'} - (\pi_{t+s})_{x_{t+s},z}^{v'} \right\| \right\}.
\end{aligned}$$

■

Proposition 4.10. *Fix any collection of observations $\{y_1,\dots,y_T\}$ and any partition $\mathscr{K}$ on the set V. There exists a region $\tilde{\mathscr{R}}_0 \subseteq (0,1)^3$, depending only on $\tilde{\Upsilon}, \Upsilon$ and $\Upsilon^{(2)}$, such that if, for given $(\epsilon_-,\epsilon_+,\kappa) \in \tilde{\mathscr{R}}_0$,*

$$\epsilon_- \leq p^v(x^v,z^v) \leq \epsilon_+ \quad \textit{and} \quad \kappa \leq g^f(x^{N(f)},y_t) \leq \frac{1}{\kappa},$$

for all $x,z \in \mathbb{X}^V, f \in F, v \in V, t \in \{1,\dots,T\}$, then for $\beta > \log(2)$ small enough depending only on $\tilde{\Upsilon}, \Upsilon, \Upsilon^{(2)}$, $\epsilon_-,\epsilon_+,\kappa$, chosen according to corollary 4.1, for any $t \in \{1,\dots,T\}$ and $m \in \{0,\dots,n\}$:

$$\left\|(\pi_t)^v_{x_1,z} - (\tilde{\pi}_t)^v_{x_1,z}\right\| \leq \frac{\tau(\beta,\kappa,m,\mathscr{K})e^\beta}{e^\beta - 2}, \quad \forall t \in \{1,\dots,T\}, \forall x_1, z \in \mathbb{X}^V,$$

where $\tilde{\pi}_t$ is the approximated filtering distribution obtained through recursion 3.12,

$$\tau(\beta,\kappa,m,\mathscr{K}) := 2\left(1-\kappa^{a(\mathscr{K})}\right) + 4e^{-\beta m}\left(1-\kappa^{b(\mathscr{K},m)}\right),$$

and $a(\mathscr{K})$ and $b(m,\mathscr{K})$ are as in proposition 4.2.

Proof. Consider the quantity of interest, it is possible to decompose it as follow:

$$\left\|(\pi_t)^v_{x_1,z} - (\tilde{\pi}_t)^v_{x_1,z}\right\| \leq \left\|(\mathsf{F}_t\pi_{t-1})^v_{x_1,z} - (\mathsf{F}_t\tilde{\pi}_{t-1})^v_{x_1,z}\right\| + \left\|(\mathsf{F}_t\tilde{\pi}_{t-1})^v_{x_1,z} - (\tilde{\mathsf{F}}^m_t\tilde{\pi}_{t-1})^v_{x_1,z}\right\|,$$

where the second quantity can be controlled using proposition 4.2 given that corollary 4.1 holds, from the choices of $\beta,\epsilon_-,\epsilon_+,\kappa$, while for the first quantity the Dobrushin machinery must be used.

Consider the probability distributions:

$$\rho(A) := \frac{\int \mathbb{1}_A(x_0,x_1^v)\prod_{f\in N(v)} g^f(x_1^{N(f)},y_t)p^v(x_0^v,x_1^v)\pi_{t-1}(dx_0)p^v(x_1^v,z^v)\psi^v(x_1^v)}{\int \prod_{f\in N(v)} g^f(x_1^{N(f)},y_t)p^v(x_0^v,x_1^v)\pi_{t-1}(dx_0)p^v(x_1^v,z^v)\psi^v(x_1^v)},$$

$$\tilde{\rho}(A) := \frac{\int \mathbb{1}_A(x_0,x_1^v)\prod_{f\in N(v)} g^f(x_1^{N(f)},y_t)p^v(x_0^v,x_1^v)\tilde{\pi}_{t-1}(dx_0)p^v(x_1^v,z^v)\psi^v(x_1^v)}{\int \prod_{f\in N(v)} g^f(x_1^{N(f)},y_t)p^v(x_0^v,x_1^v)\tilde{\pi}_{t-1}(dx_0)p^v(x_1^v,z^v)\psi^v(x_1^v)}.$$

It can be observed that:

$$\left\|\rho - \tilde{\rho}\right\|_{(1,v)} = \left\|(\mathsf{F}_t\pi_{t-1})^v_{x_1,z} - (\mathsf{F}_t\tilde{\pi}_{t-1})^v_{x_1,z}\right\|.$$

So again the Dobrushin theorem can be applied to $\rho,\tilde{\rho}$ where the index set is $I = (0,V) \cup (1,v)$.

The first step is to bound $C_{i,j}$ for all the possible combination of $i,j \in I$, as in (4.2) of theorem 4.1.

- Consider $i=(0,b)$ and $b \in V$ then:

$$
\begin{aligned}
&\tilde{\rho}^{(0,b)}_{x_0,x_1}(A)\\
&= \frac{\int \mathbb{1}_A(\tilde{x}^b_0)\mathbb{1}_{\{x_0^{V\setminus b},x_1^v\}}(\tilde{x}_0^{V\setminus b},\tilde{x}_1^v)\prod_{f\in N(v)} g^f(\tilde{x}_1^{N(f)},y_t)p^v(\tilde{x}^v_0,\tilde{x}^v_1)\tilde{\pi}_{t-1}(d\tilde{x}_0)p^v(\tilde{x}^v_1,z^v)\psi^v(\tilde{x}^v_1)}{\int \mathbb{1}_{\mathbb{X}}(\tilde{x}^b_0)\mathbb{1}_{\{x_0^{V\setminus b},x_1^v\}}(\tilde{x}_0^{V\setminus b},\tilde{x}_1^v)\prod_{f\in N(v)} g^f(\tilde{x}_1^{N(f)},y_t)p^v(\tilde{x}^v_0,\tilde{x}^v_1)\tilde{\pi}_{t-1}(d\tilde{x}_0)p^v(\tilde{x}^v_1,z^v)\psi^v(\tilde{x}^v_1)}\\
&= \frac{\prod_{f\in N(v)} g^f(x_1^{N(f)},y_t)p^v(x^v_1,z^v)}{\prod_{f\in N(v)} g^f(x_1^{N(f)},y_t)p^v(x^v_1,z^v)}\frac{\int \mathbb{1}_A(\tilde{x}^b_0)\mathbb{1}_{\{x_0^{V\setminus b},x_1^v\}}(\tilde{x}_0^{V\setminus b},\tilde{x}_1^v)p^v(\tilde{x}^v_0,\tilde{x}^v_1)\tilde{\pi}_{t-1}(d\tilde{x}_0)\psi^v(\tilde{x}^v_1)}{\int \mathbb{1}_{\mathbb{X}}(\tilde{x}^b_0)\mathbb{1}_{\{x_0^{V\setminus b},x_1^v\}}(\tilde{x}_0^{V\setminus b},\tilde{x}_1^v)p^v(\tilde{x}^v_0,\tilde{x}^v_1)\tilde{\pi}_{t-1}(d\tilde{x}_0)\psi^v(\tilde{x}^v_1)}\\
&= \frac{\int \mathbb{1}_A(x^b_0)p^b(x^b_0,x^b_1)(\tilde{\pi}_{t-1})^b_{x_0}(dx^b_0)}{\int p^b(x^b_0,x^b_1)(\tilde{\pi}_{t-1})^b_{x_0}(dx^b_0)}.
\end{aligned}
$$

  - If $j=(0,b')$ and $b' \in V$ then: $\quad C_{i,j} \le \tilde{C}^{\tilde{\pi}_{t-1}}_{b,b'}$.
  - If $j=(1,v)$ then by lemma 4.3: $\quad C_{i,j} \le \begin{cases} \left(1-\frac{\epsilon_-}{\epsilon_+}\right) & b=v \\ 0 & \text{otherwise} \end{cases}$,

    where the kernel part is upper bounded.

- Consider $i=(1,v)$ then:

$$
\begin{aligned}
\tilde{\rho}^{(1,v)}_{x_0,x_1}(A) &= \frac{\int \mathbb{1}_A(\tilde{x}^v_1)\mathbb{1}_{x_0}(\tilde{x}^V_0)\prod_{f\in N(v)} g^f(\tilde{x}_1^{N(f)},y_t)p^v(\tilde{x}^v_0,\tilde{x}^v_1)\tilde{\pi}_{t-1}(d\tilde{x}_0)p^v(\tilde{x}^v_1,z^v)\psi^v(\tilde{x}^v_1)}{\int \mathbb{1}_{\mathbb{X}}(\tilde{x}^v_1)\mathbb{1}_{x_0}(\tilde{x}^V_0)\prod_{f\in N(v)} g^f(\tilde{x}_1^{N(f)},y_t)p^v(\tilde{x}^v_0,\tilde{x}^v_1)\tilde{\pi}_{t-1}(d\tilde{x}_0)p^v(\tilde{x}^v_1,z^v)\psi^v(\tilde{x}^v_1)}\\
&= \frac{\int \mathbb{1}_A(x^v_1)\prod_{f\in N(v)} g^f(x_1^{N(f)},y_t)p^v(x^v_0,x^v_1)p^v(x^v_1,z^v)\psi^v(dx^v_1)}{\int \prod_{f\in N(v)} g^f(x_1^{N(f)},y_t)p^v(x^v_0,x^v_1)p^v(x^v_1,z^v)\psi^v(dx^v_1)}.
\end{aligned}
$$

  - If $j=(0,b)$ and $b \in V$ then by lemma 4.3: $\quad C_{i,j} \le \begin{cases} \left(1-\frac{\epsilon_-}{\epsilon_+}\right) & b=v \\ 0 & \text{otherwise} \end{cases}$,

    where the kernel part is upper bounded.
  - If $j=(1,v)$ then: $\quad C_{i,j}=0$.

But then:

$$
\max_{i\in I}\sum_{j\in I} e^{m(i,j)}C_{i,j} \le \widetilde{\mathrm{Corr}}(\tilde{\pi}_{t-1},\beta) + e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right).
$$

Given that $\beta > \log(2)$ is desired, $\epsilon_-,\epsilon_+,\kappa,\beta$ can not be chosen according to corollary 4.1. To overcome this it is enough to modify the region $\mathscr{R}_0$:

$$
\tilde{\mathscr{R}}_0 := \left\{ (\epsilon_-,\epsilon_+,\kappa)\in(0,1)^3 : \frac{\kappa^{2\Upsilon}-8\left(1-\frac{\epsilon_-}{\epsilon_+}\right)}{4\Upsilon^{(2)}\left(1-\kappa^{2\tilde{\Upsilon}}\right)\kappa^{2\Upsilon}} > 4 \quad \text{and} \right.
$$
$$
\left. \frac{1-4\left(1-\frac{\epsilon_-}{\epsilon_+}\right)}{6\Upsilon^{(2)}(1-\kappa^{2\tilde{\Upsilon}})+4\left(1-\frac{\epsilon_-}{\epsilon_+}\right)} > 4 \right\}.
$$

hence for $(\epsilon_-,\epsilon_+,\kappa)\in\tilde{\mathscr{R}}_0$ and β as in corollary 4.1:

$$\widetilde{\mathrm{Corr}}(\tilde{\pi}_{t-1},\beta)+e^{\beta}\left(1-\frac{\epsilon_-}{\epsilon_+}\right)\leq\frac{1}{2}.$$

Hence the Dobrushin theorem applies:

$$\left\|\rho-\tilde{\rho}\right\|_{(1,v)}\leq\sum_{j\in I}D_{(1,v),j}b_j.$$

The second step is to control the quantities b_j, as in (4.2) of theorem 4.1. Remark that the conditional distributions of ρ have the same form of the ones of $\tilde{\rho}$ with $\tilde{\pi}_{t-1}$ instead of π_{t-1}

- If $j=(0,b')$ and $b'\in V$ then:

$$\rho^{(0,b')}_{x_0,x_1}(A)=\frac{\int\mathbb{I}_A(x_0^{b'})p^{b'}(x_0^{b'},x_1^{b'})(\pi_{t-1})^{b'}_{x_0}(dx_0^{b'})}{\int p^{b'}(x_0^{b'},x_1^{b'})(\pi_{t-1})^{b'}_{x_0}(dx_0^{b'})},$$

  hence:

$$b_j=\sup_{x_0,x_1\in\mathbb{X}^V}\left\|(\pi_{t-1})^{b'}_{x_0,x_1}-(\tilde{\pi}_{t-1})^{b'}_{x_0,x_1}\right\|.$$

- If $j=(1,v)$ then:

$$\tilde{\rho}^{(1,v)}_{x_0,x_1}(A)=\frac{\int\mathbb{I}_A(x_1^v)\prod_{f\in N(v)}g^f(x_1^{N(f)},y_t)p^v(x_0^v,x_1^v)p^v(x_1^v,z^v)\psi^v(dx_1^v)}{\int\prod_{f\in N(v)}g^f(x_1^{N(f)},y_t)p^v(x_0^v,x_1^v)p^v(x_1^v,z^v)\psi^v(dx_1^v)}.$$

  hence:

$$b_j=0,$$

  because the only difference between $\rho,\tilde{\rho}$ is on π_{t-1} and $\tilde{\pi}_{t-1}$.

By putting all together, it can be concluded that:

$$\begin{aligned}\|(F_t\pi_{t-1})^v_{x_1,z}-(F_t\tilde{\pi}_{t-1})^v_{x_1,z}\|&\leq\sum_{j\in I}D_{(1,v),j}b_j\\&\leq\sum_{b'\in V}D_{(1,v),(0,b')}e^{\beta d(v,b')+\beta}e^{-\beta d(v,b')-\beta}\sup_{x_0,x_1\in\mathbb{X}^V}\left\|(\pi_{t-1})^{b'}_{x_0,x_1}-(\tilde{\pi}_{t-1})^{b'}_{x_0,x_1}\right\|\\&\leq2e^{-\beta}\sup_{b'\in V}\left\{e^{-\beta d(v,b')}\sup_{x_0,x_1\in\mathbb{X}^V}\left\|(\pi_{t-1})^{b'}_{x_0,x_1}-(\tilde{\pi}_{t-1})^{b'}_{x_0,x_1}\right\|\right\}.\end{aligned}$$

By joining this with proposition 4.2 in equation 1:

$$\begin{aligned}\|(\pi_t)^v_{x_1,z}-(\tilde{\pi}_t)^v_{x_1,z}\|\leq2e^{-\beta}\sup_{b'\in V}&\left\{e^{-\beta d(v,b')}\sup_{x_0,x_1\in\mathbb{X}^V}\left\|(\pi_{t-1})^{b'}_{x_0,x_1}-(\tilde{\pi}_{t-1})^{b'}_{x_0,x_1}\right\|\right\}\\&+\tau(\beta,\kappa,m,\mathscr{K}).\end{aligned}$$

This result can be iteratively applied. Indeed given that:

$$\left\|(\pi_1)^v_{x_1,z}-(\tilde{\pi}_1)^v_{x_1,z}\right\| \leq 2e^{-\beta}\sup_{b'\in V}\left\{e^{-\beta d(v,b')}\sup_{x_0,x_1\in\mathbb{X}^V}\left\|(\delta_x)^{b'}_{x_0,x_1}-(\delta_x)^{b'}_{x_0,x_1}\right\|\right\}+\tau(\beta,\kappa,m,\mathscr{K})$$
$$=\tau(\beta,\kappa,m,\mathscr{K}),$$

then:

$$\left\|(\pi_t)^v_{x_1,z}-(\tilde{\pi}_t)^v_{x_1,z}\right\| \leq \tau(\beta,\kappa,m,\mathscr{K})\sum_{j=0}^{t-1}(2e^{-\beta})^j \leq \frac{\tau(\beta,\kappa,m,\mathscr{K})e^{\beta}}{e^{\beta}-2},$$

where the last passage follow trivially from the definition of geometric sum and $\beta > \log(2)$. ■

Proposition 4.11. *Fix any collection of observations $\{y_1,\dots,y_T\}$ and any partition $\mathscr{K}$ on the set V. There exists a region $\tilde{\mathscr{R}}_0 \subseteq (0,1)^3$, depending only on $\tilde{\Upsilon},\Upsilon$ and $\Upsilon^{(2)}$, such that if, for given $(\epsilon_-,\epsilon_+,\kappa)\in\tilde{\mathscr{R}}_0$,*

$$\epsilon_- \leq p^v(x^v,z^v) \leq \epsilon_+ \quad \textit{and} \quad \kappa \leq g^f(x^{N(f)},y_t) \leq \frac{1}{\kappa},$$

for all $x,z\in\mathbb{X}^V, f\in F, v\in V, t\in\{1,\dots,T\}$, then for $\beta>\log(2)$ small enough depending only on $\tilde{\Upsilon},\Upsilon,\Upsilon^{(2)}$, $\epsilon_-,\epsilon_+,\kappa$, chosen according to corollary 4.1, for any $t\in\{1,\dots,T\}$ and $m\in\{0,\dots,n\}$:

$$\left\|(\pi_t)^v_{x_1}-(\tilde{\pi}_t)^v_{x_1}\right\| \leq \frac{\tau(\beta,\kappa,m,\mathscr{K})e^{\beta}}{e^{\beta}-2},$$

for all $t\in\{1,\dots,T\},\forall x_1,z\in\mathbb{X}^V$, where $\tilde{\pi}_t$ is the approximated filtering distribution obtained through recursion 3.12,

$$\tau(\beta,\kappa,m,\mathscr{K}) := 2\left(1-\kappa^{a(\mathscr{K})}\right)+4e^{-\beta m}\left(1-\kappa^{b(\mathscr{K},m)}\right),$$

and $a(\mathscr{K})$ and $b(m,\mathscr{K})$ are as in proposition 4.2.

Proof. The proof of this result follows the same procedure of proposition 4.10, the only difference is that $p^v(x^v_1,z^v)$ is missing. ■

4.3.3 Proof of theorem 3.2

Proof. Let $J\subseteq K\in\mathscr{K}$, then by the triangular inequality:

$$\left\|\mathsf{R}_{\tilde{\pi}_t}\tilde{\pi}_{t+1|T}-\mathsf{R}_{\pi_t}\pi_{t+1|T}\right\|_J \leq \left\|\mathsf{R}_{\tilde{\pi}_t}\dots\mathsf{R}_{\tilde{\pi}_{T-1}}\tilde{\pi}_T-\mathsf{R}_{\tilde{\pi}_t}\dots\mathsf{R}_{\tilde{\pi}_{T-1}}\pi_T\right\|_J$$
$$+\sum_{s=0}^{T-t+1}\left\|\mathsf{R}_{\tilde{\pi}_t}\dots\mathsf{R}_{\tilde{\pi}_{t+s-1}}\mathsf{R}_{\tilde{\pi}_{t+s}}\pi_{t+s+1|T}-\mathsf{R}_{\tilde{\pi}_t}\dots\mathsf{R}_{\tilde{\pi}_{t+s-1}}\mathsf{R}_{\pi_{t+s}}\pi_{t+s+1|T}\right\|_J.$$

Given that $\beta,\epsilon_-,\epsilon_+,\kappa$ are chosen such that both corollary 4.1 and proposition 4.7 and proposition 4.9 hold then:

$$\left\|\tilde{\pi}_{t|T}-\pi_{t|T}\right\|_J \leq 2e^{-\beta(T-t)}\sum_{v\in J}\max_{v'\in V}\left\{e^{-\beta d(v,v')}\sup_{x_{T-1},x_T\in\mathbb{X}^V}\left\|\overset{\tilde{\pi}_{T-1}}{\overleftarrow{(\tilde{\pi}_T)}}{}^{v'}_{x_{T-1},x_T}-\overset{\tilde{\pi}_{T-1}}{\overleftarrow{(\pi_T)}}{}^{v'}_{x_{T-1},x_T}\right\|\right\}$$
$$+\sum_{s=0}^{T-t+1}2e^{-\beta s}\sum_{v\in J}\max_{v'\in V}\left\{e^{-\beta d(v,v')}\left\|(\tilde{\pi}_{t+s})^{v'}_{x_{t+s},z}-(\pi_{t+s})^{v'}_{x_{t+s},z}\right\|\right\}.$$

Similarly also proposition 4.10, proposition 4.11 and proposition 4.8 apply:

$$
\begin{aligned}
\left\|\tilde{\pi}_{t|T}-\pi_{t|T}\right\|_J \leq\, & 4\left(\frac{\epsilon_+}{\epsilon_-}\right)^2 e^{-\beta(T-t)} \sum_{v\in J}\max_{v'\in V}\left\{e^{-\beta d(v,v')}\sup_{x_1\in\mathbb{X}^V}\left\|(\pi_T)^{v'}_{x_1}-(\tilde{\pi}_T)^{v'}_{x_1}\right\|\right\} \\
& +4\left(\frac{\epsilon_+}{\epsilon_-}\right)^2 \sum_{s=0}^{T-t-1} e^{-\beta s}\sum_{v\in J}\max_{v'\in V}\left\{e^{-\beta d(v,v')}\frac{\tau(\beta,\kappa,m,\mathscr{K})e^{\beta}}{e^{\beta}-2}\right\} \\
\leq\, & 4\left(\frac{\epsilon_+}{\epsilon_-}\right)^2 e^{-\beta(T-t)} \sum_{v\in J}\max_{v'\in V}\left\{e^{-\beta d(v,v')}\frac{\tau(\beta,\kappa,m,\mathscr{K})e^{\beta}}{e^{\beta}-2}\right\} \\
& +4\left(\frac{\epsilon_+}{\epsilon_-}\right)^2 \sum_{s=0}^{T-t-1} e^{-\beta s}\sum_{v\in J}\max_{v'\in V}\left\{e^{-\beta d(v,v')}\frac{\tau(\beta,\kappa,m,\mathscr{K})e^{\beta}}{e^{\beta}-2}\right\} \\
\leq\, & 4\left(\frac{\epsilon_+}{\epsilon_-}\right)^2 e^{-\beta(T-t)} \frac{\tau(\beta,\kappa,m,\mathscr{K})e^{\beta}}{e^{\beta}-2}\mathbf{card}(J) \\
& +4\left(\frac{\epsilon_+}{\epsilon_-}\right)^2 \frac{e^{\beta}}{e^{\beta}-1}\frac{\tau(\beta,\kappa,m,\mathscr{K})e^{\beta}}{e^{\beta}-2}\mathbf{card}(J) \\
=\, & 4\left(\frac{\epsilon_+}{\epsilon_-}\right)^2 \left[e^{-\beta(T-t)}\frac{e^{\beta}}{e^{\beta}-2}+\frac{e^{\beta}}{e^{\beta}-1}\frac{e^{\beta}}{e^{\beta}-2}\right] \\
& \mathbf{card}(J)\left[2\left(1-\kappa^{a(\mathscr{K})}\right)+4e^{-\beta m}\left(1-\kappa^{b(\mathscr{K},m)}\right)\right] \\
\leq\, & \alpha_2(\beta,\epsilon_-,\epsilon_+)\left(1-\kappa^{a(\mathscr{K})}\right)\mathbf{card}(J)+\gamma_2(\beta,\epsilon_-,\epsilon_+)\left(1-\kappa^{b(\mathscr{K},m)}\right)\mathbf{card}(J)e^{-\beta m},
\end{aligned}
$$

■

Chapter 5

Dynamic Bayesian Neural Networks

This chapter proposes a new hybrid model between an FHMM and a Bayesian neural network called a Hidden Markov neural network (HMNN). Section 5.1 and 5.2 introduces the background and all the main steps to build HMNNs. Section 5.3 compares HMNNs with different baselines on multiple scenarios. As already mentioned, a significant part of this chapter has been submitted to publication and it is available in Rimella and Whiteley [2020].

5.1 Introduction and literature review

Hidden Markov models (HMMs) are an efficient statistical tool to identify patterns in dynamic dataset, with applications ranging from speech recognition [Rabiner and Juang, 1986] to computational biology [Krogh et al., 2001]. Neural networks (NNs) are nowadays some of the most popular models in machine learning and artificial intelligence, and they have shown outstanding performances in several fields. This chapter proposes a novel hybrid model called a *Hidden Markov neural network* (HMNN), which combines a Factorial hidden Markov model [Ghahramani and Jordan, 1997] with a Bayesian neural network.

Intuitively, the aim is to perform Bayesian inference on a time-evolving NN. However, even computing the conditional distribution of the weights given the data of a single NN is a complex task and is generally intractable. Monte Carlo sampling techniques provide a way to approximate an evolving posterior distribution, however, they suffer from the curse of dimensionality [Rebeschini and Van Handel, 2015] and high computational cost [Rimella and Whiteley, 2019]. The rich literature on variational Bayes [Blei et al., 2017] and its success on Bayesian inference for NN [Graves, 2011, Kingma and Welling, 2013, Blundell et al., 2015] have motivated the use of this technique in HMNNs. In particular, the resulting procedure ends up being a sequential counterpart of the algorithm Bayes by Backprop proposed by Blundell et al. [2015]. As for Blundell et al. [2015] a pivotal role is played by the reparameterization trick [Kingma and Welling, 2013], which generates unbiased estimates of the considered gradient.

HMNNs find their place in time series forecasting and, in particular, in the continual learning field. As pointed out by Kurle et al. [2019], the majority of researchers focused on preventing forgetting [Kirkpatrick et al., 2017, Nguyen et al., 2017, Ritter et al., 2018], however sudden changes of the statistics of the data may be an intrinsic property of the generating process itself. In this case, preserving full knowledge is not desirable and the useless information should be forgotten. This operation can be done through the application of a stochastic transition kernel [Kurle et al., 2019], which can be thought of as the Markov transition kernel of an HMM. In this sense, HMNNs embrace the adaption idea of Kurle et al. [2019] and presents it through the well-established HMMs with the further generalization to a bigger class of variational approximations and stochastic transition kernels.

From NNs-HMMs hybrids to continual learning, there are multiple related works. The following paragraphs group similar works together and review them.

Combining NNs and HMMs. Multiple attempts have been accomplished in the literature to combine HMM and NN. In Franzini et al. [1990] an NN is trained to approximate the emission distribution of an HMM. Bengio et al. [1990] and Bengio et al. [1991] preprocess the data with an NN and then use the output as the observed process of a discrete HMM. Krogh and Riis [1999] proposes *Hidden neural networks* where NNs are used to parameterize Class HMM, an HMM with a distribution over classes assigned to each state. In neuroscience, Aitchison et al. [2014] explores the idea of updating measures of uncertainty over the weights in a mathematical model of a neuronal network as part of a "Bayesian Plasticity" hypothesis of how synapses take uncertainty into account during learning. However, they did not focus on artificial neural networks and the computational challenges of using them for data analysis when network weights are statistically modelled as being time-varying.

Bayesian DropConnect & DropOut. DropConnect [Wan et al., 2013] and DropOut [Srivastava et al., 2014] are well-known techniques to prevent NN from overfitting. Kingma et al. [2015] proposes variational DropOut where they combined fully factorized Gaussian variational approximation with the local reparameterization trick to re-interpret DropOut with continuous noise as a variational method. Gal and Ghahramani [2016] extensively treat the connections between DropOut and Gaussian processes, and they show how to train NNs with DropOut (or DropConnect [Mobiny et al., 2019]) through a variational Bayes setting. Variational DropConnect has several common aspects with the cited works, but the whole regularization is induced by the variational approximation's choice and the corresponding reformulation of the reparameterization trick, which is a novel approach.

Bayesian filtering. There are multiple examples of NN training through Bayesian filtering [Puskorius and Feldkamp, 1991, 1994, 2001, Shah et al., 1992, Feldkamp et al., 2003, Ollivier et al., 2018]. In particular, the recent work of Aitchison [2018] proposed AdaBayes and AdaBayes-SS where updates resembling the Kalman filter are employed to model the conditional distribution over a weight of an NN given the data and the states of all the other weights. However, the main difference with HMNN is the dynamical evolution of the underlying NN, indeed Bayesian filtering methods for NN do not consider any change in time.

Continual learning. There are multiple similarities between the proposed work and continual learning methods. This paragraph gives a quick overview of the most popular ones. Elastic Weight Consolidation (EWC) [Kirkpatrick et al., 2017] uses an L2-regularization that guarantees the weights of the NN for the new task being in the proximity of the ones from the old task. Variational continual learning (VCL) [Nguyen et al., 2017] learns a posterior distribution over the weights of an NN by approximating sequentially the true posterior distribution through variational Bayes and by propagating forward the previous variational approximation (this is like setting the transition kernel of the HMNN to a Dirac delta). Online Laplace approximation [Ritter et al., 2018] proposes a recursive update for the parameters of a Gaussian variational approximation which involves the Hessian of the newest negative log-likelihood. None of the cited techniques builds dynamic models, and even if this could be solved by storing the weights at each training step there is no forgetting, meaning that EWC, VCL, Online Laplace focus on overcoming catastrophic forgetting (see section 1.2) and they are not able to avoid outdated information. Lastly, the most similar procedure to HMNNs is the one proposed by Kurle et al. [2019], where the authors perform model adaption with Bayes forgetting through the application of a stochastic kernel. However, HMMs are not even cited and the form of the kernel and variational approximation are straightforward (they do not use mixtures). For these reasons the proposed work differs from Kurle et al. [2019] not only from a presentation point of view but also in terms of generalization properties, in the sense that Kurle et al. [2019] is a simplified HMNN.

5.1.1 Contributions

This chapter develops an evolving-in-time neural network called a Hidden Markov neural network (HMNN), which is able to forget the useless information and adapt to new changes in the data. The main contributions of HMNNS are the following.

- They are FHMMs where the Markov chain models the evolution in time of the weights of a neural network and the emission distribution is given by the output of the neural network. This motivates the forgetting procedure proposed in Kurle et al. [2019] through the well-known HMMs and extends it to any form of transition kernel.

- The algorithm Bayes by Backprop [Blundell et al., 2015] is used sequentially to train HMNNs and the reparameterization trick [Graves, 2011, Kingma and Welling, 2013, Blundell et al., 2015] is reformulated for a class of Gaussian mixtures, which induces a regularization (i.e. penalizing the complexity of the neural network) over the neural network similar to DropConnect, called a Variational DropConnect.
- Experimentally, Variational DropConnect outperforms Bayes by Backprop on MNIST, where the performance is measured in terms of classification accuracy.
- They compare favourably against multiple baselines in conceptual drift applications (i.e. the statistical properties of the data change over time).
- The proposed model can be used to forecast the next frame in a video, and it performs better than competitors like Long short-term memory (LSTM).

5.2 Hidden Markov Neural Networks

A factorial hidden Markov model $(W_t, \mathscr{D}_t)_{t\geq 0}$, as in subsection 2.1.1, with latent state-space $\mathbb{R}^V$, is called Hidden Markov Neural Network (HMNN) if the hidden process $(W_t)_{t\geq 0}$ outlines the evolution over time of the weights of a neural network. Here the finite set V collects the location of each weight, hence $v \in V$ can be thought of as a triplet (l, i, j) saying that the weight W_t^v is a weight of the NN at time t, and precisely related to the connection of the hidden unit i (or input feature i if $l-1=0$) in the layer $l-1$ with the hidden unit j in the layer l (which might be the output layer). Similarly to chapter 2, use the notation: $\lambda_0(\cdot)$ for the probability density function of W_0 (initial density); $p(w_{t-1}, \cdot)$ for the conditional probability density function of W_t given $W_{t-1} = w_{t-1}$ (transition density of the Markov chain); $g(w_t, \mathscr{D}_t)$ for the conditional probability mass or density function of $\mathscr{D}_t$ given $W_t = w_t$ (emission density). Remark that, as in subsection 2.1.1, an HMNN is also an FHMM hence the weights evolve independently from each other:

$$p(w', w) = \prod_{v \in V} p^v((w')^v, w^v), \quad w', w \in \mathbb{R}^V.$$

There is no restriction on the form of the neural network and the data $\mathscr{D}_t$, however, HMNN is presented with feed-forward neural networks and under a supervised learning scenario, where the observed process $\mathscr{D}_t$ is composed by an input x_t and an output y_t, such that the neural network associated to the weights W_t maps the input into a probability density or mass function over the output space, which is done by the emission density $g(w_t, \mathscr{D}_t)$ for $W_t = w_t$. Figure 5.1 shows the evolution over time of an HMNN and the input-output flow when $\mathscr{D}_t$ is composed by an input x_t and an output y_t.

