ECOLE
DOCTORALE
DE MATHEMATIQUES
HADAMARD

HABILITATION À DIRIGER DES RECHERCHES

Discipline : Mathématiques Appliquées (Statistiques)

présentée par

Yohan PETETIN

Generative models for time series data

Soutenue le 02/02/23 devant le jury composé de :

Pr. Randal DOUC	Telecom SudParis	Rapporteur interne
Pr. Petar DJURIC	Stony Brook University	Rapporteur
Pr. Simon GODSILL	Cambridge University	Rapporteur
Pr. Florence FORBES	Inria Grenoble Rhône-Alpes	Examinateur
Pr. Gersende FORT	Institut de Mathématiques de Toulouse	Examinateur
Pr. Arnaud GUILLIN	Laboratoire de Mathématiques Blaise Pascal	Examinateur
Pr. François SEPTIER	Université Bretagne Sud	Examinateur

Contents

Avant propos

Après quelques longs mois d'hésitation, me voici enfin en mesure de présenter mon manuscrit d'habilitation à diriger des recherches. J'avais initialement prévu de m'attaquer à sa rédaction au cours de l'année 2020. Cependant, deux évènements de nature bien différentes sont venus contrarier mes ambitions. D'une part, j'ai eu le bonheur d'accueillir mes deux petits garçons, Rafael et Simon, à la fin de l'année 2019. Dans la foulée, le confinement a fait son apparition dans notre quotidien pour une période indeterminée. Alors que pour beaucoup de collègues cette période a été propice à l'avancement de leurs projets, je dois avouer que l'énergie consacrée à mes deux nouveaux locataires, le bonheur que j'en ai retiré, et l'urgence d'adapter mes enseignements à la situation ont très vite limité mon enthousiasme quant à initier une quelconque rédaction et avancer sur ce projet qui me tient particulièrement à coeur. Essayant d'exploiter la situation tant bien que mal, j'ai décidé de profiter du temps qu'il me restait pour mettre sur pieds de nouveaux projets de recherche. Finalement, la concrétisation de ces projets de recherche m'a permis de donner une dimension nouvelle à ce manuscrit, que je n'avais pas prévue au départ. Certaines contributions post-confinement seront en effet présentées tout au long de ce manuscrit.

Lorsque l'on arrive à la concrétisation d'un projet quelconque, ici la rédaction de ce manuscrit, il convient d'admettre que ce dernier n'aurait pu aboutir sans le soutien des nombreuses personnes qui nous entourent au quotidien. Je souhaite donc adresser des remerciements à toutes les personnes qui ont contribué, de près ou de loin, à l'aboutissement de ce manuscrit d'HDR.

Bien évidemment, c'est un grand honneur pour moi que d'être lu par les trois rapporteurs que sont Simon Godsill, Petar Djuric et Randal Douc. Je les remercie infiniment de consacrer du temps à ce manuscrit, et j'espère ne pas avoir fait souffrir Randal Douc qui s'est attaqué à une première lecture en plein été caniculaire.

Je souhaite remercier mes collègues du département Communications, Images et Traitement de l'information (CITI) de Télécom SudParis avec lesquels je partage une partie mon quotidien. Ce dernier n'en est que plus agréable grâce à la bonne humeur de tous. Je remercie particulièrement Daniel Clark qui m'a relancé et encouragé à plusieurs reprises pour entamer la rédaction de mon manuscrit. À peine est t-il terminé que j'apprends que tu nous quittes pour un poste de Professeur au Royaume Uni; nous commençons néanmoins conjointement un projet de recherche des plus passionnants visant à me reconcilier avec mes premiers amours, le pistage multi-cibles, multi-capteurs. J'en profite également pour remercier Julie, qui nous quitte au moment où ce manuscrit entre dans sa dernière ligne droite; ton aide préciseuse du quotidien risque de manquer à tous.

Toujours dans le domaine académique, je souhaite remercier mes thésards de ces dernières années qui, chacun à leur façon, m'ont fait progresser lors de nos discussions, rédactions d'articles ou rédaction de leur manuscrit. Les résultats présentés dans cette synthèse sont avant tout le fruit d'un travail commun avec Jana,

Roland, Nicolas, Achille, Yazid et Katherine.

Enfin, il est temps de remercier tout ceux dont la présence suffit à me motiver dans mon activité de chercheur au quotidien: mes élèves et amis du club de Kick-Boxing de Ris-Orangis, mon frère et mes parents, et par dessus tout, Rafael et Simon. Je ne puis dire si ce sont leurs gribouillis sur mes brouillons de calcul qui m'auront permis de débloquer certaines situations, mais je suis certain que le bonheur qu'ils m'apportent quotidiennement est un catalyseur pour mes différentes activités. Je leur dédie donc ce manuscrit, en espérant qu'ils soient capables d'en comprendre le contenu dans les années à venir.

Venons en maintenant au contenu scientifique de ce manuscript. Il s'agit bien évidemment d'une synthèse de mes activités de recherche depuis ma soutenance de thèse de doctorat en 2013. Les résultats théoriques et expérimentaux, ainsi que les preuves des résultats mathématiques sont détaillés dans les articles de recherche dont une liste sélective est présentée dans un court chapitré dédié. J'ai souhaité faire refléter ma façon de réfléchir à travers la rédaction de ce manuscrit. Ma plus grande motivation dans le métier que j'exerce est avant tout de comprendre et réussir à expliquer un problème ou un résultat potentiellement compliqué de la manière la plus intuitive possible. Je considère qu'un résultat ou un algorithme n'est compris qu'à partir du moment où la solution qu'ils décrivent peut être expliquée de manière intuitive. Par conséquent, et comme nous le verrons tout au long du manuscrit, une partie de ma méthodologie consiste à repartir de solutions très populaires pour un problème donné, à chercher à les réinterpréter et voir s'il n'est pas possible d'y inclure des modifications pour améliorer et proposer des solutions originales. Il s'agit de l'esprit général de ce manuscrit, dans lequel j'essaierai d'abord d'expliquer de la façon la plus simple les problèmes auxquels je me suis attaqué ces dernières années, puis je synthétiserai mes différentes contributions à partir de ces conclusions préliminaires.

Il me reste maintenant à résumer le contenu de ce manuscrit dont le fil rouge consiste à présenter et simplifier un problème statistique lié à modélisation d'un processus stochastique observé, à décrire ses solutions puis à les remettre en question pour tenter de s'affranchir de leurs limitations.

Le chapitre 1 est une introduction aux concepts mathématiques qui seront utilisés tout au long de cette présentation. Je m'attache à introduire les modèles probabilistes qui sont au coeur de mes recherches et qui reposent sur des variables latentes ou cachées. Plus précisément, je m'interesse aux modèles de Markov cachés et aux architectures neuronales récurrentes pour le traitement des séries temporelles. On trouvera donc dans ce chapitre une vue d'ensemble de ces modèles et des traitements associés qui me permettra, au fur et à mesure, de soulever certaines questions en rapport avec la logique de leur construction, leur estimation et les problèmes de prédiction associés.

Le chapitre 2 est consacré aux algorithmes de filtrage particulaire (ou méthodes de Monte Carlo séquentielles) dans des modèles de chaines de Markov cachées. Ces algorithmes reposent principalement sur un mécanisme de base composé de trois opérations élémentaires : simulations de variables aléatoires selon une loi d'importance; pondération des échantillons générés; et enfin, réechantillonnage d'un sous ensemble d'échantillons. Ma contribution consiste à revisiter la logique de ce mécanisme dans deux directions. Dans un premier temps, je pars du fait que ce mécanisme peut être utilisé pour estimer un ratio d'espérance mathématique; auquel cas, un ensemble commun d'échantillons est utilisé pour approcher conjointement le numérateur et le dénominateur. J'exploite donc l'idée d'introduire deux lois d'importance différentes pour l'approximation d'un ratio, et je discute de la construction de ces lois et des méthodes d'échantillonnage associées. Dans un deuxième temps, je m'intéresse à la troisième étape de rééchantillonnage de ce mécanisme. Celle ci s'avère indispensable lorsque le mécanisme est appliqué de manière séquentielle, comme dans

le filtrage particulaire, en terme de stabilité. Néanmoins, le prix à payer pour cette stabilité est un appauvrissement local de l'approximation Monte Carlo qui résulte de cette opération. Je cherche donc à mesurer l'impact stastistique et computationnel d'une procédure qui viserait à garder inchangée la distribution des échantillons mais à les rendre indépendant de manière à contourner le problème d'appauvrissement. Lorsque cette procédure est appliquée de manière séquentielle, il est possible de l'interpréter comme un algorithme de filtrage particulaire (auxiliaire) particulier que l'on aurait pu déduire de manière très intuitive s'il nous avait été demandé d'élaborer à un algorithme "optimal" de ce type lorsque la loi de tirage est imposée.

Après avoir passé en revue ces algorithmes, le chapitre 3 est consacré à l'estimation de la variance asymptotique associée aux estimateurs Monte Carlo qu'ils produisent. De manière schématique, ce problème d'estimation peut être vu comme une extension du problème d'estimation de la variance associée à un estimateur empirique. En effet, dans le cas d'un tel estimateur reposant sur la simulation d'échantillons, il est possible de réeutiliser ces échantillons pour estimer, de manière non biasée, sa variance asymptotique. Cette estimateur de la variance peut lui même être réinterprété comme la différence de deux estimateurs non biaisés, celui de la moyenne du carré et du carré de la moyenne. Ces deux quantités peuvent également être vues comme deux espérances mathématiques d'une même fonction mais selon deux lois différentes, en dimension augmenté. Une fois que nous cherchons à généraliser cette méthodologie à des problèmes avec une dimension temporelle, et donc aux algorithmes de filtrage particulaire, le problème du recyclage des variables produites par l'algorithme apparait comme fondamental. La difficulté est double puisqu'elle consiste à proposer un schéma qui garantit de bonnes propriétés statistiques à l'estimateur de la variance asymptotique (convergence, vitesse de convergence) mais qui reste implémentable avec une complexité calculatoire raisonnable. Deux schémas sont donc proposés. Le premier découle d'une réinterprétation des solutions existantes et vise à robustifier les schémas de recyclage de l'état de l'art pour des algorithmes de filtrage particulaire. Le deuxième schéma vise à estimer la variance asymptotique d'estimateurs issus d'algorithmes de lissage particulaire, pour lesquels il n'existe pas, à notre connaissance, de solutions à ce jour.

Le chapitre 4 s'intéresse au problème fondamental du choix d'un modèle probabiliste pour la modélisation des séries temporelles. Comme il s'agit de la thématique générale de ce manuscrit, je me restreins à la comparaison entre modèles de Markov cachés et architectures neuronales récurrentes, qui sont deux modèles connus et exploités dans leur communauté respective. En effet, les modèles de Markov cachés sont particulièrement connus des statisticiens et de la communauté du traitement statistique du signal, tandis que les architectures neuronales font l'objet de nombreuses études dans la communauté de l'apprentissage statistique. L'idée de ce chapitre est de comparer le pouvoir modélisant des deux modèles, d'un point de vue théorique et non expérimental, tout en mettant de côté les algorithmes d'inférence associés. Pour cela, je me focalise sur les différences structurelles des deux modèles et cherche à analyser l'impact de ces différences sur la loi de probabilité d'un processus stochastique observé qu'ils construisent implicitement. Pour réaliser cette comparaison, les deux modèles sont dans un premier temps réinterprétés comme des instances particulières d'un même modèle probabiliste.

Enfin, le chapitre 5 est le fruit d'une fertilisation croisée entre les modèles de Markov cachés et les architectures neuronales récurrentes discutées jusqu'ici. Je montre que la combinaison de ces deux modèles permet de fournir des solutions nouvelles à trois problèmes relatifs à l'estimation bayésienne séquentielle. Dans un premier temps, des modèles de Markov cachés paramétrés par des architectures neuronales et associés à des algorithmes d'estimation de paramètres de type bayésien variationnel permettent de fabriquer des modèles (génératifs) puissants pour la modélisation de séries temporelles. Pour le second problème, de telles combinaisons sont étudiées mais visent à inclure une contrainte d'interprétabilité dans le modèle et son

estimation : contrairement au problème précédent dans lequel les variables latentes ne sont que des intermédiaires de calcul pour complexifier la loi des observations, je cherche maintenant à extraire une information bien précise (et donc interprétable) à partir des observations et à travers les mêmes variables latentes, et ce de manière non supervisée. Enfin, dans le dernier problème, je montre comment cette combinaison de modèles peut être utilisé pour proposer des techniques d'inférence bayésienne dans des modèles probabilistes bien connus tels que le modèle de chaine de Markov cachée linéaire et gaussien à sauts markoviens.

Research and teaching activities

A. Research activity

I start by describing my research activity since I have defended my Ph.D thesis (2010-2013). It was devoted to inference algorithms mainly based on sequential Monte Carlo methods for single and multi-object filtering.

A.1. Post-doctoral and Research engineer at CEA (2013-2015)

In november 2013, I joined the Laboratoire d'Intégration des Systèmes et des Technologies of the French Alternative Energies and Atomic Energy Commission. My research activity was guided in two directions. First, I have worked on assimilation data problems in connection with green energies. More precisely, one of the problem consisted in predicting future solar radiation for the maintenance of photovoltaic power plants. In parallel, I have been asked to co-supervise a Ph.D student, **J. Kalawoun**, who has receveived a grant from the CEA. I co-supervised her between september 2014 and september 2015, for her final year. The objective was to use hidden Markov models with regime switchings to model and evaluate the battery-state-of-charge of electrical vehicles in real time. The second direction was the most interesting since I discovered the Statistical Learning community. Through a common project with a start-up (Invensense), I got familiar with Deep Learning. The project consisted in providing a state of the art of such methods and next evaluating them for audio scene recognition. While I thought that Bayesian inference and Deep Learning were totally different, I realized that "generative" probabilistic models based on latent random variables have led to a renewal of interest of neural networks architectures. Consequently, I was curious to understand if the two points of view (Bayesian inference in hidden Markov model vs. deep learning for recurrent neural architectures) can be conciled in a given sense.

A.2. Associate Professor, Télécom SudParis, 2015-Today

In 2015, I joined Télécom SudParis as an assistant professor. In parallel of my teaching activity that will be described later, I focused my activity on the following topics.

- **Monte Carlo methods**

Of course, I have continued to investigate Monte Carlo methods. A preliminary objective was to scale some Monte Carlo algorithms for big-data. In this context, the spatial and the temporal dimensions of the data can be large and traditional Monte Carlo algorithm may be unreliable. As we will see in this manuscript, the co-supervision of the Ph.D student **R. Lamberti** (2015-2018) has been an opportunity to revise the core mechanism (sampling, weighting and resampling) of sequential Monte Carlo algorithms in two directions.

• Statistical Learning and applications

In parallel, I have further explored statistical learning approaches through the co-supervision of two Ph.D students. These works were led in an applied framework. I have first co-supervised **N. Aussel** (2015-2019) in collaboration with Tryagnosis (now Safran) and we looked for detecting anomaly from machine learning algorithms for flight data. The particularity of these data is that they are unbalanced (typically the class failure is under represented compared to the other class), so classical supervising learning algorithms suffer from poor performances. I have next co-supervised **A. Salaun** (2017-2021) with Nokia Bell-labs and we focussed on predicting failures in telecommunication networks from alarm logs.

• Bridging hidden Markov models and recurrent neural networks architectures

Once I was comfortable with statistical learning approaches and in particular with neural network architectures, I came back to my initial objective of understanding the fundamental differences between hidden Markov models and Recurrent neural networks for sequential data. The first observation was that such architectures are based on (high dimensional) variables (the neurons) and have become the reference in terms of performance for many classification and prediction problems. By contrast, statistical properties of graphical probabilistic models are well understood but associated inference algorithms (parameter estimation, computation of posterior distributions,...) become unreliable when the dimension of the involved (random) variables is large. Some topics related to the general problem of gathering both point of views are addressed in this manuscript and are the results of the co-supervision of three Ph.D. students. With **A. Salaun** (2017-2021), we first described hidden Markov models and recurrent neural networks as a particular case of a more general model and we proposed a characterization of the distribution of observations produced by each model. Currently, I am co-supervising **Y. Janati** (2020-2023) on the mutual contribution of stochastic neural architectures and Monte Carlo methods. The objective is to improve Monte Carlo algorithms through the introduction of such parameterizations (for tuning the importance distribution, for example), and reciprocally to think about Monte Carlo algorithms for Bayesian estimation in this kind of architectures. As we will see, other directions have also been exploited. I am also co-supervising **K. Morales** (2020-2023). We first extended the preliminary comparison between hidden Markov model and recurrent neural architectures of the thesis of **A. Salaun** and we have next proposed powerful hidden Markov models parameterized by neural architectures with associated Bayesian inference algorithms. The next step consists in adapting these tools for decision support in cardiac surgery through a collaboration with the Gepromed, Strasbourg.

A.3. Ph.D students (2014 - Present)

In summary, I have co-supervised 6 Ph.D students between 2014 and 2022.

- Yazid Janati, co-supervised with S. Le-Corff (Télécom SudParis), 2020-2023. *Mutual contributions of Monte-Carlo methods and stochastic neural networks architectures.*
- Katherine Morales, co-supervised with E. Monfrini (Télécom SudParis), 2020-2023. *Artificial intelligence for decision support in vascular surgery.*
- Achille Salaun, co-supervised with F. Desbouvries (Télécom SudParis) and A. Bouillard (Huawei) and M.O. Buob (Nokia), 2017-2021. *Alarms prediction in networks via the research of spatio-temporal patterns and machine learning.*

- Nicolas Aussel, co-supervised with S. Chabridon (Télécom SudParis), 2015-2019. *Real-time anomaly detection with insight data: streaming anomaly detection with heterogeneous communicating agents.*
- Roland Lamberti, co-supervised with F. Desbouvries (Télécom SudParis) and F. Septier (Université Bretagne Sud), 2015-2018. *Contributions to Monte Carlo methods and their application to statistical filtering.*
- Jana Kalawoun, co-supervised with P. Pamphile and G. Celeux (Université Paris Sud), 2014-2015. *Statistical modeling of battery-state-of-charge.*

A.4. Selected list of publications (journals and international conferences)

I finally present a selected list of publications in which more information can be found about my previous research activity and in relation with the work described in this manuscript.

[19] H. Gangloff, K. Morales, Y. Petetin, "Generalized Pairwise and Triplet Markov Chains: a deep extension for unsupervised signal processing estimation. In *hal-03584314.*

[18] Y. Janati, S. Le Corff, Y. Petetin, "'Consistent estimation of the asymptotic variance of Sequential Monte Carlo smoothers"'. In *arxiv:2204.01401.*

[17] F. Desbouvries, Y. Petetin and A. Salaun, "Expressivity of Hidden Markov models vs. Recurrent neural networks from a systems theory viewpoint". In *arXiv:2208.08175.*

[16] H. Gangloff, K. Morales, Y. Petetin, "A General parametrization framework for pairwise Markov models : an application to unsupervised image segmentation", *2021 IEEE International Workshop on Machine Learning for Signal Processing (MSLSP)*, Gold Coast, Australia, Oct. 2021.

[15] Y. Petetin, Y. Janati and F. Desbouvries, "Structured variational Bayesian inference for Gaussian state-space models with regime switching ", *IEEE Signal Processing Letters*, Volume: 28, Issue: 1, pages 1953-1957, Sept. 2021.

[14] K. Morales, Y. Petetin, "Variational Bayesian inference for pairwise Markov models ", *Proceedings of the 21th IEEE workshop on statistical signal processing (SSP '21)*, Rio De Janeiro, Brazil, pp.251-255, Jul 2021.

[13] A. Salaun, Y. Petetin and F. Desbouvries, "Comparing the modeling powers of RNN and HMM", *ICMLA 2019: 18th International Conference on Machine Learning and Applications*, Boca Raton, FL, United States. pp.1496-1499, Dec 2019.

[12] N. Aussel, Y. Petetin and S. Chabridon, "Improving performances of log mining for anomaly prediction through NLP-based log parsing", *2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Milwaukee, Wisconsin, USA , September 25-28, 2018.

[11] R. Lamberti, Y. Petetin, F. Septier and F. Desbouvries, "A Double Proposal Normalized Importance Sampling Estimator", *Proceedings of the 20th IEEE workshop on statistical signal processing (SSP '18)*, Fribourg, Germany, June 10-13, 2018.

[10] R. Lamberti, Y. Petetin, F. Desbouvries and F. Septier, "Semi-Independent Resampling for Particle filtering", *IEEE Signal Processing Letters*, Volume: 25, Issue: 1, pages 130-134, January 2018.

[9] N. Aussel, S. Jaulin, G. Gandon, Y. Petetin, E. Fazli and S.Chabridon , "Predictive Models of hard drive failures based on operational data ", *Proceedings of the 16th IEEE Conference On Machine Learning and Applications (ICMLA)*, Cancun, Mexico, December 18-21, 2017.

[8] R. Lamberti, Y. Petetin, F. Desbouvries and F. Septier, "*Independent Resampling Sequential Monte Carlo Algorithms*", IEEE Transactions on Signal Processing, Volume: 65, Issue: 20, pages 5318-5333, October 2017.

[7] Y. Petetin and F. Desbouvries, "Bayesian Conditional Monte Carlo Algorithms for non linear time-series state estimation", *IEEE Transactions on Signal Processing*, Volume 63, Issue 14, Pages 3626-3638, July 2015.

[6] Y. Petetin and F. Desbouvries, "A class of fast exact Bayesian filters in dynamical models with jumps", *IEEE Transactions on Signal Processing*, Volume 62, Issue 14, Pages 3643-3653, June 2014.

[5] Y. Petetin, M. Morelande and F. Desbouvries, "Marginalized particle PHD filters for multiple object Bayesian filtering", *IEEE Transactions on Aerospace and Electronic Systems*, Volume 50, Issue 2, Pages 1182-1196, April 2014.

[4] N. Abassi, S. Derrode, F. Desbouvries, Y. Petetin and W. Pieczynski, "Filtrage statistique rapide dans des systèmes linéaires à sauts non stationnaires", *Traitement du Signal*, num. 3-4/2014, 1-23.

[3] Y. Petetin and F. Desbouvries, "Bayesian multi-object filtering for Pairwise Markov Chains", *IEEE Transactions on Signal Processing*, Volume 61, Issue 18, Pages 4481-4490, September 2013.

[2] Y. Petetin and F. Desbouvries, "Optimal SIR algorithm vs. fully adapted auxiliary particle filter: a non asymptotical analysis", *Statistics and computing*, Volume 23, Number 6, Pages 759-775, September 2013.

[1] F. Desbouvries, Y. Petetin and B. Ait-el-Fquih, "Direct, Prediction- and Smoothing-based Kalman and Particle Filter Algorithms", *Signal Processing*, Volume 91, Number 8, Pages 2064-2077, August 2011.

B. Teaching activity

My teaching activity is mainly based in Télécom SudParis, but I also teach in some partner schools of the group *Institut Polytechnique de Paris (IPP)*. For one year, my number of teaching hours is approximately between 190h and 220h. I coordinate 5 courses; in addition to teach basic courses (Introduction to statistics, numerical analysis,...), I have created several courses in relationship with some topics addressed in this manuscript. In particular, I have introduced a course of Deep Learning in Télécom SudParis and extended it in the Master Data Science of IPP. I propose an overview of probabilistic models based on neural networks (Restricted Boltzmann Machines, Deep Belief Networks, Variational-Auto encoders,...). Finally, I am also the coordinator of the advanced research projects for the third year students of Télécom SudParis who choose a specialization in probability and statistics. An example of my teaching activity for the year 2020/2021 is

	Lecture (h)	**Tutorials (h)**	**Practical work (h)**	**Total (h)**
Probability	0	40	9	46
Introduction to statistics (coordinator)	9	16	0	29.5
Numerical analysis (coordinator)	4.5	18	4.5	27.75
Statistical filtering	6	0	12	17
Statistical learning: application to biostatistics (coordinator)	15	9	0	31.5
Deep learning (coordinator)	18	15	0	34
Deep Learning II (IP Paris, coordinator)	10.5	0	0	15.75
Probabilistic models in artificial intelligence	6	0	0	9
Initiation to computational statistics	0.5	0	4	3.4
Total	104	98	13.6	**222**

Table 1: Description of my teaching activity in 2020/2021. 1h of lecture = 1.5h of tutorial. 1h of practical work = 0.66h of tutorial. Moreover, I have also supervised **2** initiation research projects for 2nd year students (February-June) and **2** advanced research projects for 3rd year students (September-January). Finally, I have followed **4** students during their internship in companies.

given in Table 1. The total number of hours is converted in hours of tutorials according to the French university system.

CHAPTER 1

An overview of generative models for time series analysis

This chapter introduces the general tools that we will develop in this manuscript. We focus on probabilistic models based on latent random variables for describing time series. Along this general presentation, I wish to describe the framework of my research; once the general tools have been presented, I introduce the questions which have arisen these last years.

We start by recalling the (static) Bayesian estimation problem. We next adapt it when the involved random variables aim at describing a time series problem. To that end, we introduce some popular probabilistic models based on latent variables which are at the core of this synthesis. We highlight particular aspects of the structure of these models or of their associated Bayesian inference algorithms when they give rise to a fundamental statistical problem. These aspects are briefly discussed before being addressed in the next chapters. We finally present the general organization of this thesis which aims at gathering the contributions into homogeneous chapters.

1.1 A brief review of Bayesian estimation

Let X (resp. Y) be an hidden (resp. observed) random variable in a general measurable space $(\mathsf{X}, \mathcal{X})$ (resp. $(\mathsf{Y}, \mathcal{Y})$). We assume that the pair (X, Y) admits a joint probability distribution function (pdf) $p(x, y)$ w.r.t. a product measure $\nu \otimes \lambda$ on $\mathcal{X} \otimes \mathcal{Y}$. Let $h : x \mapsto h(x) \in \mathbb{R}$ be a given functional. The Bayesian problem aims at "estimating" $h(X)$ from Y. To that end, we introduce a (positive) loss function $L(.,.)$ and we look for minimizing the Bayesian risk defined as

$$R(f) = \mathbb{E}\left(L(f(Y), h(X))\right). \tag{1.1}$$

The Bayesian estimator of $h(X)$ coincides with $\widehat{h(X)} = f^\star(Y)$ where $f^\star = \arg\min_f(R(f))$. For example, if the objective is to estimate a scalar random variable X from a realization $Y = y$ (so $h(x) = x$), the Bayesian estimate which coincides with the quadratic loss $L(u, v) = (u - v)^2$ is

$$\hat{x} = \mathbb{E}(X|Y = y).$$

More generally, the minimization of (1.1) involves the posterior distribution

$$p(x|y) = \frac{p(x, y)}{p(y)} = \frac{p(x, y)}{\int_{\mathsf{X}} p(x, y)\nu(\mathrm{d}x)}.$$

However, the joint distribution $p(x, y)$ is generally unknown and two directions are possible to overcome this limitation.

1.1.1 Modelling the joint distribution

The first solution consists in introducing a parameterized distribution p_θ, where $\theta \in \Theta$ is an unknown multidimensional parameter. p_θ aims at approximating the unknown distribution p in a given sense. Once a family p_θ has been chosen, θ can be estimated from two options according to the available data. If we have at our disposal a set of independent samples

$$\mathcal{E}_1 = \{(x^i, y^i) \overset{\text{i.i.d}}{\sim} p(x, y)\}_{1 \leq i \leq n},$$

then we are in the context of *supervised* estimation; by contrast, if we only have a set of observations

$$\mathcal{E}_2 = \{y^i \overset{\text{i.i.d}}{\sim} p(y)\}_{1 \leq i \leq n}, \tag{1.2}$$

then we are in the context of *unsupervised* estimation. Due to its asymptotic properties, a popular estimator of θ is the Maximum-Likelihood (ML) estimator (Huber, 1967; White, 1982) which aims at maximizing (supervised case)

$$\mathcal{L}_1(.; \mathcal{E}_1) : \theta \mapsto \prod_{i=1}^{n} p_\theta(x^i, y^i),$$

or (unsupervised case)

$$\mathcal{L}_2(.; \mathcal{E}_2) : \theta \mapsto \prod_{i=1}^{n} p_\theta(y^i) \tag{1.3}$$

w.r.t. θ. In this last case, the optimization of (1.3) is not obvious because $p_\theta(y)$ is not necessarily available in a closed-form expression. According to the structure of $p_\theta(x, y)$, the ML estimator can be approximated with a gradient ascent method on (1.3), the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) or a variational Bayesian inference algorithm (Jordan et al., 1999; Blei et al., 2017). These algorithms will be described in the framework of our probabilistic models for time series.

Once θ has been estimated, it remains to compute or approximate the associated posterior distribution $p_\theta(x|y)$. Generally, this can be done by using a Monte-Carlo method such as normalized importance sampling (Hesterberg, 1988). Introducing an importance distribution $q(x)$ from which it is possible to sample N particles $\xi^i \overset{\text{i.i.d}}{\sim} q(x)$, the posterior measure $p(\mathrm{d}x|y)$ can be approximated by the discrete measure

$$\phi^N(\mathrm{d}x) = \sum_{i=1}^{N} \mathcal{W}^i \delta_{\xi^i}(\mathrm{d}x),$$

where

$$\mathcal{W}^i = \Omega^{-1}\omega^i, \quad \omega^i = \frac{p_\theta(\xi^i, y)}{q(\xi^i)} \quad \text{and} \quad \Omega = \sum_{i=1}^{N} \omega^i.$$

An unweighted representation can be obtained by sampling $A^i \overset{\text{i.i.d}}{\sim} \text{Categorical}(\{\mathcal{W}^l\}_{1 \leq l \leq N})$,

$$\tilde{\phi}^N(\mathrm{d}x) = \frac{1}{N} \sum_{i=1}^{N} \delta_{\xi^{A^i}}(\mathrm{d}x).$$

1.1.2 Statistical learning approach

The second approach also relies on the set of i.i.d. samples $\mathcal{E}_1$ but aims at using it for deriving the crude Monte Carlo estimate $R_n(f)$ of $R(f)$ and next minimizing it w.r.t. f. The problem of building an estimator becomes that of minimizing the empirical risk,

$$f_n^\star = \arg\min_f \frac{1}{n}\sum_{i=1}^n L(f(y^i), h(x^i)) = \arg\min_f R_n(f). \tag{1.4}$$

Since the dataset is finite, in the absence of further constraints, any function interpolating the points $(h(x^i), f(y^i))$ satisfies the optimisation problem (1.4). In such a case, the model overfits and proves unable to generalize to new observations. This problem is often overcome by choosing a family of functions $(f_\theta)_{\theta\in\Theta}$, and finally (1.4) turns into the parameter estimation problem

$$\theta_n^\star = \arg\min_\theta \frac{1}{n}\sum_{i=1}^n L(f_\theta(y^i), h(x^i)), \tag{1.5}$$

which eventually produces the estimator $\widehat{h(x)} = f_{\theta_n^\star}(y)$ (notation $\theta_n^\star$ underlines the fact that the estimator depends on the training set $\mathcal{E}_1$, which is of dimension n). The choice of the family $(f_\theta)_{\theta\in\Theta}$ should be balanced: a poor set of functions will lead to unrealistic predictions, while a rich set of functions can lead to overfitting. Moreover $(f_\theta)_{\theta\in\Theta}$ should lead to tractable learning, *i.e.* it should be possible to solve (1.5) efficiently. Classical solutions include the functions belonging to a reproducing kernel Hilbert space (RKHS) (Manton and Amblard, 2015) (Paulsen and Raghupathi, 2016) and the functions defined by neural network architectures (Jain et al., 1996) (LeCun et al., 2015). Optimizing (1.5) for these families of functions leads to well known algorithms such as (linear or kernel based) least squares (Bishop, 2006), Support Vector Machines (SVM) (Burges, 1998) (Hu et al., 2003) (Vapnik, 2013), or deep learning algorithms (Bishop, 2006) (Goodfellow et al., 2016) for regression or classification.

Example 1.1. Deep neural networks (DNNs) are particular parameterized functions which have gain in popularity these last years due to their performances on different prediction or classification tasks. Let us detail their rationale. A DNN is a succession of parameterized functions called neurons. A neuron typically computes a real multivariate function $\boldsymbol{h} \mapsto \sigma(\boldsymbol{w}\boldsymbol{h} + b)$, where $\boldsymbol{w}\boldsymbol{x}$ is the dot product of $\boldsymbol{w}$ (a vector of weights) and $\boldsymbol{x}$ (a vector of variables), b is the bias, and $\sigma(.)$ is a so-called (nonlinear) *activation function*, such as the sigmoid, hyperbolic tangent or ReLu function. Neurons can be gathered into layers which themselves can be cascaded, yielding increasingly complex functions. Some universal approximation theorems have been proposed in Cybenko (1989); Hornik (1991); Pinkus (1999); Lu et al. (2017); for instance, given any real valued continuous function f, there exists a single-layer DNN f_θ arbitrarily close to f, provided the activation function is not polynomial (Pinkus, 1999). Similar results have been proposed for multiple layers DNNs. So any Lebesgue-integrable function $f : \mathbb{R}^{d_y} \to \mathbb{R}$ can be approximated by an NN with ReLu activation functions and layers made of at least $d_y + 4$ neurons, provided the network is deep enough (Lu et al., 2017). The number of layers and of neurons per layer, as well as the activation functions, are hyperparameters which characterize the DNN architecture while the weights and biases are the model parameters learnt from a training set. For DNNs, the optimization problem (1.5) is usually approximated by a gradient descent approach; the recursive structure of the function induced by the successive DNN layers indeed enables to compute the gradient of f_θ via the backpropagation algorithm (Rumelhart et al., 1985; Hecht-Nielsen, 1992).