Throughout the chapter, it is assumed for simplicity that the considered probability measures have the Lebesgue measure on $\mathbb{R}^V$ as reference measure. Moreover, the same notation is used for

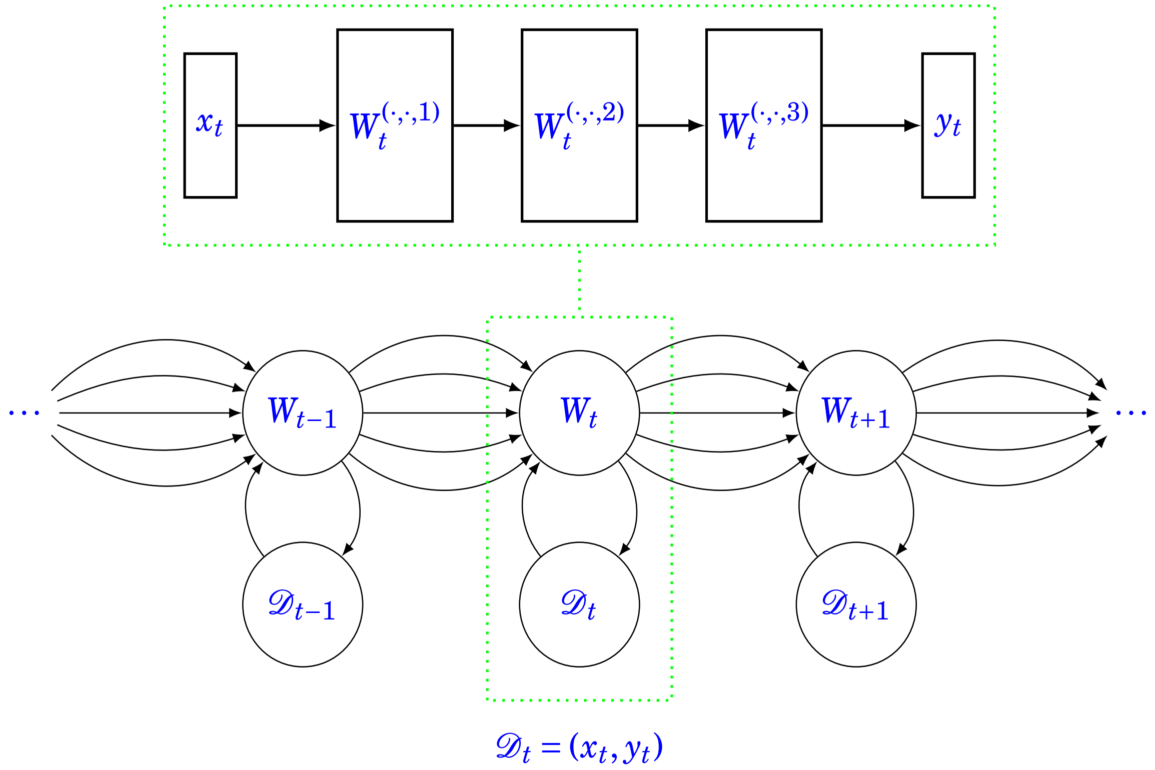


Figure 5.1: On the bottom: diagram explaining the evolution over time of an HMNN. Lines between W_{t-1} and W_t stand for the multidimensional nature of W_t, while lines between W_t and $\mathscr{D}_t$ are used to represent the input-output flow through the NN when $\mathscr{D}_t = (x_t, y_t)$. On the top: the input-output flow when the NN has three layers.

both the density and the corresponding probability measure, for instance, λ_0 refers to both the probability density and the probability measure (the use of one or the other should be clear from the context).

5.2.1 Filtering algorithm for HMNN

As in subsection 2.1.2, denote with π_t the filtering distribution of the HMNN, i.e. the conditional distributions of W_t given the data $\mathscr{D}_1, \ldots, \mathscr{D}_t$. The filtering distribution is computed through recursion (2.4), which is included below for completeness.

$$\pi_0 := \lambda_0, \qquad \pi_t := \mathsf{F}_t \pi_{t-1}, \qquad \mathsf{F}_t := \mathsf{C}_t \mathsf{P}, \qquad t \in \{1, \ldots, T\}, \tag{5.1}$$

where given the probability measure ρ (with density ρ) the "prediction" operator P and the "correction" operator C_t are defined as:

$$\mathsf{P}\rho(A) := \int \mathbb{I}_A(w)p(w',w)\rho(w')dw'dw, \qquad \mathsf{C}_t\rho(A) := \frac{\int \mathbb{I}_A(w)g(w,\mathcal{D}_t)\rho(w)dw}{\int g(w,\mathcal{D}_t)\rho(w)dw}, \qquad A \in \sigma(\mathbb{R}^V).$$

Recursion (5.1) is intractable for any chosen non-linear architecture of the underlying neural network.

The aim is then to build an approximate filtering recursion on the same flavour of the Graph Filter, see subsection 3.3.2. Variational inference can be used to approximate sequentially the target distribution π_t with a variational approximation q_{θ_t} belonging to a pre-specified class of distributions $\mathcal{Q}$. The approximate distribution q_{θ_t} is uniquely identified inside the class $\mathcal{Q}$ by a vector of parameters θ_t, which is picked up by minimizing a Kullback-Leibler (**KL**) divergence criteria:

$$q_{\theta_t} = \arg\min_{q_\theta \in \mathcal{Q}} \mathbf{KL}(q_\theta || \pi_t) = \arg\min_{q_\theta \in \mathcal{Q}} \mathbf{KL}(q_\theta || \mathsf{C}_t\mathsf{P}\pi_{t-1}), \tag{5.2}$$

the above equation is choosing the $q \in \mathcal{Q}$ closest to the filtering distribution π_t.

However, equation (5.2) is still intractable because it requires the filtering distribution at time $t-1$, i.e. π_{t-1}, which is approximated by $q_{\theta_{t-1}}$ and not available in closed form.

Under a proper minimization procedure, $q_{\theta_t} \approx \pi_t$ when π_{t-1} is known, $q_{\theta_{t-1}} \approx \pi_{t-1}$ when π_{t-2} is known, and so on. Given that π_0 is the prior knowledge λ_0 on the weights before training, a q_{θ_1} approximating π_1 using (5.2) can be found and propagated forward by following the previous logic. In this way an HMNN is trained sequentially using (5.2), by substituting the filtering distribution with the latest variational approximation. The approximated filtering recursion is then defined as follows:

$$q_{\theta_0} := \lambda_0, \quad q_{\theta_t} := \mathsf{V}_{\mathcal{Q}}\mathsf{C}_t\mathsf{P}q_{\theta_{t-1}} \quad t \in \{1,\ldots,T\}, \tag{5.3}$$

where the operators P, C_t are as in recursion (5.1) and for a probability distribution ρ the operator $\mathsf{V}_{\mathcal{Q}}$ is defined as follows:

$$\mathsf{V}_{\mathcal{Q}}(\rho) := \arg\min_{q_\theta \in \mathcal{Q}} \mathbf{KL}(q_\theta || \rho),$$

where $\mathcal{Q}$ is a chosen class of probability distributions as explained before. Note that there is the implicit assumption that $\lambda_0 \in \mathcal{Q}$.

Notes on the KL-divergence The Kullback-Leibler divergence can be rewritten as:

$$\mathbf{KL}(q_\theta || \mathsf{C}_t\mathsf{P}\pi_{t-1}) = \text{const.} + \mathbf{KL}(q_\theta || \mathsf{P}\pi_{t-1}) - \mathbb{E}_{q_\theta(w)}[\log(g(w,\mathcal{D}_t))], \tag{5.4}$$

indeed for a general time step t:

$$\begin{aligned}
\mathbf{KL}(q_\theta || \mathsf{C}_t\mathsf{P}\pi_{t-1}) &:= \mathbb{E}_{q_\theta(w)}[\log(q_\theta(w)) - \log(\mathsf{C}_t\mathsf{P}\pi_{t-1}(w))] \\
&= \text{const.} + \mathbb{E}_{q_\theta(w)}[\log(q_\theta(w)) - \log(g(w,\mathcal{D}_t)) - \log(\mathsf{P}\pi_{t-1}(w))] \\
&= \text{const.} + \mathbf{KL}(q_\theta || \mathsf{P}\pi_{t-1}) - \mathbb{E}_{q_\theta(w)}[\log(g(w,\mathcal{D}_t))],
\end{aligned}$$

where:

$$\begin{aligned}\log \mathsf{C}_t \mathsf{P}\pi_{t-1}(w) &= \log\left(\frac{g(w,\mathscr{D}_t)\mathsf{P}\pi_{t-1}(w)}{\int g(w,\mathscr{D}_t)\mathsf{P}\pi_{t-1}(w)dw}\right)\\ &= \log(g(w,\mathscr{D}_t)) + \log(\mathsf{P}\pi_{t-1}(w)) - \log\left(\int g(w,\mathscr{D}_t)\mathsf{P}\pi_{t-1}(w)dw\right).\end{aligned}$$

Note that for recursion 5.3 π_t is substituted by q_{θ_t}.

Alternatively, one could maximize the ELBO:

$$\mathbf{ELBO}(\theta;\mathscr{D}_t) := \mathbb{E}_{q_\theta(w)}[\log(g(w,\mathscr{D}_t))] - \mathbf{KL}(q_\theta||\mathsf{P}\pi_{t-1}).$$

5.2.2 Sequential reparameterization trick

The minimization procedure exploited in recursion (5.3) cannot be solved in a closed form and a suboptimal solution can be found through gradient descent. This requires an estimate of the gradient of $\mathbf{KL}(q_\theta||\mathsf{C}_t\mathsf{P}q_{\theta_{t-1}})$.

As explained in Blundell et al. [2015], derivatives of expectations can be written as expectations of derivatives under some conditions, this is summarized in the following proposition (proposition 1 of Blundell et al. [2015]).

Proposition 5.1. *Let ϵ be a random variable having a probability density given by $q_\theta(w)$ and let $w = h(\theta,\epsilon)$ where $h(\theta,\epsilon)$ is a deterministic function. Suppose further that the marginal probability density of w, $q_\theta(w)$, is such that $q_\theta(w)dw = \nu(\epsilon)d\epsilon$. Then for a function f with derivatives in w:*

$$\frac{\partial \mathbb{E}_{q_\theta}[f(w,\theta)]}{\partial\theta} = \mathbb{E}_\nu\left[\frac{\partial f(w,\theta)}{\partial w}\frac{\partial w}{\partial\theta} + \frac{\partial f(w,\theta)}{\partial\theta}\right].$$

Proof. The proof is provided in Blundell et al. [2015]. ■

Hence if the variational approximation $q_\theta(w)$ is chosen such that it can be rewritten as a probability distribution ν through a deterministic transformation h, i.e. $w = h(\theta,\epsilon)$, then:

$$\begin{aligned}\frac{\partial \mathbf{KL}(q_\theta||\mathsf{C}_t\mathsf{P}q_{\theta_{t-1}})}{\partial\theta} &= \mathbb{E}_\nu\left[\frac{\partial\log(q_\theta(w))}{\partial w}\frac{\partial w}{\partial\theta} + \frac{\partial\log(q_\theta(w))}{\partial\theta} - \frac{\partial\log\left(\mathsf{P}q_{\theta_{t-1}}(w)\right)}{\partial w}\frac{\partial w}{\partial\theta}\right]_{w=h(\theta,\epsilon)}\\ &\quad - \mathbb{E}_\nu\left[\frac{\partial\log(g(w,\mathscr{D}_t))}{\partial w}\frac{\partial w}{\partial\theta}\right]_{w=h(\theta,\epsilon)}\end{aligned} \tag{5.5}$$

where (5.4) is used with $q_{\theta_{t-1}}$ instead of π_{t-1} to propagate forward the approximation. This is a reformulation of the reparameterization trick [Opper and Archambeau, 2009, Kingma and Welling, 2013, Rezende et al., 2014, Blundell et al., 2015], and precisely an application of proposition 5.1 where f is $\log(q_\theta(w)) - \log\left(\mathsf{C}_t\mathsf{P}q_{\theta_{t-1}}(w)\right)$.

Given (5.5) the expectation $\mathbb{E}_\nu$ can be estimated via straightforward Monte Carlo sampling:

$$\begin{aligned}\frac{\partial \mathbf{KL}(q_\theta||\mathsf{C}_t\mathsf{P}q_{\theta_{t-1}})}{\partial\theta} &\approx \frac{1}{N}\sum_{i=1}^{N}\Bigg\{\left[\frac{\partial\log(q_\theta(w))}{\partial w}\frac{\partial w}{\partial\theta} + \frac{\partial\log(q_\theta(w))}{\partial\theta} - \frac{\partial\log\left(\mathsf{P}q_{\theta_{t-1}}(w)\right)}{\partial w}\frac{\partial w}{\partial\theta}\right]_{w=h(\theta,\epsilon)}\\ &\quad - \left[\frac{\partial\log(g(w,\mathscr{D}_t))}{\partial w}\frac{\partial w}{\partial\theta}\right]_{w=h(\theta,\epsilon)}\Bigg\}_{\epsilon=\epsilon^{(i)}}\end{aligned} \tag{5.6}$$

with $\epsilon^{(i)} \sim \nu$. Given the Monte Carlo estimate of the gradient, the parameters θ, related to the variational approximation at time t, can be updated according to any gradient descent technique. Algorithm 6 displays this procedure and, for the sake of simplicity, the algorithm is written with an update that follows a vanilla gradient descent.

Algorithm 6 Approximate filtering recursion

```
Require: $T; \lambda_0(\cdot); (p^v(\cdot,\cdot))_{v \in V}; (\mathscr{D}_t)_{t=1,\dots,T}; (\theta_t^{(0)})_{t=1,\dots,T}; h(\cdot,\cdot); l$
Set: $\tilde{\pi}_0 = \lambda_0$
for $t = 1,\dots,T$ do
    Set the initial condition: $\theta_t = \theta_t^{(0)}$
    repeat
        $\epsilon^{(i)} \sim \nu, \quad i = 1,\dots,N$
        Estimate the gradient $\nabla$ with (5.6) and $\theta = \theta_t$
        Update the parameters: $\theta_t = \theta_t + l\nabla$
    until maximum number of iterations
    Set: $\tilde{\pi}_t = q_{\theta_t}$
return $(\theta_t)_{t=1,\dots,T}$
```

As suggested in the literature [Graves, 2011, Blundell et al., 2015], the cost function in (5.4) is suitable for minibatches optimization. This might be useful when, at each time step, $\mathscr{D}_t$ is made of multiple data and so a full computation of the gradient is computationally prohibitive.

5.2.3 Gaussian case

A fully Gaussian model, i.e. when both the transition kernel and the variational approximation are Gaussian distributions, is not only convenient because the form of $h(\theta,\epsilon)$ is trivial, but also because there exists a closed-form solution for $\mathsf{P}q_{\theta_{t-1}}(w)$. Another appealing aspect of the Gaussian choice is that similar results hold for the scale mixture of Gaussians, which allows us to use a more complex variational approximation and a transition kernel of the same form as the prior in Blundell et al. [2015]. In this chapter we use the notation $\mathscr{N}(\cdot|m,s^2)$ for a Gaussian density with mean m and variance s^2, while when sampling use $\mathrm{N}(m,s^2)$ for the Gaussian distribution with mean m and variance s^2 and $\mathrm{Be}(\gamma)$ for the Bernoulli distribution with parameter γ.

Start by considering the variational approximation. Choose $q_\theta := \bigotimes_{v \in V} q_\theta^v$ where q_θ^v is a mixture of Gaussian with parameters $\theta^v = (m^v, s^v)$ and γ^v hyperparameter. Precisely, for a given weight w^v of the feed-forward neural network:

$$q_\theta^v(w^v) := \gamma^v \mathscr{N}\left(w^v | m^v, (s^v)^2\right) + (1-\gamma^v)\mathscr{N}\left(w^v | 0, (s^v)^2\right), \tag{5.7}$$

where $\gamma^v \in (0,1]$, $m^v \in \mathbb{R}$, $(s^v)^2 \in \mathbb{R}_+$. This technique is called variational DropConnect because it can be interpreted as setting around zero with probability $1-\gamma^v$ the weight in position v of the neural network and so it plays a role of regularization similar to Wan et al. [2013]. Under variational DropConnect the deterministic transformation $h(\theta,\epsilon)$ is still straightforward. Indeed, given that q_θ factorises, then $h(\theta,\epsilon) = (h^v(\theta^v,\epsilon^v))_{v\in V}$ (each w^v depends only on θ^v) and w^v is distributed as (5.7) which is equivalent to consider:

$$w^v = \eta m^v + \xi s^v, \quad \text{with } \eta \sim \text{Be}(\gamma^v), \xi \sim \text{N}(0,1).$$

Hence $h^v(\theta^v,\epsilon^v) = \eta m^v + \xi s^v$, where $\theta^v = (m^v, s^v)$ and $\epsilon^v = (\eta,\xi)$ with η Bernoulli with parameter γ^v and ξ standard Gaussian, meaning that it is enough to sample independently from a Bernoulli distribution and a Gaussian distribution. The collection of hyperparameters $(\gamma^v)_{v\in V}$ represents the variational DropConnect rate per each weight in the NN and generally $\gamma^v = \gamma^{\tilde{v}}$ per each $v,\tilde{v} \in V$. Remark that $(\gamma^v)_{v\in V}$ must be considered as fixed and cannot be learnt during training, because from (5.5) the distribution of ϵ has not to be dependent from the learnable parameters (this can be relaxed with the concrete distribution [Maddison et al., 2016]).

Consider now the transition kernel. It is chosen to be a scale mixture of Gaussians with parameters π,α,σ,c,μ:

$$p(w',w) := \pi\mathcal{N}\left(w|\mu+\alpha(w'-\mu),\sigma^2\mathbf{I}_V\right) + (1-\pi)\mathcal{N}\left(w|\mu+\alpha(w'-\mu),(\sigma^2/c^2)\mathbf{I}_V\right), \tag{5.8}$$

where $\pi \in (0,1)$, $\mu \in \mathbb{R}^V$, $\alpha \in (0,1)$, $\sigma \in \mathbb{R}_+$, $\mathbf{I}_V$ is the identity matrix on $\mathbb{R}^{V,V}$, $c \in \mathbb{R}_+$ and $c > 1$. Intuitively, the transition kernel tells how the weights are expected to be in the next time step given the states of the weights at the current time. It can be interpreted, along with the previous variational approximation, as playing the role of an evolving prior which constraints the new distribution in regions that are determined from the previous training step. The choice of the transition kernel is crucial. A too conservative kernel would constrain too much the training of the weights and the algorithm would not be able to learn patterns in new data. On the contrary, a too flexible kernel could just forget what was learnt before and adapt to the new data only.

The term $\mathsf{P}q_{\theta_{t-1}}(w)$ has a closed form solution when transition kernel and the variational approximation are as in (5.7) and (5.8). Consider a general weight $v \in V$ and call $(\mathsf{P}q_{\theta_{t-1}})^v$ the marginal density of $\mathsf{P}q_{\theta_{t-1}}$ on the component v. If m^v_{t-1}, s^v_{t-1} are the estimates of m^v, s^v at time $t-1$ then:

$$\begin{aligned}(\mathsf{P}q_{\theta_{t-1}})^v(w^v) = {} & \gamma^v\pi\mathcal{N}\left(w^v|\mu^v-\alpha(\mu^v-m^v_{t-1}),\sigma^2+\alpha^2(s^v_{t-1})^2\right)\\ & +(1-\gamma^v)\pi\mathcal{N}\left(w^v|\mu^v-\alpha\mu^v,\sigma^2+\alpha^2(s^v_{t-1})^2\right)\\ & +\gamma^v(1-\pi)\mathcal{N}\left(w^v|\mu^v-\alpha(\mu^v-m^v_{t-1}),\sigma^2/c^2+\alpha^2(s^v_{t-1})^2\right)\\ & +(1-\gamma^v)(1-\pi)\mathcal{N}\left(w^v|\mu^v-\alpha\mu^v,\sigma^2/c^2+\alpha^2(s^v_{t-1})^2\right).\end{aligned} \tag{5.9}$$

The closed form in equation (5.9) is derived from the following integral:

$$
\begin{aligned}
(\mathsf{P}q_{\theta_{t-1}})^v(w^v) &= \int p(\tilde{w}^v, w^v)(q_{\theta_{t-1}})^v(\tilde{w}^v)d\tilde{w}^v \\
&= \gamma^v \pi \int \mathcal{N}\left(w^v|\mu^v + \alpha(\tilde{w}^v - \mu^v), \sigma^2\right)\mathcal{N}\left(\tilde{w}^v|m^v_{t-1}, (s^v_{t-1})^2\right)d\tilde{w}^v \\
&\quad + (1-\gamma^v)\pi \int \mathcal{N}\left(w^v|\mu^v + \alpha(\tilde{w}^v - \mu^v), \sigma^2\right)\mathcal{N}\left(\tilde{w}^v|0, (s^v_{t-1})^2\right)d\tilde{w}^v \\
&\quad + \gamma^v(1-\pi) \int \mathcal{N}\left(w^v|\mu^v + \alpha(\tilde{w}^v - \mu^v), \sigma^2/c^2\right)\mathcal{N}\left(\tilde{w}^v|m^v_{t-1}, (s^v_{t-1})^2\right)d\tilde{w}^v \\
&\quad + (1-\gamma^v)(1-\pi) \int \mathcal{N}\left(w^v|\mu^v + \alpha(\tilde{w}^v - \mu^v), \sigma^2/c^2\right)\mathcal{N}\left(\tilde{w}^v|0, (s^v_{t-1})^2\right)d\tilde{w}^v,
\end{aligned}
\tag{5.10}
$$

the formulation in (5.9) can be achieved by applying the following lemma to each element of the sum in (5.10).

Lemma 5.1. *Consider a Gaussian random variable* $W \sim \mathrm{N}\left(\mu_1, \sigma_1^2\right)$ *and let:*

$$\widetilde{W} = \mu_2 - \alpha(\mu_2 - W) + \sigma_2^2 \xi, \tag{5.11}$$

with $\xi \sim \mathrm{N}(0,1)$*. Then the distribution of* $\widetilde{W}$ *is again Gaussian:*

$$\widetilde{W} \sim \mathrm{N}(\mu_2 - \alpha(\mu_2 - \mu_1), \sigma_2^2 + \alpha^2\sigma_1^2).$$

Proof. Note that the distribution of $\widetilde{W}|W$ is a Gaussian distribution, so the marginal distribution of $\widetilde{W}$ is going to be computed with an integral of the same form of the ones in (5.10). The distribution of $\widetilde{W}$ can be directly computed by noting that $\widetilde{W}$ is a linear combination of two Gaussians and a scalar, and consequently it is Gaussian itself. The lemma is proved by computing the straightforward mean and variance from formulation (5.11). ■

Note that (5.9) is again a scale mixture of Gaussians, where all the variances are influenced by the variances at the previous time step according to α^2. On the one hand, the variational DropConnect rate γ^v tells how to scale the mean of the Gaussians according to the previous estimates m^v_{t-1}. On the other hand, π controls the entity of the jumps by allowing the weights to stay in place with a small variance σ^2/c^2 and permitting big-jumps with σ^2 if necessary. As in Blundell et al. [2015] the variance is not considered in practice because of underflow issues, hence a transformation $\tilde{s}_t$ is used, precisely: $s_t = \log(1+\exp(\tilde{s}_t))$.

5.3 Numerical results

In this section, the performances of HMNNs are going to be tested empirically. The first aspect to test is about the variational approximation form: the aim is to prove that variational DropConnect, i.e. using (5.7) as variational approximation, yields better performance than Bayes by Backprop [Blundell et al., 2015] on MNIST [LeCun et al., 1998]. The second aspect concerns a concept drift scenario, which refers in the machine learning literature to the case where the statistical properties of the target variable change over time. Firstly, the ability of HMNNs to retrieve the evolution of the true parameters is tested in a simple conceptual drift scenario [Kurle et al., 2019]. In particular, this shows that using a more complex variational approximation, compared to the ones in Kurle et al. [2019], does not affect for worse the parameters retrieval procedure. The experiment is then followed by a more complex conceptual drift framework, which is built from MNIST. Under this setting HMNNs are compared with multiple continual learning baselines [Kirkpatrick et al., 2017, Nguyen et al., 2017, Kurle et al., 2019]. The final experiment shows that HMNNs can be also applied in one-step-ahead forecasting for time series, precisely a next frame prediction in the dynamic video texture of a waving flag is considered [Chan and Vasconcelos, 2007, Boots et al., 2008, Basharat and Shah, 2009]. As for Blundell et al. [2015] the studies are focused on simple feed-forward neural networks.

The experiments were run on three different clusters: BlueCrystal Phase 4 (University of Bristol), Cirrus (one of the EPSRC Tier-2 National HPC Facilities) and The Cambridge Service for Data Driven Discovery (CSD3) (University of Cambridge).

Computational cost When running the experiments, the computational cost per time step of an HMNN was comparable with Bayes by Backprop. Indeed, for a fixed time step, training HMNN is equivalent to use Bayes by Backprop with a more complicated prior and variational approximation.

5.3.1 Variational DropConnect

The experiment aims to understand if using a Gaussian mixture as variational approximation can help to improve Bayes by Backprop.

Training is performed on the MNIST [LeCun et al., 1998] dataset, consisting of 60000 images of handwritten digits with size 28 by 28, where each image is preprocessed by dividing each pixel by 126. 50000 images are used as training set and 10000 as the validation set. The test set is composed by 10000 images. MNIST dataset can be downloaded from “http://yann.lecun.com/exdb/mnist/” .

As for Blundell et al. [2015] an ordinary feed-forward neural network without any convolutional layers is used. The architecture is given by: the vectorized image as input, two hidden layers with 400 rectified linear units [Nair and Hinton, 2010, Glorot et al., 2011] and a softmax layer on 10 classes as output. A cross-entropy loss and a fully Gaussian HMNN with a single time step are considered. The Variational Dropconnect technique is applied only to the internal linear layers of the network, i.e. the initial and the final layers are excluded from variational DropConnect as for DropConnect [LeCun et al., 1998]. Observe that a single time step HMNN with $\gamma^v = 1, \alpha = 0$ and $\mu = \mathbf{0}$ (vector of zeros) is equivalent to Bayes by Backprop. For this reason, the proposed implementation of HMNN includes Bayes by Backprop, and both methods can be tested with the same algorithm by simply considering a grid of values of γ^v that includes both $\gamma^v = 1$ and $\gamma^v \neq 1$. Set then $T = 1, \alpha = 0, \mu = \mathbf{0}$ and consider $\gamma^v \in \{0.25, 0.5, 0.75, 1\}$, $\pi \in \{0.25, 0.5, 0.75\}$, $-\log(\sigma) \in \{0, 1, 2\}$, ,$\log(c) \in \{6, 7, 8\}$, learning rate $l \in \{10^{-5}, 10^{-4}, 10^{-3}\}$. Training is performed on about 50 combination of the parameters $(\gamma^v, \pi, \sigma, c)$ and the learning rate, which are randomly extracted from the pre-specified grids. For each value of γ^v we report in Figure 5.2 the performance on the validation set of the three best models.

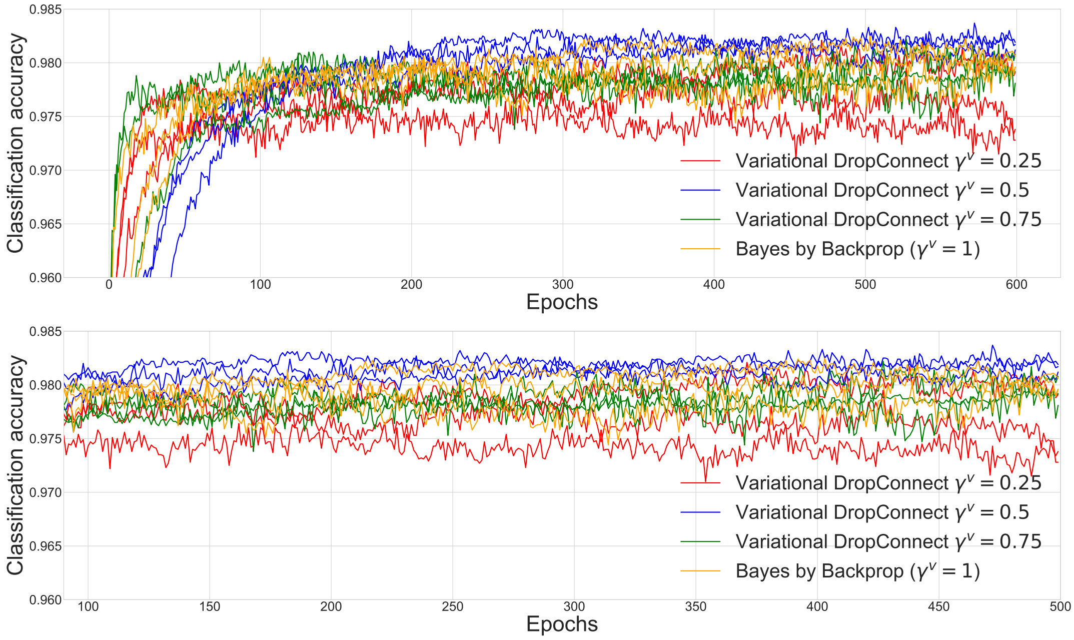


Figure 5.2: Performance of HMNN (and Bayes by Backprop) on the validation set of a Bayes by Backprop with and without variational DropConnect. The plot on the bottom is a zoom-in of the plot on the top.

Model selection is done according to the validation score for each possible value of γ^v. Test performances are reported in Table 5.1. We find out that the smaller the values of γ^v the better the performance.

Table 5.1: Classification accuracy of HMNN (and Bayes by Backprop) on MNIST's test set (the bigger is the better). $\gamma^v = 1$ refers to the case of Bayes by Backprop without variational DropConnect.

Parameter value	**Accuracy**
$\gamma^v = 0.25$	0.9838
$\gamma^v = 0.5$	0.9827
$\gamma^v = 0.75$	0.9825
$\gamma^v = 1$ [Blundell et al., 2015]	0.9814

5.3.2 Concept drift: Logistic regression

In this subsection, HMNN and the adaption with Ornstein-Uhlenbeck process proposed by Kurle et al. [2019] are compared on a simple concept drift scenario. As in Kurle et al. [2019] a 2-dimensional logistic regression problem with evolving weights is considered. Precisely, there are two weights which evolves with the following laws:

$$w_t^{(1)} = 10\sin(\beta t) \quad \text{and} \quad w_t^{(2)} = 10\cos(\beta t),$$

where $\beta = 5deg/sec$ (degrees over seconds) and $t \geq 0$.

Consider then 700 time steps and each time step being composed by 10000 data which are generated in the following manner:

$$x_t^k \sim \mathrm{U}(-3,3), \quad y_t^k \sim \mathrm{Be}(\mathrm{sigmoid}(x_t^k, w_t))$$

where $\mathrm{U}(a,b)$ stands for the Uniform distribution on the interval $[a,b] \subseteq \mathbb{R}$, $w_t = (w_t^{(1)}, w_t^{(2)})$ and $\mathrm{sigmoid}(x_t^k, w_t)$ is the sigmoid function with weights w_t and input x_t^k. The above procedure builds the training set to be used to train HMNN and the method proposed by Kurle et al. [2019]. Note that this is enough for the experiment, indeed the aim of the section is to empirically test how HMNNs do in recovering the oscillating nature of the true weights compared to the method in Kurle et al. [2019]. Studies on the prediction quality of HMNNs for concept drift are left to the next subsection.

Given the training set, a fully Gaussian HMNN is run under multiple combinations of hyperparameters. The experiments showed recovery of the sinusoidal evolution of the weights in each combination. The hyperparameters setting is given by: $T = 700, \alpha = 1, \mu = m_{t-1}$ (to encourage a strong memory of the past), $\gamma^v = 0.75$, $\pi = 0.5$, $-\log(\sigma) = 0$, $-\log(c) = 6$, learning rate $l = 10^{-3}$, the results are reported in figure 5.3, along with the output from Kurle et al. [2019]. Figure 5.3 not only displays that both methods are able to recover the oscillating nature of the ground truth, but also that the mean estimates from HMNNs are generally noisier than the ones from Kurle et al. [2019]. It can be concluded that using a more complicated variational approximation does not affect the recovery of the ground truth, or at least not in the case of a simple conceptual drift.

5.3.3 Concept drift: Evolving classifier on MNIST

In this subsection, HMNNs are compared with continual learning baselines when the data generating distribution is dynamic. This is similar to the previous subsection, indeed the data generating distribution changes over time. However, a simple conceptual drift is not enough, hence a dataset is artificially generated from MNIST with the following procedure.

1. As for subsection 5.3.1 each image is preprocessed by dividing each pixel by 126.
2. Define two labellers: $\mathscr{C}_1$, naming each digit with its label in MNIST; $\mathscr{C}_2$, labelling each digit with its MNIST's label shifted by one unit, i.e. 0 is classified as 1, 1 is classified as 2, ..., 9 is classified as 0.