1.1.3 Discussion

Let us now discuss on these two approaches. A main advantage of optimizing (1.5) is that we do not need to model the distribution $p(x, y)$ by $p_\theta(x, y)$. Under some assumptions about the family $(f_\theta)_{\theta\in\Theta}$ (in particular in some RKHSs), it is possible to derive concentration inequalities

$$\mathbb{P}\left(|R_n(f_{\theta_n^\star}) - R(f_{\theta_n^\star})| > \epsilon\right) \leq \delta_{\epsilon,n},$$

where $\delta_{\epsilon,n}$ depends on ϵ, n and on the characteristics of the family $(f_\theta)_{\theta\in\Theta}$ (Vapnik, 1998; Bousquet et al., 2003). If $\delta_{\epsilon,n}$ tends to 0 as a function of n, it means that the estimated function $f_{\theta_n^\star}$ remains valid for unseen data since the training error $R_n(f_{\theta_n^\star})$ is close to the Bayesian risk $R(f_{\theta_n^\star})$, in probability. $\delta_{\epsilon,n}$ also provides a convergence rate.

By contrast, the first approach described in section 1.1.1 relies on the relevance of the parameterized joint distribution $p_\theta(x, y)$. However, it can be estimated from realizations $\{y^{(i)}\}_{1\leq i\leq n}$. This is particularly interesting, even in machine learning problems, because the introduction of a non observed and artificial hidden random variable X can be used to build an implicit distribution $p_\theta(y)$ defined as the marginal of $p_\theta(x, y)$. In addition, the distribution $p_\theta(x|y)$ is richer than a point estimator such as the conditional expectation. Consequently, it is possible to quantify the uncertainty of a prediction. Finally, once we have at our disposal the posterior distribution $p_\theta(x|y)$, it can be used to estimate $h(X)$ for a large class of functions h without running a new estimation algorithm. In summary, modelling the joint distribution can address two issues: the estimation of an hidden random variable X from an observation Y; or the design of a generative model based on a latent random variable which aims at modelling implicitly the distribution of the observations. From now on, we will mainly focus on this methodology in the context of time series. Indeed, generally we only have at our disposal a set of unlabelled observations (1.2). However, we would like to underline that some solutions developed in the framework of Section 1.1.2 are not incompatible with those of Section 1.1.1 as we see in the following example.

Example 1.2. In this manuscript, we will exploit the fact that it is possible to describe the parameters of a given conditional distribution $p_\theta(y|x)$ by a DNN $f_\theta(x)$. This idea has been exploited by the successful work of Kingma and Welling (2014) through the variational auto-encoder (VAE). The VAE methodology aims at building an implicit but powerful generative models in which

$$p_\theta(x, y) = p(x)p_\theta(y|x),$$

where

$$\begin{aligned} p(x) &= \mathcal{N}(x; 0; I), \\ p_\theta(y|x) &= \mathcal{N}\left(y; f_\theta(x); g_\theta(x)\right), \end{aligned}$$

and where $f_\theta(x)$ and $g_\theta(x)$ are the outputs of a DNN described by a set of weights and biases θ ($\mathcal{N}(x; \mu; \Sigma)$ denotes the Gaussian pdf with mean μ and covariance matrix Σ evaluated in x). The estimation of θ from a set of observations (1.2) will be discussed later and relies on variational Bayesian inference.

1.2 Statistical models for time series

Let us now turn to time series problems. Let $\{Y_t\}_{t\in\mathbb{N}}$ be a sequence of observed random variables with unknown distribution. Following the conclusion of the previous section, two related estimation problems

can be considered. The first one consists in predicting future observations of a time series. For example, applying the framework of Paragraph 1.1.1 with $Y \leftarrow Y_{0:t}$ and $X \leftarrow Y_{t+1}$, the prediction of Y_{t+1} relies on

$$p(y_{t+1}|y_{0:t}) = \frac{p(y_{0:t+1})}{p(y_{0:t})},$$

and so on modelling the joint distribution of the observations $(Y_0, \cdots, Y_t) = Y_{0:t}$ by a family of distributions $p_\theta(y_{0:t})$, for all $t \in \mathbb{N}$. However, as we have just seen, it can be relevant to introduce an hidden process $\{X_t\}_{t\in\mathbb{N}}$ which depends on $\{Y_t\}_{t\in\mathbb{N}}$; in this case, the generative distribution $p_\theta(y_{0:t})$ becomes the marginal of $p_\theta(x_{0:t}, y_{0:t})$. As a direct consequence, if $\{X_t\}_{t\in\mathbb{N}}$ is a process of interest (*i.e.* a physical interpretable process), then we can also look for estimating a (real) functional $h(X_{0:t})$ from a realization $y_{0:t}$.

As we see, whether it is to predict a set of future observations or to estimate an hidden random process dependent of the observations, the problem can be cast into the framework of Bayesian estimation in a latent data model $p_\theta(x_{0:t}, y_{0:t})$. It gives rise to three fundamentals problems that we approach in this thesis:

P.1 Which family of parametric distribution $p_\theta(x_{0:t}, y_{0:t})$ should we choose?

P.2 For a given family p_θ, how to estimate θ from a given realization $Y_{0:t} = y_{0:t}$?

P.3 For a given θ, how to compute or approximate quantities of interest such as any posterior/predictive distribution $p_\theta(x_{t'}|y_{0:t})$ or the predictive likelihoods $p_\theta(y_{t+1:t+t'}|y_{0:t})$?

Actually, these problems are related with each other and should not be addressed independently. For **P**.1, note that in the context of time series, the model p_θ should be able to consider sequences of any length, so θ should not depend on t (otherwise, the model cannot be used with any sequence of observations). Next, a thorny issue is to choose an expressive distribution which is able to describe accurately the statistical properties of the observations, but also the dependencies between the observed and the hidden processes when this last one is interpretable and should be estimated. However, we have to take into account that the Bayesian quantities of interest should be computed or approximated in a reasonable computational time, particularly in a context where the observations can arrive over time. This last constraint is the reason why the tools developed for static Bayesian estimation cannot be directly applied: when the sequence of observations is large, the computation of quantities of interest have to be done sequentially. In summary, a compromise between the model p_θ and the computational problems involved in Bayesian estimation has to be found.

Of course, this very general problem is not new and has been considered for decades. In particular, many statistical models based on latent random variables have been proposed. Some of them are the cornerstone of this synthesis: Markovian models (in a wide sense) and Recurrent Neural Networks (RNNs) which have been particularly developed and used in the machine learning community. We now give a short description of these models and some problems connecting them to **P**.1-**P**.3 that we have addressed (or at least tried to address) these last years.

1.2.1 Hidden Markov models

Let us briefly review the rationale of Hidden Markov models. Taking into account that the objective is to model the distribution of a time series $\{Y_t\}_{t\in\mathbb{N}}$, assuming that the observations are independent would be irrelevant. An alternative would be to consider a Markov chain

$$p_\theta(y_{0:t}) = p_\theta(y_0) \prod_{s=1}^{t} \underbrace{p_\theta(y_s|y_{s-1})}_{p_\theta(y_s|y_{0:s-1})}, \quad \text{for all } t \in \mathbb{N}.$$

However, this model is also very limited since a direct consequence is that Y_t only depends on Y_{t-1} given $Y_{0:t-1}$ and so that it is not possible to model dependencies between a current observation and the past ones when $Y_{t-1} = y_{t-1}$ is observed.

Construction (P.1) - An efficient way to introduce dependency between all the observations is to consider a simple additional hidden process $\{X_t\}_{t\in\mathbb{N}}$ which is assumed to be a Markov chain. We also assume that given $X_{0:t} = x_{0:t}$, the observations $Y_{0:t}$ are independent and Y_s only depends on x_s , for all $s \leq t$ and for all $t \in \mathbb{N}$. The deduced joint distribution is called an Hidden Markov Chain (HMC) and satisfies

$$p_\theta(x_{0:t}, y_{0:t}) \overset{\text{HMC}}{=} \underbrace{p_\theta(x_0)\prod_{s=1}^{t} p_\theta(x_s|x_{s-1})}_{p_\theta(x_{0:t})} \underbrace{\prod_{s=0}^{t} p_\theta(y_s|x_s)}_{p_\theta(y_{0:t}|x_{0:t})}, \text{ for all } t \in \mathbb{N}. \tag{1.6}$$

The pdfs $p_\theta(x_t|x_{t-1})$ describe the transitions of the Markov chain $\{X_t\}_{t\in\mathbb{N}}$ (w.r.t. a measure ν) while the distributions $p_\theta(y_t|x_t)$ are the conditional likelihoods and are assumed to be measurable w.r.t. ν as a function of x_t. Even if we have motivated the introduction of (1.6) to model time series, it appears that it can also be used to describe an interpretable hidden process dependent on observations. Indeed, this model has found many applications in signal processing such as tracking (X_t represents the state vector of a target at time t and Y_t the associated noisy range bearing measurement) (Jazwinski, 1970), financial problems (X_t represents the volatility of a financial time series) (Pitt and Shephard, 1999) or image segmentation (X_t represents the class associated to a noisy observed pixel Y_t) (Derrode and Pieczynski, 2004). While the HMC model is quite simple, it involves many challenges.

Prediction in an HMC (P.3) - When θ is known (so we remove the dependency in the notation θ), let us observe that our two initial estimation problems are directly connected. Indeed, in model (1.6), $p(x_t|x_{t-1}) = p(x_t|x_{t-1}, y_{0:t-1})$ so the predictive likelihood can be sequentially computed from

$$p(y_{t+1}|y_{0:t}) = \int_{\mathsf{X}^2} p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t)p(x_t|y_{0:t})\nu^{\otimes 2}(\mathrm{d}x_{t:t+1}). \tag{1.7}$$

The computation of (1.7) relies on the filtering density, denoted as

$$\phi_t(x_t) = p(x_t|y_{0:t}),$$

which can be itself computed sequentially from

$$\phi_{t+1}(x_{t+1}) = \frac{p(y_{t+1}|x_{t+1})\int_{\mathsf{X}} p(x_{t+1}|x_t)\phi_t(x_t)\nu(\mathrm{d}x_t)}{\int_{\mathsf{X}^2} p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t)\phi_t(x_t)\nu^{\otimes 2}(\mathrm{d}x_{t:t+1})}.$$

From $\phi_t(x_t)$, it is also possible to estimate a functional $h(X_t)$ from the past observations and we denote the expectation of $h(X_t)$ w.r.t. ϕ_t as

$$\phi_t(h) = \int_{\mathsf{X}} h(x_t)\phi_t(x_t)\nu(\mathrm{d}x_t).$$

These key distributions are computable in a few cases such as discrete state-space models (Rabiner, 1989) or linear and Gaussian HMCs (Jazwinski, 1970). In the general case, one needs to resort to approximations.

A popular class of approximations is the Sequential Monte Carlo methods. These methods approximate the filtering measure $\phi_t(\mathrm{d}x_t)$ by a random discrete measure

$$\phi_t^N(\mathrm{d}x_t) = \sum_{i=1}^{N} \mathcal{W}_t^i \delta_{\xi_t^i}(\mathrm{d}x_t) \tag{1.8}$$

from which we deduce an approximation of $\phi_t(h)$,

$$\phi_t^N(h) = \sum_{i=1}^{N} \mathcal{W}_t^i h(\xi_t^i), \tag{1.9}$$

and so one of $p(y_{t+1}|y_{0:t})$. The sequential computation of $\{(\mathcal{W}_t^i, \xi_t^i)\}_{1\leq i\leq N}$ relies on particle filters. Most of them are based on the sequential application of the Rubin's Sampling Importance Resampling (SIR) mechanism (Rubin, 1988; Smith and Gelfand, 1992). The mechanism consists of the three steps described at the end of Paragraph 1.1.1: a sampling step according to a given importance distribution; a weighting step which involves the target and the importance distributions; a resampling step from which an unweighted representation of the filtering distribution is obtained. In a sequential context, this last step is critical and is a rescue against the weight degeneracy phenomenon over time. However, while it ensures the stability of the particle filter overtime, this step tends to shrink severely the support of the unweighted representation when the weights in (1.8) are imbalanced and has consequences on future steps. An additional and general problem is to evaluate the reliability of estimator (1.9). Consequently, we have proposed some contributions in these directions; they are related to problem $\mathbf{P}.3$ through sequential Monte Carlo algorithms and aim at addressing the following questions.

Q.1 As stated above, particle filters rely on the Rubin's SIR mechanism. Rather than looking for tuning some steps of the mechanism (*e.g.* optimizing the conditional importance distribution), we revisit it in two directions. First, note that the rationale of this mechanism is to approximate a ratio of two integrals by a Monte Carlo method which uses the same importance distribution and the same samples. What happens when we introduce different importance distributions? Next, can we revisit the global mechanism (and not only its resampling step) without losing its rationale but with the objective to limit the degeneration phenomenon induced by classical resampling algorithms?

Q.2 The reliability of the estimator $\phi_t^N(h)$ is related to its variance. Under some assumptions, $\phi_t^N(h)$ satisfies a Central Limit Theorem (CLT) from which we can deduce its (theoretical) asymptotic variance (in the number of samples N). Can we propose an estimator of this asymptotic variance based on the samples already produced by the particle filter algorithm, for computational cost reasons?

Parameter estimation in an HMC ($\mathbf{P}.2$) - When θ is unknown and we have at our disposal a sequence of dependent observations $y_{0:t}$, the likelihood (1.3) becomes $p_\theta(y_{0:t})$. In such models, the statistical properties of the ML estimator

$$\hat{\theta}^{\mathrm{ML}} = \arg\max_{\theta\in\Theta} \quad \log(p_\theta(y_0)) + \sum_{s=1}^{t} \log(p_\theta(y_s|y_{0:s-1})),$$

where $p_\theta(y_s|y_{0:s-1})$ is computed from (1.7), has been theoretically studied in Douc et al. (2004) and Douc and Moulines (2012). However, $p_\theta(y_{0:t})$ (and so its gradient w.r.t. θ) is generally not computable. As discussed above,

$p_\theta(y_{0:t})$ can be approximated with an SMC algorithm; however, obtaining a differentiable Monte Carlo approximation to use a gradient ascent method is a thorny issue due to the resampling steps of such algorithms (Kantas et al., 2015).

A well known alternative is the application of the EM algorithm for the joint distribution (1.6) (Dempster et al., 1977; Rabiner, 1989; Cappé et al., 2005). It consists of two steps. For a given $\theta = \theta^{(i)}$, the E-step computes

$$\begin{aligned} Q(\theta, \theta^{(i)}) &= \mathbb{E}_{\theta^{(i)}}\left(\log\left(p_\theta(X_{0:t}, Y_{0:t})\right)|Y_{0:t} = y_{0:t}\right), \\ &= \sum_{s=1}^{t} \int_{\mathsf{X}^2} \left[\log(p_\theta(x_s|x_{s-1})) + \log(p_\theta(y_s|x_s))\right] p_{\theta^{(i)}}(x_{s-1:s}|y_{0:t})\nu^{\otimes 2}(\mathrm{d}x_{s-1:s}) \\ &+ \int_{\mathsf{X}} \log(p_\theta(x_0)) p_{\theta^{(i)}}(x_0|y_{0:t})\nu(\mathrm{d}x_0). \end{aligned}$$

Next, $\theta^{(i)}$ is updated by computing

$$\theta^{(i+1)} = \arg\max_\theta Q(\theta, \theta^{(i)}).$$

Note that the posterior distributions $p_{\theta^{(i)}}(x_{s-1:s}|y_{0:t})$ are generally not computable except in models where $p_\theta(y_{0:t})$ is available. For such models, choosing between a gradient ascent method or the EM-algorithm remains an open question that we do not address. Statistical properties of the EM algorithm are discussed in Balakrishnan et al. (2017). In the general case, approximations should be used and are based on the observation that $Q(\theta, \theta^{(i)})$ computes a mathematical expectation of a functional

$$h_{0:t}(x_{0:t}) = \sum_{s=1}^{t} h_s(x_{s-1}, x_s)$$

according to the smoothing distribution

$$\phi_{0:t|t}(x_{0:t}) = p_\theta(x_{0:t}|y_{0:t}).$$

In the case of the E-step of the EM algorithm, $h_s(x_{s-1}, x_s) = \log(p_\theta(x_s|x_{s-1})) + \log(p_\theta(y_s|x_s))$. This problem is more complicated than the previous filtering problem but sequential Monte Carlo algorithms have been extended for the smoothing problem. A popular smoother is the Forward Filtering Backward Smoothing (FFBS) algorithm (Tanizaki and Mariano, 1994; Doucet et al., 2000) which has the advantage to be computable online (*i.e.* in a forward way) for additive functionals (Del Moral et al., 2010). Consequently, **Q**.2 can be extended for the smoothing problem:

Q.2′ If we have at our disposal a Monte Carlo estimator $\phi^N_{0:t|t}(h_{0:t})$ based on the FFBS algorithm for additive functionals, can we estimate its asymptotic variance?

1.2.2 Extensions of HMC models

As we have just seen, problems **P**.2 and **P**.3 have been deeply tackled for HMCs. Let us now question **P**.1 in HMCs. Remember that a motivation to introduce the HMC (1.6) is that a Markovian hypothesis on the observations $\{Y_t\}_{t\in\mathbb{N}}$ is unsatisfying from a modelling point of view. So one can wonder if in the case where the hidden process $\{X_t\}_{t\in\mathbb{N}}$ is of interest, the Markovian assumption related to this process is reasonable. The models we now introduce aim at revising the answer to problem **P**.1 brought by the HMC construction.

Construction (P.1) - A direct way to relax the Markovian assumption related to $\{X_t\}_{t\in\mathbb{N}}$ is to consider the Pairwise Markov Chain (PMC) of Pieczynski (2003); Derrode and Pieczynski (2004) in which the joint distribution of the hidden and observed processes reads

$$p_\theta(x_{0:t}, y_{0:t}) \overset{\text{PMC}}{=} p_\theta(x_0, y_0) \prod_{s=1}^{t} p_\theta(x_s, y_s|x_{s-1}, y_{s-1}), \quad \text{for all } t \in \mathbb{N}. \tag{1.10}$$

It is easy to check that if

$$p_\theta(x_t, y_t|x_{t-1}, y_{t-1}) = p_\theta(x_t|x_{t-1})p_\theta(y_t|x_t), \quad \text{for all } t \in \mathbb{N},$$

the PMC (1.10) coincides with the HMC (1.6).

Since the motivation in this context is to provide a relevant model for the joint process $\{X_t, Y_t\}_{t\in\mathbb{N}}$ (and not only $\{Y_t\}_{t\in\mathbb{N}}$), we can follow the same path as in section 1.2.1 and introduce a latent process $\{Z_t\}_{t\in\mathbb{N}}$ which aims at building an implicit (but more realistic) distribution for $\{X_t, Y_t\}_{t\in\mathbb{N}}$. Thus, the PMC model can be further extended by the Triplet Markov Chain (TMC) (Pieczynski, 2002), a probabilistic model in which the distribution of $\{Z_t, X_t, Y_t\}_{t\in\mathbb{N}}$ reads

$$p_\theta(z_{0:t}, x_{0:t}, y_{0:t}) \overset{\text{TMC}}{=} p_\theta(z_0, x_0, y_0) \prod_{s=1}^{t} p_\theta(z_s, x_s, y_s|z_{s-1}, x_{s-1}, y_{s-1}), \quad \text{for all } t \in \mathbb{N}. \tag{1.11}$$

From a mathematical point of view, a TMC can be seen as a PMC in an augmented dimension in the sense that the role of the hidden random process is now played by $\{Z_t, X_t\}_{t\in\mathbb{N}}$. However, in the case where $\{X_t\}_{t\in\mathbb{N}}$ is a physical process of interest, the role played by $\{X_t\}_{t\in\mathbb{N}}$ and $\{Z_t\}_{t\in\mathbb{N}}$ is different and their separation in two distinct processes is critical as we will see later. The introduction of these models gives rise to a fundamental question from a modelling point of view.

Q.3 Even if we have introduced the PMC to strengthen the statistical properties of the hidden process, we can wonder if it has an impact on the distribution of the observations; more precisely, can we evaluate the relevance of the generative model $p_\theta(y_{0:t})$ induced by (1.10) w.r.t. that induced by (1.6)? Note that according to the previous remark, this question is irrelevant in the context of TMCs since the fact that the hidden process is interpretable or not does not impact the distribution of the observations.

While these models are more general, the choice of the transition distributions in (1.10)-(1.11) for a given application is not obvious. When an HMC has been validated for a given application, how can we improve it with models (1.10)-(1.11)? For example, it is not clear how to model the new dependencies between the current hidden state and the past observation via $p_\theta(x_t|x_{t-1}, y_{t-1})$, or that between the current and the past observations via $p_\theta(y_t|x_t, y_{t-1})$.

Q.4 Such models have been used in some applications such that unsupervised image segmentation but the problem of modelling their core distributions has been circumvented by the introduction of some assumptions which may reduce the modelling power of these generalized models (Derrode and Pieczynski, 2004; Gorynin et al., 2018). Based on the observation that DNNs can be seen as universal approximators, can we propose a general parameterized approach for models (1.10)-(1.11) which enables us to embed DNNs in these models?

Prediction and parameter estimation ($\mathbf{P}.2$ and $\mathbf{P}.3$) - The Bayesian estimation algorithms developed for HMCs can be extended to PMCs and TMCs by replacing the transition $p_\theta(x_t|x_{t-1})p_\theta(y_t|x_t)$ which appears in these algorithms by $p_\theta(x_t, y_t|x_{t-1}, y_{t-1})$ or $p_\theta(z_t, x_t, y_t|z_{t-1}, x_{t-1}, y_{t-1})$ (Desbouvries and Pieczynski, 2003a,b; Ait-El-Fquih and Desbouvries, 2006; Abbassi et al., 2011). This can be seen from the fact that PMCs and TMCs are nothing more than a partially observed Markov Chain and so an HMC in which one of the component of the (augmented) hidden state is perfectly observed. Thus, we do not further discuss on these direct extensions.

However, when these models are used for a Bayesian problem in which we focus on the estimation of an interpretable hidden process $\{X_t\}_{t\in\mathbb{N}}$, the direct adaptation of unsupervised estimation methods can be detrimental in terms of classification or prediction. The reason why is that when the complexity of the model increases (in terms of dependencies between the random variables and of number of parameters), nothing ensures that the hidden process $\{X_t\}_{t\in\mathbb{N}}$ associated to the estimated model has the desired physical properties of interest. For example, in the case of the PMC, the observation Y_t depends on X_t but also on X_{t-1} given the past; at first glance, it is not clear if Y_t is explained by X_t or X_{t-1}, contrary to a simple HMC.

$\mathbf{Q}.5$ When we introduce the PMC and the TMC models for the Bayesian estimation problem of an hidden process of interest, the estimation parameters methods should be tuned in a such way that the interpretability of the hidden process that we have with a fundamental HMC is kept. This problem is all the more important when we introduce highly parameterized models as suggested by question $\mathbf{Q}.4$. How to include this constraint in the parameter estimation of models (1.10)-(1.11) and how to exploit the interpretability of an already used model such as the HMC?

Finally, in the general class of TMC models, Pieczynski (2011a) has highlighted a particular TMC model which presents interesting computational properties. When Z_t is discrete and the model satisfies

$$\begin{aligned} p_\theta(z_t, x_t, y_t|z_{t-1}, x_{t-1}, y_{t-1}) &= p_\theta(z_t, y_t|z_{t-1}, y_{t-1})p_\theta(x_t|z_t, x_{t-1}, y_{t-1:t}), \\ p_\theta(x_t|z_t, x_{t-1}, y_{t-1:t}) &= \mathcal{N}\left(x_t; C_\theta\left(z_t, y_{t-1:t}\right)x_{t-1} + h_\theta\left(z_t, y_{t-1:t}\right); \Sigma_\theta\left(z_t, y_{t-1:t}\right)\right), \end{aligned} \tag{1.12}$$

$\mathbb{E}(h(X_s)|y_{0:t})$ can be exactly computed if h is a quadratic function, at cost linear in the number of observations (Pieczynski, 2011a,b; Derrode and Pieczynski, 2013).

$\mathbf{Q}.6$ Model (1.12) is interesting from a computational point of view but is clearly non-identifiable since the parameters related to $p_\theta(x_t|z_t, x_{t-1}, y_{t-1:t})$ do not depend on the associated likelihood. It is also close to the popular linear and Gaussian Jump Markov State Space System (JMSS), *i.e.* a linear and Gaussian HMC which depends on a Markovian discrete random process (Tugnait, 1982). The JMSS has been used for many applications (Doucet et al., 2001b) but the computation of $\mathbb{E}(h(x_s)|y_{0:t})$ is an NP-hard problem for quadratic functionals. How to use model (1.12) (in which the posterior distribution is also unknown) to address in an alternative way the Bayesian inference problems $\mathbf{P}.1$ and $\mathbf{P}.2$ in the linear and Gaussian JMSS?

1.2.3 Recurrent Neural Networks

Initially, RNNs aimed at solving the statistical learning approach of section 1.1.2 in the case of sequential data by building parameterized functions which take into account all observations until time t. They are an adaptation of DNNs (see Ex. 1.1) for time series, but they can be used to produce generative models as we now see.

Construction (P.1) - A DNN is not relevant to predict an observation Y_{t+1} from $Y_{0:t} = y_{0:t}$ because its input has a fixed size and cannot take into account an increasing sequence of observations. Even if it can be used with a predefined window size of observations, it is unsatisfying from a modelling point of view. The idea underlying RNNs is to store a summary of all the past observations in a variable of fixed dimension. At a given time t, the so-called latent state h_t is a function of all the past observations $y_{0:t}$. In order to respect the constraint that the parameters of the model do not depend on time, the latent state is computed sequentially as

$$h_t = f_\theta(h_{t-1}, y_t), \quad \text{for all } t \in \mathbb{N}, \tag{1.13}$$

where f_θ is a parameterized activation function (*e.g.* a DNN). A prediction of interest at time t is deduced from the latent state through a function $g_\theta(h_t)$ which can also be parameterized by a DNN. However, as discussed previously, we focus on generative models to quantify the uncertainty of our predictions. Endowing the observations a pdf turns the RNN into a generative model. To do this, g_θ is replaced by a parameterized pdf $p_\theta(y_{t+1}|h_t)$. This yields a generative model based on the conditional distribution $p_\theta(y_{t+1}|h_t)$ and described as

$$p_\theta(y_{0:t}) \overset{\text{RNN}}{=} p_\theta(y_0) \prod_{s=1}^{t} \underbrace{p_\theta(y_s|h_{s-1})}_{p_\theta(y_s|y_{0:s-1})}, \quad \text{for all } t \in \mathbb{N}, \tag{1.14}$$

where h_s is computed from (1.13) for all s, $s \leq t$. So the construction of a generative model based on an RNN also relies on a latent but conditionally deterministic process $\{H_t\}_{t\in\mathbb{N}}$. Actually, in the time series framework, model (1.13)-(1.14) is called an observation driven model while the HMC (1.6) is a parameter driven model (Cox et al., 1981; Koopman et al., 2016).

Q.7 Starting from the observation that Markovian models and generative RNNs share a common construction (the distribution of the observations is deduced from a random or deterministic variable), can we extend the question **Q.3** to the RNN? More precisely, how to compare the modelling power of Markovian and RNN models? We will see that this comparison can be done under the general PMC framework introduced in Paragraph 1.2.2.

Prediction in an RNN (P.2) - By construction of (1.14), the prediction of an observation Y_{t+1} from $y_{0:t}$ is obtained by computing the latent state h_t from (1.13) and next $p_\theta(y_{t+1}|h_t)$. More generally, it is possible to obtain a discrete approximation of $p_\theta(y_{t+1:t+t'}|y_{0:t})$ by sampling sequentially y^i_{t+s} according to from $p_\theta(y_{t+s}|h^i_{t+s-1})$, where h^i_{t+s-1} is computed from (1.13) for all s, $1 \leq s \leq t'$.

Estimation of θ (P.3) - The parameter estimation in a RNN is often computed from a gradient ascent method since the log-likelihood

$$\log\left(p_\theta(y_{0:t})\right) = \log\left(p_\theta(y_0)\right) + \sum_{s=1}^{t-1} \log\left(p_\theta(y_{s+1}|h_s)\right)$$

is computable in a generative RNN. However, note that the computation of the gradient of $\log(p_\theta(y_{0:t}))$ w.r.t. θ is not direct since h_s also depends on θ via (1.13). However, gradient backpropagation can still be used. By constrast with feedforward DNN, there can be as many computed gradients as observations for a given parameter. In that case, a parameter is updated according to the sum of partial derivatives computed at each time instant. This adaptation of the backpropagation algorithm is called *backpropagation through*

time (Robinson and Fallside, 1987; Werbos, 1990; Mozer, 1995). In practice, the gradients computed for a given parameter geometrically tend to infinity or to zero when we get back into the past. These phenomena are called *exploding gradient* and *vanishing gradient*. The exploding gradient phenomenon is often due to the repeated multiplication of high weights. Learning the RNN becomes particularly unstable. An efficient way to limit this behavior is to bound the values taken by the gradient (Goodfellow et al., 2016; Goldberg, 2017). One can also include a regularization term to the cost function in order to penalize weights that are too large (Pascanu et al., 2013). By contrast, the vanishing gradient phenomenon results from the repeated multiplication or small size weights, as well as the iterated use of activation functions which have derivates bounded by 1 in magnitude (*e.g.* the sigmoid). In that case, the oldest observations are not taken into account in the learning phase, so it is difficult to learn long term dependencies. Consequently more sophisticated parameterizations of f_θ in (1.13) have been proposed, such as the *Long Short Term Memories* (LSTM) (Hochreiter and Schmidhuber, 1997) and the *Gated Recurrent Units* (GRU) (Chung et al., 2014). In particular, LSTM and GRU render the returning loop of the RNN more complex, in order to mitigate the vanishing gradient phenomenon.

1.3 About the notations

Let us now clarify our notations for the different processes introduced in the previous sections. From now on, $\{Y_t\}_{t\in\mathbb{N}}$ refers to an observed time series; $\{X_t\}_{t\in\mathbb{N}}$ refers to an hidden process of physical interest, *i.e.* a process that we aim at estimating (even partially). We will sometimes say that $\{X_t\}_{t\in\mathbb{N}}$ is interpretable (*e.g.* X_t is a 4-dimensional vector and consists of the position and the velocity of a target in the Euclidean plane at time t). $\{H_t\}_{t\in\mathbb{N}}$ refers to an intermediate hidden process (whether it is deterministic or not) which only aims at building an implicit distribution to model the observations; so its estimation is not the main purpose when it is considered. Finally, $\{Z_t\}_{t\in\mathbb{N}}$ also refers to an intermediate hidden process but will always be introduced jointly with $\{X_t\}_{t\in\mathbb{N}}$ or $\{H_t\}_{t\in\mathbb{N}}$; it aims at emphasizing its different nature w.r.t. $\{X_t\}_{t\in\mathbb{N}}$ or $\{H_t\}_{t\in\mathbb{N}}$. As an example, X_t can be a continuous random variable while Z_t is a discrete one; or $\{H_t\}_{t\in\mathbb{N}}$ is random while $\{Z_t\}_{t\in\mathbb{N}}$ is deterministic, given the observations.

1.4 Organization of the thesis

In this brief review of latent data models for time series, we have highlighted some problems related to $\mathbf{P}.1$-$\mathbf{P}.3$ (construction, estimation, prediction) through questions $\mathbf{Q}.1$-$\mathbf{Q}.7$. This manuscript proposes a synthesis of my contributions to address some of these questions. It is organized as follows.

Chapter 2 is devoted to alternative (sequential) Monte Carlo algorithms when the probabilistic model is known. More precisely, we focus on $\mathbf{P}.3$ in an HMC and we address the points raised in $\mathbf{Q}.1$. For these questions, we first revisit the Rubin's SIR mechanism (Rubin, 1988; Smith and Gelfand, 1992) by proposing an importance sampling estimator of $\mathbb{E}(h(X)|Y = y)$ based on two importance distributions; we propose a procedure to tune the introduced importance distributions with the objective to minimize the asymptotic variance of the resulting estimator. In a second stage, we carefully study the resampling scheme associated to the mechanism and which is critical in the sequential case. In pathological models (*e.g.* informative HMCs), when the resampling steps tend to eliminate all the samples except one, we propose a resampling mechanism which produces independent conditional samples. Actually, our methodology can be interpreted as an implicit auxiliary particle filter. Since the computational cost associated to our particle filter increases com-

pared to the original mechanism, we compare our scheme with traditional methods at a fixed computational cost and we also propose an intermediate solution to decrease it.

Chapter 3 also focusses on Problem $\mathbf{P}.3$ in an HMC but aims at estimating the asymptotic variance of some estimators based on particle filter; so it addresses $\mathbf{Q}.2 - 2'$. We focus on two popular sequential Monte Carlo algorithms: the bootstrap particle filter of Gordon et al. (1993) and the FFBS algorithm of Tanizaki and Mariano (1994); Doucet et al. (2000). For each of these algorithms, we propose estimators of the asymptotic variance of $\phi_t^N(h)$ and of $\phi_{0:t}^N(h_{0:t})$. Our estimators satisfy the following constraints: (i) they are built from the output of these algorithms (*i.e.* no additional simulations are required); (ii) they are sequentially computable; (iii) they converge to the theoretical asymptotic variances. When it is possible, a convergence rate is given.

While the two previous chapters are related to $\mathbf{P}.3$, Chapter 4 is devoted to $\mathbf{P}.1$ and aims at understanding the impact of the structural differences of the HMC, the PMC and the RNN on their associated generative distribution $p_\theta(y_{0:t})$, for all $t \in \mathbb{N}$. We thus address $\mathbf{Q}.3$ and $\mathbf{Q}.7$ and we show that under some assumptions, these questions can be approached from a system theory perspective (Chen, 1970).