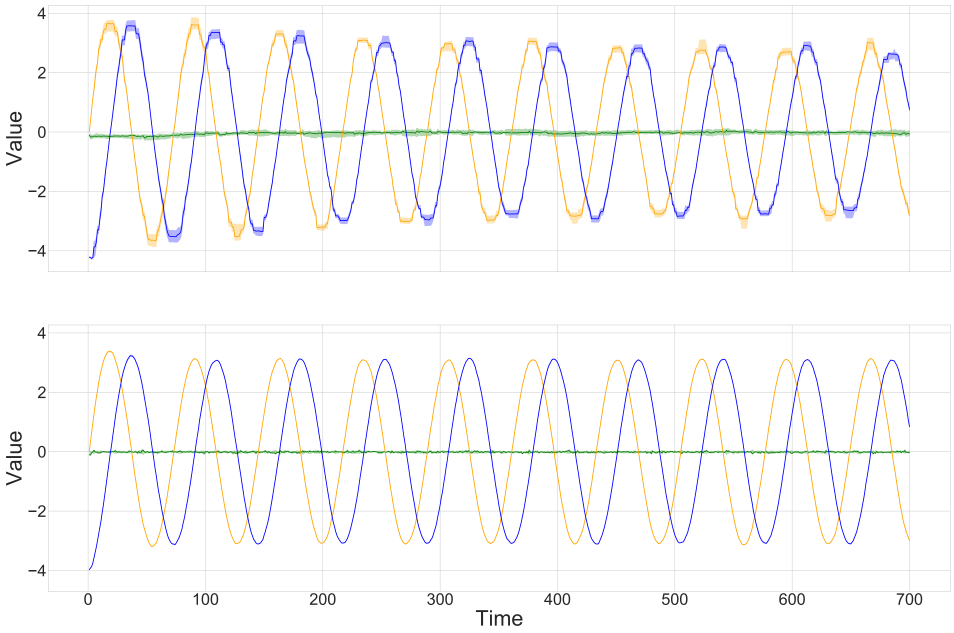


Figure 5.3: Mean of the approximate posterior distributions over 700 steps, credible intervals are built from multiple runs. Orange stands for $w_t^{(1)}$, blue stands for $w_t^{(2)}$ and green is used for the bias. On the top, HMNN. On the bottom, adaption with Ornstein-Uhlenbeck process by Kurle et al. [2019].

3. Consider 19 time steps where each time step t is associated with a probability $f_t \in [0,1]$ and a portion of the MNIST's dataset $\mathscr{D}_t$.

4. At each time step, t randomly label each digit in $\mathscr{D}_t$ with either $\mathscr{C}_1$ or $\mathscr{C}_2$ according to the probabilities $f_t, 1-f_t$.

The resulting $(\mathscr{D}_t)_{t=1,\dots,19}$ is a collection of images where the labels evolve in t by switching randomly from $\mathscr{C}_1$ to $\mathscr{C}_2$ and vice-versa. The above procedure builds both training, validation and test sets. The sample size of each time step is 10000 for training, 5000 for validation and and 5000 for test (the desired sample size is obtained by resampling from MNIST). To validate and test the models the mean classification accuracy over time is considered:

$$\mathscr{A}(\mathscr{D}_{1:T}, \hat{\mathscr{D}}_{1:T}) := \frac{1}{T}\sum_{t=1}^{T}\frac{1}{\mathbf{card}(\mathscr{D}_t)}\sum_{x,y\in\mathscr{D}_t}\mathbb{1}_y(\hat{y}(x)), \tag{5.12}$$

where $\mathscr{D}_t$ is the generated dataset of images x and labels y, $\hat{\mathscr{D}}_t$ is the collection of images x and predictions $\hat{y}(x)$ on the images x using the considered model, $\mathbf{card}(\mathscr{D}_t)$ is the number of elements in $\mathscr{D}_t$, i.e. the total number of labels or images.

In such a scenario, one would ideally want to be able to predict the correct labels by learning sequentially a classifier that is capable of inferring part of the information from the previous time step and forgetting the outdated one. Remark that when $f_t = 0.5$ the best is a classification accuracy of 0.5, because $\mathscr{C}_1$ and $\mathscr{C}_2$ are indistinguishable.

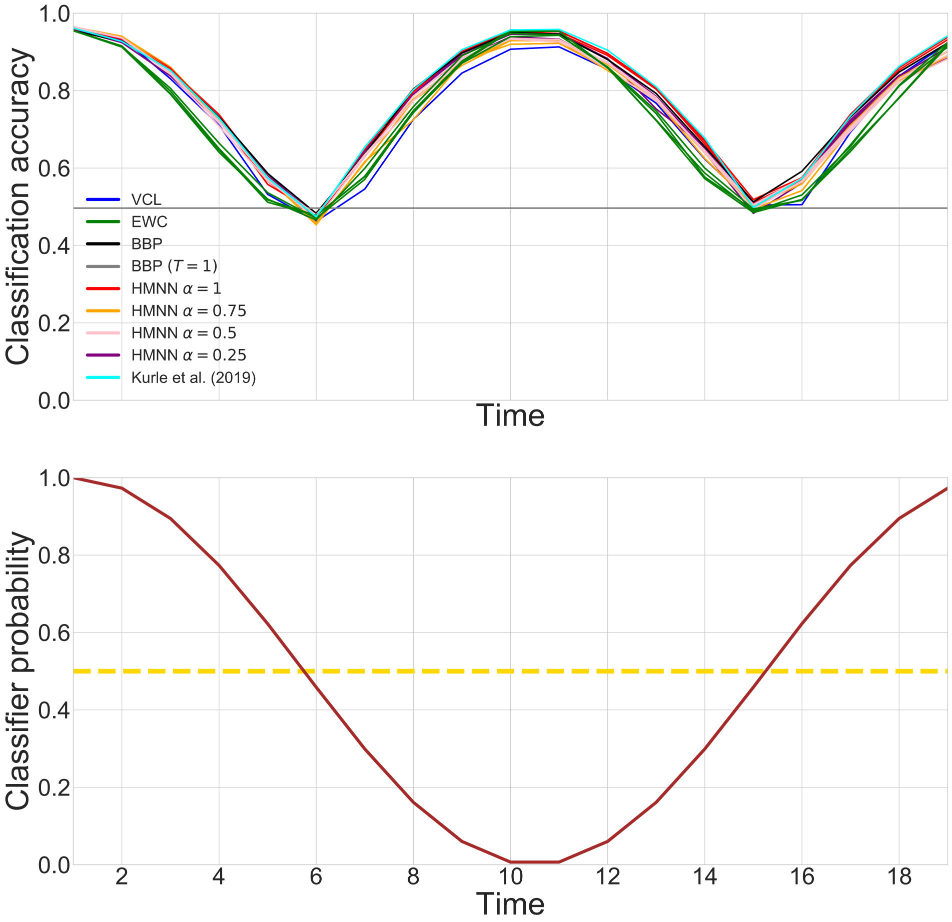


Figure 5.4: On the top, performances on the validation set of time of evolving classifiers obtained with different algorithms. BBP refers to Bayes by Backprop trained sequentially. BBP (T=1) refers to training of Bayes by Backprop on the whole dataset. On the bottom, in brown evolution in time of the probability f_t of choosing the labeller $\mathscr{C}_1$; in yellow the value 0.5.

Consider a fully Gaussian HMNN with $\mu = m_{t-1}$, to encourage a strong memory on the previous posterior distribution. The evolving in time NN is composed by the vectorize image as input, two hidden layers with 100 rectified linear units and a softmax layer on 10 classes as output with a cross-entropy loss. The Variational DropConnect technique is again applied to the internal linear layers of the network only. The parameters $\alpha \in \{0.25, 0.5, 0.75, 1\}$ and $\gamma \in \{0.25, 0.5, 0.75, 1\}$ are selected through the validation set (using the metric (5.12)) while the other parameters are: $\pi = 0.5, -\log(\sigma) = 2, \log(c) = 4, \gamma = 10^{-3}, N = 1$. Other values for $\pi, \sigma, c, \gamma, N$ were tested, but there were no significant changes in the performances. All the possible combinations of α, γ are used and training is pursued for 19 time steps and 100 epochs per each $\mathscr{D}_t$. Remark that a single $\mathscr{D}_t$ is a collection of images and labels, which can be seen as a whole dataset itself.

The method is compared with four algorithms. The architecture of the NN is the same as for the HMNN.

- Sequential Bayes by Backprop. At each time step t training is pursued for 100 epochs on $\mathscr{D}_t$. The parameters are $\pi = 0.5, -\log(\sigma) = 2, \log(c) = 4, \gamma = 10^{-3}, N = 1$. The Bayesian NN at time t is initialize with the previous estimates m_{t-1}, s_{t-1}.
- Bayes by Backprop on the whole dataset. Training is run for 100 epochs on the whole dataset (no time, the sample size is 190000) a Bayesian NN with Bayes by Backprop. The parameters are $\pi = 0.5, -\log(\sigma) = 2, \log(c) = 4, \gamma = 10^{-3}, N = 1$.
- Elastic Weight Consolidation. At each time step t training is run for 100 epochs on $\mathscr{D}_t$. The tuning parameter is chosen from the grid $\{10, 100, 1000, 10000\}$ through the validation set and the metric (5.12). ADAM is used to update the parameters. Remark that this method is not Bayesian, but it is a well-known baseline for continual learning.
- Variational Continual Learning. At each time step t training is pursued for 100 epochs on $\mathscr{D}_t$. The learning rate is extracted from the grid $\{10^{-3}, 10^{-4}, 10^{-5}\}$ through validation and the metric (5.12). The method does not use a coreset, because none of the considered methods is rehearsal.

To test the method the mean over time of the classification accuracy is reported in Table 5.2. For HMNN, Kurle et al. [2019], Bayes by Backprop, EWC and VCL the parameters that perform the best on validation are chosen and the corresponding accuracy is then reported in the table. HMNN, the model by Kurle et al. [2019] and a sequential training of Bayes by Backprop perform the best. It is not surprising that continual learning methods fall behind. Indeed, EWC and VCL are built to preserve knowledge on the previous tasks, which might mix up $\mathscr{C}_1$ and $\mathscr{C}_2$ and confuse the network.

Table 5.2: Classification accuracy for the evolving classifier (the bigger is the better). BBP refers to Bayes by Backprop trained sequentially. BBP (T=1) refers to a training of Bayes by Backprop on the whole dataset.

Model	**Accuracy**
BBP (T=1)	0.503
VCL	0.744
EWC	0.760
BBP	0.780
Kurle et al. Kurle et al. [2019]	0.784
HMNN	0.786

5.3.4 One-step ahead prediction for flag waving

The latest step is testing HMNNs in predicting the frames of a video. The dataset is a sequence of images extracted from a video of a waving flag [Basharat and Shah, 2009, Venkatraman et al., 2015].

The idea is to create an HMNN where the neural network at time t can predict the next frame, i.e. the NN maps frame t to frame $t+1$. To measure the performance the metric suggested in Venkatraman et al. [2015] is used. This is a standardized version of the RMSE on a chosen test trajectory:

$$\mathscr{M}(y_{1:T}, \hat{y}_{1:T}) := \sqrt{\frac{\sum_{t=1}^{T} \|y_t - \hat{y}_t\|_2^2}{\sum_{t=1}^{T} \|y_t\|_2^2}}, \tag{5.13}$$

where $y_{1:T}$ is the ground truth on frames $1, \ldots, T$ and $\hat{y}_{1:T}$ are the predicted frames. Unless specified differently, $\hat{y}_{1:T}$ is a sequence of one step ahead predictions. To have a proper learning procedure multiple frames per time step are need, otherwise, the neural network would just learn the current frame. To overcome this problem a sliding window with 36 frames is used, meaning that at time step t training is pursued on predicting frames $t-35, \ldots, t$ from frames $t-36, \ldots, t-1$, with $t > 36$. The choice of the length for the sliding window was empirical, multiple lengths were considered and 36 was the first one that was not overfitting on the data inside the window (HMNNs along with baselines are all considered in such choice).

The preprocessing phase is similar to MNIST: the video is converted in a sequence of frames in grayscale, in addition, each frame is divided by 126 and converted in a vector form. The dimension is reduced with PCA (130 principal components). A total of 300 frames is used, the first 100 are used for training, validation is pursued on the frames from 100 to 150 and the last 150 frames are used for testing. Note that a single video is available, hence validation and test must be performed online, meaning that validation and test sets are also part of the training, but they are not seen in advance. Precisely, during validation training is pursued on the full path from 1 to 150, then the prediction is made on the frames from 100 to 150 using HMNN from time 99 to 149. The validation score is given by the metric (5.13) on the considered path, which allows model selection. Then training flows on the next frames and the test performance equivalently.

Table 5.3: Metric $\mathscr{M}$ value on the test set (the smaller is the better).

Model	$\mathscr{M}$
Trivial Predictor	0.2162
LSTM	0.2080
DropConnect	0.2063
BBP	0.1932
HMNN	0.1891

Consider a fully Gaussian HMNN with a simple architecture of three layers with 500, 20, 500 rectified linear units, the vectorized previous frame as input and the vectorized current frame as output, and an MSE loss. Apply Variational DropConnect to all the linear layers. The parameters

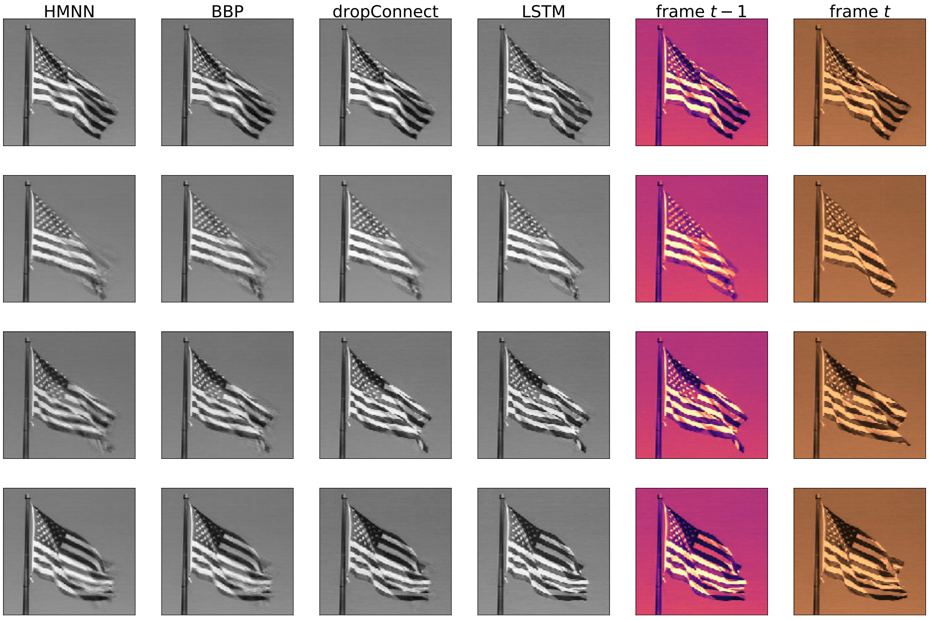


Figure 5.5: The grey columns show the prediction for different algorithms. The orange column shows the target frame for prediction and the purple column shows the frame at the previous time step. The rows display different frames, i.e. different time steps.

$\alpha \in \{0.25, 0.5, 0.75\}$, $\gamma \in \{0.3, 0.8, 1\}$, $\pi = \{0.25, 0.5, 0.75\}$ are chosen according to the validation set while $\mu = m_{t-1}$ to encourage a strong memory of the past. For the other parameters, the setting $-\log(\sigma) = 2, \log(c) = 8, l = 10^{-4}, N = 1$ was found to perform the best. 150 epochs are used on each sliding window.

HMNNs are compared with four models: Bayes by Backprop trained sequentially on the sliding windows, DropConnect trained sequentially on the sliding windows, LSTM trained with the same sliding window size, a trivial predictor that uses the previous frame as forecasting for the current frame. The architectures for sequential Bayes by Backprop and sequential DropConnect are the same as HMNN. For implementation purposes, an architecture of three layers with 500, 500, 500 rectified linear units is used for the LSTM.

- Sequential Bayes by Backprop. At each time step t training is pursued on the current sliding window for 150 epochs. The parameter $\pi = \{0.25, 0.5, 0.75\}$ is chosen with grid search and $-\log(\sigma) = 2, \log(c) = 8, l = 10^{-4}, N = 1$. Other choices of σ, c, l, N do not improve the performance. The Bayesian neural network at time t is initialized with the estimates at time $t-1$.

- Sequential DropConnect. At each time step t training is done on the current sliding window for 150 epochs. The learning rate $l = \{10^{-3}, 10^{-4}, 10^{-5}\}$ is chosen with grid search using the validation score. The neural network at time t is initialized with the estimates at time $t-1$. This is not a Bayesian method.

- LSTM. The window size is set to 36 and the learning rate $l \in \{10^{-3}, 10^{-4}, 10^{-5}\}$ is chosen with grid search using the validation score. This is not a Bayesian method.

- Trivial predictor. Frame t is predicted with frame $t-1$.This trivial baseline is included as an indicator of overfitting on the current window.

Figure 5.5 compares HMNN predictions with the considered baselines: Bayes by Backprop trained sequentially on the sliding windows (column named BBP), DropConnect trained sequentially on the sliding windows (column named DropConnect), LSTM trained with the same sliding window size (column named LSTM), a trivial predictor that uses the previous frame as forecasting for the current frame (column named: frame $t-1$). Notice that LSTM is prone to overfit on the sliding window and so to predict frame t with the latest frame seen without any uncertainty. Similar issues appear in sequential DropConnect. This unwanted behaviour is probably due to the absence of uncertainty quantification and so overconfidence in the considered predictions. HMNN and sequential BBP are less certain about prediction and they create blurred regions where they expect the image to change. This phenomenon is particularly evident in the last row of Figure 5.5. Table 5.3 summarizes the performances using metric (5.13). Overall, HMNN performs better than the baselines and it is directly followed by the sequential BBP.

Chapter 6

Inference in Stochastic Epidemic models via Multinomial Approximations

Section 6.1 introduces the problem, what are the common approaches in the literature to perform inference in compartmental models and the contribution of this thesis in this framework. Section 6.2 sets the mathematical framework. Section 6.3 proposes the multinomial approximation, by showing what are the main step and results to be proved. Section 6.4 concludes the chapter with epidemiological experiments. As already mentioned, a significant part of this chapter has been published in AISTAT2021 and it is available in Whiteley and Rimella [2021].

6.1 Introduction and literature review

Compartmental models are used for predicting the scale and duration of epidemics, estimating epidemiological parameters such as reproduction numbers, and guiding outbreak control measures [Brauer, 2008, O'Neill, 2010, Kucharski et al., 2020]. They are increasingly important because they allow joint modelling of disease dynamics and multimodal data, such as medical test results, cell phone and transport flow data [Rubrichi et al., 2018, Wu et al., 2020], census and demographic information [Prem et al., 2020]. However, statistical inference in stochastic variants of compartmental models is a major computational challenge [Bretó, 2018]. The likelihood function for model parameters is usually intractable because it involves summation over a prohibitively large number of configurations of latent variables representing counts of subpopulations in disease states which cannot be observed directly.

This has lead to the recent development of sophisticated computational methods for approximate inference involving various forms of stochastic simulation [Funk and King, 2020]. Examples include Approximate Bayesian Computation (ABC) [Kypraios et al., 2017, McKinley et al., 2018, Brown et al., 2018, 2016], Data Augmentation Markov Chain Monte Carlo (MCMC) [Lekone and Finkenstädt, 2006], Particle Filters [Murray et al., 2018], Iterated Filtering [Stocks, 2019], and Synthetic Likelihood [Fasiolo et al., 2016]. These methods continue to have a real public health impact, for example, the ABSEIR R package of [Brown et al., 2018] features in the UK COVID-19 surveillance protocols [de Lusignan et al., 2020]. However, the intricacy of these methods, their substantial computational cost arising from the use of stochastic simulation, and their dependence on tuning parameters are obstacles to their wider use and scalability.

6.1.1 Contributions

This chapter introduces a new approach to inference in compartmental epidemic models, through the approximation of the posterior distribution over the compartments with a Multinomial distribution. The main advantages of this method are listed below.

- It applies to a class of finite population, partially observed, discrete-time, stochastic models. In contrast to ODE models, these models can account for statistical variability in disease dynamics.

- It allows approximate evaluation of the likelihood function for model parameters and of filtering /smoothing for compartment occupation numbers, without any stochastic simulation or algorithm tuning parameters, in contrast to state-of-the-art techniques such as ABC.

- It revolves around a computationally simple filtering recursion. The resulting likelihood and smoothing approximations can be combined with e.g., MCMC or Expectation Maximization techniques for parameter estimation.

- It is shown to recover ground truth parameter values from synthetic data, and to compare favourably against Data Augmentation MCMC [Lekone and Finkenstädt, 2006], ABC using the ABSEIR R package [Brown et al., 2018] and ODE [Chowell et al., 2004] alternatives analyzing real Ebola outbreak data under a model from Lekone and Finkenstädt [2006].

- It is used to extend a method of Kucharski et al. [2020] for estimating the time-varying reproduction number of COVID-19 in Wuhan, China, from an ODE compartmental model to a stochastic model.

6.2 Preliminaries

The well-known Susceptible-Exposed-Infective-Recovered (SEIR) model is used as a simple running example. The new methods proposed in this chapter are then applied to more realistic and complex models in section 6.4.

Perhaps the most widely applied formulation of a compartmental model is as a system of ordinary differential equations.

SEIR example. For a population of size n, the SEIR ODE model is:

$$\frac{\mathrm{d}S}{\mathrm{d}t} = -\frac{\beta SI}{n}, \qquad \frac{\mathrm{d}E}{\mathrm{d}t} = \frac{\beta SI}{n} - \rho E, \qquad \frac{\mathrm{d}I}{\mathrm{d}t} = \rho E - \gamma I, \qquad \frac{\mathrm{d}R}{\mathrm{d}t} = \gamma I,$$

initialized with nonnegative integers in each of the compartments (S_0, E_0, I_0, R_0) such that $S_0 + E_0 + I_0 + R_0 = n$.

The most obvious drawback of ODE models is that, once model parameters and the initial population are fixed, any discrepancy between observed data and the solution of the ODE has to be explained as observation error, which is a serious restriction from a modelling point of view. In practice, one can try to estimate unknown parameters and/or the initial condition by numerically minimizing this discrepancy, e.g. under squared error loss. Standard errors for parameter estimates can be derived using asymptotic theory for nonlinear least squares, but the calculation of them in practice involves numerical differentiation of the ODE solution flow w.r.t. parameters [Chowell et al., 2004].

When a probabilistic observation model is specified, Bayesian approaches allow for uncertainty quantification over parameters via posterior distributions, but evaluating the likelihood function for model parameters still involves numerical solution of the ODE.

6.2.1 Difficulties of inference in stochastic compartmental models

In epidemiology, one typically observes times series of count data associated with some subset of the compartments, perhaps subject to random error, or under-reporting. Given such data, when using stochastic models, evaluating the likelihood function of the model parameters and initial condition requires the variables associated with unobserved compartments to be marginalized out. This operation involves summing over all the possible configurations of the unobserved compartments, which depends on the total population size n and the number of compartments, making it unfeasible for models with anything but a small n and a small number of compartments.

SEIR example. The discrete-time stochastic SEIR model is:

$$\begin{aligned} S_{t+1} &= S_t - B_t, \\ E_{t+1} &= E_t + B_t - C_t, \\ I_{t+1} &= I_t + C_t - D_t, \\ R_{t+1} &= R_t + D_t, \end{aligned} \tag{6.1}$$

with conditionally independent, binomially-distributed random variables:

$$\begin{aligned} B_t &\sim \mathrm{Bin}(S_t, 1 - e^{-h\beta I_t/n}) \\ C_t &\sim \mathrm{Bin}(E_t, 1 - e^{-h\rho}), \\ D_t &\sim \mathrm{Bin}(I_t, 1 - e^{-h\gamma}), \end{aligned} \tag{6.2}$$

where $h > 0$ is a time-step size, β, ρ, γ are model parameters, and the process is initialized with nonnegative integers in each of the compartments (S_0, E_0, I_0, R_0) such that $S_0 + E_0 + I_0 + R_0 = n$ and n is the total population size. The interpretation of β is the rate at which an interaction between a susceptible individual and the infective proportion of the population results in the disease being passed to that individual. The mean exposure and infective periods are respectively $1/\rho$ and $1/\gamma$. The sequence $(S_t, E_t, I_t, R_t)_{t \geq 0}$ is a Markov chain.

Stochastic compartmental models also commonly arise in the form of continuous-time pure jump Markov processes, in which transitions of individuals between compartments occur in an asynchronous manner [Bretó, 2018]. Likelihood-based inference for such processes is similarly intractable in general.

6.2.2 Notation

In the remainder of chapter 6, bold upper-case and bold lower-case characters are respectively matrices and column vectors, e.g., $\mathbf{A}$ and $\mathbf{b}$, with generic elements $a^{(i,j)}$ and $b^{(i)}$. The length-m column vector of 1's is denoted $\mathbf{1}_m$. A vector is called a probability vector if its elements are nonnegative and sum to 1. A matrix is called row-stochastic if its elements are nonnegative and its row sums are all 1. The indicator function is denoted $\mathbb{I}[\cdot]$. The element-wise product of matrices is denoted $\mathbf{A} \circ \mathbf{B}$ and the outer product of vectors is denoted $\mathbf{a} \otimes \mathbf{b}$. Element-wise natural logarithm and factorial are denoted $\log \mathbf{A}$ and $\mathbf{A}!$. For positive integers m and n, define $\mathscr{C}_m := \{1, \ldots, m\}$ and $\mathscr{S}_{m,n} := \{\mathbf{x} = [x^{(1)} \cdots x^{(m)}]^{\mathrm{T}} : x^{(i)} \geq 0, i = 1, \ldots, m; \sum_{i=1}^{m} x^{(i)} = n\}$. For $\mathbf{x} \in \mathscr{S}_{m,n}$, define $\boldsymbol{\eta}(\mathbf{x}) := [x^{(1)}/n \cdots x^{(m)}/n]^{\mathrm{T}}$. For a matrix $\mathbf{P}$ (resp. a vector $\boldsymbol{\pi}$) with nonnegative elements summing to 1, $\mathrm{Mult}(n, \mathbf{P})$ (resp. $\mathrm{Mult}(n, \boldsymbol{\pi})$) denotes the distribution of the random matrix (resp. vector) whose elements are the incidence counts obtained from sampling n times with replacement according to $\mathbf{P}$ (resp. $\boldsymbol{\pi}$). This is the usual definition of a multinomial distribution.

This chapter focus on the specific case of distributions over $\mathscr{S}_{m,n}$, such as $\mathrm{Mult}(n, \cdot)$, and probability vectors on $\mathscr{C}_m$. Moreover, when considering a stochastic process $(\mathbf{x}_t)_{t \geq 0}$ the notation $p(\mathbf{x}_t)$ is used for the probability distribution of $\mathbf{x}_t$ and $p(\mathbf{x}_t|\cdot)$ is used for the conditional probability distribution of $\mathbf{x}_t$ given $\cdot$.

6.2.3 Compartmental models as hidden Markov models

Let m be the number of compartments and n be the total population size. Define the location of individual $k \in \{1, \ldots, n\}$ at time $t \geq 0$ as the random variable $\xi_t^{(k)}$, then the population at time $t \geq 0$ is the set of n random variables $\{\xi_t^{(1)}, \ldots, \xi_t^{(n)}\}$, each valued in $\mathscr{C}_m$. Collect the counts of individuals in each of the m compartments at time t in a vector $\mathbf{x}_t = [x_t^{(1)} \cdots x_t^{(m)}]^{\mathrm{T}} \in \mathscr{S}_{m,n}$, where:

$$x_t^{(i)} := \sum_{k=1}^{n} \mathbb{I}[\xi_t^{(k)} = i], \quad i = 1, \ldots, m.$$

For $t \geq 1$ let $\mathbf{Z}_t$ be the $m \times m$ matrix with elements $z_t^{(i,j)}$ counting the individuals transitioning from compartment i at $t-1$ to j at t, hence:

$$z_t^{(i,j)} := \sum_{k=1}^{n} \mathbb{I}[\xi_{t-1}^{(k)} = i, \xi_t^{(k)} = j], \quad i,j = 1,\ldots,m.$$

Let $\boldsymbol{\pi}_0$ be a length-m probability vector and for each $t \geq 1$ let $\boldsymbol{\eta} \mapsto \mathbf{K}_{t,\boldsymbol{\eta}}$ be a mapping from length-m probability vectors to $m \times m$ row-stochastic matrices. Given $\boldsymbol{\pi}_0$ and $\mathbf{K}_{t,\boldsymbol{\eta}}$, the sequence $\{\xi_t^{(1)},\ldots,\xi_t^{(n)}\}_{t\geq 0}$ is constructed to be a Markov chain where $\{\xi_0^{(1)},\ldots,\xi_0^{(n)}\}$ are i.i.d. with:

$$p(\xi_0^{(k)} = j) = \boldsymbol{\pi}_0^{(j)}$$

and $\{\xi_t^{(1)},\ldots,\xi_t^{(n)}\}$ are conditionally independent given $\{\xi_{t-1}^{(1)},\ldots,\xi_{t-1}^{(n)}\}$, with $\xi_t^{(k)}$ drawn from the $\xi_{t-1}^{(k)}$'th row of $\mathbf{K}_{t,\boldsymbol{\eta}(\mathbf{x}_{t-1})}$, i.e.:

$$p(\xi_t^{(k)} = j | \xi_{t-1}^{(1)},\ldots,\xi_{t-1}^{(n)}) = \mathbf{K}_{t,\boldsymbol{\eta}(\mathbf{x}_{t-1})}^{(\xi_{t-1}^{(k)},j)}$$

It follows from this prescription that the sequence of matrices $(\mathbf{Z}_t)_{t\geq 0}$ is also a Markov chain. Indeed, conditional on $\mathbf{Z}_{t-1}$, and hence automatically on $\mathbf{x}_{t-1}$ since $\mathbf{Z}_t\mathbf{1}_m = \mathbf{x}_{t-1}$, the rows of $\mathbf{Z}_t$ are independent, and the distribution of the i-th row of $\mathbf{Z}_t$ is $\text{Mult}(x_{t-1}^{(i)}, \mathbf{K}_{t,\boldsymbol{\eta}(\mathbf{x}_{t-1})}^{(i,\cdot)})$, where $\mathbf{K}_{t,\boldsymbol{\eta}(\mathbf{x}_{t-1})}^{(i,\cdot)}$ is the ith row of $\mathbf{K}_{t,\boldsymbol{\eta}(\mathbf{x}_{t-1})}$. Moreover, noting $\mathbf{1}_m^{\mathrm{T}}\mathbf{Z}_t = \mathbf{x}_t^{\mathrm{T}}$, it can be observed that $(\mathbf{x}_t)_{t\geq 1}$ is also a Markov chain, but an explicit formula for its transition probabilities is not going to be needed.

SEIR example The SEIR model in (6.1)-(6.2) is equivalent to the above described compartmental model with $m = 4$ and

$$(6.3) \qquad \left(\mathbf{K}_{t,\boldsymbol{\eta}}\right)^{(i,j)} = \begin{pmatrix} e^{-h\beta\eta^{(3)}} & 1-e^{-h\beta\eta^{(3)}} & 0 & 0 \\ 0 & e^{-h\rho} & 1-e^{-h\rho} & 0 \\ 0 & 0 & e^{-h\gamma} & 1-e^{-h\gamma} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

for all $t \geq 1$, and identifying $[x_t^{(1)}\, x_t^{(2)}\, x_t^{(3)}\, x_t^{(4)}]^{\mathrm{T}}$ with respectively the counts of susceptible, exposed, infective and recovered individuals at time t. The h in $\mathbf{K}_{t,\boldsymbol{\eta}}$ represents the time resolution. Generally, h is set to 1, but one could even do multistep ahead predictions with lower resolution.

Consider now two observations models. One relates to $(\mathbf{x}_t)_{t\geq 1}$ and the other to $(\mathbf{Z}_t)_{t\geq 1}$. These are going to be the emission distribution of the HMMs.

Observations derived from $(\mathbf{x}_t)_{t\geq 1}$ In this scenario, the observation at time $t \geq 1$ is a length-m vector $\mathbf{y}_t$ with elements $y_t^{(i)}$ which are conditionally independent given $\mathbf{x}_t$, and:

$$(6.4) \qquad y_t^{(i)} \sim \text{Bin}(x_t^{(i)}, q_t^{(i)}).$$

The parameters $q_t^{(i)} \in [0,1]$ are collected in a length-m vector $\mathbf{q}_t$. When conducting likelihood-based inference for $\mathbf{x}_t$ using this model, if $y_t^{(i)}$ is a missing observation, then in the likelihood function associated with (6.4) one should take $y_t^{(i)}$ to be 0, and set $q_t^{(i)} = 0$.

Observations derived from $(\mathbf{Z}_t)_{t\geq 1}$ In this scenario, the observation at time $t \geq 1$ is a $m \times m$ matrix $\mathbf{Y}_t$ with elements $y_t^{(i,j)}$ which are conditionally independent given $\mathbf{Z}_t$, and:

$$y_t^{(i,j)} \sim \mathrm{Bin}(z_t^{(i,j)}, q_t^{(i,j)}). \tag{6.5}$$

The parameters $q_t^{(i,j)} \in [0,1]$ from (6.5) are collected into a $m \times m$ matrix $\mathbf{Q}_t$. Missing data are handled by putting a 0 in place of the missing $y_t^{(i,j)}$ and setting $q_t^{(i,j)} = 0$.