Finally, Chapter 5 is a cross-fertilization of the reflexions which have emerged from the previous chapters. And indeed, we deal with the complete chain $\mathbf{P}.1$ - $\mathbf{P}.3$ for Bayesian classification and prediction of time series. We show that by unifying the two point of views (hidden Markov models and recurrent neural architectures), we can address $\mathbf{Q}.4$ - $\mathbf{Q}.6$ with powerful probabilistic models ($\mathbf{P}.1$) in which we propose some Bayesian inference methods ($\mathbf{P}.2$ - $\mathbf{P}.3$). Three applications are proposed. We first introduce new generative models based on the combination of PMCs and RNNs which are able to learn the distribution of complex time series. These models are next used in the case where we want to classify each observation of a time-series in an unsupervised way. In this case, the hidden process becomes of physical interest and parameter estimation algorithms should be guided in order to take into account this constraint. Finally, we show that such combinations can be used to propose a fast alternative to Monte Carlo methods for Bayesian estimation in linear and Gaussian JMSS (Tugnait, 1982).

At the end of the manuscript, we summarize the contributions and describe the unanswered questions related to our solutions. We also give a short description of future projects in which I will be involved.

CHAPTER 2

Revisiting some (sequential) Monte Carlo methods

This chapter focuses on Monte Carlo methods based on the Rubin's SIR mechanism (Rubin, 1988). We assume that the probabilistic models are known so we give up temporarily the dependency in θ of the involved distributions. Moreover, in the sequential case, we particularly focus on the HMC and we emphasize the fact that this model is built from two class of conditional distributions: the transitions of the Markov chain $\{X_t\}_{t\in\mathbb{N}}$ denoted as $f_t(x_t|x_{t-1})$ at time t, and the conditional likelihoods of the observations with the hidden states denoted as $g_t(y_t|x_t)$ at time t. In other words, the joint distribution of an HMC reads

$$p(x_{0:t}, y_{0:t}) \overset{\text{HMC}}{=} p(x_0) \prod_{s=1}^{t} f_s(x_s|x_{s-1}) \prod_{s=0}^{t} g_s(y_s|x_s), \quad \text{for all } t \in \mathbb{N}. \tag{2.1}$$

In model (2.1), we discuss on sampling strategies which aim at improving current Monte Carlo estimators

$$\phi_t^N(h) = \sum_{i=1}^{N} \mathcal{W}_t^i h(\xi_t^i) \tag{2.2}$$

of

$$\phi_t(h) = \int_{\mathsf{X}} h(x_t)\phi_t(x_t)\nu(\mathrm{d}x_t)$$

(remember that the filtering distribution is denoted as $\phi_t(x_t) = p(x_t|y_{0:t})$). After recalling the rationale of Monte Carlo methods and their sequential application, we propose two improvements. In section 2.2, we discuss on the relevance of introducing two importance distributions for normalized importance sampling. In section 2.3, we propose to use resampled but independent particles to compute Monte Carlo estimators. This last alternative is extended in the sequential case.

The methods proposed in this chapter are mainly a synthesis of the work I realized during the supervision of R. Lamberti (2015-2018), with F. Desbouvries and F. Septier. More details can be found in [6,7,10,11], see paragraph A.4. of my research activities.

2.1 Background

The Rubin's SIR mechanism (Alg. 2.1) - Let $\pi \propto p$ be a pdf known up to a constant and q be an importance distribution which satisfies $q(x) = 0$ when $p(x) = 0$; The objective is to compute

$$\pi(h) = \int_{\mathsf{X}} h(x)\pi(x)\nu(\mathrm{d}x) = \frac{\int_{\mathsf{X}} h(x)p(x)\nu(\mathrm{d}x)}{\int_{\mathsf{X}} p(x)\nu(\mathrm{d}x)} = \frac{p(h)}{p(1)} = \frac{q\left(\frac{ph}{q}\right)}{q\left(\frac{p}{q}\right)}. \tag{2.3}$$

The so-called importance sampling estimator of this ratio is obtained by drawing i.i.d. samples $\{\xi^i\}_{i=1}^N$ according to q,

$$\pi_{\mathrm{IS}}^N(h) = \sum_{i=1}^N \mathcal{W}^i h(\xi^i), \tag{2.4}$$

where

$$\mathcal{W}^i = \Omega^{-1}\omega^i, \quad \omega^i = \frac{p(\xi^i)}{q(\xi^i)} \quad \text{and} \quad \Omega = \sum_{i=1}^N \omega^i, \qquad \text{for all } i \in [1:N]. \tag{2.5}$$

If we want an unweighted representation of π, particles $\{\xi^i\}_{i=1}^N$ can be resampled according to the normalized weights. The resampling step is equivalent to sample independently discrete variables $\{A^i\}_{i=1}^N$ according to the categorical distribution defined by $\{\mathcal{W}^i\}_{i=1}^N$. We thus obtain an alternative estimator

$$\pi_{\mathrm{SIR}}^N(h) = \sum_{i=1}^N \frac{1}{N} h(\xi^{A^i}). \tag{2.6}$$

From a computational point of view, it is easy to check that $\pi_{\mathrm{IS}}^N(h)$ is a better estimator than $\pi_{\mathrm{SIR}}^N(h)$ because $\mathbb{V}\mathrm{ar}(\pi_{\mathrm{IS}}^N(h)) \leq \mathbb{V}\mathrm{ar}(\pi_{\mathrm{SIR}}^N(h))$. However, studying $\pi_{\mathrm{SIR}}^N(h)$ can be interesting to evaluate the consequences of the resampling step. It is well known that a sample ξ^{A^i} converges in distribution to π when $N \to \infty$ (Geweke, 1989). Setting

$$\begin{aligned} \mathcal{V}_q^\infty(h) &= q\left(\frac{\pi^2}{q^2}(h-\pi(h))^2\right), \\ \widetilde{\mathcal{V}}_q^\infty(h) &= \mathcal{V}_q^\infty(h) + \pi\left((h-\pi(h))^2\right), \end{aligned} \tag{2.7}$$

and assuming that $\mathcal{V}_q^\infty(h) < \infty$ and $\widetilde{\mathcal{V}}_q^\infty(h) < \infty$, we have

$$\begin{aligned} \sqrt{N}\left(\pi_{\mathrm{IS}}^N(h) - \pi(h)\right) &\underset{N\to\infty}{\Longrightarrow} \mathcal{N}\left(0, \mathcal{V}_q^\infty(h)\right), \\ \sqrt{N}\left(\pi_{\mathrm{SIR}}^N(h) - \pi(h)\right) &\underset{N\to\infty}{\Longrightarrow} \mathcal{N}\left(0, \widetilde{\mathcal{V}}_q^\infty(h)\right), \end{aligned}$$

where $\Longrightarrow$ denotes the convergence in distribution. This result suggests some ideas to improve the estimators $\pi_{\mathrm{IS}}^N(h)$ and $\pi_{\mathrm{SIR}}^N(h)$.

- It is well known that

  $$q^*(x) \propto |h(x)|\pi(x)$$

  minimizes the asymptotic variance $\mathcal{V}_q^\infty(h)$ of $\pi_{\mathrm{IS}}^N(h)$. However, the fact that the variance cannot be further decreased is the consequence that the same samples (and so the same importance distributions) are used to compute the numerator and denominator of (2.3). In order to break this classical scheme, we propose to introduce an importance distribution in augmented dimension on X^2 (so two marginal importance distributions) and next to tune them by taking into account the computational cost and the asymptotic variance of the resulting Monte Carlo estimator;

- The resampling step is critical in sequential problems. It can be observed in (2.7) that $\pi_{\mathrm{SIR}}^N(h)$ is poor when $\pi_{\mathrm{IS}}^N(h)$ is poor or $\mathbb{V}\mathrm{ar}_\pi(h)$ is large. In order to limit the increase of variance w.r.t. $\pi_{\mathrm{IS}}^N(h)$, several alternative schemes such as the residual, systematic or stratified resampling have been proposed

(Hol et al., 2006; Douc et al., 2005; Li et al., 2015) but the improvement w.r.t. the multinomial resampling is very limited if the importance distribution is not well chosen. Starting from the observation that the degree of dependence of the samples $\{\xi^{A^i}\}_{i=1}^N$ is related to the quality of the multinomial resampling (in a degenerate case, samples $\{\xi^{A^i}\}_{i=1}^N$ tend to be all equal), we propose a mechanism which produces samples according to the same marginal distribution as that of $\{\xi^{A^i}\}_{i=1}^N$, but which are independent. As we will see, the obtained mechanism can be interpreted as particular importance sampling algorithm with an implicit importance distribution, provided the samples are reweighted.

Algorithm 2.1 Rubin's SIR mechanism

Require: $p(x)$ such that $\pi(x) \propto p(x)$, an importance distribution q
  for $i \in [1:N]$ **do**
    Sample $\xi^i \sim q(x)$.
    Set $\omega^i = \frac{p(\xi^i)}{q(\xi^i)}$.
  end for
  for $i \in [1:N]$ **do**
    Sample $A^i \overset{\text{i.i.d.}}{\sim} \text{Categorical}\left(\left\{\mathcal{W}^l = \omega^l \times \left(\sum_{j=1}^N \omega^j\right)^{-1}\right\}_{l\in[1:N]}\right)$.
  end for
  return $\{\omega^i, \xi^i, A^i\}_{i=1}^N$

Sequential Importance Sampling with Resampling (Alg. 2.2) - Let us now introduce the HMC (2.1) and set

$$\begin{aligned} \pi(x) &\leftarrow \phi_{0:t|t}(x_{0:t}) = p(x_{0:t}|y_{0:t}) \propto p(x_{0:t}, y_{0:t}) \\ q(x) &\leftarrow q_t(x_{0:t}). \end{aligned}$$

In order to update $\phi_{t|t}^N(h)$ from $\phi_{t-1|t-1}^N(h)$ when a new observation y_t is available, the importance distribution q_t satisfies

$$q_t(x_{0:t}) = q_{t-1}(x_{0:t-1}) q_t(x_t|x_{0:t-1}).$$

In practice, we set $q_t(x_t|x_{0:t-1}) = q_t(x_t|x_{t-1})$ but note that it can depend on the observations $y_{0:t}$. Applying importance sampling with this particular setting allows to sample new particles and to compute the importance weights sequentially. However, this direct sequential application tends to degenerate when t grows: all the normalized weights $\{\mathcal{W}_t^i\}_{i=1}^N$ associated to $\phi_{t|t}^N(h)$ in (2.2) are equal to 0, except one. The reason why is that the dimension of the space on which evolves the target distribution $\phi_{0:t|t}(x_{0:t})$ increases overtime. This phenomenon can be addressed by introducing the resampling step of the Alg. 2.1. The resulting algorithm known as the sequential importance sampling with resampling algorithm (Doucet et al., 2001a) coincides with Alg. 2.2 and consists of a sequential application of Alg. 2.1.

Auxiliary Particle Filters (Alg. 2.3) - The rationale of Auxiliary Particle Filters (APFs) is based on the sequential expression of the filtering density

$$\phi_t(x_t) \propto g_t(y_t|x_t) \int_{\mathsf{X}} f_t(x_t|x_{t-1}) \phi_{t-1}(x_{t-1}) \nu(\mathrm{d}x_{t-1}).$$

Algorithm 2.2 SIR particle filter

Require: $\{\omega_{t-1}^i, \xi_{t-1}^i\}_{i=1}^N$, y_t, a conditional importance distribution $q_t(x_t|x_{t-1})$
for $i \in [1:N]$ **do**
Sample $A_{t-1}^i \overset{\text{i.i.d}}{\sim} \text{Categorical}\left(\{\mathcal{W}_{t-1}^l = \omega_{t-1}^l \left(\sum_{j=1}^N \omega_{t-1}^j\right)^{-1}\}_{l\in[1:N]}\right)$.
Sample $\xi_t^i \sim q_t(x_t|\xi_{t-1}^{A_{t-1}^i})$.
Set $\omega_t^i = \frac{f_t(\xi_t^i|\xi_{t-1}^{A_{t-1}^i}) g_t(y_t|\xi_t^i)}{q_t(\xi_t^i|\xi_{t-1}^{A_{t-1}^i})}$.
end for
return $\{\omega_t^i, \xi_t^i\}_{i=1}^N$

Plugging a Monte Carlo approximation $\phi_{t-1}^N(x_{t-1})$, we obtain a mixture approximation of $\phi_t(x_t)$,

$$\tilde{\phi}_t^N(x_t) \propto \sum_{i=1}^N \omega_{t-1}^i p(y_t|\xi_{t-1}^i) p(x_t|\xi_{t-1}^i, y_t) = \sum_{i=1}^N \omega_{t-1}^i g_t(y_t|x_t) f_t(x_t|\xi_{t-1}^i).$$

APF algorithms target the mixture $\tilde{\phi}_t^N(x_t)$ by resorting to importance sampling in augmented dimension (Pitt and Shephard, 1999; Cappé et al., 2007). It relies on an importance mixture

$$q_t^N(x_t) \propto \sum_{i=1}^N \mu(\xi_{t-1}^i) q_t(x_t|\xi_{t-1}^i). \tag{2.8}$$

An iteration of the APF is given in Alg. 2.3. Note that under this point of view, the resampling step appears naturally when we sample according to (2.8). Alg. 2.3 also generalizes Alg. 2.2 which corresponds to the particular setting $\mu(\xi_{t-1}^i) = \omega_{t-1}^i$. Finally, the particular setting $\mu(x_{t-1}) = p(y_t|x_{t-1})$ and $q_t(x_t|x_{t-1}) = p(x_t|x_{t-1}, y_t) \propto f_t(x_t|x_{t-1}) g_t(y_t|x_t)$ coincides with the Fully Adapted APF (FA-APF); in this case, the particles of time $t-1$ are first resampled according to the likelihood of ξ_{t-1}^i with new observation y_t; they are next extended with new samples drawn from the so-called optimal distribution which takes into account observation y_t.

Algorithm 2.3 Auxiliary Particle Filter

Require: $\{\omega_{t-1}^i, \xi_{t-1}^i\}_{i=1}^N$, y_t, $\mu(x_{t-1})$, $q_t(x_t|x_{t-1})$
for $i \in [1:N]$ **do**
Sample $A_t^i \overset{\text{i.i.d.}}{\sim} \text{Categorical}\left(\left\{\mu(\xi_{t-1}^l) \times \left(\sum_{j=1}^N \mu(\xi_{t-1}^j)\right)^{-1}\right\}_{l\in[1:N]}\right)$.
Sample $\xi_t^i \sim q_t(x_t|\xi_{t-1}^{A_t^i})$.
Set $\omega_t^i = \frac{\omega_{t-1}^{A_t^i} f_t(\xi_t^i|\xi_{t-1}^{A_t^i}) g_t(y_t|\xi_t^i)}{\mu(\xi_{t-1}^{A_t^i}) q_t(\xi_t^i|\xi_{t-1}^{A_t^i})}$.
end for
return $\{\omega_t^i, \xi_t^i\}_{i=1}^N$

The asymptotic results of the static case can be extended to the sequential one but are not presented in this chapter (see e.g. (Cappé et al., 2005; Chopin and Papaspiliopoulos, 2020)). Here, we rather focus on the critical resampling step of Alg. 2.2. Even if it ensures the stability of the algorithm overtime, there are

pathological cases for which the resampling step is severe since it may eliminate all the particles except one (when N is finite). Even if diversity is recreated during the next sampling step, the particles at a given time t have the same ancestor. Such cases can appear in informative or high dimensional HMCs in which it is difficult to draw samples in relevant regions of $g_t(.|x_t)$. A sequential version of the revisited SIR mechanism can improve the particle filters estimators and actually provides a relevant and implicit importance distribution for the APF which mimics the rationale of the FA-APF with any conditional importance distribution $q_t(x_t|x_{t-1})$. Our experiments show that it can be used to mitigate the shrinkage phenomenon of the traditional resampling scheme at the same computational cost of classical particle filters.

2.2 Double Proposal Importance Sampling

Construction of the estimator - We first turn back to the problem of approximating $\pi(h)$. Let $q_{1,2}(x_1, x_2)$ be an importance distribution on X^2 such that its marginal distributions $q_1(x_1)$ and $q_2(x_2)$ are known. The main idea of the Double Proposal Importance Sampling (DPIS) idea is to rewrite (2.3) as

$$\pi(h) = \frac{q_1\left(\frac{ph}{q_1}\right)}{q_2\left(\frac{p}{q_2}\right)}.$$

A natural Monte Carlo estimator reads

$$\pi^N_{\mathrm{DPIS}}(h) = \frac{\sum_{i=1}^N \omega_1^i h(\xi_1^i)}{\sum_{i=1}^N \omega_2^i}, \quad (\xi_1^i, \xi_2^i) \overset{\text{i.i.d.}}{\sim} q_{1,2}, \text{ for all } i \in [1:N], \tag{2.9}$$

where

$$\omega_1^i = \frac{p(\xi_1^i)}{q_1(\xi_1^i)} \quad \text{and} \quad \omega_2^i = \frac{p(\xi_2^i)}{q_2(\xi_2^i)}, \qquad \text{for all } i \in [1:N].$$

Note that we do not make any assumption about the statistical dependency between ξ_1^i and ξ_2^i for a given i. Estimator (2.9) generalizes (2.4) which coincides with the particular setting $q_{1,2}(\mathrm{d}x_2|x_1) = \delta_{x_1}(\mathrm{d}x_2)$, *i.e.* $q_2 = q_1$ and $\xi_1^i = \xi_2^i$ for all $i \in [1:N]$. The following Proposition describes the asymptotic properties of $\pi^N_{\mathrm{DPIS}}(h)$.

Proposition 2.1. Let $q_{1,2}(x_1, x_2)$ be the (importance) distribution associated to samples $\{\xi_1^i, \xi_2^i\}_{i=1}^N$. We note

$$g(x_1, x_2) = \frac{p(x_1)}{q_1(x_1)} h(x_1) - \frac{\pi(x_2)}{q_2(x_2)} \pi(h), \tag{2.10}$$

$$\mathcal{V}^\infty_{q_{1,2}}(h) = q_{1,2}\left(g^2\right). \tag{2.11}$$

Then if $\mathcal{V}^\infty_{q_{1,2}}(h) < \infty$,

$$\sqrt{N}\left(\pi^N_{\mathrm{DPIS}}(h) - \pi(h)\right) \underset{N\to\infty}{\Longrightarrow} \mathcal{N}\left(0; \mathcal{V}^\infty_{q_{1,2}}(h)\right).$$

Of course, $\mathcal{V}^\infty_{q_{1,2}}(h)$ coincides with $\mathcal{V}^\infty_q(h)$ if $q_{1,2}(\mathrm{d}x_2|x_1) = \delta_{x_1}(\mathrm{d}x_2)$. The introduction of two importance distributions gives an additional degree of freedom. Indeed, let us assume that h is a constant sign function. Then it is easy to check that $\mathcal{V}^\infty_{q^*_{1,2}}(h) = 0$ if $q^*_{1,2}$ satisfies $q_1^* \propto |h|p$ and $q_2^* \propto p$, contrary to the classical importance sampling estimator where $\mathcal{V}^\infty_{q^*}(h) \geq 0$ when $q^* \propto |h|q$ (except if h is a constant). Of course, the distributions q_1^* and q_2^* are not computable in practice and the practical computation of (2.9) is related to the following constraints:

1. The choice of q_1 and q_2;
2. The dependency between ξ_1^i and ξ_2^i (so the choice of $q_{1,2}$ such that the marginals coincide with q_1 and q_2);
3. The computational cost associated to $\pi^N_{\mathrm{DPIS}}(h)$ w.r.t. $\pi^N_{\mathrm{IS}}(h)$.

In the light of Prop. 2.1, a natural objective is to minimize $\mathcal{V}^{\infty}_{q_{1,2}}(h)$. Setting $(X_1, X_2) \sim q_{1,2}$, let us first remark that the asymptotic variance can be rewritten as

$$\mathcal{V}^{\infty}_{q_{1,2}}(h) = \mathbb{V}\mathrm{ar}\left(\frac{\pi(X_1)}{q_1(X_1)}h(X_1)\right) - 2\pi(h)\mathbb{C}\mathrm{ov}\left(\frac{\pi(X_1)}{q_1(X_1)}h(X_1), \frac{\pi(X_2)}{q_2(X_2)}\right) + \mathbb{V}\mathrm{ar}\left(\frac{\pi(X_2)}{q_2(X_2)}\right). \quad (2.12)$$

Consequently, if the marginals q_1 and q_2 are first fixed (remember that we need them to compute the importance weights), the minimization of $\mathcal{V}^{\infty}_{q_{1,2}}(h)$ relies on the covariance term and so on the dependency between X_1 and X_2. A preliminary study in the Gaussian case (π and $q_{1,2}$ are Gaussian) where all the quantities are computable has shown that the optimal relation of dependency depends on the parameters of q_1 and on q_2; the optimal choice of $q_{1,2}$ from q_1 and q_2 actually remains an open problem. However, in the sequel we constraint X_1 and X_2 to be deterministic transformations of a given random variable. The reason why is that it can be interpreted as direct extension of the computation of the estimator $\pi^N_{\mathrm{IS}}(h)$, where $X_1 = X_2$, and the associated computational cost remains unchanged in terms of sampling steps. Indeed, in the general case, the computation of $\pi^N_{\mathrm{DPIS}}(h)$ would require $2N$ sampling steps.

A Practical DPIS estimator (Alg. 2.4) - It remains to deal with the first point. We propose an easy way to build relevant importance distributions q_1 and q_2. We start from a common importance distribution q used to compute $\pi^N_{\mathrm{DPIS}}(h)$ and its associated samples $\{\xi^i\}_{i=1}^N$. The idea is to move samples $\{\xi^i\}_{i=1}^N$ in order to optimize $\mathcal{V}^{\infty}_{q_{1,2}}(h)$. As an illustration, we focus on the case where $x \in \mathbb{R}$ and we propose a linear transformation of the original samples $\{\xi^i\}_{i=1}^N$. Note that any differentiable transformation such that q_1 and q_2 are computable remains valid. In the linear case, we have

$$\xi_1^i = \alpha_1\xi^i + \beta_1 \quad \text{and} \quad \xi_1^2 = \alpha_2\xi^i + \beta_2,$$

and so

$$\begin{aligned} q_{\phi,1}(x_1) &= \frac{1}{|\alpha_1|}q\left(\frac{x_1 - \beta_1}{\alpha_1}\right), \\ q_{\phi,2}(x_2) &= \frac{1}{|\alpha_2|}q\left(\frac{x_2 - \beta_2}{\alpha_2}\right). \end{aligned}$$

When $\phi = (\alpha_1, \alpha_2, \beta_1, \beta_2) = (1, 1, 0, 0)$, $\pi^N_{\mathrm{DPIS}}(h) = \pi^N_{\mathrm{IS}}(h)$. Setting

$$p_{\phi,1}(x) = p(\alpha_1 x + \beta_1), \quad p_{\phi,2}(x) = p(\alpha_2 x + \beta_2) \quad \text{and} \quad h_{\phi,1}(x) = h(\alpha_1 x + \beta_1),$$

the asymptotic variance (2.11) now depends on ϕ and reads

$$\mathcal{V}^{\infty}_{\phi,1,2}(h) = \frac{1}{p(1)^2} \times q\left(\frac{1}{q^2}\left(|\alpha_1|p_{\phi,1}h_{\phi,1} - |\alpha_2|p_{\phi,2}\pi(h)\right)^2\right).$$

Let us remark that we have transformed a mathematical expectation according to $q_{\phi,1,2}$ in (2.11) to an expectation according to q (which does not depend on ϕ). Consequently, it can be approximated by a Monte Carlo estimator in which the samples are drawn according to q and do not depend on ϕ,

$$\mathcal{V}^{N,\infty}_{\phi,1,2}(h) = \frac{N}{\left(\sum_{i=1}^N \omega^i_{\phi,2}\right)^2} \sum_{i=1}^N \frac{1}{q(\xi^i)^2} \left(|\alpha_1| p_{1,\phi}(\xi^i) h_{1,\phi}(\xi^i) - |\alpha_2| p_{\phi,2}(\xi^i) \pi^N_{\phi,\mathrm{DPIS}}(h)\right)^2. \tag{2.13}$$

The fact that the samples do not depend on ϕ is particularly desirable in optimization problems. Actually, the reparametrization of (ξ^i_1, ξ^i_2) as a differentiable function (w.r.t. ϕ) of a random variable is a particular case of the popular reparametrization trick in machine learning and introduced in Kingma and Welling (2014). Finally, (2.13) can be optimized w.r.t. ϕ with a given optimization method. In particular, any gradient descent method can be used if p and h are also differentiable. Alg. 2.4 gives an example of the tuning of the DPIS estimator with a basic gradient descent method.

Algorithm 2.4 DPIS Algorithm

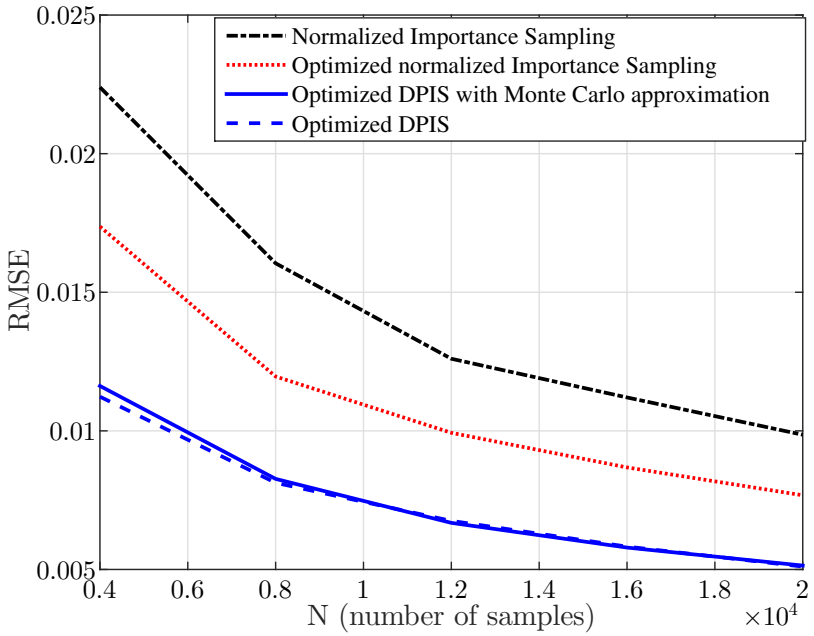
Require: q, a learning rate ϵ, a threshold S
  for $i \in [1:N]$ **do**
    Sample $\xi^i \sim q(x)$.
  end for
  Initialize $\phi = \phi^{(*)}$ randomly.
  while $|| \nabla \mathcal{V}^{N,\infty}_{\phi,1,2}(h)\Big|_{\phi=\phi^{(*)}} || \geq S$ **do**
    Set $\phi^{(*)} = \phi^{(*)} - \epsilon \nabla \mathcal{V}^{N,\infty}_{\phi,1,2}(h)\Big|_{\phi=\phi^{(*)}}$
  end while
  return $\phi^{(*)} = (\alpha_1^{(*)}, \alpha_2^{(*)}, \beta_1^{(*)}, \beta_2^{(*)})$

Experimental result (Fig. 2.1) - We compare the performance of our optimized DPIS estimator based on an initial distribution q with the classical importance sampling estimator (2.4) based on q but also with an optimized importance sampling estimator in two scenarios. In the first scenario (Fig. 2.1a), we compare the resulting DPIS estimator based on the optimization of the Monte Carlo approximation (2.13) of the asymptotic variance with an exact optimization of (2.11) (it is computable for this particular case). In the second scenario, we compute our estimator in a Bayesian scenario where $\pi(x) \propto p(x)p(y|x)$.

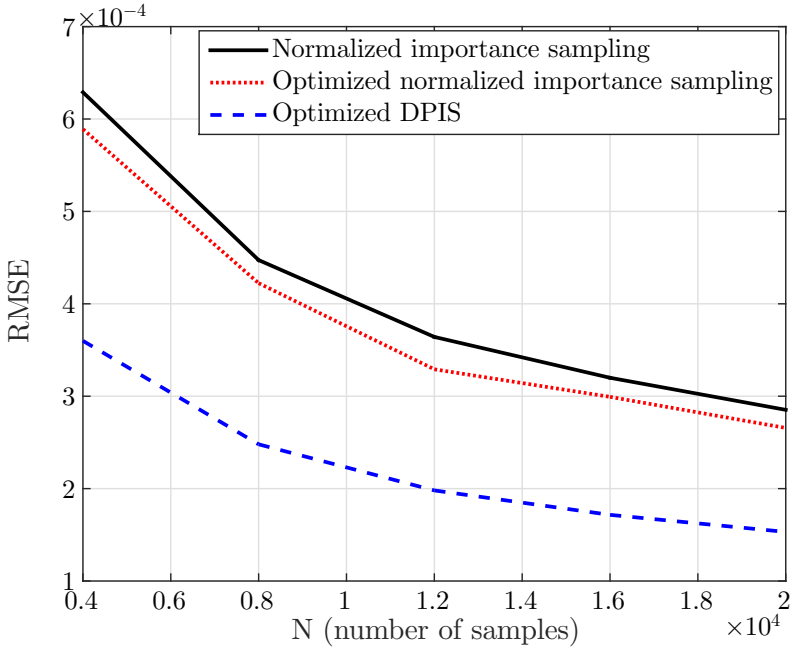
2.3 The Rubin's independent resampling mechanism

The DPIS idea is promising in the static case but remains difficult to apply in the sequential one because it would involve an optimization procedure at each time step. We revisit the Rubin's SIR mechanism in an alternative direction. We first propose a procedure to mitigate the support shrinkage of the (multinomial) resampling step of particle filters. This enables us to build an implicit relevant importance distribution. It is next exploited for the sequential case.

Main Idea - We still consider the problem of approximating $\pi(h)$. In Alg. 2.1, the local effect of the resampling step can be measured through the estimator $\pi^N_{\mathrm{SIR}}(h)$ in (2.6). It involves discrete index variables



(a) Toy model : $h(x) = x^2$, $\pi(x) = q(x) = \mathcal{N}(x; 0; 1)$. Simulation with 100 MC runs comparing DPIS estimators, using either exact or approximated asymptotic variance expressions for minimization, with classical importance sampling and optimized classical importance sampling estimators.



(b) Bayesian non-linear Gaussian model, $h(x) = \mathcal{N}(x; 3; 1)$, $p(x) = q(x) = \mathcal{N}(x; 1; 1)$, $p(y|x) = \mathcal{N}(y; x^2; 1)$. Simulation with 100 Monte Carlo runs comparing the DPIS estimator with classical importance sampling and optimized importance sampling estimators.

Figure 2.1

$\{A^i\}_{i=1}^N$ sampled according to the categorical distribution $\mathrm{Cat}(\{\mathcal{W}^l\}_{l\in[1:N]})$. So we focus on the samples

$$\tilde{\xi}^i = \xi^{A^i}, \quad \text{for all } i \in [1:N]$$

produced by the SIR mechanism and we have the following Proposition.

Proposition 2.2. Let us consider the samples $\{\tilde{\xi}^i\}_{i=1}^N$ produced by the SIR mechanism and let us emphasize that the normalized weights in (2.5) are a realization of a random variable function of all the particles $\{\xi^i\}_{i=1}^N$, *i.e.*

$$\mathcal{W}(\xi^1, \cdots, \xi^n) = \frac{\frac{p(\xi^1)}{q(\xi^1)}}{\sum_{i=1}^N \frac{p(\xi^i)}{q(\xi^i)}} \tag{2.14}$$

(so $\mathcal{W}^i = \mathcal{W}(\xi^i, \xi^1, \cdots, \xi^{i-1}, \xi^{i+1}, \cdots, \xi^n)$ in (2.5)). Then $\{\tilde{\xi}^i\}_{i=1}^N$ are identically distributed according to $\tilde{q}^N$ with

$$\tilde{q}^N(x) = N\mathbb{E}\left(\mathcal{W}(x, \xi^2, \cdots, \xi^n)|\xi^1 = x\right) q(x)$$

but are dependent.

To illustrate this result, let us consider the particular case where π and q are pdfs w.r.t. the Lebesgue measure on $\mathbb{R}$. Then

$$\mathbb{E}\left(\mathcal{W}(x, \xi^2, \cdots, \xi^N)|\xi^1 = x\right) = \int_{\mathbb{R}^{N-1}} \frac{\frac{p(x)}{q(x)}}{\frac{p(x)}{q(x)} + \sum_{i=2}^N \frac{p(x^i)}{q(x^i)}} \prod_{i=2}^N q(x^i) \mathrm{d}x^2 \cdots \mathrm{d}x^N$$

and $\tilde{q}^N(x)$ is also a pdf w.r.t. the Lebesgue measure on $\mathbb{R}$. However, since there is a non null probability that two final samples are equal, it means that $\{\tilde{\xi}^i\}_{i=1}^N$ are dependent. Now, what happens if we have at our disposal i.i.d samples according to $\tilde{q}^N(x)$? To measure the impact, we introduce the (fictional, for the moment) estimator

$$\pi^N_{\mathrm{I-SIR}}(h) = \frac{1}{N}\sum_{i=1}^N h(\xi^i), \quad \{\xi^i\}_{i=1}^N \overset{\text{i.i.d}}{\sim} \tilde{q}^N \tag{2.15}$$

We then have the following results.