SEIR example In practice, one typically observes, at each time step, counts of *new* infectives rather than the total number of infectives, subject to some random under-reporting or missing data. How can such data be represented in the model? Due to the definition of $\mathbf{Z}_t$, the number of new infectives at time t is exactly $z_t^{(2,3)}$, since the only way an individual can transition to being infective (compartment 3) is by first being exposed (compartment 2). Therefore in this case $y_t^{(2,3)}$ following (6.5) is a count of newly infectives at time t, subject to binomial random under-reporting with parameter $q_t^{(2,3)}$, as required.

It can be concluded that $(\mathbf{x}_t, \mathbf{y}_t)_{t\geq 1}$ and $(\mathbf{Z}_t, \mathbf{Y}_t)_{t\geq 1}$ are HMMs with initial distribution, transition kernel and emission distribution derived from the previous construction. One can then run the forward-filtering and backward-smoothing as in (2.4)-(2.6) to compute the filtering and the smoothing distributions. However, as already mentioned, the cost of this operations is cubic in the cardinality of the state-space, which is $\mathbf{card}(\mathscr{S}_{m,n})$ for the HMM $(\mathbf{x}_t, \mathbf{y}_t)_{t\geq 1}$, and depends on the total population size n.

6.3 Approximate filtering and smoothing

Given the HMMs $(\mathbf{x}_t, \mathbf{y}_t)_{t\geq 1}$ and $(\mathbf{Z}_t, \mathbf{Y}_t)_{t\geq 1}$ defined in subsection 6.2.3, this section proposes methods for approximating: the filtering distributions $p(\mathbf{x}_t|\mathbf{y}_{1:t}), p(\mathbf{Z}_t|\mathbf{Y}_{1:t})$; the marginal likelihoods $p(\mathbf{y}_{1:t}), p(\mathbf{Y}_{1:t})$; the smoothing distributions $p(\mathbf{x}_t|\mathbf{y}_{1:T}), p(\mathbf{Z}_t|\mathbf{Y}_{1:T})$ for a fixed horizon T. Note that the considered setting is taking in to account a discrete state-space, hence the measure theory setting used in chapter 2 is simplified, i.e. the probability measures are probability vectors, transition kernels are stochastic matrices, integrals are sums, and so on.

6.3.1 Multinomial filtering

Start from the HMM $(\mathbf{x}_t, \mathbf{y}_t)_{t\geq 1}$ with emission density as in (6.4), in principle the filtering distributions can be computed through a two-step recursion as explained in (2.4):

$$p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \overset{\text{prediction}}{\longrightarrow} p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) \overset{\text{correction}}{\longrightarrow} p(\mathbf{x}_t|\mathbf{y}_{1:t}).$$

However, the sums in the prediction and correction steps are prohibitively expensive, since they involve summing over all possible values of $\mathbf{x}_{t-1}$ and require computing a probability vector over $\mathscr{S}_{m,n}$.

Approximate prediction step for $\mathbf{x}_t$ For each $\mathbf{x} = [x^{(1)} \cdots x^{(m)}]^{\mathrm{T}} \in \mathscr{S}_{m,n}$ and length-m probability vector $\boldsymbol{\eta}$, let $M_t(\mathbf{x}, \boldsymbol{\eta}, \cdot)$ be the probability mass function on $\mathscr{S}_{m,n}$ of $(\mathbf{1}_m^{\mathrm{T}}\mathbf{Z})^{\mathrm{T}}$, where $\mathbf{Z}$ is a random $m \times m$ matrix whose rows are independent, and whose i-th row has distribution $\mathrm{Mult}(x^{(i)}, \mathbf{K}_{t,\boldsymbol{\eta}}^{(i,\cdot)})$. So by construction $M_t(\mathbf{x}_{t-1}, \boldsymbol{\eta}(\mathbf{x}_{t-1}), \mathbf{x}_t)$ is the transition kernel for the Markov chain $(\mathbf{x}_t)_{t\geq 0}$ defined in subsection 6.2.3. Thus the prediction operation can be written in terms of M_t:

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) &= \sum_{\mathbf{x}_{t-1}\in\mathscr{S}_{m,n}} p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) p(\mathbf{x}_t|\mathbf{x}_{t-1}) \\ &= \sum_{\mathbf{x}_{t-1}\in\mathscr{S}_{m,n}} p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) M_t(\mathbf{x}_{t-1}, \boldsymbol{\eta}(\mathbf{x}_{t-1}), \mathbf{x}_t). \end{aligned} \tag{6.6}$$

It is proposed to approximate $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ with a multinomial distribution and then to propagate forward such approximation to the correction step. Precisely, assume a multinomial distribution approximation to $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ is available, then: replace $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ with this approximation in (6.6) and replace the vector $\boldsymbol{\eta}(\mathbf{x}_{t-1})$ by its expectation under the given multinomial distribution. The outcome of this procedure is a multinomial distribution which approximates $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$. The following lemma formalizes this recipe. The proof is given in the appendix B.

Lemma 6.1. *If for a given length-m probability vector $\boldsymbol{\pi}$, $\mu(\cdot)$ is the probability mass function on $\mathscr{S}_{m,n}$ associated with* $\mathrm{Mult}(n, \boldsymbol{\pi})$ *and* $\mathbb{E}_\mu[\boldsymbol{\eta}(\mathbf{x})]$ *is the expected value of $\boldsymbol{\eta}(\mathbf{x})$ when $\mathbf{x} \sim \mu$, then* $\sum_{\mathbf{x}\in\mathscr{S}_{m,n}} \mu(\mathbf{x}) M_t(\mathbf{x}, \mathbb{E}_\mu[\boldsymbol{\eta}(\mathbf{x})], \cdot)$ *is the probability mass function associated with* $\mathrm{Mult}(n, \boldsymbol{\pi}^{\mathrm{T}}\mathbf{K}_{t,\boldsymbol{\pi}})$.

Approximate correction step for $\mathbf{x}_t$ For the compartmental model described in subsection 6.2.3, the correction operator in (2.4) is:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})}, \quad p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \sum_{\mathbf{x}_t\in\mathscr{S}_{m,n}} p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1}). \tag{6.7}$$

Again the logic is to approximate $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ with a multinomial distribution and then propagate forward to the next prediction step. The core idea is to assume that a multinomial distribution approximation to $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ is available (this is given by the approximate prediction step) and it is going to take the place of $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ in (6.7), resulting in a shifted-multinomial distribution whose mean vector is used to define a multinomial distribution approximation to $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. The following lemma formalizes this recipe. The proof is given in the appendix B.

Lemma 6.2. *Suppose that* $\mathbf{x} \sim \mathrm{Mult}(n, \boldsymbol{\pi})$ *for a given length-m probability vector $\boldsymbol{\pi}$, and assume that given $\mathbf{x}$, $\mathbf{y}$ is a vector with conditionally independent elements distributed: $y^{(i)} \sim$* $\mathrm{Bin}(x^{(i)}, q^{(i)})$. *Then the conditional distribution of* $\mathbf{x}$ *given* $\mathbf{y}$ *is equal to that of* $\mathbf{y} + \mathbf{x}^\star$*, where*

$$\mathbf{x}^\star \sim \mathrm{Mult}\left(n - \mathbf{1}_m^{\mathrm{T}}\mathbf{y}, \frac{\boldsymbol{\pi} \circ (\mathbf{1}_m - \mathbf{q})}{1 - \boldsymbol{\pi}^{\mathrm{T}}\mathbf{q}}\right) \tag{6.8}$$

with $\mathbf{q} = [q^{(1)} \cdots q^{(m)}]^{\mathrm{T}}$*, and the conditional mean of* $\mathbf{x}$ *given* $\mathbf{y}$ *is:*

$$\mathbb{E}[\mathbf{x}|\mathbf{y}] = y + (n - \mathbf{1}_m^{\mathrm{T}}\mathbf{y})\left(\frac{\boldsymbol{\pi} \circ (\mathbf{1}_m - \mathbf{q})}{1 - \boldsymbol{\pi}^{\mathrm{T}}\mathbf{q}}\right). \tag{6.9}$$

Moreover, the marginal distribution of $\mathbf{y}$ *has probability mass function given by:*

$$\begin{aligned}\log p(\mathbf{y}) = &\log(n!) + \mathbf{y}^{\mathrm{T}}(\log\boldsymbol{\pi} + \log\mathbf{q}) - \mathbf{1}_m^{\mathrm{T}}\log(\mathbf{y}!) + (n - \mathbf{1}_m^{\mathrm{T}}\mathbf{y})\log(1 - \boldsymbol{\pi}^{\mathrm{T}}\mathbf{q})\\ &- \log((n - \mathbf{1}_m^{\mathrm{T}}\mathbf{y})!),\end{aligned} \tag{6.10}$$

with the convention $0\log 0 \equiv 0$.

Putting together the results of lemma 6.1 and lemma 6.2 in a recursive fashion leads to algorithm 7; line 3 is motivated by lemma 6.1, line 4 is motivated by (6.8)-(6.9).

Algorithm 7 Multinomial filtering with observations derived from $(\mathbf{x}_t)_{t\geq 1}$

```
1: initialize $\boldsymbol{\pi}_{0|0} \leftarrow \boldsymbol{\pi}_0$
2: for $t \geq 1$ do
3:     $\boldsymbol{\pi}_{t|t-1} \leftarrow (\boldsymbol{\pi}_{t-1|t-1}^{\mathrm{T}}\mathbf{K}_{t,\boldsymbol{\pi}_{t-1|t-1}})^{\mathrm{T}}$
4:     $\boldsymbol{\pi}_{t|t} \leftarrow \frac{\mathbf{y}_t}{n} + \left(1 - \frac{\mathbf{1}_m^{\mathrm{T}}\mathbf{y}_t}{n}\right)\frac{\boldsymbol{\pi}_{t|t-1}\circ(\mathbf{1}_m - \mathbf{q}_t)}{1 - \boldsymbol{\pi}_{t|t-1}^{\mathrm{T}}\mathbf{q}_t}$
5:     $\log w_t \leftarrow \log(n!) + \mathbf{y}_t^{\mathrm{T}}(\log\boldsymbol{\pi}_{t|t-1} + \log\mathbf{q}_t) - \mathbf{1}_m^{\mathrm{T}}\log(\mathbf{y_t}!)$
6:             $+(n - \mathbf{1}_m^{\mathrm{T}}\mathbf{y}_t)\log(1 - \boldsymbol{\pi}_{t|t-1}^{\mathrm{T}}\mathbf{q}_t) - \log((n - \mathbf{1}_m^{\mathrm{T}}\mathbf{y}_t)!)$
```

One may take as output from algorithm 7 the approximations:

$$\begin{aligned}p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) &\approx \mathrm{Mult}(n, \boldsymbol{\pi}_{t|t-1}),\\ p(\mathbf{x}_t|\mathbf{y}_{1:t}) &\overset{d}{\approx} \mathbf{y}_t + \mathbf{x}_t^{\star},\end{aligned}$$

where the $\overset{d}{\approx}$ term indicates approximation of $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ by the distribution of the sum of $\mathbf{y}_t$ (regarded as a constant) and a random variable $\mathbf{x}_t^{\star}$ which is defined to have distribution:

$$\mathbf{x}_t^{\star} \sim \mathrm{Mult}\left(n - \mathbf{1}_m^{\mathrm{T}}\mathbf{y}_t, \frac{\boldsymbol{\pi}_{t|t-1}\circ(\mathbf{1}_m - \mathbf{q}_t)}{1 - \boldsymbol{\pi}_{t|t-1}^{\mathrm{T}}\mathbf{q}_t}\right).$$

In view of (6.10), the quantities w_t computed in algorithm 7 can be used to approximate the marginal likelihood as follows:

$$p(\mathbf{y}_{1:t}) = p(\mathbf{y}_1)\prod_{s=2}^{t} p(\mathbf{y}_s|\mathbf{y}_{1:s-1}) \approx \prod_{s=1}^{t} w_s. \tag{6.11}$$

Turning to the HMM $(\mathbf{Z}_t, \mathbf{Y}_t)_{t\geq 1}$ with emission density as in (6.5), the filtering recursion as in (2.4) is:

$$p(\mathbf{Z}_{t-1}|\mathbf{Y}_{1:t-1}) \overset{\text{prediction}}{\longrightarrow} p(\mathbf{Z}_t|\mathbf{Y}_{1:t-1}) \overset{\text{correction}}{\longrightarrow} p(\mathbf{Z}_t|\mathbf{Y}_{1:t}),$$

and it shows the same computational issues as for $(\mathbf{x}_t)_{t\geq 1}$. To build multinomial approximations for $(\mathbf{Z}_t, \mathbf{Y}_t)_{t\geq 1}$ the procedure is similar to the one above.

Approximate prediction step for $\mathbf{Z}_t$ Given $\mathbf{Z}$ and $\boldsymbol{\eta}$, let $\overline{M}_t(\mathbf{Z},\boldsymbol{\eta},\cdot)$ be the probability mass function of a random $m\times m$ matrix, say $\widetilde{\mathbf{Z}}$, such that $\mathbf{1}_m^{\mathrm{T}}\mathbf{Z}=(\widetilde{\mathbf{Z}}\mathbf{1}_m)^{\mathrm{T}}$ with probability 1, and such that given the row-sums $\widetilde{\mathbf{Z}}\mathbf{1}_m$, the rows of $\widetilde{\mathbf{Z}}$ are independent and the conditional distribution of the ith row is $\mathrm{Mult}(x^{(i)},\mathbf{K}_{t,\boldsymbol{\eta}}^{(i,\cdot)})$ where $\mathbf{x}^{\mathrm{T}}=\mathbf{1}_m^{\mathrm{T}}\mathbf{Z}$. So by construction $\overline{M}_t(\mathbf{Z}_{t-1},\boldsymbol{\eta}(\mathbf{1}_m^{\mathrm{T}}\mathbf{Z}_{t-1}),\mathbf{Z}_t)$ gives the transition probabilities of the Markov chain $(\mathbf{Z}_t)_{t\geq 1}$ defined in subsection 6.2.3. Similarly to the $\mathbf{x}_t$ case, it is proposed to approximate the outcome of the prediction with a multinomial. The following lemma formalizes this recipe. The proof is given in the appendix B.

Lemma 6.3. *If for a given $m\times m$ matrix $\mathbf{P}$, $\overline{\mu}$ is the probability mass function associated with* $\mathrm{Mult}(n,\mathbf{P})$ *and $\mathbb{E}_{\overline{\mu}}[\boldsymbol{\eta}(\mathbf{1}_m^{\mathrm{T}}\mathbf{Z})]$ is the expected value of $\mathbf{1}_m^{\mathrm{T}}\mathbf{Z}$ when $\mathbf{Z}\sim\mathrm{Mult}(n,\mathbf{P})$, then* $\sum_{\mathbf{Z}}\overline{\mu}(\mathbf{Z})\overline{M}_t(\mathbf{Z},\mathbb{E}_{\overline{\mu}}[\boldsymbol{\eta}(\mathbf{1}_m^{\mathrm{T}}\mathbf{Z})],\cdot)$ *is the probability mass function associated with* $\mathrm{Mult}(n,(\boldsymbol{\pi}\otimes\mathbf{1}_m)\circ\mathbf{K}_{t,\boldsymbol{\pi}})$ *where $\boldsymbol{\pi}^{\mathrm{T}}=\mathbf{1}_m^{\mathrm{T}}\mathbf{P}$.*

Approximate correction for $\mathbf{Z}_t$ The correction operation is a Bayes' rule update applied to $p(\mathbf{Z}_t|\mathbf{Y}_{1:t-1})$. Again as for $\mathbf{x}_t$ the correction operation applied to a multinomial generates a shifted multinomial, which is going to be approximated with a multinomial and propagated forward. The following lemma formalizes this recipe. The proof is given in the appendix B.

Lemma 6.4. *Suppose that $\mathbf{Z}\sim\mathrm{Mult}(n,\mathbf{P})$ for a given $m\times m$ matrix $\mathbf{P}$, and given $\mathbf{Z}$, $\mathbf{Y}$ is a matrix with conditionally independent entries distributed: $y^{(i,j)}\sim\mathrm{Bin}(z^{(i,j)},q^{(i,j)})$. Then the conditional distribution of $\mathbf{Z}$ given $\mathbf{Y}$ is equal to that of $\mathbf{Y}+\mathbf{Z}^\star$ where*

$$\mathbf{Z}^\star\sim\mathrm{Mult}\left(n-\mathbf{1}_m^{\mathrm{T}}\mathbf{Y}\mathbf{1}_m,\frac{\mathbf{P}\circ(\mathbf{1}_m\otimes\mathbf{1}_m-\mathbf{Q})}{1-\mathbf{1}_m^{\mathrm{T}}(\mathbf{P}\circ\mathbf{Q})\mathbf{1}_m}\right)$$

and

$$\mathbb{E}[\mathbf{Z}|\mathbf{Y}]=\mathbf{Y}+(n-\mathbf{1}_m^{\mathrm{T}}\mathbf{Y}\mathbf{1}_m)\frac{\mathbf{P}\circ(\mathbf{1}_m\otimes\mathbf{1}_m-\mathbf{Q})}{1-\mathbf{1}_m^{\mathrm{T}}(\mathbf{P}\circ\mathbf{Q})\mathbf{1}_m}.$$

Moreover

$$\begin{aligned}\log p(\mathbf{Y})=&\log(n!)+\mathbf{1}_m^{\mathrm{T}}(\mathbf{Y}\circ\log\mathbf{P})\mathbf{1}_m+\mathbf{1}_m^{\mathrm{T}}(\mathbf{Y}\circ\log\mathbf{Q})\mathbf{1}_m-\mathbf{1}_m^{\mathrm{T}}\log(\mathbf{Y}!)\mathbf{1}_m\\&+(n-\mathbf{1}_m^{\mathrm{T}}\mathbf{Y}\mathbf{1}_m)\log(1-\mathbf{1}_m^{\mathrm{T}}(\mathbf{P}\circ\mathbf{Q})\mathbf{1}_m)-\log((n-\mathbf{1}_m^{\mathrm{T}}\mathbf{Y}\mathbf{1}_m)!).\end{aligned}$$

Algorithm 8 is derived from lemma 6.3 and lemma 6.4. From algorithm 8 one may take the approximations:

$$\begin{aligned}&p(\mathbf{Z}_t|\mathbf{Y}_{1:t-1})\approx\mathrm{Mult}(n,\mathbf{P}_{t|t-1}),\\&p(\mathbf{Z}_t|\mathbf{Y}_{1:t})\overset{d}{\approx}\mathbf{Y}_t+\mathbf{Z}_t^\star,\end{aligned}$$

where

$$\mathbf{Z}_t^\star\sim\mathrm{Mult}\left(n-\mathbf{1}_m^{\mathrm{T}}\mathbf{Y}_t\mathbf{1}_m,\frac{\mathbf{P}_{t|t-1}\circ(\mathbf{1}_m\otimes\mathbf{1}_m-\mathbf{Q}_t)}{1-\mathbf{1}_m^{\mathrm{T}}(\mathbf{P}_{t|t-1}\circ\mathbf{Q}_t)\mathbf{1}_m}\right).$$

The marginal likelihood is approximated using the same formula as in (6.11) but with the w_t's computed as per algorithm 8.

Algorithm 8 Multinomial filtering with observations derived from $(\mathbf{Z}_t)_{t\geq 1}$

```
initialize $\boldsymbol{\pi}_{0|0} \leftarrow \boldsymbol{\pi}_0$
for $t \geq 1$ do
    $\mathbf{P}_{t|t-1} \leftarrow (\boldsymbol{\pi}_{t-1|t-1} \otimes \mathbf{1}_m) \circ \mathbf{K}_{t,\boldsymbol{\pi}_{t-1|t-1}}$
    $\mathbf{P}_{t|t} \leftarrow \frac{\mathbf{Y}_t}{n} + \frac{\mathbf{P}_{t|t-1} \circ (\mathbf{1}_m \otimes \mathbf{1}_m - \mathbf{Q}_t)}{1 - \mathbf{1}_m^{\mathrm{T}}(\mathbf{P}_{t|t-1} \circ \mathbf{Q}_t)\mathbf{1}_m} - \left(\frac{\mathbf{1}_m^{\mathrm{T}}\mathbf{Y}_t\mathbf{1}_m}{n}\right)\frac{\mathbf{P}_{t|t-1} \circ (\mathbf{1}_m \otimes \mathbf{1}_m - \mathbf{Q}_t)}{1 - \mathbf{1}_m^{\mathrm{T}}(\mathbf{P}_{t|t-1} \circ \mathbf{Q}_t)\mathbf{1}_m}$
    $\log w_t \leftarrow \log(n!) + \mathbf{1}_m^{\mathrm{T}}(\mathbf{Y}_t \circ \log \mathbf{P}_{t|t-1})\mathbf{1}_m$
              $+\mathbf{1}_m^{\mathrm{T}}(\mathbf{Y}_t \circ \log \mathbf{Q}_t)\mathbf{1}_m - \mathbf{1}_m^{\mathrm{T}} \log(\mathbf{Y}_t!)\mathbf{1}_m$
              $+(n - \mathbf{1}_m^{\mathrm{T}}\mathbf{Y}_t\mathbf{1}_m)\log(1 - \mathbf{1}_m^{\mathrm{T}}(\mathbf{P}_{t|t-1} \circ \mathbf{Q}_t)\mathbf{1}_m) - \log((n - \mathbf{1}_m^{\mathrm{T}}\mathbf{Y}_t\mathbf{1}_m)!)$
    $\boldsymbol{\pi}_{t|t} \leftarrow (\mathbf{1}_m^{\mathrm{T}}\mathbf{P}_{t|t})^{\mathrm{T}}$
```

6.3.2 Multinomial smoothing

Given a time horizon $T \in \mathbb{N}$ (use the same notation as in the previous chapter, this should not be confused with the transpose symbol T), start again from the HMM $(\mathbf{x}_t, \mathbf{y}_t)_{t=1,\ldots,T}$ with emission density as in (6.4), the smoothing distributions can be computed through the recursion (2.6), which takes the form of an application of a reverse kernel:

$$p(\mathbf{x}_t|\mathbf{y}_{1:T}) \overset{\text{reverse}}{\longleftarrow} p(\mathbf{x}_{t+1}|\mathbf{y}_{1:T}).$$

Precisely, consider the identities:

$$\begin{aligned} p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T}) &= p(\mathbf{x}_T|\mathbf{y}_{1:T}) \prod_{t=0}^{T-1} p(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{y}_{1:t}) \\ p(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{y}_{1:t}) &= \frac{p(\mathbf{x}_t|\mathbf{y}_{1:t})p(\mathbf{x}_{t+1}|\mathbf{x}_t)}{p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t})} \\ p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) &= \sum_{\mathbf{x}_t \in \mathcal{S}_{m,n}} p(\mathbf{x}_t|\mathbf{y}_{1:t})p(\mathbf{x}_{t+1}|\mathbf{x}_t), \end{aligned} \tag{6.14}$$

with the conventions $p(\mathbf{x}_0|\mathbf{x}_1, \mathbf{y}_{1:0}) \equiv p(\mathbf{x}_0|\mathbf{x}_1)$, $p(\mathbf{x}_0|\mathbf{y}_{1:0}) \equiv p(\mathbf{x}_0)$. The smoothing distributions $p(\mathbf{x}_t|\mathbf{y}_{1:T})$, $t = 0, \ldots, T$, satisfy the backward recursion.

$$\sum_{\mathbf{x}_{t+1} \in \mathcal{S}_{m,n}} p(\mathbf{x}_{t+1}|\mathbf{y}_{1:T})p(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{y}_{1:t}) = p(\mathbf{x}_t|\mathbf{y}_{1:T}). \tag{6.15}$$

All these formulae are standard identities for hidden Markov models [Briers et al., 2010], and they are the discrete counterpart of recursion 2.6.

In order to approximate (6.15) each of the terms $p(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{y}_{1:t})$ is approximated. Consider first the numerator in (6.14). Recall from subsection 6.3.1 that the transition probabilities of the $(\mathbf{x}_t)_{t\geq 0}$ process can be written in terms of $(M_t)_{t\geq 1}$, so:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t})p(\mathbf{x}_{t+1}|\mathbf{x}_t) = p(\mathbf{x}_t|\mathbf{y}_{1:t})M_{t+1}(\mathbf{x}_t, \boldsymbol{\eta}(\mathbf{x}_t), \mathbf{x}_{t+1}). \tag{6.16}$$

Replace $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ in (6.16) by its multinomial approximation $\mathrm{Mult}(n, \boldsymbol{\pi}_{t|t})$ obtained using algorithm 7, and replace $\boldsymbol{\eta}(\mathbf{x}_t)$ in (6.16) by its expected value under this multinomial distribution, i.e.

$\boldsymbol{\pi}_{t|t}$, to give

$$p(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{y}_{1:t}) \approx \frac{\mu_{t|t}(\mathbf{x}_t)M_{t+1}(\mathbf{x}_t,\boldsymbol{\pi}_{t|t},\mathbf{x}_{t+1})}{\sum_{\tilde{\mathbf{x}}\in\mathscr{S}_{m,n}}\mu_{t|t}(\tilde{\mathbf{x}})M_{t+1}(\tilde{\mathbf{x}},\boldsymbol{\pi}_{t|t},\mathbf{x}_{t+1})}, \tag{6.17}$$

where $\mu_{t|t}(\cdot)$ is the probability mass function associated with $\mathrm{Mult}(n,\boldsymbol{\pi}_{t|t})$. The following lemma gives a multinomial formulation of (6.17), the proof is available in the appendix B.

Lemma 6.5. *With $\mathbf{x}_{t+1}$ considered fixed, the function which maps $\mathbf{x}_t$ to the right hand side of* (6.17) *is the probability mass function associated with $\mathbf{1}_m^{\mathrm{T}}\widetilde{\mathbf{Z}}$, where $\widetilde{\mathbf{Z}}$ is an $m\times m$ random matrix whose ith row has distribution* $\mathrm{Mult}(x_{t+1}^{(i)},\mathbf{L}_t^{(i,\cdot)})$ *and $\mathbf{L}_t$ is the row-stochastic matrix with entries*

$$l_t^{(i,j)} = \frac{\pi_{t|t}^{(j)}k_{t+1,\boldsymbol{\pi}_{t|t}}^{(j,i)}}{(\boldsymbol{\pi}_{t|t}^{\mathrm{T}}\mathbf{K}_{t+1,\boldsymbol{\pi}_{t|t}})^{(i)}},$$

where $k_{t+1,\boldsymbol{\pi}_{t|t}}^{(i,j)}$ are the elements of $\mathbf{K}_{t+1,\boldsymbol{\pi}_{t|t}}$.

Now considering (6.15) and the approximation (6.17), define the probability mass functions:

$$\mu_{t|T}(\mathbf{x}_t) := \sum_{\mathbf{x}_{t+1}\in\mathscr{S}_{m,n}}\mu_{t+1|T}(\mathbf{x}_{t+1})\frac{\mu_{t|t}(\mathbf{x}_t)M_{t+1}(\mathbf{x}_t,\boldsymbol{\pi}_{t|t},\mathbf{x}_{t+1})}{\sum_{\mathbf{z}\in\mathscr{S}_{m,n}}\mu_{t|t}(\mathbf{z})M_{t+1}(\mathbf{z},\boldsymbol{\pi}_{t|t},\mathbf{x}_{t+1})}, \quad t<T,$$

recalling from (6.17) that $\mu_{T|T}(\cdot)$ is the probability mass function associated with $\mathrm{Mult}(n,\boldsymbol{\pi}_{T|T})$. The following lemma gives a multinomial approximation for the smoothing distribution, the proof is available in the appendix B.

Lemma 6.6. *For $0\leq t\leq T$, $\mu_{t|T}(\cdot)$ is the probability mass function associated with* $\mathrm{Mult}(n,\boldsymbol{\pi}_{t|T})$*, where $\boldsymbol{\pi}_{t|T}$ is computed as per algorithm 9.*

Assuming that algorithm 7 has already been run up to a given time T and using lemma 6.6, algorithm 9 can be derived, and it gives the approximation:

$$p(\mathbf{x}_t|\mathbf{y}_{1:T}) \approx \mathrm{Mult}(n,\boldsymbol{\pi}_{t|T}).$$

Algorithm 9 Multinomial smoothing with observations derived from $(\mathbf{x}_t)_{t\geq 1}$

```
for $t = T-1,\dots,0$ do
    let $\mathbf{L}_t$ be the matrix with elements $l_t^{(i,j)} \leftarrow \pi_{t|t}^{(j)}k_{t+1,\boldsymbol{\pi}_{t|t}}^{(j,i)}/(\boldsymbol{\pi}_{t|t}^{\mathrm{T}}\mathbf{K}_{t+1,\boldsymbol{\pi}_{t|t}})^{(i)}$
    $\boldsymbol{\pi}_{t|T} \leftarrow (\boldsymbol{\pi}_{t+1|T}^{\mathrm{T}}\mathbf{L}_t)^{\mathrm{T}}$
```

Turning to the HMM $(\mathbf{Z}_t,\mathbf{Y}_t)_{t=1,\dots,T}$ with emission density as in (6.5) the smoothing distributions can be computed through the recursion (2.6), which takes the form of an application of a reverse kernel:

$$p(\mathbf{Z}_t|\mathbf{Y}_{1:T}) \xleftarrow{\text{reverse}} p(\mathbf{Z}_{t+1}|\mathbf{Y}_{1:T}).$$

As for the $\mathbf{x}_t$ case, to derive our approximations start from the fact that $(\mathbf{Z}_t, \mathbf{Y}_t)_{t\geq 1}$ is a hidden Markov model, and consider the identities:

$$\begin{aligned}
p(\mathbf{Z}_{1:T}|\mathbf{Y}_{1:T}) &= p(\mathbf{Z}_T|\mathbf{Y}_{1:T}) \prod_{t=1}^{T-1} p(\mathbf{Z}_t|\mathbf{Z}_{t+1}, \mathbf{Y}_{1:t}) \\
p(\mathbf{Z}_t|\mathbf{Z}_{t+1}, \mathbf{Y}_{1:t}) &= \frac{p(\mathbf{Z}_t|\mathbf{Y}_{1:t}) p(\mathbf{Z}_{t+1}|\mathbf{Z}_t)}{p(\mathbf{Z}_{t+1}|\mathbf{Y}_{1:t})} \qquad (6.18) \\
p(\mathbf{Z}_{t+1}|\mathbf{Y}_{1:t}) &= \sum_{\mathbf{Z}_t} p(\mathbf{Z}_t|\mathbf{Y}_{1:t}) p(\mathbf{Z}_{t+1}|\mathbf{Z}_t).
\end{aligned}$$

The backward recursion is in this case is:

$$\sum_{\mathbf{Z}_{t+1}} p(\mathbf{Z}_{t+1}|\mathbf{Y}_{1:T}) p(\mathbf{Z}_t|\mathbf{Z}_{t+1}, \mathbf{Y}_{1:t}) = p(\mathbf{Z}_t|\mathbf{Y}_{1:T}). \qquad (6.19)$$

Writing $\overline{\mu}_{t|t}(\cdot)$ for the probability mass function associated with $\mathrm{Mult}(n, \mathbf{P}_{t|t})$, our approximation to (6.18) is:

$$p(\mathbf{Z}_t|\mathbf{Z}_{t+1}, \mathbf{Y}_{1:t}) \approx \frac{\overline{\mu}_{t|t}(\mathbf{Z}_t)\overline{M}_{t+1}(\mathbf{Z}_t, \boldsymbol{\pi}_{t|t}, \mathbf{Z}_{t+1})}{\sum_{\widetilde{\mathbf{Z}}} \overline{\mu}_{t|t}(\widetilde{\mathbf{Z}})\overline{M}_{t+1}(\widetilde{\mathbf{Z}}, \boldsymbol{\pi}_{t|t}, \mathbf{Z}_{t+1})}. \qquad (6.20)$$

where $\overline{M}_{t+1}$ was introduced in subsection 6.3.1 when building the approximate filtering for $\mathbf{Z}_t$ and in the setting of (6.20) has the explicit formula:

$$\overline{M}_{t+1}(\mathbf{Z}_t, \boldsymbol{\pi}_{t|t}, \mathbf{Z}_{t+1}) = \mathbb{I}[\mathbf{1}_m^{\mathrm{T}}\mathbf{Z}_t = (\mathbf{Z}_{t+1}\mathbf{1}_m)^{\mathrm{T}}] \left(\prod_{j=1}^{m} (\mathbf{Z}_{t+1}\mathbf{1}_m)^{(j)}! \prod_{\ell=1}^{m} \frac{\left(k_{t+1,\boldsymbol{\pi}_{t|t}}^{(j,\ell)}\right)^{z_{t+1}^{(j,\ell)}}}{z_{t+1}^{(j,\ell)}!} \right).$$

The following lemma gives a formulation for (6.20), and it is proved in appendix B.

Lemma 6.7. *With $\mathbf{Z}_{t+1}$ considered fixed, the function which maps $\mathbf{Z}_t$ to the right hand side of* (6.20) *is the probability mass function such that the columns of $\mathbf{Z}_t$ are independent and the distribution of the ith column is* $\mathrm{Mult}((\mathbf{Z}_{t+1}\mathbf{1}_m)^{(i)}, \overline{\mathbf{L}}_t^{(i,\cdot)})$*, where $\overline{\mathbf{L}}_t$ is the row-stochastic matrix with entries*

$$\overline{l}_s^{(i,j)} = p_{t|t}^{(j,i)} / \pi_{t|t}^{(i)}$$

where $p_{t|t}^{(i,j)}$ are the elements of $\mathbf{P}_{t|t}$, and $\pi_{t|t}^{(i)}$ are the elements of $\boldsymbol{\pi}_{t|t} := (\mathbf{1}_m^{\mathrm{T}}\mathbf{P}_{t|t})^{\mathrm{T}}$ with $\mathbf{P}_{t|t}$ computed in algorithm 8.

Now considering (6.19) and the approximation (6.20), define the probability mass functions:

$$\overline{\mu}_{t|T}(\mathbf{Z}_t) := \sum_{\mathbf{Z}_{t+1}} \overline{\mu}_{t+1|T}(\mathbf{Z}_{t+1}) \frac{\overline{\mu}_{t|t}(\mathbf{Z}_t)\overline{M}_{t+1}(\mathbf{Z}_t, \boldsymbol{\pi}_{t|t}, \mathbf{Z}_{t+1})}{\sum_{\widetilde{\mathbf{Z}}} \overline{\mu}_{t|t}(\widetilde{\mathbf{Z}})\overline{M}_{t+1}(\widetilde{\mathbf{Z}}, \boldsymbol{\pi}_{t|t}, \mathbf{Z}_{t+1})}, \quad t < T, \qquad (6.21)$$

where $\overline{\mu}_{T|T}(\cdot)$ is defined to be the probability mass function associated with $\mathrm{Mult}(n, \mathbf{P}_{T|T})$.

The following lemma is the key result to build the approximate smoothing, and it is proved in the appendix B.

Lemma 6.8. *For $0 \le t \le T$, $\overline{\mu}_{t|T}(\cdot)$ is the probability mass function associated with* $\mathrm{Mult}(n, \mathbf{P}_{t|T})$*, where $\mathbf{P}_{t|T}$ is computed as per algorithm 10.*

Assuming algorithm 8 has already been run up to a given time T, the above lemma is used to derive algorithm 10, from which the following approximation is obtained:

$$p(\mathbf{Z}_t|\mathbf{y}_{1:T}) \approx \mathrm{Mult}(n, \mathbf{P}_{t|T}).$$

Algorithm 10 Multinomial smoothing with observations derived from $(\mathbf{Z}_t)_{t\ge1}$

```
for $t = T-1, \ldots, 1$ do
    $\boldsymbol{\pi}_{t|T} \leftarrow \mathbf{P}_{t+1|T}\mathbf{1}_m$
    let $\overline{\mathbf{L}}_t$ be the matrix with elements $\overline{l}_t^{(i,j)} \leftarrow p_{t|t}^{(j,i)}/\pi_{t|t}^{(i)}$
    $\mathbf{P}_{t|T} \leftarrow (\mathbf{1}_m \otimes \boldsymbol{\pi}_{t|T}) \circ \overline{\mathbf{L}}_t^{\mathrm{T}}$
```

In algorithm 10, $p_{t|t}^{(i,j)}$ are the elements of $\mathbf{P}_{t|t}$ and $\pi_{t|t}^{(i)}$ are the elements of $\boldsymbol{\pi}_{t|t} := (\mathbf{1}_m^{\mathrm{T}}\mathbf{P}_{t|t})^{\mathrm{T}}$, with $\mathbf{P}_{t|t}$ computed in algorithm 8.

6.3.3 Computational cost

The computational cost of algorithms 7 and 8 is independent of the overall population size n, except through factorial terms such as $\log(n!)$ and $\log((n-\mathbf{1}_m^{\mathrm{T}}\mathbf{y})!)$. However these terms do not depend on the model parameters $\mathbf{K}_{t,\boldsymbol{\eta}}$, $\mathbf{q}_t$ etc., so can be pre-computed or even not computed at all if the approximate marginal likelihood needs to be evaluated only up to a constant of proportionality independent of model parameters. Leaving these factorial terms out the worst-case costs of algorithms 7 and 8 are therefore respectively $\mathcal{O}(Tm^2)$ and $\mathcal{O}(Tm^3)$. Costs may be substantially lower in practice as $\mathbf{K}_{t,\boldsymbol{\eta}}$ and $\mathbf{q}_t$ are typically sparse. Similar observations hold for the smoothing algorithms.

This compares to $O(Tmf(n))$ to simulate $(\mathbf{x}_t)_{t\ge0}$ from the model where $f(n)$ is the complexity of sampling from $\mathrm{Bin}(n,p)$, assuming no more than two non-zero entries in each row of $\mathbf{K}_{t,\boldsymbol{\eta}}$. A larger number of non-zero entries would imply a higher cost. Such a simulation is necessary (but usually not sufficient) to approximately evaluate the likelihood in ABSEIR [Brown et al., 2018]. The worst case is $f(n) = O(n)$, but modest improvements are available if one accepts 'with high probability' performance measures [Farach-Colton and Tsai, 2015]. The worst-case time complexity of the Data Augmentation MCMC method [Lekone and Finkenstädt, 2006] is also linear in n. Whilst the wall-clock time of any given algorithm is of course heavily dependent on exactly how it is implemented, these considerations suggest that the proposed methods will have attractive computational costs in many applications, where m is often many orders of magnitude smaller than n

6.4 Numerical results

This section analyses the numerical results obtained with the multinomial approximations. The experiments treat different aspects of the approximation: retrieve the ground truth parameters from a synthetic data scenario; estimate the effect of control measures in the 1995 Ebola outbreak in the Democratic Republic of Congo; assess the accuracy of the multinomial approximation for the filtering distribution; estimate the reproduction number of COVID-19 during its initial stage in Wuhan.

6.4.1 The 1995 Ebola outbreak in the Democratic Republic of Congo

Simulated and real data on the 1995 outbreak of Ebola in the Democratic Republic of Congo under a discrete-time SEIR model used by [Lekone and Finkenstädt, 2006] are analysed to investigate the impact of control interventions. The experiments follow closely those in [Lekone and Finkenstädt, 2006] to allow comparisons with their Data Augmentation MCMC method. Comparisons to the ABC method from the ABSEIR R package [Brown et al., 2018], and results of least-squares fitting of an ODE model from [Chowell et al., 2004], which [Lekone and Finkenstädt, 2006] used as a benchmark, are included in the analysis.

Model The model of [Lekone and Finkenstädt, 2006] is the same as the SEIR model in (6.1) with $h = 1$, except that β is replaced by a time-varying parameter:

$$\beta_t = \begin{cases} \beta & t < t_\star \\ \beta e^{-\lambda(t-t_\star)} & t \geq t_\star \end{cases},$$

where $t_\star$ is the time at which control measures began. Thus $\mathbf{K}_{t,\boldsymbol{\eta}}$ is as in (6.3) but with β replaced by this β_t. Also following [Lekone and Finkenstädt, 2006], the data consist of daily counts of new cases (i.e. new infectives) and new deaths (i.e. new removals). In [Lekone and Finkenstädt, 2006] it was assumed these counts are observed directly, subject to known proportions of missing data. In this thesis, a slightly more general observation model is chosen. As per subsection 6.2.3 the observation model for $\mathbf{Z}_t$ is considered with $q_t^{(i,j)} = 0$ for all (i,j) except $(2,3)$ and $(3,4)$, and where $q_t^{(2,3)}$ and $q_t^{(3,4)}$ are treated as constant-in-t but otherwise unknown and to be estimated. Thus the parameters of the model to be estimated are:

$$\Theta = (\beta, \lambda, \rho, \gamma, q^{(2,3)}, q^{(3,4)}).$$

Note that the initial distribution is given and set as in Lekone and Finkenstädt [2006] for both the synthetic and real scenario.

Synthetic data Using the following settings from [Lekone and Finkenstädt, 2006]:

$$\beta = 0.2, \quad \lambda = 0.2, \quad \rho = 0.2, \quad \gamma = 0.143, \quad t_\star = 130,$$

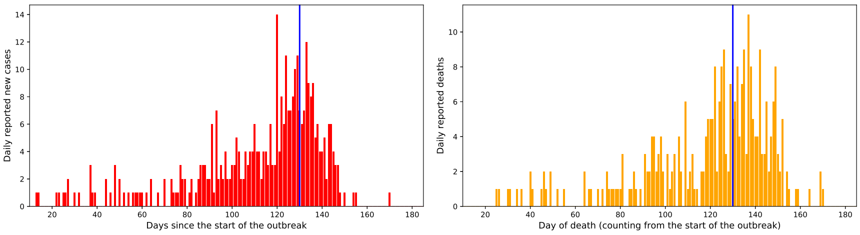


Figure 6.1: Synthetic data: in red $(y_t^{(2,3)})_{t\geq 1}$, which are daily numbers of reported new cases, and in orange $(y_t^{(3,4)})_{t\geq 1}$, which are the daily numbers of reported new deaths, simulated from the Ebola model of Lekone and Finkenstädt [2006]. Blue lines indicate the day $t_\star = 130$ at which control measures were introduced.

and $S_0 = 5{,}364{,}500$, $E_0 = 1$, $I_0 = R_0 = 0$, plus $q_t^{(2,3)} = 291/316$ and $q_t^{(3,4)} = 236/316$ for all $t \geq 1$ informed by realistic proportions of non-missing data [Lekone and Finkenstädt, 2006], a SEIR model is simulated until extinction, which took 175 time steps. The data are shown in figure 6.1.

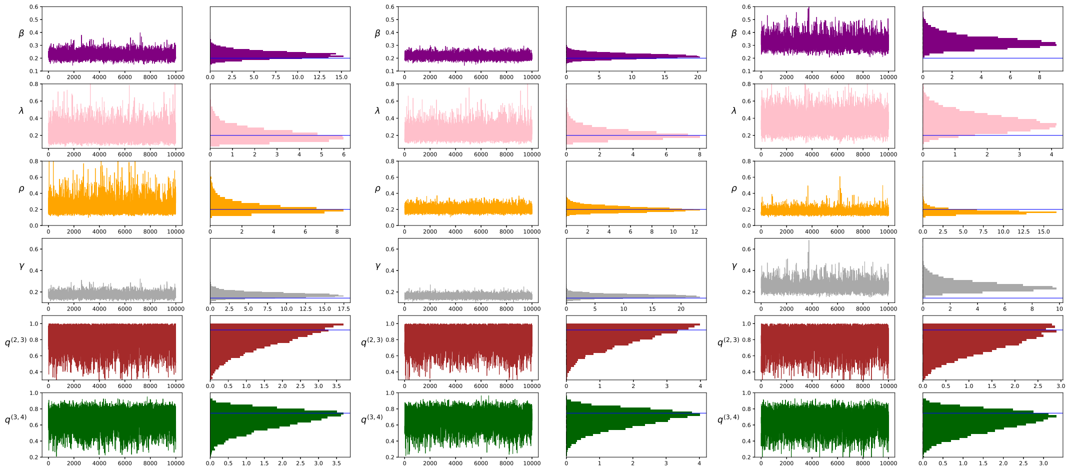


Figure 6.2: Traceplots and histograms for the MCMC method applied to synthetic data from the Ebola model, with the 'vague' set of priors (left), the 'informative' set of priors (centre) and the 'noncentered' (right) set of priors specified by Lekone and Finkenstädt [2006]. Blue lines show true parameter values.

The parameters are estimated either through MLE's from an EM algorithm that uses the approximate filtering and smoothing methods, or through marginal posterior means and standard deviations, estimated using a Metropolis-within-Gibbs MCMC algorithm which incorporates the approximate marginal likelihood from the multinomial approximation. Figure 6.2 shows traceplots and histograms of the MCMC output from which posterior means and posterior standard deviations are calculated. The MCMC chain is run for 5×10^5 iterations, the first 10^5 iterations are discarded for burn-in, and the remaining samples thinned to result in a sample size of 10^4. Table 6.1 shows posterior means and standard deviations under three sets of prior distributions over $(\beta, \lambda, \rho, \gamma)$ labelled 'vague', 'informative' and 'noncentered' by [Lekone and Finkenstädt, 2006]. The basic reproduction number is $R_0 = \beta/\gamma$. The results show an accurate recovery of the true parameter values. More information about the MCMC and the EM algorithm are available in appendix B.

Real data This paragraph analyses the same real Congo Ebola data as in [Lekone and Finkenstädt, 2006]. As for the previous paragraph, the parameters are estimated either through MLE's from an EM algorithm that uses the approximate filtering-smoothing method or through marginal posterior means and standard deviations estimated using a Metropolis-within-Gibbs MCMC algorithm which incorporates the approximate marginal likelihood from the multinomial approximation. Traceplots and histograms for the MCMC method are displayed in figure 6.4, the 'vague' and 'uninformative' sets of priors are considered only.

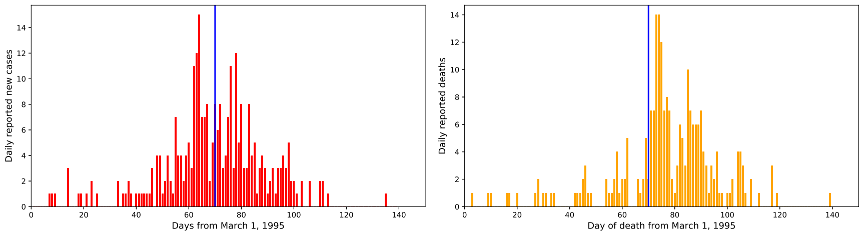


Figure 6.3: Data from the 1995 Ebola outbreak in the Democratic Republic of Congo per day from March 1, 1995, to July 16, 1995. In red daily numbers of reported new cases and in orange daily number of reported new deaths. Blue lines indicate the 9th of May when control measurements were introduced.

Table 6.2 summarises the results of the multinomial approximation and the considered baselines. There are several interesting findings. 1) The results from the proposed methods are generally closer to those from the Data Augmentation MCMC sampler of [Lekone and Finkenstädt, 2006] than those from the ABC method of [Brown et al., 2018]; the former targets the true posterior distribution whilst the latter does so only approximately. 2) Under the 'vague'

Table 6.1: Parameter estimates for synthetic data under the Ebola model using our EM and MCMC methods under three sets of prior distributions specified by [Lekone and Finkenstädt, 2006]. For the MCMC results, the posterior means is reported as the point estimate and the numbers in parentheses are posterior standard deviations.

Parameter	β	λ	ρ	γ	$\mathbf{q^{(2,3)}}$	$\mathbf{q^{(3,4)}}$	$\mathbf{R_0}$
True value	0.2	0.2	0.2	0.143	0.92	0.75	1.40
MLE (EM-alg.)	0.20	0.18	0.21	0.139	1.00	0.81	1.44
MCMC (vague)	0.23 (0.028)	0.21 (0.080)	0.22 (0.076)	0.173 (0.024)	0.81 (0.140)	0.66 (0.119)	1.31 (0.088)
MCMC (infor.)	0.22 (0.020)	0.22 (0.065)	0.20 (0.035)	0.162 (0.017)	0.83 (0.130)	0.67 (0.112)	1.34 (0.082)
MCMC (noncent.)	0.32 (0.048)	0.35 (0.101)	0.17 (0.031)	0.256 (0.049)	0.79 (0.147)	0.64 (0.125)	1.28 (0.084)

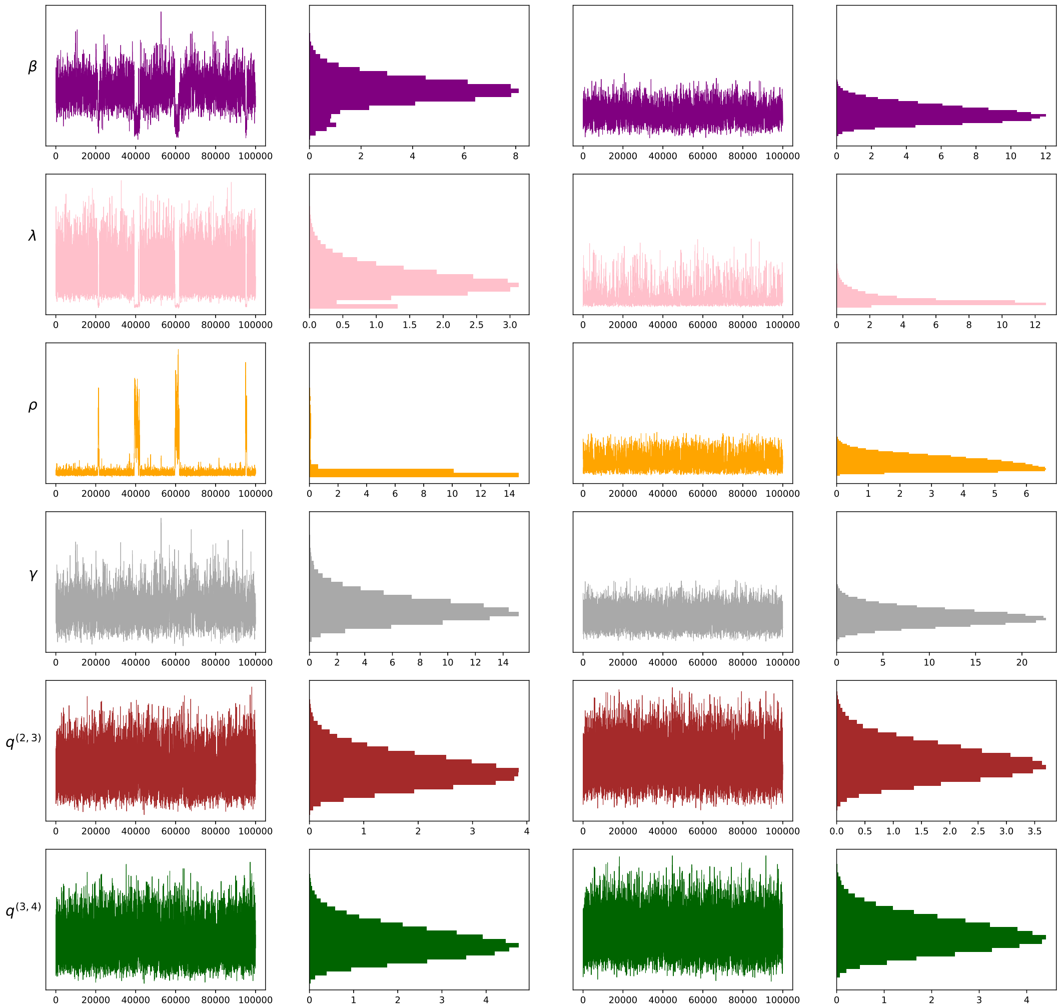


Figure 6.4: Traceplots and histogram for the MCMC on the Ebola data.

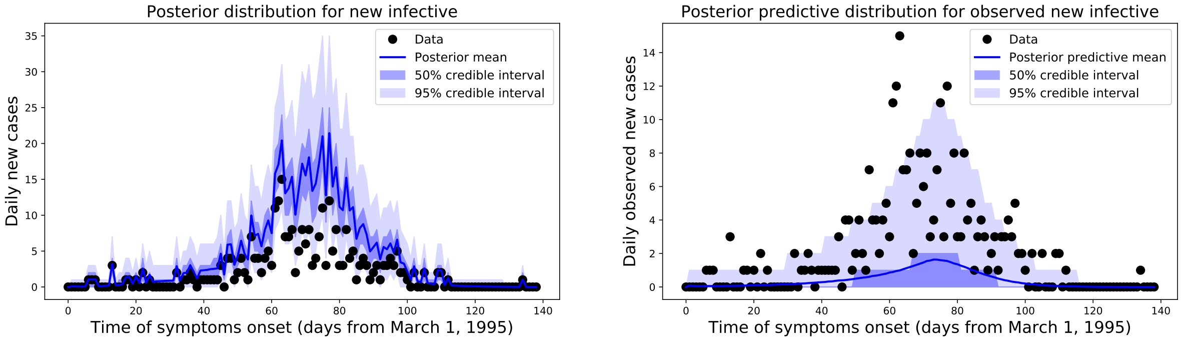


Figure 6.5: Analysis of real Ebola data with our method. On the left, posterior smoothing distributions for the number of new infectives per day and on the right posterior predictive distributions for the associated observations, i.e., subject to under-reporting. Control measures were introduced on day 70.

Table 6.2: Parameter estimates for the real Ebola data. Numbers in parentheses in column 5 are standard errors, for all other columns they are posterior standard deviations. For columns 2,3,4,6 the parameter estimates are posterior means. For each of β, λ, and $1/\rho$ the pairs of estimates in column 1 were obtained from the respective bi-modal posteriors by applying k-means clustering, with $k = 2$, to the MCMC output.

Parameter	β	λ	$1/\rho$	$1/\gamma$	$\mathbf{q^{(2,3)}}$	$\mathbf{q^{(3,4)}}$	$\mathbf{R_0}$
Our MCMC method vague prior	0.36 (0.049) 0.22 (0.025)	0.32 (0.140) 0.05 (0.008)	10.39 (1.554) 1.86 (0.487)	6.17 (1.042)	0.44 (0.103)	0.36 (0.088)	2.18 (0.227) 1.42 (0.102)
Our MCMC method informative prior	0.26 (0.033)	0.12 (0.064)	6.07 (1.919)	6.86 (0.834)	0.50 (0.109)	0.41 (0.093)	1.64 (0.696)
[Lekone and Finkenstädt, 2006] vague prior	0.24 (0.020)	0.16 (0.009)	9.43 (0.620)	5.71 (0.548)	_	_	1.38 (0.127)
[Lekone and Finkenstädt, 2006] informative prior	0.21 (0.017)	0.15 (0.010)	10.11 (0.713)	6.52 (0.564)	_	_	1.36 (0.128)
ODE + least squares [Chowell et al., 2004]	0.33 (0.006)	0.98 (unknown)	5.30 (0.230)	5.61 (0.190)	_	_	1.83 (0.060)
ABC ABSEIR [Brown et al., 2018]	0.3 (0.088)	0.36 (0.325)	7.91 (2.703)	15.01 (32.863)	_	_	3.66 (6.592)

prior the proposed method finds bi-modal posterior distributions of β, λ, and $1/\rho$. For β, one of the modes roughly matches the posterior mean obtained using [Lekone and Finkenstädt, 2006] whilst the other is more similar to the least-squares estimate from [Chowell et al., 2004]; one can conjecture that the proposed MCMC sampler has better mixing than that of [Lekone and Finkenstädt, 2006], allowing it to find these two modes. 3) The proposed method can report estimates for $q^{(2,3)}$ and $q^{(3,4)}$, whilst the other methods do not. Figure 6.5 shows posterior and posterior-predictive distributions for the counts of new infectives each day. The former estimates for the true numbers gave rise to the under-reported data, whilst the latter shows coverage of the data hence a good model fit [Gelman et al., 1996].

6.4.2 Accuracy: filtering bias and credible interval coverage

The purpose of this subsection is to study the accuracy of the approximate filtering distributions obtained from algorithm 8 when applied to the Ebola model described in subsection 6.4.1. The ground truth parameter values $(\beta, \lambda, \rho, \gamma)$ in the synthetic data experiment were taken together with $q_t^{(2,3)} = 291/316$, $q_t^{(3,4)} = 236/316$. Three population sizes $n = 5 \times 10^2, 5 \times 10^4, 5 \times 10^6$ are considered, and in each case the initial distribution was $\boldsymbol{\pi}_0 = [1 - 1/n, 1/n, 0, 0]^{\mathrm{T}}$. For each value of n, 2×10^4 data sets are simulated from the model, each over 200 time steps.

To assess accuracy, consider bias and credible-interval coverage. The former is calculated from the empirical bias associated with the mean vector of the approximation to $p(\mathbf{x}_t | \mathbf{Y}_{1:t})$ obtained from algorithm 8 as an estimator of $\mathbf{x}_t$. The latter is obtained from the empirical coverage of the nominal 95%-credible interval for the marginal over each $x_t^{(i)}$, $i = 1, 2, 3, 4$. For the true (i.e. approximation-free) filtering distributions, asymptotically in the number of simulated data sets the bias would be zero and the coverage would be 95%.

Figure 6.6 shows that for all three values of n, the bias at every time step and for every compartment is less than 0.1 in magnitude. This shows the approximation is very accurate: the

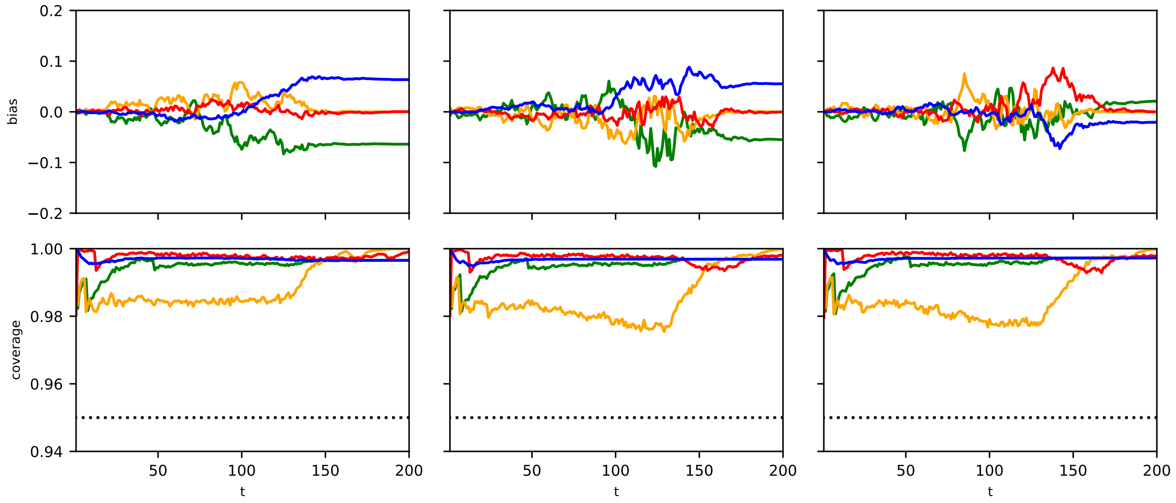


Figure 6.6: Empirical bias and empirical coverage of nominal 95%-credible intervals from 2×10^4 simulations over 200 time steps of the Ebola model. Columns from left to right: $n = 5 \times 10^2, 5 \times 10^4, 5 \times 10^6$. Top row: bias, bottom row: coverage. Red, yellow, blue, green correspond to $x_t^{(i)}$, $i = 1,2,3,4$, i.e. susceptible, exposed, infective, removed.

true values of $x_t^{(i)}$, $i = 1,2,3,4$ are always integers, and a bias less than 0.5 in magnitude means that, on average, if the estimated number of individuals is rounded to the nearest integer, the true number of individuals is recovered. The credible interval coverage reported in figure 6.6 shows that the approximate filtering distributions tend to over-represent uncertainty: the empirical coverage at all time steps for all compartments of the nominal 95% interval is between 97% and 100%. The bias and coverage appear robust to population size.

6.4.3 Estimating the time-varying reproduction number of COVID-19 in Wuhan, China

A compartmental model for estimating the time-varying reproduction number of COVID-19 in Wuhan, China, has recently been published in [Kucharski et al., 2020]. The model has 15 compartments: susceptibles in Wuhan become exposed and either stay in Wuhan or depart internationally, then, in either case, pass through further stages being exposed, infective, symptomatic and confirmed. The transmission rate is modelled as time-varying $(\beta_t)_{t\geq 0}$, a-priori by a geometric random walk, and β_t is considered proportional to the reproductive number R_t. [Kucharski et al., 2020] proposed a Sequential Monte Carlo (SMC) algorithm to estimate $(R_t)_{t\geq 0}$ which weights samples of $(\beta_t)_{t\geq 0}$ by the likelihood of the associated ODE solution under a Poisson observation model.

The multinomial approximation can be used to replace the ODE model in Kucharski et al. [2020] with a discrete-time stochastic version of the compartmental model, and their Poisson model can be replaced by the binomial observation model from section 6.2.3. Precisely, the compartments are the following:

- S_t which is the number of susceptible individuals in Wuhan (compartment 1);
- $E_t^{(1,W)}, E_t^{(2,W)}$ which are the numbers of exposed individuals in Wuhan in the first and second stage of the incubation period (compartments 2&3);
- $I_t^{(1,W)}, I_t^{(2,W)}$ which are the numbers of infective individuals in Wuhan in the first and second stage of the disease (compartments 3&4);
- $E_t^{(1,T)}, E_t^{(2,T)}$ which are the numbers of individuals who were initially exposed whilst in Wuhan and subsequently travelled to other countries, in their first and second stage of the incubation period (compartments 5&6);
- $I_t^{(1,T)}, I_t^{(2,T)}$ which are the numbers of infective individuals who were initially exposed whilst in Wuhan and subsequently travelled to other countries, in their first and second stage of the disease (compartments 7&8);
- R_t which is the number of removed individuals (compartment 9).

The evolution of the compartments are then:

$$\begin{aligned}
S_{t+1} &= S_t - B_t^{(1,W)} - B_t^{(1,T)},\\
E_{t+1}^{(1,W)} &= E_t^{(1,W)} + B_t^{(1,W)} - B_t^{(2,W)},\\
E_{t+1}^{(2,W)} &= E_t^{(2,W)} + B_t^{(2,W)} - C_t^{(1,W)},\\
I_{t+1}^{(1,W)} &= I_t^{(1,W)} + C_t^{(1,W)} - C_t^{(2,W)},\\
I_{t+1}^{(2,W)} &= I_t^{(2,W)} + C_t^{(2,W)} - D_t^{W},\\
E_{t+1}^{(1,T)} &= E_t^{(1,T)} + B_t^{(1,T)} - B_t^{(2,T)},\\
E_{t+1}^{(2,T)} &= E_t^{(2,T)} + B_t^{(2,T)} - C_t^{(1,T)},\\
I_{t+1}^{(1,T)} &= I_t^{(1,T)} + C_t^{(1,T)} - C_t^{(2,T)},\\
I_{t+1}^{(2,T)} &= I_t^{(2,T)} + C_t^{(2,T)} - D_t^{T},\\
R_{t+1} &= R_t + D_t^{(W)} + D_t^{(T)},
\end{aligned}$$

where:

$$\begin{bmatrix} B_t^{(1,W)} \\ B_t^{(1,T)} \\ S_t - B_t^{(1,W)} - B_t^{(1,T)} \end{bmatrix} \sim \text{Mult}\left(S_t, \begin{bmatrix} (1-f_t)p_t \\ f_t p_t \\ 1-p_t \end{bmatrix} \right),$$

$$\begin{aligned}
B_t^{(2,W)} &\sim \text{Bin}\left(E_t^{(1,W)}, p_C\right), & B_t^{(2,T)} &\sim \text{Bin}\left(E_t^{(1,T)}, p_C\right),\\
C_t^{(1,W)} &\sim \text{Bin}\left(E_t^{(2,W)}, p_C\right), & C_t^{(1,T)} &\sim \text{Bin}\left(E_t^{(2,T)}, p_C\right),\\
C_t^{(2,W)} &\sim \text{Bin}\left(I_t^{(1,W)}, p_R\right), & C_t^{(2,T)} &\sim \text{Bin}\left(I_t^{(1,T)}, p_R\right),\\
D_t^{(W)} &\sim \text{Bin}\left(I_t^{(2,W)}, p_R\right), & D_t^{(T)} &\sim \text{Bin}\left(I_t^{(2,T)}, p_R\right),
\end{aligned}$$

with

$$p_t = 1 - e^{-h\beta_t(I_t^{(1,W)}+I_t^{(2,W)})/n}, \quad p_C = 1 - e^{-h2\rho}, \quad p_R = 1 - e^{-h2\gamma},$$

where f_t is the fraction of cases that depart from Wuhan to other countries at time t and h is generally chosen to be 1. The observations consist of the new infectives in Wuhan, $y_t^{(3,4)}$, and internationally, $y_t^{(7,8)}$, at each time step, subject to random under-reporting:

$$y_t^{(3,4)} \sim \mathrm{Bin}\left(C_t^{(1,W)}, q^{(W)}\right), \quad y_t^{(7,8)} \sim \mathrm{Bin}\left(C_t^{(1,T)}, q^{(T)}\right).$$

The resulting model is a compartmental model with $m = 10$ as the class of models described in subsection 6.2.3. Remark that the ODE model of Kucharski et al. [2020] incorporates a number of other compartments which are used to accumulate the numbers of individuals who have passed through certain states, but which otherwise do not play an active role in the model, hence they are not specified here.

The SMC algorithm from [Kucharski et al., 2020] is then modified such that the weights samples of $(\beta_t)_{t\geq 0}$are substituted by their approximate marginal likelihoods, computed using the multinomial filtering techniques. The full algorithm is shown in algorithm 11. Two of three data sets from [Kucharski et al., 2020] are analysed: daily counts of new infectives by date of symptom onset in Wuhan, and internationally exported from Wuhan.