Proposition 2.3. Let us consider the three estimators $\pi^N_{\mathrm{IS}}(h)$, $\pi^N_{\mathrm{SIR}}(h)$ and $\pi^N_{\mathrm{I-SIR}}(h)$ in (2.4), (2.6) and (2.15), respectively. Then

$$\begin{aligned} \mathbb{E}(\pi^N_{\mathrm{IS}}(h)) = \mathbb{E}(\pi^N_{\mathrm{SIR}}(h)) = \mathbb{E}(\pi^N_{\mathrm{I-SIR}}(h)), \\ \mathbb{V}\mathrm{ar}(\pi^N_{\mathrm{SIR}}(h)) = \mathbb{V}\mathrm{ar}(\pi^N_{\mathrm{I-SIR}}(h)) + \frac{N-1}{N}\mathbb{V}\mathrm{ar}(\pi^N_{\mathrm{IS}}(h)). \end{aligned} \tag{2.16}$$

In addition, if $\mathbb{E}(\pi^N_{\mathrm{I-SIR}}(h)^2) < \infty$, then $\pi^N_{\mathrm{I-SIR}}(h)$ satisfies a CLT,

$$\sqrt{N}(\pi^N_{\mathrm{I-SIR}}(h) - \pi(h)) \underset{N\to\infty}{\Longrightarrow} \mathcal{N}(0; \pi(h - \pi(h))^2). \tag{2.17}$$

Let us comment this result. First, it shows that the variance of $\pi^N_{\mathrm{I-SIR}}(h)$ is lower than that of $\pi^N_{\mathrm{SIR}}(h)$. This is not surprising since $\frac{N-1}{N}\mathbb{V}\mathrm{ar}(\pi^N(h))$ in (2.16) actually coincides with the sum of (positive) covariance terms between the samples involved in the dependent resampling scheme. In addition, the CLT (2.17)

shows that in an asymptotic regime, getting i.i.d. samples according to $\tilde{q}^N$ is equivalent to sampling according to the target distribution π. The comparison between $\pi^N_{\mathrm{I-SIR}}(h)$ and $\pi^N_{\mathrm{IS}}(h)$ depends on h and on q and coincides with the traditional discussion between a crude and an importance sampling estimator. In particular, if q has been tuned in function of h, then $\pi^N_{\mathrm{IS}}(h)$ may be more interesting than $\pi^N_{\mathrm{I-SIR}}(h)$. By contrast, $\pi^N_{\mathrm{I-SIR}}(h)$ is more adapted for a large class of functions h.

As we have just seen, i.i.d. samples according to $\tilde{q}^N$ provide an alternative and interesting estimator, at least from an asymptotic point of view. However, considering that $\tilde{q}^N$ can be seen as a particular importance distribution for a given N, we are able to propose an importance sampling estimator based on $\tilde{q}^N$,

$$\pi^N_{\mathrm{I-SIR-w}}(h) = \sum_{i=1}^N \mathcal{W}^i h(\xi^i), \quad \{\xi^i\}_{i=1}^N \overset{\text{i.i.d}}{\sim} \tilde{q}$$

where

$$\mathcal{W}^i = \Omega^{-1}\omega^i, \quad \omega^i = \frac{p(\xi^i)}{\tilde{q}^N(\xi^i)} \quad \text{and} \quad \Omega = \sum_{i=1}^N \omega^i, \text{ for all } i \in [1:N]. \tag{2.18}$$

Note that the importance distribution $\tilde{q}^N$ and so the importance weights $\{\omega^i\}_{i=1}^N$ now depend on N. The objective of this weighted estimator is to correct the fact that $\tilde{q}^N$ is different from π, from a non asymptotical point of view. The statistical interest of $\pi^N_{\mathrm{I-SIR-w}}(h)$ w.r.t. $\pi^N_{\mathrm{IS}}(h)$ can be highlighted in the case where the number of samples obtained from $\tilde{q}^N$ is different from N.

Drawing i.i.d. samples from $\tilde{q}^N$ (Alg. 2.5) - As we have just seen, if we replace the classical dependent resampling by a resampling scheme which achieves to produce independent samples without affecting the marginal distribution of the resampled particles, then we eliminate the support shrinkage but we also obtain estimators which may be more interesting than the estimator $\pi^N_{\mathrm{IS}}(h)$. However, obtaining such samples and computing their weights in (2.18). is not direct. We provide a procedure in Alg. 2.5 with a computational cost in $\mathcal{O}(N^2)$. It is based on N^2 samples according to q which can be latter recycled to approximate $\tilde{q}^N(x)$ and so the new importance weights in (2.18). An unbiased estimator of $\mathbb{E}\left(\mathcal{W}(\xi^i, \xi^{i_1}, \cdots, \xi^{i_{N-1}})|\xi^i\right)$ is indeed given by

$$\mathbb{E}\left(\mathcal{W}(\xi^i, \xi^{i_1}, \cdots, \xi^{i_{N-1}})|\xi^i\right) \approx \frac{1}{N}\sum_{j=1}^N \mathcal{W}(\xi^i, \xi^{1,j}, \cdots, \xi^{i-1,j}, \xi^{i+1,j}, \cdots \xi^{N,j}), \quad \{\xi^{i,j}\}_{i,j=1}^N \overset{\text{i.i.d.}}{\sim} q \tag{2.19}$$

and deduced from the computed weights $\{\omega^{i,j}\}_{i,j=1}^N$ of Alg. 2.5.

Due to its computational cost, Alg. 2.5 should also be compared to a SIR algorithm which first samples N^2 particles and next resamples N particles among the N^2 ones. In this case, the N resampled particles are dependent according to $\tilde{q}^{N^2}$ and it is easy to check that it satisfies the same CLT (2.17) as the I-SIR estimator. However, an advantage of the I-SIR estimator is that it can be parallelized since the N resampling steps are independent.

Experiments (Fig. 2.2) - We consider a static linear and Gaussian model, $\pi(x) \propto p(x)p(y|x) \propto \mathcal{N}(x;0;10)\mathcal{N}(y;x;3)$, $h(x) = x$. We compute several estimators based on the same number of final samples N: the SIR estimator $\pi^N_{\mathrm{SIR}}(h)$ based on N intermediate samples; the SIR estimator $\pi^N_{\mathrm{SIR-2}}(h)$ based on N^2 intermediate samples; the importance sampling estimator $\pi^N_{\mathrm{IS}}(h)$; our estimators $\pi^N_{\mathrm{I-SIR}}(h)$ and $\pi^N_{\mathrm{I-SIR-w}}(h)$ which have the same computational cost as $\pi^N_{\mathrm{SIR-2}}(h)$ but which can be parallelized.

Algorithm 2.5 I-SIR mechanism

Require: $p(x)$ such that $\pi(x) \propto p(x)$
  for $i \in [1:N]$ **do**
    for $j \in [1:N]$ **do**
      Sample $\xi^{i,j} \sim q(x)$.
      Set $\omega^{i,j} = \frac{p(\xi^{i,j})}{q(\xi^{i,j})}$.
    end for
    Sample $A^i \sim \text{Categorical}\left(\left\{\omega^{i,l} \times \left(\sum_{j=1}^N \omega^{i,j}\right)^{-1}\right\}_{l\in[1:N]}\right)$.
  end for
  return $\{\omega^{i,j}, \xi^i = \xi^{i,A^i}\}_{i,j=1}^N$

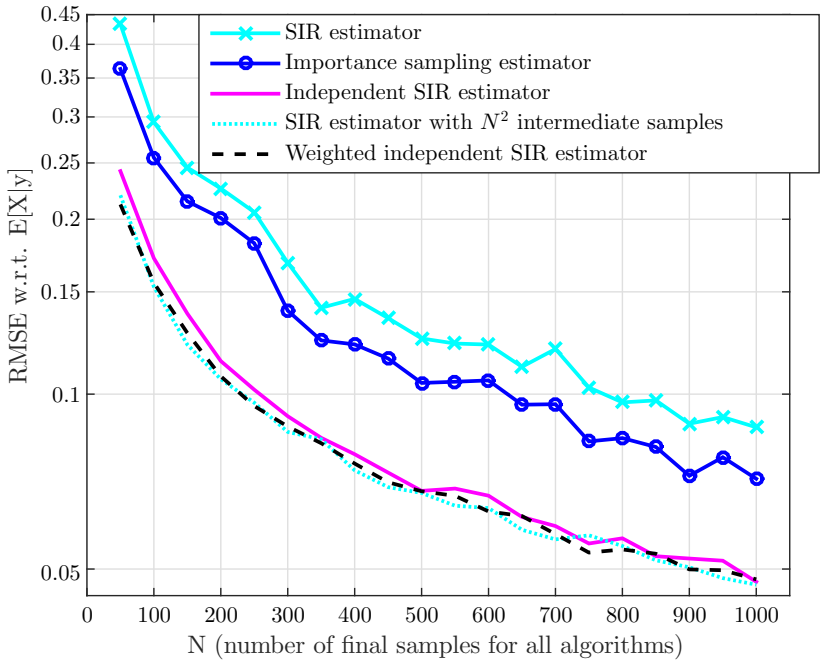


Figure 2.2: Static linear and Gaussian model - Bayesian estimates of $\pi(x)$ based on the independent resampling mechanism outperform the estimators based on the traditional IS and SIR mechanisms although they require an extra computational cost. Other simulations show that for small N ($N < 100$), the estimator based on weighted i.i.d. samples from $\tilde{q}^N$ slightly outperforms the estimator based on dependent samples from $\tilde{q}^{N^2}$ (which uses the same overall computational cost), while for large N the performances coincide.

2.4 Sequential independent resampling mechanism: an implicit APF

In this section, we extend our resampling scheme to the sequential case. It requires some adaptation since Alg. 2.2 produces *conditionally* identically distributed particles at time t.

An adaptation to the sequential case (Alg. 2.6) - In order to evaluate the impact of the resampling step of Alg. 2.2, we consider the *sampling* $\rightarrow$ *weighting* $\rightarrow$ *resampling* loop, that is to say the mechanism which transforms $\{\xi_{t-1}^{A_{t-1}^i}\}_{i=1}^N$ in $\{\xi_t^{A_t^i}\}_{i=1}^N$. It is possible to extend Proposition 2.3 for the sequential case; former pdf $q^{\tilde{N}}$ now coincides with the conditional distribution of $\xi_t^{A_t^i}$ given $\{\xi_{t-1}^{A_{t-1}^j}\}_{j=1}^N$.

Proposition 2.4. Let us consider the samples produced by Alg. 2.2. Given $\{\xi_{t-1}^{A_{t-1}^j}\}_{j=1}^N$, the *sampling* $\rightarrow$ *weighting* $\rightarrow$ *resampling* loop produces identically distributed (but dependent) particles $\{\xi_t^{A_t^i}\}_{i=1}^N$ according to

$$\tilde{q}_t^N(x) = \sum_{i=1}^N m_t^i(x) q_t(x|\xi_{t-1}^{A_{t-1}^i}),$$

where $m_t^i(x)$ is the conditional expectation of the i-th normalized importance weight

$$\mathcal{W}_t^i(\xi_t^i, \xi_t^1, \cdots, \xi_t^{i-1}, \xi_t^{i+1}, \cdots, \xi_t^N) = \frac{\frac{f_t(\xi_t^i|\xi_{t-1}^i) g_t(y_t|\xi_t^i)}{q_t(\xi_t^i|\xi_{t-1}^i)}}{\frac{f_t(\xi_t^i|\xi_{t-1}^i) g_t(y_t|\xi_t^i)}{q_t(\xi_t^i|\xi_{t-1}^i)} + \sum_{j\neq i} \frac{f_t(\xi_t^j|\xi_{t-1}^j) g_t(y_t|\xi_t^j)}{q_t(\xi_t^j|\xi_{t-1}^j)}} \tag{2.20}$$

given $\{\xi_{t-1}^{A_{t-1}^j}\}_{j=1}^N$ and $\xi_t^i = x$,

$$m_t^i(x) = \mathbb{E}\left(\mathcal{W}_t^i(\xi_t^i = x, \xi_t^1, \cdots, \xi_t^{i-1}, \xi_t^{i+1}, \cdots, \xi_t^N)|x, \{\xi_{t-1}^{A_{t-1}^j}\}_{j=1}^N\right). \tag{2.21}$$

For the same reasons as those presented in the static case, it is interesting to deal with (conditionally) i.i.d. samples: starting from a common set of samples, a direct adaptation of Prop. 2.3 shows that an estimator based on i.i.d. samples according to $\tilde{q}_t^N(x)$ has a lower variance than that computed after the classical resampling step of Alg. 2.2. Consequently, obtaining conditionally i.i.d. samples is a way to to keep the marginal distribution of each sample $\xi_t^{A_t^i}$ but also to cancel locally the detrimental effect of the resampling step. Note that it is a conditional property and that the global trajectories remain dependent. As in the static case, Alg. 2.6 describes a procedure to obtain such i.i.d. samples according to $\tilde{q}_t^N(x)$ and Fig. 2.5 provides an explicative scheme. It also requires the sampling of N^2 particles $\{\xi_t^{i,j}\}_{i,j=1}^N$ but it can be parallelized in the light of the Island particle filter (Vergé et al., 2015). From now on, in the framework the I-SIR particle filter, we note

$$\xi_t^i = \xi_t^{A_t^i, i}, \text{ for all } t \in \mathbb{N} \text{ and for all } i \in [1:N].$$

Relation with APF Alg. 2.3 - We now show that the *sampling* $\rightarrow$ *weighting* $\rightarrow$ *resampling* loop of Alg. 2.6 can be interpreted as a *resampling* $\rightarrow$ *sampling* step of the APF. Indeed, the conditional importance distribution $\tilde{q}_t^N(x)$ can be rewritten as a mixture

$$\tilde{q}_t^N(x) = \sum_{i=1}^N \mathbb{E}\left(\mathcal{W}_t^i \middle| \{\xi_{t-1}^j\}_{j=1}^N\right) \frac{m_t^i(x) q_t(x|\xi_{t-1}^i)}{\mathbb{E}\left(\mathcal{W}_t^i|\{\xi_{t-1}^j\}_{j=1}^N\right)}. \tag{2.22}$$

Algorithm 2.6 I-SIR particle filter

Require: $\{\omega_{t-1}^{i,j}, \xi_{t-1}^{i,j}\}_{i=1}^N$, y_t
  for $i \in [1:N]$ **do**
    Sample $A_{t-1}^i \sim \text{Categorical}\left(\{\mathcal{W}_{t-1}^{l,i} = \omega_{t-1}^{l,i} \times \left(\sum_{l=1}^N \omega_{t-1}^{l,i}\right)^{-1}\}_{l\in[1:N]}\right)$.
    for $j \in [1:N]$ **do**
      Sample $\xi_t^{i,j} \sim q_t(x_t|\xi_{t-1}^{A_{t-1}^i,i})$.
      Set $\omega_t^{i,j} = \frac{f_t(\xi_t^{i,j}|\xi_{t-1}^{i,A_{t-1}^i}) g_t(y_t|\xi_t^{i,j})}{q_t(\xi_t^{i,j}|\xi_{t-1}^{i,A_{t-1}^i})}$
    end for
  end for
  return $\{\omega_t^{i,j}, \xi_t^{i,j}\}_{i,j=1}^N$

And indeed, the pair $(A_t^i, \xi_t^i = \xi_t^{A_t^i,i})$ produced by Alg. 2.6 is distributed (in augmented dimension) according to $\tilde{q}_t^N(x)$: it is easy to check that A_t^i is distributed according to $\text{Cat}(\{\mathbb{E}\left(\mathcal{W}_t^l|\{\xi_{t-1}^j\}_{j=1}^N\right)\}_{l\in[1:N]})$, where $\mathcal{W}_t^l$ is defined in (2.14); and given A_t^i, $\xi_t^i = \xi_t^{A_t^i,i}$ is distributed according to $m_t^i(x) q_t(x|\xi_{t-1}^{A_{t-1}^i})$ $\mathbb{E}(\mathcal{W}_t^i|\{\xi_{t-1}^j\}_{j=1}^N)^{-1}$ (Alg. 2.6 describes the distribution of $\{\xi_t^{i,j}\}_{i,j=1}^N$ and next that of A_t^i given $\{\xi_t^{i,j}\}_{i,j=1}^N$). Let us now interpret the sense of sampling (indirectly) according to this mixture. Starting from a given set of initial particles $\{\xi_{t-1}^i\}_{i=1}^N$, they are first resampled according to the expectation of the normalized importance weights of the SIR algorithm at time t. In other words, samples ξ_{t-1}^i which will likely produce large important weights tend to be selected; but even if such samples have been selected, it not ensured that the final associated weights will be large, so they are extended according to a distribution proportional to $m_t(x_t) q_t(x_t|x_{t-1})$, *i.e.* a distribution which puts mass where the conditional expectation of the normalized weight and the sampling distribution are large. Thus, the mixture (2.22) can be interpreted as a kind of optimal importance distribution which aims at selecting and next guiding the samples in an optimal sense when we only have at our disposal an importance distribution $q_t(x_t|x_{t-1})$. As the FA-APF which produces a mixture from the SIR algorithm with the optimal importance distribution $p(x_t|x_{t-1}, y_t)$ (and so with the associated weights $p(y_t|x_{t-1})$), our mixture is deduced from any importance distribution of the the SIR algorithm and coincides with the FA-APF in the particular case of the optimal importance distribution.