The procedure of Kucharski et al. [2020] is then followed in producing the remainder of the results: the algorithm is ran for 100 times with $n_{\text{part}} = 3 \times 10^3$, resulting in 100 samples of $(\tilde{\beta}_t, \tilde{\mathbf{P}}_{t|T}, \tilde{\mathbf{Z}}_t)_{t=1,\ldots,T}$ (more sophisticated approaches to particle smoothing are available in the literature, but they are not used in order to make fair comparisons with results from Kucharski et al. [2020]).

Figure 6.7 shows results in the format of [Kucharski et al., 2020], figure 6.8 shows the results for the SMC algorithm by Kucharski et al. [2020] applied to the same data as in this thesis method, i.e. with the evacuation flights data left out of the analysis. The quantities reported in figures 6.7 and 6.8 are specified in the following list, where κ is the rate of reporting (the same numerical value as in Kucharski et al. [2020]) and $F^{(W)}$, $F^{(T)}$ are auxiliary compartments.

1. The time-varying reproduction number (first row of the figures) $R_t = \tilde{\beta}_t/\gamma$ for $t = 1, \ldots, T$;

2. The new confirmed cases by date of onset in Wuhan and China (left plot in the second row of the figures) $\tilde{y}_t^{(3,4)}$ for $t = 1, \ldots, T$:

$$\hat{z}_t \sim \mathrm{Bin}\left(n, \tilde{p}_{t|T}^{(3,4)}\right) \quad \text{and} \quad \tilde{y}_t^{(3,4)} \sim \mathrm{Bin}\left(\hat{z}_t, q^{(W)}\right).$$

3. The new confirmed cases by date of onset internationally (right plot in the second row of the figures) $\tilde{y}_t^{(7,8)}$ for $t = 1, \ldots, T$:

$$\hat{z}_t \sim \mathrm{Bin}\left(n, \tilde{p}_{t|T}^{(7,8)}\right) \quad \text{and} \quad \tilde{y}_t^{(7,8)} \sim \mathrm{Bin}\left(\hat{z}_t, q^{(T)}\right).$$

Algorithm 11 Particle filter and backward sampler for COVID-19 application

```
**initialize** $\boldsymbol{\pi}_{0|0}^{(i)} \leftarrow \boldsymbol{\pi}_0, \quad \beta_0^{(i)} = \beta_0, \quad \text{for } i = 1, \dots, n_{\text{part}}$
**for** $s = 1$ to $t$ **do**
    **for** $i = 1$ to $n_{\text{part}}$ **do**
        $\beta_s^{(i)} \leftarrow \beta_{s-1}^{(i)} \exp(V^{(i)}), \quad V^{(i)} \sim \mathrm{N}(0, \sigma_V^2)$
        $\mathbf{P}_{s|s-1}^{(i)} \leftarrow (\boldsymbol{\pi}_{s-1|s-1}^{(i)} \otimes \mathbf{1}_m) \circ \mathbf{K}_{t, \boldsymbol{\pi}_{s-1|s-1}^{(i)}, \beta_t^{(i)}}$
        $\mathbf{P}_{s|s}^{(i)} \leftarrow \frac{\mathbf{Y}_s}{n} + \left(1 - \frac{\mathbf{1}_m^{\mathrm{T}} \mathbf{Y}_s \mathbf{1}_m}{n}\right) \frac{\mathbf{P}_{s|s-1}^{(i)} \circ (\mathbf{1}_m \otimes \mathbf{1}_m - \mathbf{Q}_s)}{1 - \mathbf{1}_m^{\mathrm{T}} (\mathbf{P}_{s|s-1}^{(i)} \circ \mathbf{Q}_s) \mathbf{1}_m}$
        $\log w_s^{(i)} \leftarrow \log(n!) + \mathbf{1}_m^{\mathrm{T}} (\mathbf{Y}_s \circ \log \mathbf{P}_{s|s-1}^{(i)}) \mathbf{1}_m + \mathbf{1}_m^{\mathrm{T}} (\mathbf{Y}_s \circ \log \mathbf{Q}_s) \mathbf{1}_m - \mathbf{1}_m^{\mathrm{T}} \log(\mathbf{Y}_s!) \mathbf{1}_m$
            $+ (n - \mathbf{1}_m^{\mathrm{T}} \mathbf{Y}_s \mathbf{1}_m) \log(1 - \mathbf{1}_m^{\mathrm{T}} (\mathbf{P}_{s|s-1}^{(i)} \circ \mathbf{Q}_s) \mathbf{1}_m) - \log((n - \mathbf{1}_m^{\mathrm{T}} \mathbf{Y}_s \mathbf{1}_m)!)$
        $\boldsymbol{\pi}_{s|s}^{(p)} \leftarrow (\mathbf{1}_m^{\mathrm{T}} \mathbf{P}_{s|s}^{(i)})^{\mathrm{T}}$
    $\bar{w}_s^{(i)} \leftarrow w_i^{(s)} / \sum_j w_s^{(j)}, \quad i = i, \dots, n_{\text{part}}$
    **resample** $\{\beta_s^{(i)}, \boldsymbol{\pi}_{s|s}^{(i)}\}_{i=1}^{n_{\text{part}}}$ according to $\{\bar{w}_s^{(i)}\}_{i=1}^{n_{\text{part}}}$ and keep track of ancestors in $\boldsymbol{a}_s = [a_s^{(1)} \cdots a_s^{(n_{\text{parts}})}]^{\mathrm{T}}$

**sample** $\zeta$ according to $\{\bar{w}_t^{(i)}\}_{i=1}^{n_{\text{part}}}$
$\tilde{\boldsymbol{\pi}}_{t|t} \leftarrow \boldsymbol{\pi}_{t|t}^{(\zeta)}, \quad \tilde{\mathbf{P}}_{t|t} \leftarrow \mathbf{P}_{t|t}^{(\zeta)}, \quad \tilde{\beta}_t \leftarrow \beta_t^{(\zeta)}$
**sample** $\tilde{\mathbf{Z}}_t$ from $\mathrm{Mult}(n, \tilde{\mathbf{P}}_{t|t})$
**for** $s = t-1, \dots, 1$ **do**
    $\tilde{\boldsymbol{\pi}}_{s|t} \leftarrow \tilde{\mathbf{P}}_{s+1|t} \mathbf{1}_m$
    $\zeta \leftarrow a_s^{(\zeta)}, \quad \tilde{\boldsymbol{\pi}}_{s|s} \leftarrow \boldsymbol{\pi}_{s|s}^{(\zeta)}, \quad \tilde{\mathbf{P}}_{s|s} \leftarrow \mathbf{P}_{s|s}^{(\zeta)}, \quad \tilde{\beta}_s \leftarrow \beta_s^{(\zeta)}$
    Let $\overline{\mathbf{L}}_s$ be the matrix with elements $\bar{l}_s^{(i,j)} \leftarrow \tilde{p}_{s|s}^{(j,i)} / \tilde{\pi}_{s|s}^{(i)}$
    **for** $i = 1, \dots, m$ **do**
        **sample** $\tilde{\mathbf{Z}}_s^{(\cdot,i)}$ from $\mathrm{Mult}((\tilde{\mathbf{Z}}_{s+1} \mathbf{1}_m)^{(i)}, \overline{\mathbf{L}}_s^{(i,\cdot)})$
    $\tilde{\mathbf{P}}_{s|t} \leftarrow (\mathbf{1}_m \otimes \tilde{\boldsymbol{\pi}}_{s|t}) \circ \overline{\mathbf{L}}_s^{\mathrm{T}}$
     **return** $\{\tilde{\beta}_s, \tilde{\mathbf{P}}_{s|t}, \tilde{\mathbf{Z}}_s\}_{s=1}^{t}$
```

4. The new confirmed cases by date in Wuhan (left plot in the third row of the figures) $\Delta\widetilde{\mathrm{Conf}}_t^{(W)}$ for $t = 1, \dots, T$:

$$\begin{aligned}
&\tilde{F}_0^{(W)} = 0, \\
&\Delta\tilde{F}_t^{(W)} \sim \mathrm{Bin}\left(\tilde{z}_t^{(3,4)}, 1 - e^{-e^{-\gamma\kappa}}\right), \quad \Delta\widetilde{\mathrm{Conf}}_t^{(W)} \sim \mathrm{Bin}\left(\tilde{F}_t^{(W)}, 1 - e^{-\kappa}\right), \\
&\tilde{F}_{t+1}^{(W)} = \tilde{F}_t^{(W)} + \Delta\tilde{F}_t^{(W)} - \Delta\widetilde{\mathrm{Conf}}_t^{(W)}.
\end{aligned}$$

5. The new confirmed cases by date internationally (right plot in the third row of the figures)

$\Delta\widetilde{\mathrm{Conf}}_t^{(T)}$ for $t = 1, \ldots, T$:

$$\begin{aligned}
&\tilde{F}_0^{(T)} = 0,\\
&\Delta\tilde{F}_t^{(T)} \sim \mathrm{Bin}\left(\tilde{z}_t^{(7,8)}, 1 - e^{-e^{-\gamma\kappa}}\right), \quad \Delta\widetilde{\mathrm{Conf}}_t^{(T)} \sim \mathrm{Bin}\left(\tilde{F}_s^{(T)}, 1 - e^{-\kappa}\right),\\
&\tilde{F}_{t+1}^{(T)} = \tilde{F}_t^{(T)} + \Delta\tilde{F}_t^{(T)} - \Delta\widetilde{\mathrm{Conf}}_t^{(T)}.
\end{aligned}$$

Given figures 6.7 and 6.8, it can be noticed that the estimates obtained from algorithm 11 of R_t are generally lower, and closer to 1 than the ones from Kucharski et al. [2020] for the period after travel restrictions are introduced; and the credible intervals for the in-sample plots are generally wider, reflecting the stochastic nature of the proposed compartmental model. In the bottom two plots, the posterior distributions are mostly concentrated on lower values than those from the method of [Kucharski et al., 2020].

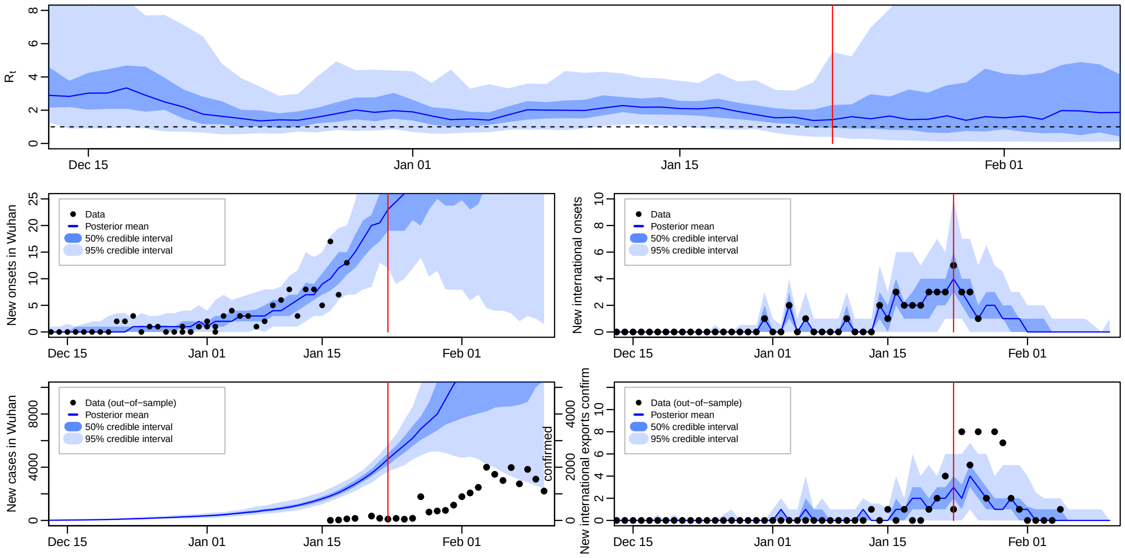


Figure 6.7: Results for the COVID-19 model using algorithm 11. The red line shows the date at which travel restrictions were introduced. Top: estimated reproduction number. Middle row: estimated daily new confirmed cases in Wuhan (left) and internationally (right), both with in-sample data by date of symptom onset. Bottom row, left: estimated new symptomatic but possibly unconfirmed cases (left axis) and out-of-sample new confirmed cases data (right axis); right: estimated confirmed international cases by date of confirmation and out-of-sample data.

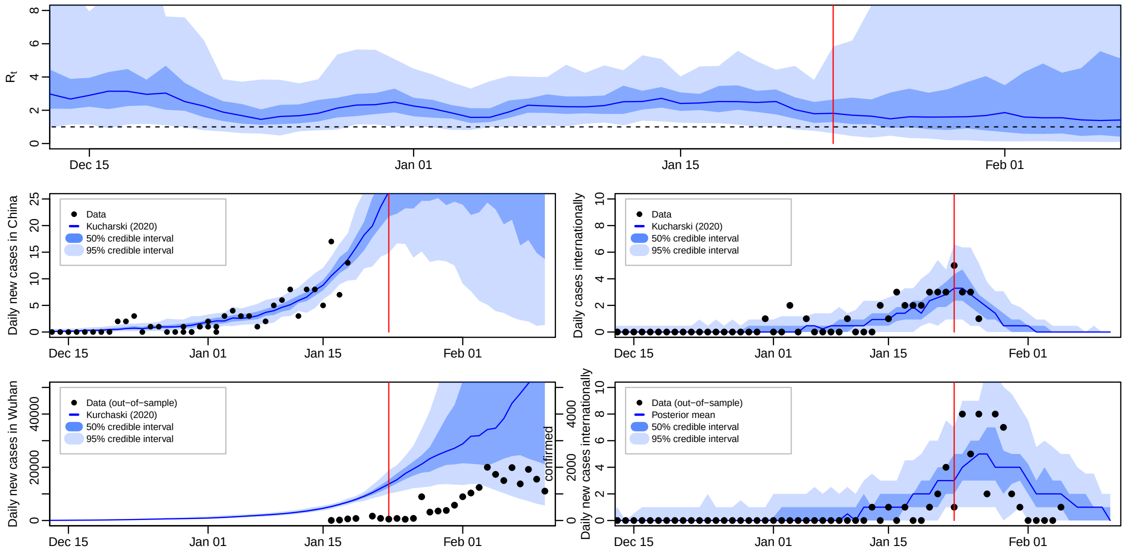


Figure 6.8: Results for the COVID-19 model using Kucharski et al. [2020] methods without rescue flights data. The red line shows the date at which travel restrictions were introduced. Top: estimated reproduction number. Middle row: estimated daily new confirmed cases in Wuhan (left) and internationally (right), both with in-sample data by date of symptom onset. Bottom row, left: estimated new symptomatic but possibly unconfirmed cases (left axis) and out-of-sample new confirmed cases data (right axis); right: estimated confirmed international cases by date of confirmation and out-of-sample data.

Chapter 7

Conclusion

This thesis extensively treated hidden Markov models by presenting the main background and issues (chapter 2) arising when scaling-up the state-space. Novel methods to perform inference over high-dimensional HMMs are developed (chapters 3, 5, 6), with experiments touching several strands of applications: traffic modelling, image classification, next-frame prediction, epidemiology.

In chapter 3, an approximate procedure for filtering and smoothing distributions in high-dimensional FHMM is proposed. The procedure avoids the exponential-in-dimension computational cost of the forward-backward algorithm for FHMMs [Ghahramani and Jordan, 1997] and guarantees the local total variation distance between the approximations and the filtering-smoothing distributions to have dimension-free upper bounds.
There are number of ways in which this work could be extended or generalized. From an algorithmic point of view, it seems natural to explore hybrids between variational methods of the sort proposed by Ghahramani and Jordan [1997] and the Graph Filter-Smoother. In the spirit of variational methods, could a layer of optimization in Kullback-Leibler divergence be somehow combined with the approximations in Graph Filter-Smoother, with the aim of yielding even more accurate approximations? From a theoretical point of view, it would be interesting to investigate whether the kind of mathematical tools used to study the Graph Filter and Smoother could also help rigorously quantify the approximation error in variational methods for FHMMs, which appears to be an unsolved problem. Another interesting question beyond the scope of the present work is whether the generalized Dobrushin Comparison Theorems developed by Rebeschini and van Handel [2014] might help loosen some of the technical assumptions on which the results rely. From a modelling point of view, there are many directions in which the London Underground example could be further investigated. The model considered is a simple prototype,

with straightforward spacial interactions among lanes and stations. It would be interesting to see how it performs with spacial interactions over time, i.e. lanes do not evolve independently from each other, and with a richer state space. Even the emission distribution could be chosen more carefully, a zero-inflated Poisson could be used to model more precisely the quiet periods.

Chapter 5 introduces a novel hybrid model between Bayesian neural networks and FHMMs called Hidden Markov neural network (HMNN), where the posterior over the weights is estimated sequentially through variational Bayes with an evolving prior, that is obtained by propagating forward through a stochastic transition kernel the approximation of the posterior at the previous time step. Moreover, the considered variational approximation induces a regularization technique called variational DropConnect, which resembles DropConnect Wan et al. [2013] and variational DropOut [Kingma et al., 2015].
The idea behind HMNNs is simple and it opens multiple research questions that can be answered in future works. The quality of the approximation is not treated and the rate of accumulation of the error over time is unknown. Theoretical studies on variational approximations could be used to answer this matter [Yang et al., 2017, Chérief-Abdellatif, 2019]. A smoothing algorithm is not considered, this could improve the performance of HMNN and open the avenue for an approximate EM algorithm, which would avoid the heavy initial tuning of the parameters. The neural networks architecture deployed in HMNNs are simple and they could be extended to recurrent neural network and convolutional neural network, to tackle more complicated applications. The variational DropConnect idea suggests that using mixtures of Gaussians as variational approximations can improve the performance, it would be interesting to see if that is the case in other variational Bayes scenarios.

Chapter 6 proposes an approximation for filtering and smoothing distribution in compartmental models with fixed population. The general argument is to modify the prediction and correction step in the forward-backward algorithm in such a way that they preserve the form of the chosen approximation, i.e. after applying prediction /correction to a Multinomial distribution the results is still a Multinomial distribution. The algorithm is then used in an epidemiological framework. Several paths can be undertaken to extend this work. Little is known about the approximation, surely a detailed mathematical study on both the quality of the approximation and the behaviour of the approximate marginal likelihood is needed. This can be done, for example, for an increasing population size given what is suggested by figure 6.6 in the experimental session. As showed during the experiments, the method can be nested in MCMC and SMC algorithm to infer the parameters of the epidemics, but how far can someone go in creating hybrids algorithms? Interestingly, one option could be to use the multinomial approximation to inform the proposal in an SMC. The Multinomial choice seems a bit restrictive, it is likely that other distributions could be used to improve dispersion or relax the fixed population size assumption. Finally, applications in epidemiology are presented, however compartmental models have a wide range of applications that can be explored, e.g. pharmacokinetics, biomedicine, engineering.

Appendix A

Exploiting locality in high-dimensional factorial hidden Markov models

This appendix provides some extra details on chapter 3. In particular, section A.1 shows the formulation of the EM algorithm for Graph Filter-Smoother under Gaussian and Poisson emission. Section A.2 gives some additional details on the traffic flow experiment. Section A.3 formulates a particle filter algorithm derived from the Graph Filter-Smoother.

A.1 EM algorithm for Graph Filter-Smoother

The parameters of the FHMMs considered in section 3.4 can be estimated with an EM algorithm where the exact smoothing distribution is substituted with the *Graph Filter-Smoother* approximation. This appendix contains details of the considered maximization step for both the Gaussian and the Poisson model.

A.1.1 Gaussian emission

From the Gaussian emission model as in subsection 3.4.1 it can be seen that the parameters are $\theta = (\hat{\mu}_0, \hat{p}, c, \sigma^2)$. The log-likelihood is then:

$$\begin{aligned}\log L(\theta; X_{0:T}, y_{0:T}) = \text{const.} &+ \sum_{v=1}^{M} \log[\hat{\mu}_0(X_0^v)] \\ &+ \sum_{t=1}^{T}\sum_{v=1}^{M} \log[\hat{p}(X_{t-1}^v, X_t^v)] \\ &+ \sum_{t=1}^{T}\sum_{f=1}^{M-1} \left\{ -\frac{1}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\left[y_t^f - c\left(X_t^f - X_t^{f+1}\right)\right]^2 \right\}.\end{aligned}$$

Consider the EM scenario as explained in subsection 2.1.3, the aim is to maximize the expected log-likelihood, however this expectation is taken under the approximated smoothing distribution $\tilde{\pi}_{t|T}$ given the parameters θ'. To make the notation lighter the dependence of $\tilde{\pi}_{t|T}$ on θ' is dropped. The expected log-likelihood is:

$$\begin{aligned} Q_T(\theta,\theta') = &\sum_{v=1}^{M}\sum_{x\in\mathbb{X}} \log[\hat{\mu}_0(x)]\tilde{\pi}^v_{0|T}(x) + \sum_{t=1}^{T}\sum_{v=1}^{M}\sum_{x\in\mathbb{X}}\sum_{z\in\mathbb{X}} \log[\hat{p}(x,z)]\tilde{\pi}^v_{t-1,t|T}(x,z) \\ &+ \sum_{t=1}^{T}\sum_{v=1}^{M-1}\sum_{x\in\mathbb{X}}\sum_{z\in\mathbb{X}} \left\{-\frac{1}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\left[y_t^f - c\cdot x - c\cdot z\right]^2\right\}\tilde{\pi}^{v,v+1}_{t|T}(x,z), \end{aligned}$$

where $\tilde{\pi}^{v,v+1}_{t|T}$ is the joint smoothing distribution of X^v_t, X^{v+1}_t given θ' and $\tilde{\pi}^v_{t-1,t|T}$ is the joint smoothing distribution of X^v_{t-1}, X^v_t given θ':

$$\tilde{\pi}^v_{t-1,t|T}(x,z) = \frac{p'(x,z)\tilde{\pi}^v_{t-1}(x)\tilde{\pi}^v_{t|T}(z)}{\sum_{\tilde{x}\in\mathbb{X}} p'(\tilde{x},z)\tilde{\pi}^v_{t-1}(\tilde{x})},$$

where p' is the current estimate of $\hat{p}$. Given the target function the next step is to compute the gradient of Q_T:

$$\begin{aligned} \frac{\partial Q_T}{\partial \hat{\mu}_0(x)} &= \frac{1}{\hat{\mu}_0(x)}\sum_{v=1}^{M}\tilde{\pi}^v_{0|T}(x) - \frac{1}{\hat{\mu}_0(\tilde{x})}\sum_{v=1}^{M}\tilde{\pi}^v_{0|T}(\tilde{x}); \\ \frac{\partial Q_T}{\partial \hat{p}(x,z)} &= \frac{1}{\hat{p}(x,z)}\sum_{t=1}^{T}\sum_{v=1}^{M}\tilde{\pi}^v_{t-1,t|T}(x,z) - \frac{1}{\hat{p}(x,\tilde{x})}\sum_{t=1}^{T}\sum_{v=1}^{M}\tilde{\pi}^v_{t-1,t|T}(x,\tilde{x}); \\ \frac{\partial Q_T}{\partial c} &= \sum_{t=1}^{T}\sum_{f=1}^{M-1}\sum_{x\in\mathbb{X}}\sum_{z\in\mathbb{X}}\left\{\frac{(x+z)}{\sigma^2}\left[y_t^f - c\cdot x - c\cdot z\right]\right\}\tilde{\pi}^{f,f+1}_{t|T}(x,z); \\ \frac{\partial Q_T}{\partial \sigma^2} &= \sum_{t=1}^{T}\sum_{f=1}^{M-1}\sum_{x\in\mathbb{X}}\sum_{z\in\mathbb{X}}\left[-\frac{1}{2\sigma^2} + \frac{1}{2(\sigma^2)^2}\left(y_t^f - c\cdot x - c\cdot z\right)^2\right]\tilde{\pi}^{f,f+1}_{t|T}(x,z); \end{aligned}$$

remark that deriving Q_T is not enough, all the constraint on the probability masses must be satisfied, i.e.:

$$\hat{\mu}_0(\tilde{x}) = 1 - \sum_{\tilde{z}\in\mathbb{X}}\hat{\mu}_0(\tilde{z}) \quad \text{and} \quad \hat{p}(x,\tilde{x}) = 1 - \sum_{\tilde{z}\in\mathbb{X}}\hat{p}(x,\tilde{z}).$$

The M-step is then obtained with $\nabla Q_T(\hat{\mu}_0, \hat{p}, c, \sigma^2) = 0$, precisely:

$$\begin{aligned} \hat{\mu}_0(x) &= \frac{1}{M}\sum_{v=1}^{M}\tilde{\pi}^v_{0|T}(x); \\ \hat{p}(x,z) &= \frac{\sum_{t=1}^{T}\sum_{v=1}^{M}\tilde{\pi}^v_{t-1,t|T}(x,z)}{\sum_{t=1}^{T}\sum_{v=1}^{M}\tilde{\pi}^v_{t-1|T}(x)}; \\ c &= \frac{\sum_{t=1}^{T}\sum_{f=1}^{M-1}\sum_{x\in\mathbb{X}}\sum_{z\in\mathbb{X}} y_t^f(x+z)\tilde{\pi}^{f,f+1}_{t|T}(x,z)}{\sum_{t=1}^{T}\sum_{f=1}^{M-1}\sum_{x\in\mathbb{X}}\sum_{z\in\mathbb{X}}(x+z)^2\tilde{\pi}^{f,f+1}_{t|T}(x,z)}; \\ \sigma^2 &= \frac{1}{T(M-1)}\sum_{t=1}^{T}\sum_{f=1}^{M-1}\sum_{x\in\mathbb{X}}\sum_{z\in\mathbb{X}}[y_t - c(x+z)]^2\tilde{\pi}^{f,f+1}_{t|T}(x,z). \end{aligned}$$

A.1.2 Poisson emission

Consider the model described in subsection 3.4.2. A graphical representation is available in figure A.1.

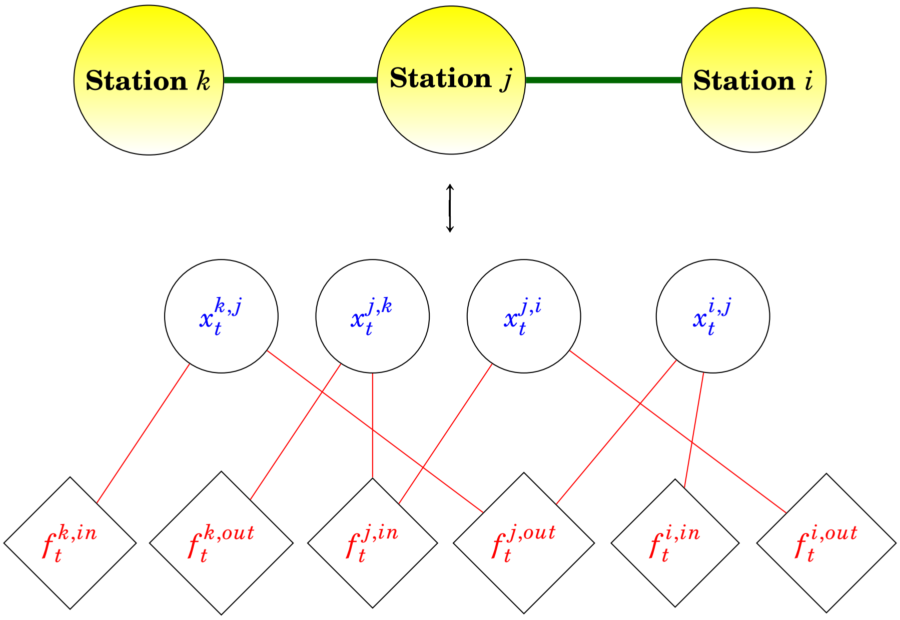


Figure A.1: Top: a simple tube's network with three stations. Bottom: the corresponding factor graph build as explained in section 3.4.2.

The parameters of the model are $\theta = (\mu_0, p, \lambda)$. The log-likelihood of the model is:

$$\begin{aligned}\log L(\theta; X_{0:T}, y_{0:T}) = &\sum_{i,j=1}^{F} \log[\mu_0^{i,j}(X_0^{i,j})] \\ &+ \sum_{t=1}^{T}\sum_{i,j=1}^{F} \log[p^{i,j}(X_{t-1}^{i,j}, X_t^{i,j})] \\ &+ \sum_{t=1}^{T}\sum_{i=1}^{F} \left\{ y_t^{i,in} \log(\lambda^{i,in}) - \lambda^{i,in} \left(\sum_{j \in N(i)} X_t^{i,j} \right) \right\} \\ &+ \sum_{t=1}^{T}\sum_{i=1}^{F} \left\{ y_t^{i,out} \log(\lambda^{i,out}) - \lambda^{i,out} \left(\sum_{j \in N(i)} X_t^{j,i} \right) \right\} \\ &+ const.\end{aligned}$$

As for the previous section, in EM the expected log-likelihood must be maximise, where the expectation is taken under the approximated smoothing distribution $\tilde{\pi}_{t|T}$ given the current estimates of the parameters θ' (as for the previous subsection the dependence of the approximate smoothing on θ' is dropped). Precisely, the expected log-likelihood is:

$$
\begin{aligned}
Q_T(\theta,\theta') = &\sum_{i,j=1}^{F}\sum_{x\in\mathbb{X}}\log[\mu_0^{i,j}(x)]\tilde{\pi}_{0|T}^{i,j}(x)+\sum_{t=1}^{T}\sum_{v=1}^{M}\sum_{x\in\mathbb{X}}\sum_{z\in\mathbb{X}}\log[p^{i,j}(x,z)]\tilde{\pi}_{t-1,t|T}^{i,j}(x,z)\\
&+\sum_{t=1}^{T}\sum_{i=1}^{F}\left\{y_t^{i,in}\log(\lambda^{i,in})-\sum_{x\in\mathbb{X}^{i,N(i)}}\lambda^{i,in}\left(\sum_{j\in N(i)}x_t^{i,j}\right)\tilde{\pi}_{t|T}^{i,N(i)}(x^{i,N(i)})\right\}\\
&+\sum_{t=1}^{T}\sum_{i=1}^{F}\left\{y_t^{i,out}\log(\lambda^{i,out})-\sum_{x\in\mathbb{X}^{i,N(i)}}\lambda^{i,out}\left(\sum_{j\in N(i)}x_t^{j,i}\right)\tilde{\pi}_{t|T}^{N(i),i}(x^{N(i),i})\right\}.
\end{aligned}
$$

where $\tilde{\pi}_{t|T}^{i,N(i)}$ is the joint smoothing distribution of $(X_t^{i,j})_{j\in N(i)}$ and $\tilde{\pi}_{t-1,t|T}^{i,j}$ is the joint smoothing distribution of $X_{t-1}^{i,j}, X_t^{i,j}$. Given the target function the gradient of Q_T can be computed:

$$
\begin{aligned}
&\frac{\partial Q_T}{\partial\mu_0^{i,j}(x)}=\frac{1}{\mu_0^{i,j}(x)}\tilde{\pi}_{0|T}^{i,j}(x)-\frac{1}{\mu_0^{i,j}(\tilde{x})}\tilde{\pi}_{0|T}^{i,j}(\tilde{x});\\
&\frac{\partial Q_T}{\partial Q(\mu_0,p,\lambda)\partial p^{i,j}(x,z)}=\frac{1}{p^{i,j}(x,z)}\sum_{t=1}^{T}\tilde{\pi}_{t-1,t|T}^{i,j}(x,z)-\frac{1}{p^{i,j}(x,\tilde{x})}\sum_{t=1}^{T}\tilde{\pi}_{t-1,t|T}^{i,j}(x,\tilde{x});\\
&\frac{\partial Q_T}{\partial\lambda^{i,in}}=\sum_{t=1}^{T}\left\{\frac{y_t^{i,in}}{\lambda^{i,in}}-\sum_{x\in\mathbb{X}^{i,N(i)}}\left(\sum_{j\in N(i)}x_t^{i,j}\right)\tilde{\pi}_{t|T}^{i,N(i)}(x)\right\};\\
&\frac{\partial Q_T}{\partial\lambda^{i,out}}=\sum_{t=1}^{T}\left\{\frac{y_t^{i,out}}{\lambda^{i,out}}-\sum_{x\in\mathbb{X}^{i,N(i)}}\left(\sum_{j\in N(i)}x_t^{j,i}\right)\tilde{\pi}_{t|T}^{N(i),i}(x)\right\}.
\end{aligned}
$$

remark that deriving Q_T is not enough, all the constraint on the probability masses must be satisfied, i.e.:

$$
\mu_0^{i,j}(\tilde{x})=1-\sum_{\tilde{z}\in\mathbb{X}}\mu^{i,j}{}_0(\tilde{z})\quad\text{and}\quad p^{i,j}(x,\tilde{x})=1-\sum_{\tilde{z}\in\mathbb{X}}p^{i,j}(x,\tilde{z}).
$$

The M-step is then obtained by setting $\nabla Q_T(\theta)=0$, precisely:

$$
\begin{aligned}
&\mu_0^{i,j}(x)=\tilde{\pi}_{0|T}^{i,j}(x);\\
&p^{i,j}(x,z)=\frac{\sum_{t=1}^{T}\tilde{\pi}_{t-1,t|T}^{i,j}(x,z)}{\sum_{t=1}^{T}\tilde{\pi}_{t-1|T}^{i,j}(x)};\\
&\lambda^{i,in}=\frac{\sum_{t=1}^{T}y_t^{i,in}}{\sum_{x\in\mathbb{X}^{i,N(i)}}\left(\sum_{j\in N(i)}x_t^{i,j}\right)\tilde{\pi}_{t|T}^{i,N(i)}(x)};\\
&\lambda^{i,out}=\frac{\sum_{t=1}^{T}y_t^{i,out}}{\sum_{x\in\mathbb{X}^{i,N(i)}}\left(\sum_{j\in N(i)}x_t^{j,i}\right)\tilde{\pi}_{t|T}^{N(i),i}(x)}.
\end{aligned}
$$

A.2 Analyzing traffic flows on the London Underground: supplementary

This section provides some additional details about the experiments in section 3.4.2. Section A.2.1 describes the training procedure of the LSTM. Section A.2.2 provides complete plots about the experiments.

A.2.1 LSTM details

The training set is normalized between 0 and 1. The LSTM is trained on Keras with architecture given by an LSTM cell with 10 hidden units and input the inflow-outflow for all the twenty stations, i.e. a 40-dimensional input, and an additional dense layer with output dimension 40 (the inflow-outflow for all the twenty stations). The total number of parameters is then 850. More complicated (and simpler) architecture have been tested, but the chosen one was performing the best. The weights of the neural network are trained over 100 epochs and ADAM optimizer is used. The experiment is repeated 100 times with different seeds.

A.2.2 Plots on all the stations

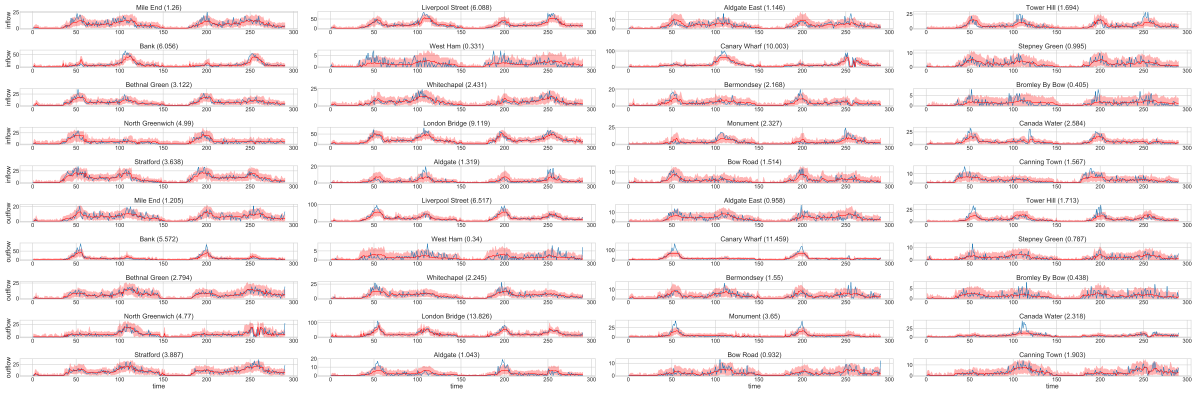


Figure A.2: Posterior predictive performance with Graph Filter on all the considered stations. The first five rows show the inflow, the remaining five rows show the outflow. In blue: the inflow and outflow per station. In red: one step-ahead posterior predictive mean (solid red line) and 0.95 credible intervals (red bands) using the Graph Filter-Smoother.

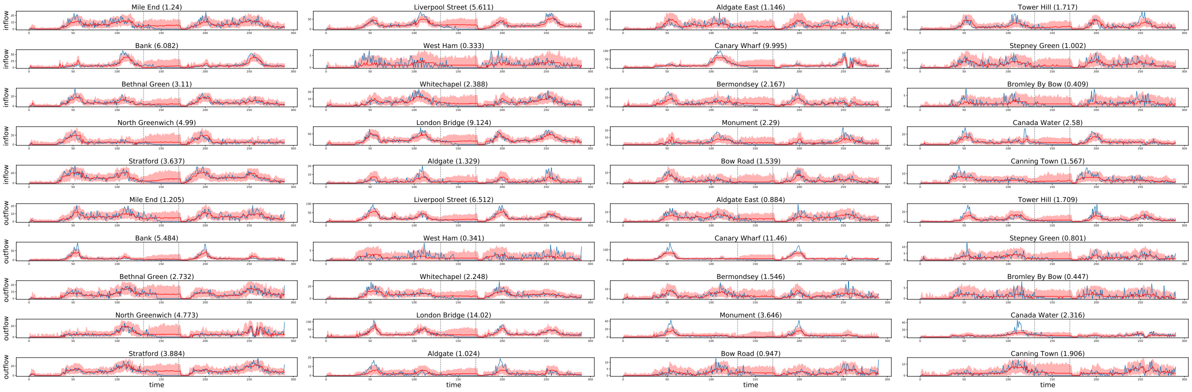


Figure A.3: Posterior predictive performance with Graph Filter on all the considered stations with missing data in a quiet period. The first five rows show the inflow, the remaining five rows show the outflow. In blue: the inflow and outflow per station. In red: one step-ahead posterior predictive mean (solid red line) and 0.95 credible intervals (red bands) using the Graph Filter-Smoother. Grey vertical dashed lines show the start and the end of the missing data window.

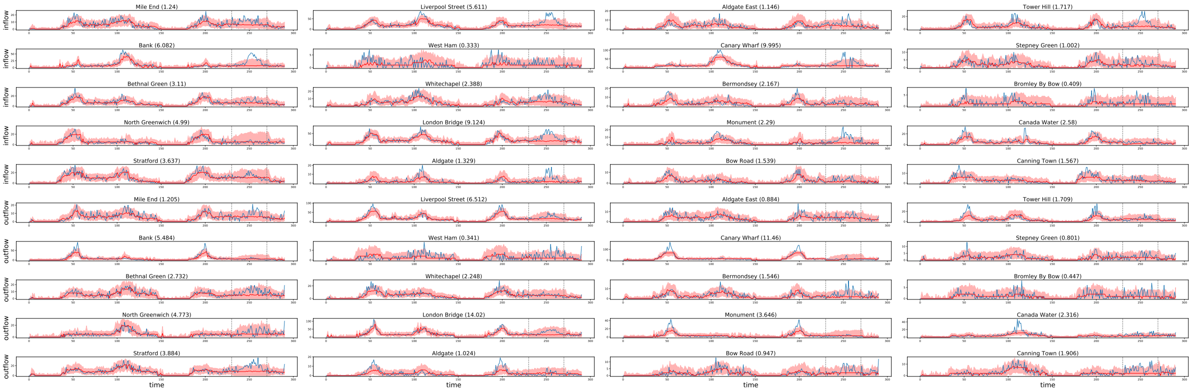


Figure A.4: Posterior predictive performance with Graph Filter on all the considered stations with missing data in a quiet period. The first five rows show the inflow, the remaining five rows show the outflow. In blue: the inflow and outflow per station. In red: one step-ahead posterior predictive mean (solid red line) and 0.95 credible intervals (red bands) using the Graph Filter-Smoother. Grey vertical dashed lines show the start and the end of the missing data window.

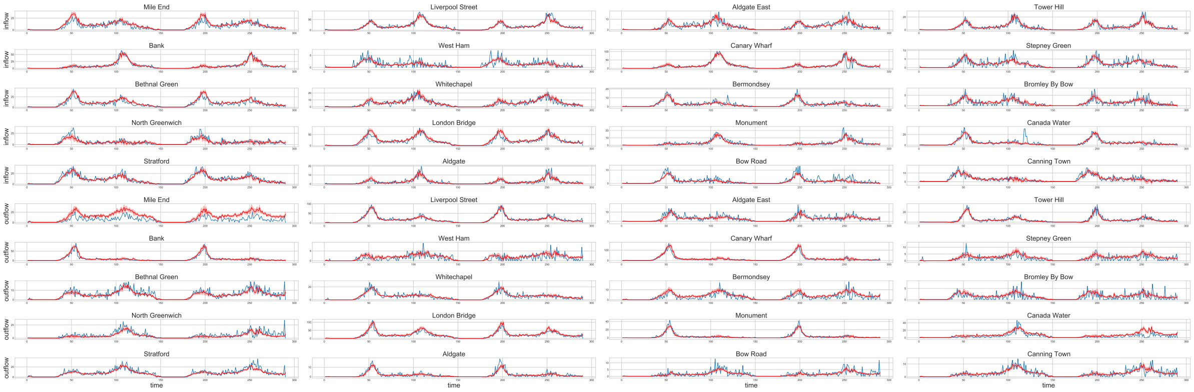


Figure A.5: LSTM performance on all the considered stations with missing data in a quiet period. The first five rows show the inflow, the remaining five rows show the outflow. In blue: the inflow and outflow per station. In red: one-step-ahead prediction with the LSTM, per time step, a sample of size 100 is built over different training of the LSTM with solid red lines showing the mean and red bands showing the region between the 0.025 and the 0.975 quantile.

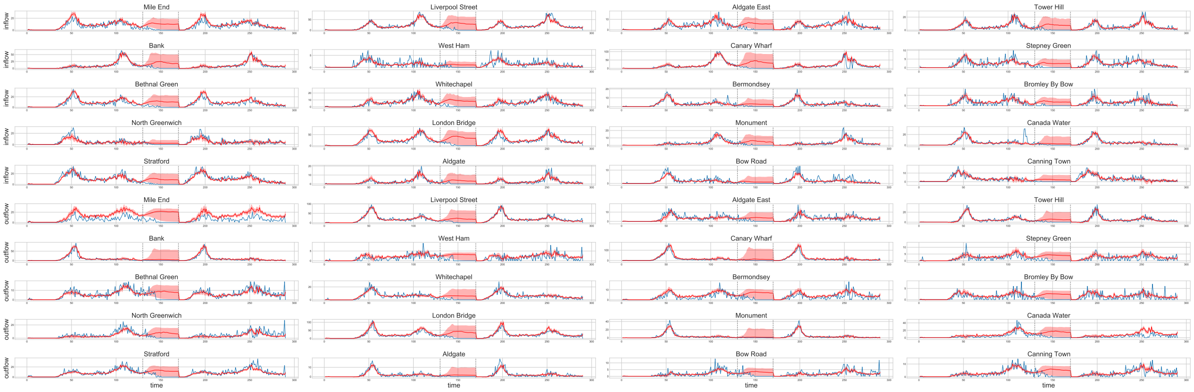


Figure A.6: LSTM performance on all the considered stations with missing data in a quiet period. The first five rows show the inflow, the remaining five rows show the outflow. In blue: the inflow and outflow per station. In red: multi-step-ahead prediction with the LSTM, per time step, a sample of size 100 is built over different training of the LSTM with solid red lines showing the mean and red bands showing the region between the 0.025 and the 0.975 quantile.

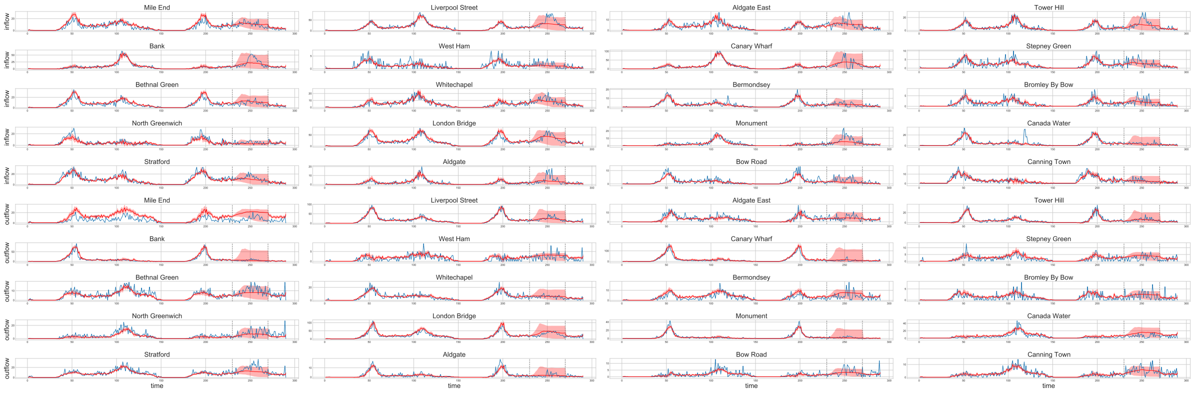


Figure A.7: LSTM performance on all the considered stations with missing data in a quiet period. The first five rows show the inflow, the remaining five rows show the outflow. In blue: the inflow and outflow per station. In red: multi-step-ahead prediction with the LSTM, per time step, a sample of size 100 is built over different training of the LSTM with solid red lines showing the mean and red bands showing the region between the 0.025 and the 0.975 quantile.

A.3 Graph Particle Filter

Chapter 3 is focused on a finite state-space scenario, however algorithm 4 can be reformulated as a particle filter as for Block Particle Filter in Rebeschini and Van Handel [2015]. The key point is to compute locally the weights associated with the elements of the partition. In such a way, per each element of the partition only a subset of factors is considered according to the chosen parameter m. As for Rebeschini and Van Handel [2015] algorithm 12 can be proved to be accurate, but this is left to future works.

Algorithm 12 Graph Particle Filter

```
Require: $\mathcal{K}, (N_f^m(K))_{K\in\mathcal{K}}, (N_v^m(K))_{K\in\mathcal{K}}, (\mu_0^K)_{K\in\mathcal{K}}, (p^v(\cdot,\cdot))_{v\in V}, (g^f(\cdot,\cdot))_{f\in F}, (y_t)_{t=\{1,\dots,T\}}$
for $K \in \mathcal{K}$ do
    $\tilde{\pi}_0^K \leftarrow \mu_0^K$
for $t \in \{1,\dots,T\}$ do
    for $i \in \{1,\dots,N\}$ do
        Sample i.i.d. $\tilde{x}_{t-1}(i)$ from the distribution $\tilde{\pi}_{t-1}$
        Sample $\tilde{x}_t^v(i) \sim p^v(\tilde{x}_{t-1}^v(i),\cdot), v \in V$
        Compute $w_t^K(i) = \prod_{f\in N_f^m(K)} g^f(\tilde{x}_t^{N(f)}(i), y_t), \quad K \in \mathcal{K}$
    Normalize the weights $w_t^K(i) = \frac{w_t^K(i)}{\sum_{j=1}^N w_t^K(j)}, \quad K \in \mathcal{K}$
    Let $\tilde{\pi}_t \leftarrow \bigotimes_{K\in\mathcal{K}} \sum_{i=1}^N w_t^K(i)\delta_{\tilde{x}_t^K(i)}$
return $((\tilde{x}_t(i))_{i=1,\dots,N})_{t=0,\dots,T}$
```

Appendix B

Inference in Stochastic Epidemic models via Multinomial Approximations

Section B.1 discusses some of the shortcomings of deterministic compartmental models, expanding on the discussion in chapter 6. Section B.2 contains proofs of lemmas 6.1 and 6.2, plus corresponding results and proofs for the observation model from subsection 6.2.3. Section B.3 provides additional details and numerical results for the Ebola example in subsection 6.4.1. Additional details and numerical results for the COVID-19 example in subsection 6.4.3 are given in section B.4.

B.1 Stochastic vs. deterministic SEIR models

Perhaps the most widely applied formulation of a compartmental model is as a system of ordinary differential equations.

SEIR example. For a population of size n, the SEIR ODE model is:

$$\frac{\mathrm{d}S}{\mathrm{d}t} = -\frac{\beta SI}{n}, \qquad \frac{\mathrm{d}E}{\mathrm{d}t} = \frac{\beta SI}{n} - \rho E, \qquad \frac{\mathrm{d}I}{\mathrm{d}t} = \rho E - \gamma I, \qquad \frac{\mathrm{d}R}{\mathrm{d}t} = \gamma I,$$

initialized with nonnegative integers in each of the compartments (S_0, E_0, I_0, R_0) such that $S_0 + E_0 + I_0 + R_0 = n$.

The most obvious drawback of ODE models is that, once model parameters and the initial population are fixed, any discrepancy between observed data and the solution of the ODE has to be explained as observation error, which is a serious restriction from a modelling point of view. In practice one can try to estimate unknown parameters and/or the initial condition by numerically minimizing this discrepancy, e.g. under squared error loss. Standard errors for parameter estimates can be derived using asymptotic theory for nonlinear least squares, but calculation of them in practice involves numerical differentiation of the ODE solution flow w.r.t. parameters [Chowell et al., 2004]. When a probabilistic observation model is specified, Bayesian approaches allow for uncertainty quantification over parameters via posterior distributions, but evaluating the likelihood function for model parameters still involves numerical solution of the ODE.

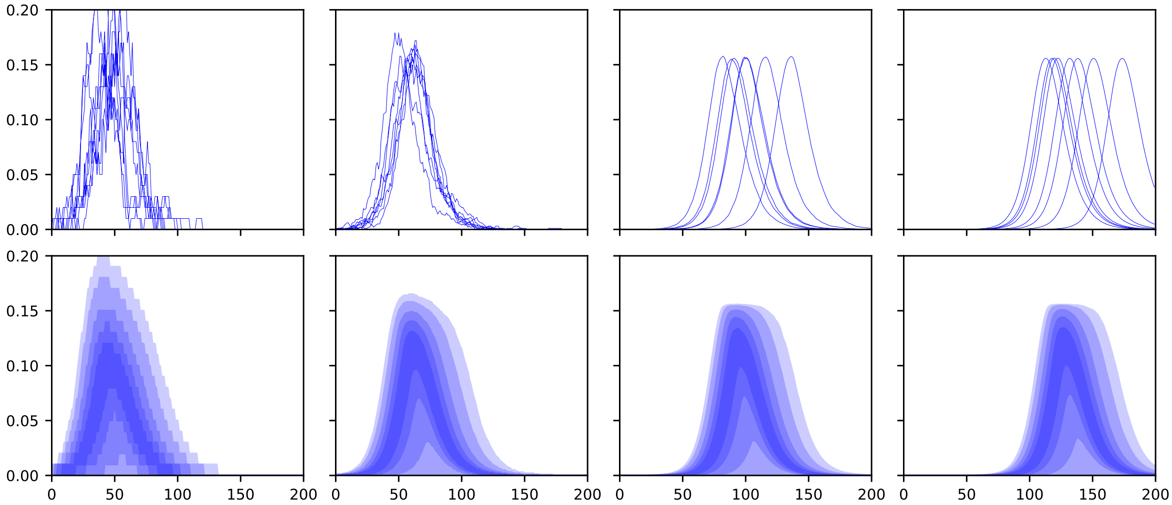


Figure B.1: Proportion of infective individuals against time in simulations from the discrete-time stochastic SEIR model with $\beta = 0.8$, $\rho = 1/5$, $\gamma = 1/9$ and $h = 1$ over 200 time steps. Columns left to right: $n = 10^2, 10^3, 10^5, 10^7$ and initial conditions $(S_0, E_0, I_0, R_0) = (n-1, 1, 0, 0)$. Top row: 10 realizations from the model. Bottom row: at each time step shaded regions indicate percentile intervals of the form $[\alpha, 1-\alpha] \times 100\%$, for $\alpha = 0.05, 0.1, 0.2, 0.3, 0.4$ estimated from 10^4 realizations from the model.

Figure B.1 shows simulation output for the proportion of infective individuals in the discrete-time SEIR model with $n = 10^2, 10^3, 10^5, 10^7$ and initial conditions $(S_0, E_0, I_0, R_0) = (n-1, 1, 0, 0)$, with $\beta = 0.8$, $\rho = 1/9$ and $\gamma = 0.2$. It is evident that the sample paths become smoother as n grows, but there is still substantial variability across sample paths even with $n = 10^7$. This can be explained by the fact that since $(E_0, I_0) = (1, 0)$ independently of n, the numbers of exposed and infective individuals in the first few time periods of the epidemic are typically very small, despite the fact that the overall population size may be large, and the statistical variability associated with these small numbers has a lasting effect on the overall timing of the outbreak.

To explain how this relates to ODE limits, for $n \geq 1$ and initial proportions of the population (s_0, e_0, i_0, r_0), i.e., $s_0 + e_0 + i_0 + r_0 = 1$, let us write $\mathscr{D}_n(s_0, e_0, i_0, r_0)$ for the collection of ODEs in (B.1) together with the initial condition (ns_0, ne_0, ni_0, nr_0). It can be checked by substitution that $t \mapsto (S_t, E_t, I_t, R_t)$ is a solution of $\mathscr{D}_1(s_0, e_0, i_0, r_0)$ if and only if $t \mapsto (nS_t, nE_t, nI_t, nR_t)$ is a solution of $\mathscr{D}_n(s_0, e_0, i_0, r_0)$. Thus n plays a trivial role in the ODE model: it is just a scaling factor for the solution.

The limit theorems of Kurtz [1970, 1971] applied in this situation pertain to the probabilistic convergence on a finite time-window as $n \to \infty$ of the path of the continuous-time SEIR Markov process with initial condition $(S_0, E_0, I_0, R_0) = (n-1, 1, 0, 0)$ and compartment counts normalized by n, to the solution of $\mathscr{D}_1(s_0, e_0, i_0, r_0)$, where $(s_0, e_0, i_0, r_0) = \lim_{n\to\infty}(n-1,1,0,0)/n = (1,0,0,0)$. However in the solution of $\mathscr{D}_1(1,0,0,0)$, the exposed and infective compartments are always empty, i.e. an epidemic never occurs. This can be reconciled with figure B.1 by observing that there the peak in the number of infectives typically occurs later as n grows: the limiting $n \to \infty$ case is that in which a peak *never* occurs.

This illustrates that sequences of well-behaved stochastic models can have ODE limits which are unrealistic to the point of being pathological, and therefore these limits are not always a sensible justification for using ODE models.

B.2 Supplementary information about multinomial filtering and smoothing

B.2.1 Proofs of lemma 6.1 and lemma 6.2

Proof of Lemma 6.1 Since μ is the probability mass function associated with $\text{Mult}(n, \boldsymbol{\pi})$, $\mathbb{E}_\mu[\boldsymbol{\eta}(\mathbf{x})] = \boldsymbol{\pi}$, so it is needed to be proved that $\sum_{\mathbf{x}\in\mathscr{S}_{m,n}} \mu(\mathbf{x}) M_t(\mathbf{x}, \boldsymbol{\pi}, \cdot)$ is the probability mass function associated with $\text{Mult}(n, \boldsymbol{\pi}^{\mathrm{T}}\mathbf{K}_{t,\boldsymbol{\pi}})$. This can be achieved using the unique characterization of the probability mass function by its moment generating function.

With $\mathbf{x} \sim \mu$, let $\tilde{\mathbf{x}} \sim M_t(\mathbf{x}, \boldsymbol{\pi}, \cdot)$, so by construction $\sum_{\mathbf{x}\in\mathscr{S}_{m,n}} \mu(\mathbf{x}) M_t(\mathbf{x}, \boldsymbol{\pi}, \cdot)$ is the marginal probability mass function of $\tilde{\mathbf{x}}$. Therefore by the definition of M_t in section 6.3.1, $\tilde{\mathbf{x}} = (\mathbf{1}_m^{\mathrm{T}}\mathbf{Z})^{\mathrm{T}}$, where the rows of $\mathbf{Z}$ are conditionally independent given $\mathbf{x}$, and the conditional distribution of the ith row of $\mathbf{Z}$ given $\mathbf{x}$ is $\text{Mult}(x^{(i)}, \mathbf{K}_{t,\boldsymbol{\pi}}^{(i,\cdot)})$. Using these facts, the moment generating function of $\tilde{\mathbf{x}}$ can be written:

$$\begin{aligned}
\mathbb{E}\left[\exp(\tilde{\mathbf{x}}^{\mathrm{T}}\mathbf{b})\right] &= \mathbb{E}\left[\exp(\mathbf{1}_m^{\mathrm{T}}\mathbf{Z}\mathbf{b})\right] \\
&= \mathbb{E}\left[\exp\left(\sum_{i,j=1}^{m} Z^{(i,j)} b^{(j)}\right)\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\exp\left(\sum_{i,j=1}^{m} Z^{(i,j)} b^{(j)}\right)\middle| \mathbf{x}\right]\right] \\
&= \mathbb{E}\left[\prod_{i=1}^{m} \mathbb{E}\left[\exp\left(\sum_{j=1}^{m} Z^{(i,j)} b^{(j)}\right)\middle| x^{(i)}\right]\right].
\end{aligned}$$

Now use the fact that, again by definition of M_t, $\mathbb{E}\left[\exp\left(\sum_{j=1}^{m} Z^{(i,j)} b^{(j)}\right)\middle| x^{(i)}\right]$ is the m.g.f. of $\text{Mult}(x^{(i)}, \mathbf{K}_{t,\boldsymbol{\pi}}^{(i,\cdot)})$, where $\mathbf{K}_{t,\boldsymbol{\pi}}$ has elements $k_{t,\boldsymbol{\pi}}^{(i,j)}$, i.e. :

$$\mathbb{E}\left[\exp\left(\sum_{j=1}^{m} Z^{(i,j)} b^{(j)}\right)\middle| x^{(i)}\right] = \left(\sum_{j=1}^{m} k_{t,\boldsymbol{\pi}}^{(i,j)} e^{b^{(j)}}\right)^{x^{(i)}}.$$

Substituting this into (1) and then using $\mathbf{x} \sim \mu = \mathrm{Mult}(n, \boldsymbol{\pi})$,

$$\begin{aligned}
\mathbb{E}\left[\exp(\widetilde{\mathbf{x}}^{\mathrm{T}}\mathbf{b})\right] &= \mathbb{E}\left[\prod_{i=1}^{m} \mathbb{E}\left[\exp\left(\sum_{j=1}^{m} Z^{(i,j)} b^{(j)}\right)\middle| x^{(i)}\right]\right] = \mathbb{E}\left[\prod_{i=1}^{m}\left(\sum_{j=1}^{m} k_{t,\boldsymbol{\pi}}^{(i,j)} e^{b^{(j)}}\right)^{x^{(i)}}\right] \\
&= \sum_{(x^{(1)},\ldots,x^{(m)})\in\mathscr{S}_{m,n}} n! \prod_{i=1}^{m} \frac{(\pi^{(i)})^{x^{(i)}}}{x^{(i)}!}\left(\sum_{j=1}^{m} k_{t,\boldsymbol{\pi}}^{(i,j)} e^{b^{(j)}}\right)^{x^{(i)}} \\
&= \sum_{(x^{(1)},\ldots,x^{(m)})\in\mathscr{S}_{m,n}} n! \prod_{i=1}^{m} \frac{1}{x^{(i)}!}\left(\sum_{j=1}^{m} \pi^{(i)} k_{t,\boldsymbol{\pi}}^{(i,j)} e^{b^{(j)}}\right)^{x^{(i)}} \\
&= \left(\sum_{i=1}^{m}\sum_{j=1}^{m} \pi^{(i)} k_{t,\boldsymbol{\pi}}^{(i,j)} e^{b^{(j)}}\right)^{n} = \left(\sum_{j=1}^{m} (\boldsymbol{\pi}^{\mathrm{T}}\mathbf{K}_{t,\boldsymbol{\pi}})^{(j)} e^{b^{(j)}}\right)^{n},
\end{aligned}$$

where the penultimate equality holds by the multinomial theorem. The proof is completed by noticing that $\left(\sum_{j=1}^{m}(\boldsymbol{\pi}^{\mathrm{T}}\mathbf{K}_{t,\boldsymbol{\pi}})^{(j)} e^{b^{(j)}}\right)^{n}$ is the moment generating function of $\mathrm{Mult}(n, \boldsymbol{\pi}^{\mathrm{T}}\mathbf{K}_{t,\boldsymbol{\pi}})$. ■

Proof of lemma 6.2 Under the distributional assumptions in the statement of the lemma, for $\mathbf{x} \in \mathscr{S}_{m,n}$,

$$p(\mathbf{x}) = \frac{n!}{\prod_{j=1}^{m} x^{(j)}!}\prod_{j=1}^{m}(\pi^{(j)})^{x^{(j)}},$$

and for $0 \le y^{(j)} \le x^{(j)}$, $j = 1, \ldots, m$,

$$p(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{m} \frac{x^{(j)}!}{y^{(j)}!(x^{(j)} - y^{(j)})!}(q^{(j)})^{y^{(j)}}(1 - q^{(j)})^{x^{(j)} - y^{(j)}}.$$

Therefore

(B.1) $$p(\mathbf{x}, \mathbf{y}) = n! \prod_{j=1}^{m} \frac{(\pi^{(j)})^{x^{(j)}}(q^{(j)})^{y^{(j)}}(1 - q^{(j)})^{x^{(j)} - y^{(j)}}}{y^{(j)}!(x^{(j)} - y^{(j)})!},$$

and

$$\begin{aligned}
p(\mathbf{y}) &= \sum_{\{x^{(j)}: x^{(j)} \ge y^{(j)}, \sum_j x^{(j)} = n\}} n! \prod_{j=1}^{m} \frac{(\pi^{(j)})^{x^{(j)}}(q^{(j)})^{y^{(j)}}(1 - q^{(j)})^{x^{(j)} - y^{(j)}}}{y^{(j)}!(x^{(j)} - y^{(j)})!} \\
&= n! \left(\prod_{j=1}^{m} \frac{(q^{(j)})^{y^{(j)}}(\pi^{(j)})^{y^{(j)}}}{y^{(j)}!}\right) \\
&\quad \times \sum_{\{x^{(j)}: x^{(j)} \ge y^{(j)}, \sum_j x^{(j)} = n\}} \left(\prod_{j=1}^{m} \frac{(\pi^{(j)})^{x^{(j)} - y^{(j)}}(1 - q^{(j)})^{x^{(j)} - y^{(j)}}}{(x^{(j)} - y^{(j)})!}\right)
\end{aligned}$$

(B.2) $$= n! \left(\prod_{j=1}^{m} \frac{(q^{(j)})^{y^{(j)}}(\pi^{(j)})^{y^{(j)}}}{y^{(j)}!}\right) \frac{\left(\sum_{j=1}^{m} \pi^{(j)}(1 - q^{(j)})\right)^{n - \sum_{j=1}^{m} y^{(j)}}}{(n - \sum_{j=1}^{m} y^{(j)})!},$$

where the final equality holds by the multinomial theorem. Dividing (B.1) by (B.2) gives

$$p(\mathbf{x}|\mathbf{y}) = \left(n - \sum_{j=1}^{m} y^{(j)}\right)! \prod_{j=1}^{m} \frac{1}{(x^{(j)} - y^{(j)})!}\left(\frac{\pi^{(j)}(1 - q^{(j)})}{\sum_{i=1}^{m} \pi^{(i)}(1 - q^{(i)})}\right)^{x^{(j)} - y^{(j)}},$$

which is the probability mass function of $\mathbf{y} + \mathbf{x}^{\star}$ given in the statement of the lemma. ■

Remark B.1. *The probability mass function in (B.2) has the interpretation of being a multinomial distribution over $n+1$ compartments, where the count variable associated with the $(m+1)$th compartment is $n - \sum_{j=1}^{m} y^{(j)}$.*

B.2.2 Proofs on lemma 6.3 and lemma 6.4

Proof of lemma 6.3 The proof is similar to the proof of Lemma 1, so some steps and commentary are omitted. Note $\mathbb{E}_{\overline{\mu}}[\boldsymbol{\eta}((\mathbf{1}_m^{\mathrm{T}}\mathbf{Z})^{\mathrm{T}})] = (\mathbf{1}_m^{\mathrm{T}}\mathbf{P})^{\mathrm{T}} = \boldsymbol{\pi}$, and let $\widetilde{\mathbf{Z}} \sim \overline{M}(\mathbf{Z}, \boldsymbol{\pi}, \cdot)$. The moment generating function of $\widetilde{\mathbf{Z}}$ is:

$$
\begin{aligned}
\mathbb{E}[\exp(\mathbf{1}_m^{\mathrm{T}}(\widetilde{\mathbf{Z}} \circ \mathbf{B})\mathbf{1}_m)] &= \mathbb{E}\left[\prod_{i=1}^{m} \exp\left(\sum_{j=1}^{m} \widetilde{z}^{(i,j)} b^{(i,j)}\right)\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\prod_{i=1}^{m} \exp\left(\sum_{j=1}^{m} \widetilde{z}^{(i,j)} b^{(i,j)}\right) \middle| \mathbf{Z}\right]\right] \\
&= \mathbb{E}\left[\prod_{i=1}^{m} \mathbb{E}\left[\exp\left(\sum_{j=1}^{m} \widetilde{z}^{(i,j)} b^{(i,j)}\right) \middle| (\mathbf{1}_m^{\mathrm{T}}\mathbf{Z})^{(i)}\right]\right] \\
&= \mathbb{E}\left[\prod_{i=1}^{m} \left(\sum_{j=1}^{m} k_{t,\boldsymbol{\pi}}^{(i,j)} b^{(i,j)}\right)^{(\mathbf{1}_m^{\mathrm{T}}\mathbf{Z})^{(i)}}\right] \\
&= n! \sum_{(x^{(1)},\ldots,x^{(m)}) \in \mathscr{S}_{m,n}} \prod_{i=1}^{m} \frac{(\pi^{(i)})^{x^{(i)}}}{x^{(i)}!} \left(\sum_{j=1}^{m} k_{t,\boldsymbol{\pi}}^{(i,j)} e^{b^{(i,j)}}\right)^{x^{(i)}} \\
&= \left(\sum_{i=1}^{m}\sum_{j=1}^{m} \pi^{(i)} k_{t,\boldsymbol{\pi}}^{(i,j)} e^{b^{(i,j)}}\right)^{n}.
\end{aligned}
$$

■

Proof of lemma 6.4 The proof is very similar to the proof of Lemma 2 so is omitted. ■

B.2.3 Proofs of lemma 6.5 and lemma 6.6

Proof of lemma 6.5 Recalling the definition of M_{s+1} from subsection 6.3.1, the numerator in (6.17) is:

$$
\begin{aligned}
&\mu_{s|s}(\mathbf{x}_s) M_{s+1}(\mathbf{x}_s, \boldsymbol{\pi}_{s|s}, \mathbf{x}_{s+1}) \\
&= \left(n! \prod_{i=1}^{m} \frac{(\pi_{s|s}^{(i)})^{x_s^{(i)}}}{x_s^{(i)}!}\right)\left(\sum_{\mathbf{Z} \in \mathscr{T}_{m,n}(\mathbf{x}_s,\mathbf{x}_{s+1})} \prod_{i=1}^{m} x_s^{(i)}! \prod_{j=1}^{m} \frac{(k_{s+1,\boldsymbol{\pi}_{s|s}}^{(i,j)})^{z^{(i,j)}}}{z^{(i,j)}!}\right) \\
&= n! \left(\prod_{i=1}^{m} (\pi_{s|s}^{(i)})^{x_s^{(i)}}\right) \sum_{\mathbf{Z} \in \mathscr{T}_{m,n}(\mathbf{x}_s,\mathbf{x}_{s+1})} \prod_{i,j=1}^{m} \frac{(k_{s+1,\boldsymbol{\pi}_{s|s}}^{(i,j)})^{z^{(i,j)}}}{z^{(i,j)}!} \\
&= n! \sum_{\mathbf{Z} \in \mathscr{T}_{m,n}(\mathbf{x}_s,\mathbf{x}_{s+1})} \prod_{i,j=1}^{m} \frac{(\pi_{s|s}^{(i)})^{z^{(i,j)}} (k_{s+1,\boldsymbol{\pi}_{s|s}}^{(i,j)})^{z^{(i,j)}}}{z^{(i,j)}!}, \qquad \text{(B.3)}
\end{aligned}
$$

where $\mathscr{T}_{m,n}(\mathbf{x}_s,\mathbf{x}_{s+1})$ is the set of $m \times m$ matrices, say $\mathbf{Z}$, with nonnegative entries such that $\mathbf{Z}\mathbf{1}_m = \mathbf{x}_s$, $(\mathbf{1}_m^{\mathrm{T}}\mathbf{Z}) = \mathbf{x}_{s+1}^{\mathrm{T}}$, and $\mathbf{1}_m^{\mathrm{T}}\mathbf{Z}\mathbf{1}_m = n$.

Now in order to derive an expression for the the denominator in (6.17), observe that (B.3) can be disintegrated to give:

$$n!\prod_{i,j=1}^{m}\frac{(\pi_{s|s}^{(i)})^{z^{(i,j)}}(k_{s+1,\boldsymbol{\pi}_{s|s}}^{(i,j)})^{z^{(i,j)}}}{z^{(i,j)}!},$$

which is the probability mass function of $\mathbf{Z} \sim \mathrm{Mult}(n,(\boldsymbol{\pi}_{s|s}\otimes\mathbf{1}_m)\circ\mathbf{K}_{s+1,\boldsymbol{\pi}_{s|s}})$. Therefore using the fact the marginal of this multinomial distribution over $\mathbf{1}_m^{\mathrm{T}}\mathbf{Z}$ is $\mathrm{Mult}(n,\boldsymbol{\pi}_{s|s}^{\mathrm{T}}\mathbf{K}_{s+1,\boldsymbol{\pi}_{s|s}})$, the denominator in (6.17) is

$$\sum_{\mathbf{x}_s\in\mathscr{S}_{m,n}}\mu_{s|s}(\mathbf{x}_s)M_{s+1}(\mathbf{x}_s,\boldsymbol{\pi}_{s|s},\mathbf{x}_{s+1}) = n!\prod_{j=1}^{m}\frac{((\boldsymbol{\pi}_{s|s}^{\mathrm{T}}\mathbf{K}_{s+1,\boldsymbol{\pi}_{s|s}})^{(j)})^{x_{s+1}^{(j)}}}{x_{s+1}^{(j)}!}. \tag{B.4}$$

Dividing (B.3) by (B.4) gives:

$$\sum_{\mathbf{Z}\in\mathscr{T}_{m,n}(\mathbf{x}_s,\mathbf{x}_{s+1})}\prod_{i=1}^{m}x_{s+1}^{(j)}!\prod_{j=1}^{m}\left(\frac{\pi_{s|s}^{(i)}k_{s+1,\boldsymbol{\pi}_{s|s}}^{(i,j)}}{(\boldsymbol{\pi}_{s|s}^{\mathrm{T}}\mathbf{K}_{s+1,\boldsymbol{\pi}_{s|s}})^{(j)}}\right)^{z^{(i,j)}}\frac{1}{z^{(i,j)}!}.$$

Re-writing this sum with the change of variable $\widetilde{\mathbf{Z}} := \mathbf{Z}^{\mathrm{T}}$ and interchanging i and j yields the result. ■

Proof of lemma 6.6 The result can be proved by induction. The induction is initialized using the fact that $\mu_{t|t}(\cdot)$ is by definition the probability mass function associated with $\mathrm{Mult}(n,\boldsymbol{\pi}_{t|t})$, and then proceeds by combining the result of lemma 6.5 with moment generating function techniques similar to those used in the proof of lemma 6.1. The details are omitted to avoid repetition. ■

B.2.4 Proofs of lemma 6.7 and lemma 6.6

Proof of lemma 6.7 For the numerator in (6.20) is

$$\begin{aligned}&\overline{\mu}_{s|s}(\mathbf{Z}_s)\overline{M}_{s+1}(\mathbf{Z}_s,\boldsymbol{\pi}_{s|s},\mathbf{Z}_{s+1})\\ &\quad=\left(n!\prod_{i,j=1}^{m}\frac{(p_{s|s}^{(i,j)})^{z_s^{(i,j)}}}{z_s^{(i,j)}!}\right)\mathbb{I}[\mathbf{1}_m^{\mathrm{T}}\mathbf{Z}_s = (\mathbf{Z}_{s+1}\mathbf{1}_m)^{\mathrm{T}}]\\ &\quad\times\left(\prod_{j=1}^{m}(\mathbf{Z}_{s+1}\mathbf{1}_m)^{(j)}!\prod_{\ell=1}^{m}\frac{\left(k_{s+1,\boldsymbol{\pi}_{s|s}}^{(j,\ell)}\right)^{z_{s+1}^{(j,\ell)}}}{z_{s+1}^{(j,\ell)}!}\right),\end{aligned} \tag{B.5}$$

and for the denominator in (6.20),

$$\begin{aligned}&\sum_{\mathbf{Z}_s}\overline{\mu}_{s|s}(\mathbf{Z}_s)\overline{M}_{s+1}(\mathbf{Z}_s,\boldsymbol{\pi}_{s|s},\mathbf{Z}_{s+1})\\ &\quad=\left(n!\prod_{j=1}^{m}\frac{(\pi_{s|s}^{(j)})^{(\mathbf{Z}_{s+1}\mathbf{1}_m)^{(j)}}}{(\mathbf{Z}_{s+1}\mathbf{1}_m)^{(j)}!}\right)\left(\prod_{j=1}^{m}(\mathbf{Z}_{s+1}\mathbf{1}_m)^{(j)}!\prod_{\ell=1}^{m}\frac{\left(k_{s+1,\boldsymbol{\pi}_{s|s}}^{(j,\ell)}\right)^{z_{s+1}^{(j,\ell)}}}{z_{s+1}^{(j,\ell)}!}\right),\end{aligned} \tag{B.6}$$

where the equality in (B.6) holds by combining (B.5) with the fact that the marginal of $\overline{\mu}_{s|s}$ over $\mathbf{1}_m^{\mathrm{T}}\mathbf{Z}_s$ is $\mathrm{Mult}(n, \mathbf{1}_m^{\mathrm{T}}\mathbf{P}_{s|s}) = \mathrm{Mult}(n, \boldsymbol{\pi}_{s|s})$.

Dividing (B.5) by (B.6) results in

$$\mathbb{I}[\mathbf{1}_m^{\mathrm{T}}\mathbf{Z}_s = (\mathbf{Z}_{s+1}\mathbf{1}_m)^{\mathrm{T}}] \prod_{j=1}^{m} (\mathbf{Z}_{s+1}\mathbf{1}_m)^{(j)}! \prod_{i=1}^{m} \left(\frac{p_{s|s}^{(i,j)}}{\pi_{s|s}^{(j)}} \right)^{z_s^{(i,j)}!} \frac{1}{z_s^{(i,j)}!}, \tag{B.7}$$

where $p_{s|s}^{(i,j)}$ are the elements of $\mathbf{P}_{s|s}$, $\overline{\mathbf{L}}_s$ is the matrix with elements $\overline{l}_s^{(i,j)} = p_{s|s}^{(j,i)}/\pi_{s|s}^{(i)}$. ■

Proof of lemma 6.6 The proof is by induction for $s = t, t-1, \ldots$, initialized using the fact that $\overline{\mu}_{t|t}(\cdot)$ is defined to be the probability mass function associated with $\mathrm{Mult}(n, \mathbf{P}_{t|t})$, and for the induction step plugging (B.7), which is an explicit expression for the right hand side of (6.20), into (6.21), and using the fact the marginal of $\overline{\mu}_{s+1|t}(\mathbf{Z}_{s+1})$ over $\mathbf{Z}_{s+1}\mathbf{1}_m$ is $\mathrm{Mult}(n, \mathbf{P}_{s+1|t}\mathbf{1}_m)$. ■

B.3 Ebola example: further details and numerical results

This section provides more information about the numerical results from section 6.4.1.

B.3.1 Details of the EM algorithm

In numerical experiments it has been found that a robust approach to approximate maximum likelihood estimation of Θ was to take a profile-likelihood approach using an EM algorithm: 1) choose a grid of values for (β, λ); 2) for each point on this grid, say $(\hat{\beta}, \hat{\lambda})$, run an EM algorithm to approximately maximize $p(\mathbf{Y}_{1:t}|\hat{\beta}, \hat{\lambda}, \rho, \gamma, q^{(2,3)}, q^{(3,4)})$ with respect to $(\rho, \gamma, q^{(2,3)}, q^{(3,4)})$ then evaluate the marginal likelihood at the resulting parameter values using algorithm 8; 3) maximize over the grid.

The EM component of this procedure follows the usual steps for a hidden Markov model [Cappé et al., 2006], so it is just provided an outline. One step of the EM procedure is as follows: given Θ one performs forward filtering using algorithm 8 then backward smoothing using algorithm 10 resulting in $(\mathbf{P}_{t|T})_{t \le T}$. The expected complete data log-likelihood is then maximized with respect to the parameters of interest. It turns out that for the Ebola model, the maximization steps for $(\rho, \gamma, q^{(2,3)}, q^{(3,4)})$ have closed-form solutions, leading to the update equations:

$$\rho \leftarrow \log\left(1 + \frac{\sum_{t=1}^{T} p_{t|T}^{(2,3)}}{\sum_{t=1}^{T} p_{t|T}^{(2,2)}}\right), \qquad \gamma \leftarrow \log\left(1 + \frac{\sum_{t=1}^{T} p_{t|T}^{(3,4)}}{\sum_{t=1}^{T} p_{t|T}^{(3,4)}}\right),$$

$$q^{(2,3)} \leftarrow 1 \wedge \frac{\sum_{t=1}^{T} y_s^{(2,3)}/n}{p_{t|T}^{(2,3)}}, \qquad q^{(3,4)} \leftarrow 1 \wedge \frac{\sum_{t=1}^{T} y_s^{(3,4)}/n}{p_{t|T}^{(3,4)}},$$

where $p_{t|T}^{(i,j)}$ are the elements of $\mathbf{P}_{t|T}$.

B.3.2 Details of the MCMC algorithm

A Metropolis-within-Gibbs MCMC is implemented. The algorithm is targeting the approximate posterior distribution $\widehat{p}(\Theta|\mathbf{Y}_{1:t}) \propto \widehat{p}(\mathbf{Y}_{1:t}|\Theta)p(\Theta)$, where $\widehat{p}(\mathbf{Y}_{1:t}|\Theta)$ is the approximate marginal likelihood computed using algorithm 8, and

$$p(\Theta) = p(\beta)p(\lambda)p(\rho)p(\gamma)p(q^{(2,3)})p(q^{(3,4)}).$$

Three sets of Gamma prior distributions over $\beta, \lambda, \rho, \gamma$ are considered, precisely these are the ones specified in section 3.3 of Lekone and Finkenstädt [2006] and referred to as 'vague', 'informative' and 'non-centered'. The priors $p(q^{(2,3)}), p(q^{(3,4)}))$ were taken to be uniform densities on $[0,1]$.

Gaussian random walk proposals were applied to each parameter, with variances manually tuned to give acceptance rates between 20% and 40% [Roberts et al., 2001].

B.4 COVID-19 example: further details and numerical results

This section provides additional information about the COVID-19 experiment treated in subsection 6.4.3.

B.4.1 Data and Parameter settings

The series $(y_t^{(3,4)})_{t\geq 0}$ and $(y_t^{(7,8)})_{t\geq 0}$, i.e. the reported numbers of new infectives in Wuhan and internationally, constitute two of the three data sets considered for inference by Kucharski et al. [2020]. They additionally considered a third data set consisting of information about prevalence of infections on evacuation flights. This prevalence data are excluded from the analysis, since the structure of the observation model required fall outside the class of models considered in this thesis.

f_t, n, ρ, γ are set to the same values used in Kucharski et al. [2020], including the fact that f_t is set to zero after the date when travel restrictions were introduced. $q^{(W)} = 0.00175, q^{(T)} = 0.8$ are estimated via approximate maximum likelihood over a grid and $h = 1$.

B.4.2 Implementation

The implementation of this experiment is base on the R code accompanying Kucharski et al. [2020], which is available at `https://github.com/adamkucharski/2020-ncov/`. The experiments reported in Kucharski et al. [2020] are re-ran using their method, but excluding the evacuation flight data mentioned in subsection 6.4.3. This allows for like-for-like comparisons of the proposed method results with theirs.

B.4.3 Inference and results

Algorithm 11 is based directly on the Sequential Monte Carlo algorithm of Kucharski et al. [2020], incorporates the proposed discrete-time stochastic model instead of their ODE model. The first stage consists of a particle filter where for time step t, the ith of n_{part} particles consists of $\beta_t^{(i)}$ and $\mathbf{P}_{t|t}^{(i)}$, and the unnormalized importance weight $w_t^{(i)}$ is computed similarly to algorithm 8. The second stage samples from the smoothing distribution of β_t by tracing back the ancestors of a selected particle - see [Andrieu et al., 2010] for details of the role of ancestors in resampling. In addition backward steps as in algorithm 10 compute the corresponding smoothing distribution over $\mathbf{Z}_t$, for $t = T, \ldots, 1$.

The effective sample size for algorithm 11 and the SMC method of Kucharski et al. [2020] applied to the same data are reported in figure B.2.

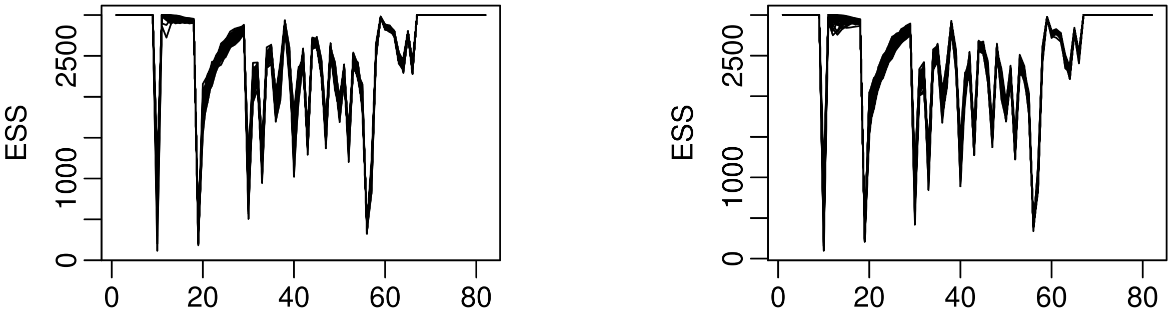


Figure B.2: Effective sample size for algorithm 11 on the left, and for the particle filter of Kucharski et al. [2020] on the right. Remark the evacuation flights data from Kucharski et al. [2020] are excluded from the latter to make a fair comparison with the model proposed in this thesis.

BIBLIOGRAPHY

L. Aitchison. Bayesian filtering unifies adaptive and non-adaptive neural network optimization methods. *arXiv preprint arXiv:1807.07540*, 2018.

L. Aitchison, A. Pouget, and P. E. Latham. Probabilistic synapses. *arXiv preprint arXiv:1410.1029*, 2014.

C. Andrieu, A. Doucet, and R. Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

A. Basharat and M. Shah. Time series prediction by chaotic modeling of nonlinear dynamical systems. In *2009 IEEE 12th international conference on computer vision*, pages 1941–1948. IEEE, 2009.

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.

L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

Y. Bengio, R. Cardin, R. De Mori, and Y. Normandin. A hybrid coder for hidden markov models using a recurrent neural networks. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 537–540. IEEE, 1990.

Y. Bengio, R. De Mori, G. Flammia, and R. Kompe. Global optimization of a neural network-hidden markov model hybrid. In *IJCNN-91-Seattle International Joint Conference on Neural Networks*, volume 2, pages 789–794. IEEE, 1991.

T. Bengtsson, P. Bickel, B. Li, et al. Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In *Probability and statistics: Essays in honor of David A. Freedman*, pages 316–334. Institute of Mathematical Statistics, 2008.

E. Besada-Portas, S. M. Plis, M. Jesus, and T. Lane. Parallel subspace sampling for particle filtering in dynamic bayesian networks. In *Machine Learning and Knowledge Discovery in Databases*, pages 131–146. Springer, 2009.

P. Bickel, B. Li, T. Bengtsson, et al. Sharp failure rates for the bootstrap particle filter in high dimensions. In *Pushing the limits of contemporary statistics: Contributions in honor of Jayanta K. Ghosh*, pages 318–329. Institute of Mathematical Statistics, 2008.

P. Bickel, D. Choi, X. Chang, and H. Zhang. Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. *The Annals of Statistics*, 41(4):1922–1943, 2013.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

B. Boots, G. J. Gordon, and S. M. Siddiqi. A constraint generation approach to learning stable linear dynamical systems. In *Advances in neural information processing systems*, pages 1329–1336, 2008.

C. B. Bose and S.-S. Kuo. Connected and degraded text recognition using hidden Markov model. *Pattern Recognition*, 27(10):1345–1363, 1994.

X. Boyen and D. Koller. Tractable inference for complex stochastic processes. In *Proc. 14th Conf. Uncertainty Artif. Intell.*, pages 33–42. Morgan Kaufmann Publishers Inc., 1998.

X. Boyen and D. Koller. Exploiting the architecture of dynamic systems. In *Proc. 16th Nat. Conf. Artif. Intell.*, pages 313–320, 1999.

B. C. Brandao, J. Wainer, and S. K. Goldenstein. Subspace hierarchical particle filter. In *Proc. 19th Brazilian Symp. Comput. Graph. Process*, pages 194–204. IEEE, 2006.

F. Brauer. *Compartmental models in epidemiology*. Springer, 2008.

C. Bretó. Modeling and inference for infectious disease dynamics: a likelihood-based approach. *Statistical Science: a review journal of the Institute of Mathematical Statistics*, 33(1):57–69, 2018.

M. Briers, A. Doucet, and S. Maskell. Smoothing algorithms for state–space models. *Annals of the Institute of Statistical Mathematics*, 62(1):61, 2010.

G. D. Brown, J. J. Oleson, and A. T. Porter. An empirically adjusted approach to reproductive number estimation for stochastic compartmental models: A case study of two Ebola outbreaks. *Biometrics*, 72(2):335–343, 2016.

G. D. Brown, A. T. Porter, J. J. Oleson, and J. A. Hinman. Approximate Bayesian Computation for spatial SEIR(S) epidemic models. *Spatial and Spatio-temporal Epidemiology*, 24:27–37, 2018.

O. Cappé, E. Moulines, and T. Rydén. *Inference in hidden Markov models*. Springer Science & Business Media, 2006.

A. Celisse, J.-J. Daudin, and L. Pierre. Consistency of maximum-likelihood and variational estimators in the stochastic block model. *Electronic Journal of Statistics*, 6:1847–1899, 2012.

A. B. Chan and N. Vasconcelos. Classifying video with kernel dynamic textures. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6. IEEE, 2007.

G. Chowell, N. W. Hengartner, C. Castillo-Chavez, P. W. Fenimore, and J. M. Hyman. The basic reproductive number of Ebola and the effects of public health measures: the cases of Congo and Uganda. *Journal of Theoretical Biology*, 229(1):119–126, 2004.

B.-E. Chérief-Abdellatif. Convergence rates of variational inference in sparse deep learning, 2019.

V. Colizza, A. Barrat, M. Barthélemy, and A. Vespignani. The role of the airline transportation network in the prediction and predictability of global epidemics. *Proceedings of the National Academy of Sciences*, 103(7):2015–2020, 2006.

S. de Lusignan, J. L. Bernal, M. Zambon, O. Akinyemi, G. Amirthalingam, N. Andrews, R. Borrow, R. Byford, A. Charlett, G. Dabrera, et al. Emergence of a novel coronavirus (COVID-19): protocol for extending surveillance used by the Royal College of general practitioners research and surveillance centre and public health England. *JMIR Public Health and Surveillance*, 6 (2):e18606, 2020.

P. Del Moral, A. Doucet, and S. Singh. Forward smoothing using sequential monte carlo. *arXiv preprint arXiv:1012.5390*, 2010.

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.

A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

M. Farach-Colton and M.-T. Tsai. Exact sublinear binomial sampling. *Algorithmica*, 73(4): 637–651, 2015.

M. Fasiolo, N. Pya, and S. N. Wood. A comparison of inferential methods for highly nonlinear state space models in ecology and epidemiology. *Statistical Science*, 31(1):96–118, 2016.

L. A. Feldkamp, D. V. Prokhorov, and T. M. Feldkamp. Simple and conditioned adaptive behavior from kalman filter trained recurrent networks. *Neural Networks*, 16(5-6):683–689, 2003.

A. Finke and S. S. Singh. Approximate smoothing and parameter estimation in high-dimensional state-space models. *IEEE Transactions on Signal Processing*, 65(22):5982–5994, 2017.

M. Franzini, K.-F. Lee, and A. Waibel. Connectionist viterbi training: a new hybrid method for continuous speech recognition. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 425–428. IEEE, 1990.

S. Funk and A. A. King. Choices and trade-offs in inference with infectious disease models. *Epidemics*, 30:100383, 2020.

Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

A. Gelman, X.-L. Meng, and H. Stern. Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica*, 6(4):733–760, 1996.

H.-O. Georgii. *Gibbs measures and phase transitions*, volume 9. Walter de Gruyter, 2011.

Z. Ghahramani and M. I. Jordan. Factorial hidden Markov models. *Machine Learning*, 29(2): 245–273, Nov 1997. ISSN 1573-0565. doi: 10.1023/A:1007425814087. URL `https://doi.org/10.1023/A:1007425814087`.

X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.

S. Godsill and T. Clapp. Improvement strategies for monte carlo particle filters. In *Sequential Monte Carlo methods in practice*, pages 139–158. Springer, 2001.

S. J. Godsill, A. Doucet, and M. West. Monte carlo smoothing for nonlinear time series. *Journal of the american statistical association*, 99(465):156–168, 2004.

I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

A. Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356, 2011.

R. Guimera, S. Mossa, A. Turtschi, and L. N. Amaral. The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. *Proceedings of the National Academy of Sciences*, 102(22):7794–7799, 2005.

R. Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992.

A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen. Learning the dynamics of arterial traffic from probe data using a dynamic Bayesian network. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1679–1693, 2012.

J. P. Hughes, P. Guttorp, and S. P. Charles. A non-homogeneous hidden Markov model for precipitation occurrence. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 48(1):15–30, 1999.

D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2013. *URL https://arxiv.org/abs/1312.6114*, 2013.

D. P. Kingma, T. Salimans, and M. Welling. Variational dropout and the local reparameterization trick. In *Advances in neural information processing systems*, pages 2575–2583, 2015.

J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

G. Kitagawa. Non-gaussian state-space modeling of nonstationary time series. *Journal of the American statistical association*, 82(400):1032–1041, 1987.

A. Krogh and S. K. Riis. Hidden neural networks. *Neural Computation*, 11(2):541–563, 1999.

A. Krogh, B. Larsson, G. Von Heijne, and E. L. Sonnhammer. Predicting transmembrane protein topology with a hidden markov model: application to complete genomes. *Journal of molecular biology*, 305(3):567–580, 2001.

A. J. Kucharski, T. W. Russell, C. Diamond, Y. Liu, J. Edmunds, S. Funk, R. M. Eggo, F. Sun, M. Jit, J. D. Munday, et al. Early dynamics of transmission and control of COVID-19: a mathematical modelling study. *The Lancet Infectious Diseases*, 20(5):553–558, 2020.

R. Kurle, B. Cseke, A. Klushyn, P. van der Smagt, and S. Günnemann. Continual learning with bayesian neural networks for non-stationary data. In *International Conference on Learning Representations*, 2019.

T. G. Kurtz. Solutions of ordinary differential equations as limits of pure jump Markov processes. *Journal of Applied Probability*, 7(1):49–58, 1970.

T. G. Kurtz. Limit theorems for sequences of jump Markov processes approximating ordinary differential processes. *Journal of Applied Probability*, 8(2):344–356, 1971.

T. Kypraios, P. Neal, and D. Prangle. A tutorial introduction to Bayesian inference for stochastic epidemic models using Approximate Bayesian Computation. *Mathematical Biosciences*, 287: 42–53, 2017.

Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

P. E. Lekone and B. F. Finkenstädt. Statistical inference in a stochastic epidemic SEIR model with control intervention: Ebola as a case study. *Biometrics*, 62(4):1170–1177, 2006.

M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6): 861–867, 1993.

C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

E. Marinari and G. Parisi. Simulated tempering: a new monte carlo scheme. *EPL (Europhysics Letters)*, 19(6):451, 1992.

M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

T. J. McKinley, I. Vernon, I. Andrianakis, N. McCreesh, J. E. Oakley, R. N. Nsubuga, M. Goldstein, and R. G. White. Approximate Bayesian computation and Simulation-Based Inference for Complex Stochastic Epidemic Models. *Statistical Science*, 33(1):4–18, 2018.

A. Mobiny, H. V. Nguyen, S. Moulik, N. Garg, and C. C. Wu. Dropconnect is effective in modeling uncertainty of bayesian deep networks. *arXiv preprint arXiv:1906.04569*, 2019.

L. Murray, D. Lundén, J. Kudlicka, D. Broman, and T. Schön. Delayed sampling and automatic Rao-Blackwellization of probabilistic programs. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2018.

V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

M. Newman, A.-L. Barabasi, and D. J. Watts. *The structure and dynamics of networks*, volume 19. Princeton University Press, 2011.

B. Ng, L. Peshkin, and A. Pfeffer. Factored particles for scalable monitoring. In *Proc. 18th Conf. Uncertainty Artif. Intell.*, pages 370–377. Morgan Kaufmann Publishers Inc., 2002.

C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner. Variational continual learning. *arXiv preprint arXiv:1710.10628*, 2017.

Y. Ollivier et al. Online natural gradient as a kalman filter. *Electronic Journal of Statistics*, 12 (2):2930–2961, 2018.

P. D. O'Neill. Introduction and snapshot review: relating infectious disease transmission models to data. *Statistics in Medicine*, 29(20):2069–2077, 2010.

M. Opper and C. Archambeau. The variational gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009.

K. Prem, Y. Liu, T. W. Russell, A. J. Kucharski, R. M. Eggo, N. Davies, et al. The effect of control strategies to reduce social mixing on outcomes of the COVID-19 epidemic in Wuhan, China: a modelling study. *The Lancet Public Health*, 2020.

G. V. Puskorius and L. A. Feldkamp. Decoupled extended kalman filter training of feedforward layered networks. In *IJCNN-91-Seattle International Joint Conference on Neural Networks*, volume 1, pages 771–777. IEEE, 1991.

G. V. Puskorius and L. A. Feldkamp. Neurocontrol of nonlinear dynamical systems with kalman filter trained recurrent networks. *IEEE Transactions on neural networks*, 5(2):279–297, 1994.

G. V. Puskorius and L. A. Feldkamp. Parameter-based kalman filter training: Theory and implementation. *Kalman filtering and neural networks*, page 23, 2001.

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

L. R. Rabiner and B.-H. Juang. An introduction to hidden markov models. *ieee assp magazine*, 3 (1):4–16, 1986.

P. Ramachandran, B. Zoph, and Q. V. Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

R. Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

P. Rebeschini and R. van Handel. Comparison theorems for gibbs measures. *Journal of Statistical Physics*, 157(2):234–281, 2014.

P. Rebeschini and R. Van Handel. Can local particle filters beat the curse of dimensionality? *The Annals of Applied Probability*, 25(5):2809–2866, 2015.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

L. Rimella and N. Whiteley. Exploiting locality in high-dimensional factorial hidden Markov models. *arXiv preprint arXiv:1902.01639*, 2019.

L. Rimella and N. Whiteley. Dynamic Bayesian Neural Networks. *arXiv preprint arXiv:2004.06963*, 2020.

H. Ritter, A. Botev, and D. Barber. Online structured laplace approximations for overcoming catastrophic forgetting. In *Advances in Neural Information Processing Systems*, pages 3738–3748, 2018.

G. O. Roberts, J. S. Rosenthal, et al. Optimal scaling for various Metropolis-Hastings algorithms. *Statistical science*, 16(4):351–367, 2001.

S. Rubrichi, Z. Smoreda, and M. Musolesi. A comparison of spatial-based targeted disease mitigation strategies using mobile phone data. *EPJ Data Science*, 7(1):1–15, 2018.

S. Shah, F. Palmieri, and M. Datum. Optimal filtering algorithms for fast learning in feedforward neural networks. *Neural networks*, 5(5):779–787, 1992.

R. Silva, S. M. Kang, and E. M. Airoldi. Predicting traffic volumes and estimating the effects of shocks in massive transportation systems. *Proceedings of the National Academy of Sciences*, page 201412908, 2015.

C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640, 2008.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15 (1):1929–1958, 2014.

M. Stanke and S. Waack. Gene prediction with a hidden Markov model and a new intron submodel. *Bioinformatics*, 19(suppl_2):ii215–ii225, 2003.

T. Stocks. Iterated filtering methods for Markov process epidemic models. In L. Held, N. Hens, P. D O'Neill, and J. Wallinga, editors, *Handbook of Infectious Disease Data Analysis*, chapter 11, pages 199–220. CRC Press, 2019.

Transport for London. Oyster card journey information, 2018. `https://api-portal.tfl.gov.uk/docs`, section: "Oyster card data".

R. van Handel. Lecture notes: Hidden markov models, July 2008.

A. Venkatraman, M. Hebert, and J. A. Bagnell. Improving multi-step prediction of learned time series models. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066, 2013.

B. Wang and D. Titterington. Convergence properties of a general algorithm for calculating variational bayesian estimates for a normal mixture model. *Bayesian Analysis*, 1(3):625–650, 2006.

G. Welch, G. Bishop, et al. An introduction to the kalman filter. 1995.

N. Whiteley and L. Rimella. Inference in stochastic epidemic models via multinomial approximations. In *International Conference on Artificial Intelligence and Statistics*, pages 1297–1305. PMLR, 2021.

D. Woodard, G. Nogin, P. Koch, D. Racz, M. Goldszmidt, and E. Horvitz. Predicting travel time reliability using mobile phone GPS data. *Transportation Research Part C: Emerging Technologies*, 75:30–44, 2017.

J. T. Wu, K. Leung, and G. M. Leung. Nowcasting and forecasting the potential domestic and international spread of the 2019-nCoV outbreak originating in Wuhan, China: a modelling study. *The Lancet*, 395(10225):689–697, 2020.

Y. Yang, D. Pati, and A. Bhattacharya. α-variational inference with statistical guarantees. *arXiv preprint arXiv:1710.03266*, 2017.