In conclusion, if we now start from a set of weighting samples $\{\omega_{t-1}^i, \xi_{t-1}^i\}_{i=1}^N$, it is possible to apply the sampling, weighting and resampling steps of Alg. 2.5 (up to the introduction of ω_{t-1}^j in the expression of $\omega_t^{i,j}$) to obtain i.i.d. samples according to a relevant mixture in the APF framework. It remains to weight samples (ξ_t^i, A_t^i) produced by Alg. 2.5 by the APF importance weights,

$$\omega_t^i = \frac{f_t(\xi_t^i|\xi_{t-1}^{A_t^i}) g_t(y_t|\xi_t^i)}{m_t^{A_t^i}(\xi_t^i) q_t(\xi_t^i|\xi_{t-1}^{A_{t-1}})}.$$

As in the static case, ω_t^i is not computable because it relies on $m_t^{A_t^i}(\xi_t^i)$. However, remember that it coincides with the conditional expectation of the A_t^i-th normalized importance weight so a crude Monte Carlo estimator of $m_t^{A_t^i}(\xi_t^i)$ can be estimated from the intermediate samples $\{\xi_t^{l,j}\}$, $j \in [1:N]$ and $l \neq A_t^i$. This is nothing more that the extension of the computation (2.19) in the static case. In summary, let us retain the following conclusions:

- when we study the *sampling* $\rightarrow$ *weighting* $\rightarrow$ *resampling* loop of Alg. 2.2, we obtain dependent and unweighted samples according to an implicit mixture; this loop can be reinterpreted as a degenerated iteration of the APF;
- if the objective is to reduce the effect of the support shrinkage in the context of particle filters based on the SIR mechanism, Alg. 2.6 can be applied and produces unweighted but conditionnally i.i.d. samples according to the previous mixture;
- based on the observation that this mixture has a relevant interpretation, it can be used in the APF framework. In this case, the final samples are (approximately) weighted.

Simulations (Figs. 2.3 and 2.4) - We consider two simulations. In the first one, we consider the ARCH model

$$\begin{aligned} f_t(x_t|x_{t-1}) &= \mathcal{N}\left(x_t; 0; \beta_0 + \beta_1 x_{t-1}^2\right), \\ g_t(y_t|x_t) &= \mathcal{N}\left(y_t; x_t; 1\right) \end{aligned}$$

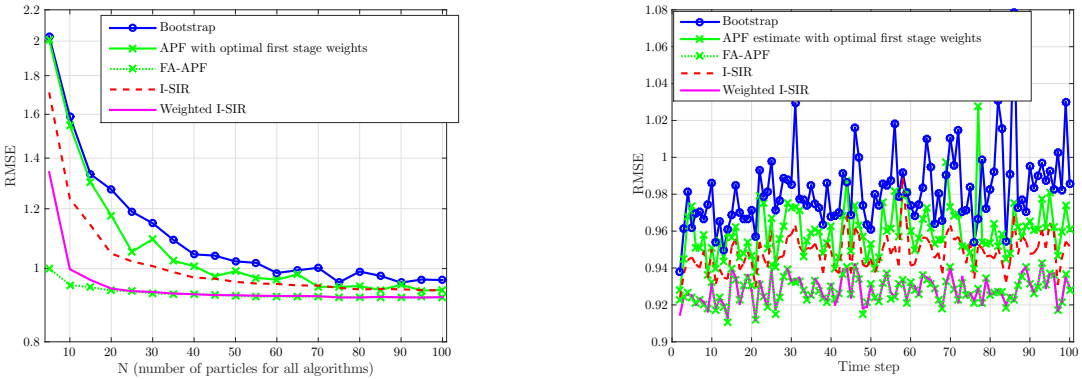
in which the FA-APF is computable and we compare all the discussed algorithms for the estimation of $\phi_t(h)$ with $h(x) = x$, without discussing of the extra computational cost involved by our algorithm. The reason why is that we want to compare the implicit mixture $\tilde{q}_t^N$ obtained from $q_t(x_t|x_{t-1})$ with those used in the APF framework. For our algorithms, we use the conditional importance distribution of the bootstrap algorithm, $q_t(x_t|x_{t-1}) = f_t(x_t|x_{t-1})$.

In the second scenario, we consider a tracking scenario (*i.e.* we estimate the position and the velocity of a target X_t) from informative range-bearing measurements,

$$\begin{aligned} f_t(x_t|x_{t-1}) &= \mathcal{N}\left(x_t; \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \otimes \mathbf{I}_2 x_{t-1}; \begin{bmatrix} 1/3 & 1/2 \\ 1/2 & 1 \end{bmatrix} \otimes \mathbf{I}_2\right), \\ g_t(y_t|x_t) &= \mathcal{N}\left(y_t; g(x_t); \begin{bmatrix} \sigma_\rho^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix}\right), \end{aligned}$$

where g is the non linear function which computes the polar coordinates from the Cartesian coordinates x_t. In Fig. 2.4, we compare different estimators for a given budget of samples. If the size of the support after our resampling step is M, Alg. 2.6 is based on $M^2 + M$ sampling operations which can be parallelized; on the other hand Alg. 2.2 requires $2N$ sampling operations, so we set $N = \frac{M^2+M}{2}$.

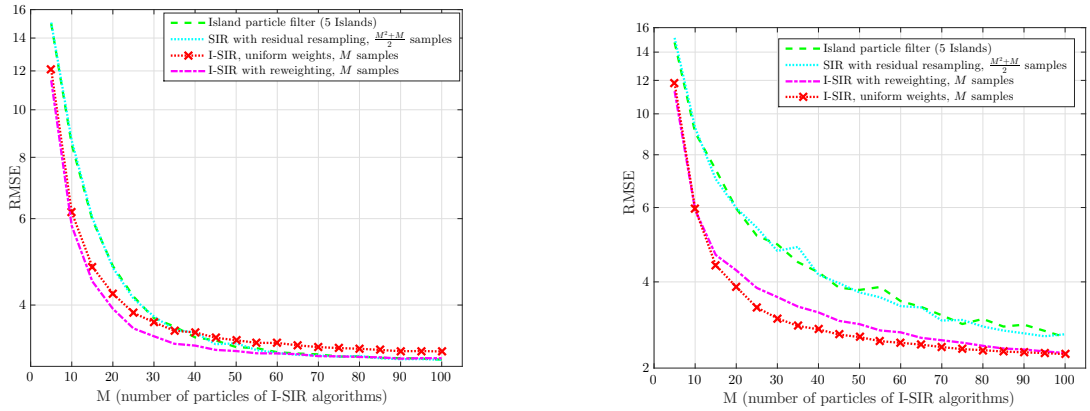
Reducing the computational cost from semi-independent resampling (Fig. 2.5) - Alg. 2.6 requires N^2 sampling operations. We propose an iterative procedure which can be seen as an intermediate solution between the SIR and I-SIR particle filters. This procedure is called *semi-independent resampling* and relies on an hyperparameter k, $0 \leq k \leq N$. It coincides with the degree of diversity that we want to introduce after the resampling step. Rather than presenting an algorithm, we explain the core idea of the procedure through Fig. 2.5. In this figure, we have represented the set of N^2 intermediate particles $\{\xi_t^{i,j}\}_{i,j=1}^N$ produced by Alg. 2.6; remember that for each block $i \in [1:N]$, a particle $\xi_t^i = \xi_t^{A_t^i, i}$ is resampled. Note also that the classical SIR particle filter can be seen as a particular case of this figure in which the blocks are replicated, *i.e.* $\xi_t^{i,j} = \xi_t^{i,j'}$ for all $(j, j') \in [1:N] \times [1:N]$. The semi-independent resampling solution



(a) RMSE w.r.t. number of particles

(b) RMSE w.r.t. time

Figure 2.3: ARCH model - $R = 1$, $\beta_0 = 3$ and $\beta_1 = 0.75$ - (a) The estimator based on the independent resampling mechanism with a final reweighting has the same performances as the estimator deduced from the FA-APF. The final reweighting mechanism is beneficial when compared to the use of uniform weights. - (b) RMSE w.r.t time of the various estimates for $N = 100$



(a) $\sigma_Q = \sqrt{10}$, $\sigma_\rho = 0.25$ and $\sigma_\theta = \frac{\pi}{720}$

(b) $\sigma_Q = \sqrt{10}$, $\sigma_\rho = 0.05$ and $\sigma_\theta = \frac{\pi}{3600}$

Figure 2.4: Target tracking model from range-bearing measurements - (a) the independent resampling procedure with final weighting outperforms the other estimators and is particularly interesting when the number of final samples is weak - (b) in the informative case, all estimates suffer from the degeneration of the importance weights except that based on the unweighted independent resampling algorithm. To achieve the same performances as $\widehat{\Theta}_k^{\mathrm{I-SIR-w},M}$ with $M = 20$, the classical particle filter uses $N = (50^2 + 50)/2 = 1275$ samples.

that we propose is an intermediate solution in which the $j+1$-th block is first replicated from the j-th one; next among the N samples of this $j+1$-th block, k samples are selected uniformly and are next replaced by a new sample drawing from the corresponding conditional importance distribution. Thus, the diversity in introduced iteratively over the construction of the blocks. We can show that starting from a common set of samples, the variance of the semi-independent resampled estimators is a decreasing function of k. Experimentally, we can observe that a value of $k = N/2$ produces approximately the same performance as the independent resampling procedure ($k = N$). A parallelized version can be also deduced, provided that the j blocks, for $j \in [2 : N]$, are built from the first one. A direct consequence is that the diversity is reduced w.r.t. the previous construction. Experimentally, a value of $k = 4N/5$ offers similar performances to the independent resampling procedure.

In Fig. 2.6, we consider again the range-bearing tracking scenario and we compare different algorithms with a given budget of samples. Since the complexity of the semi-independent resampling in terms of sampling operations is $N + k(N-1) + N$, for $N = 100$ and $k = N/2$, we compare it with Alg. 2.6 with 72 samples and with Alg. 2.2 with $N + (N-1)k/2 = 2575$ particles.

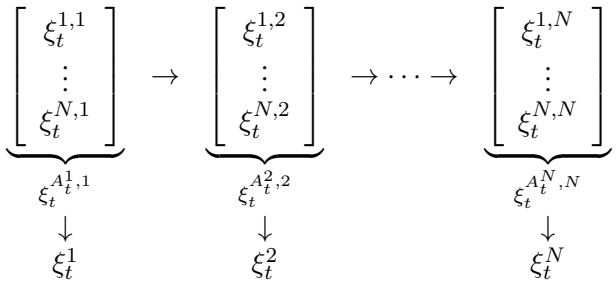


Figure 2.5: The classical, independent and semi-independent resampling mechanisms. Each scheme draws N supports $\xi_t^{i,:}$ and redraws one sample ξ_t^i out of each support. The difference lies in the way $\xi_t^{i,:}$ is built from $\xi_t^{i-1,:}$: $\xi_t^{i,j}$ is a copy of $\xi_t^{i-1,j}$ in the classical case; is a new particle in the independent case; and can be either copied of redrawn in the intermediate, semi-independent case.

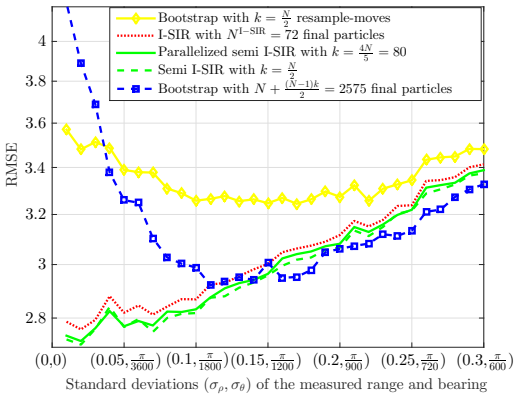


Figure 2.6: Tracking model, $\sigma_\rho \in [0.01, 0.3]$ and $\sigma_\theta \in [\frac{\pi}{18000}, \frac{\pi}{600}]$. Our estimators based on the I-SIR and the semi-I-SIR algorithm are compared with traditional SMC algorithms for a fixed budget of sampling operations.

CHAPTER 3

Estimating asymptotic variances with recycled particles

We continue to investigate SMC methods in the HMC model. However, instead of proposing alternative sampling algorithms, we rather exploit and recycle the current outputs of traditional sequential Monte Carlo methods such as the bootstrap and the FFBS algorithms for estimating the asymptotic variance of their associated estimators. We also provide a statistical analysis of the variance estimators and practical algorithms to compute them.

The analysis proposed in this Chapter are a synthesis of some work I realized during the supervision of Y. Janati (2020-2023) with S. Le Corff. The proofs are omitted for clarity but can be found in [18].

3.1 Background

Before recalling the two estimators for which we look an estimator of their asymptotic variance, we introduce some notations with the objective of simplifying the presentation of our main results.

General notations - In this chapter, the unit function $\mathbb{1}$ satisfies $\mathbb{1}(x) = 1$ for all x in X. When consider an augmented space such as $\mathsf{X}^{t+1} \times \mathsf{X}^{t+1}$, the functionals from this augmented space are denoted as $\mathbf{h}$. From a given augmented space, we also define a functional $h \otimes h'$ which satisfies $(h \otimes h')(x, x') = h(x)h'(x')$. For $(a, b) \in \mathbb{N}^2$, $a \leq b$, the set of integer between a and b is denoted as $[a : b]$, and we define $[b] = [1 : b]$. Given a set of particles $\{\xi_s^i\}_{s\in[0:t],i\in[1:N]}$, we write $\xi_t^{1:N} = (\xi_t^1, \cdots, \xi_t^N)$ the set of particles at time t, $\xi_{0:t}^{k_{0:t}} = (\xi_0^{k_0}, \cdots, \xi_t^{k_t})$ a particular trajectory indexed by $(k_0, \cdots, k_t)$ and $\boldsymbol{\xi}_{0:t}^{1:N} = \{\xi_{0:t}^{k_{0:t}}\}_{k_{0:t}\in[N]^{t+1}}$ the set of all trajectories.

Notation in HMCs - Since our estimators are conditional to a given set of observations, we give up the dependence in y_t and we introduce some notations related to the Feynmann-Kac framework (Del Moral, 2004). In model (1.6), the transitions of the Markov chain and the conditional likelihoods are denoted as

$$\begin{aligned} f_t(x_{t-1}, x_t) &= p(x_t|x_{t-1}), \\ g_t(x_t) &= p(y_t|x_t). \end{aligned}$$

In addition, the moment of a given functional h w.r.t. a conditional pdf (e.g. $f_t(x_{t-1}, x_t)$) is noted

$$f_t[h](x_{t-1}) = \int_{\mathsf{X}} f_t(x_{t-1}, x_t)h(x_t)\mathrm{d}\nu(x_t).$$

It is easy to check that in an HMC, the pair $\{X_t, Y_{t-1}\}_{t\in\mathbb{N}^*}$ is a Markov chain; its transitions are denoted as $\mathbf{Q}_t(x_{t-1}, x_t)$ and satisfy

$$\mathbf{Q}_t(x_{t-1}, x_t) = p(x_t, y_{t-1}|x_{t-1}) = g_{t-1}(x_{t-1})f_t(x_{t-1}, x_t).$$

By extension, we note $\mathbf{Q}_{s:t}(x_{s-1}, x_{s:t}) = p(x_{s:t}, y_{s-1:t-1}|x_{s-1})$ and we have

$$\mathbf{Q}_{s:t}(x_{s-1}, x_{s:t}) = \mathbf{Q}_s(x_{s-1}, x_s) \times \cdots \times \mathbf{Q}_t(x_{t-1}, x_t), \quad \text{if } s \leq t.$$

When $s > t$, we set $\mathbf{Q}_{s:t}[h](x_{s-1}) = h(x_{s-1})$. In our problem, the sequence of unormalized distributions

$$\gamma_{0:t}(x_{0:t}) = p(x_{0:t}, y_{0:t-1})$$

plays a key role and can be defined from

$$\gamma_0(x_0) = p(x_0), \quad \gamma_{0:t}(x_{0:t}) = \gamma_{0:t-1}(x_{0:t-1})\mathbf{Q}_t(x_{t-1}, x_t).$$

The marginal $p(x_t, y_{0:t-1})$ is denoted as $\gamma_t(x_t)$. The predictive and the filtering distributions, $\eta_t(x_t) = p(x_t|y_{0:t-1})$ and $\phi_t(x_t) = p(x_t|y_{0:t})$, satisfy

$$\eta_t(x_t) = \gamma_t^{-1}(1)\gamma_t(x_t), \quad \phi_t(x_t) = g_t(x_t)\eta_t(x_t)/\eta_t(g_t).$$

When we deal with the smoothing problem, it is interesting to consider the HMC in a backward way. Given $\{Y_t\}_{t\in\mathbb{N}}$, $\{X_t\}_{t\in\mathbb{N}}$ satisfies a backward Markov property since the smoothing distribution $\phi_{0:t|t}(x_{0:t}) = p(x_{0:t}|y_{0:t})$ can be written as

$$\phi_{0:t|t}(x_{0:t}) = \phi_t(x_t) \prod_{s=0}^{t-1} \underbrace{p(x_s|x_{s+1}, y_{0:s})}_{p(x_s|x_{s+1:t}, y_{0:t})}, \quad \text{for all } t \in \mathbb{N}.$$

We thus introduce the so-called Backward kernels $\mathbf{B}_{\phi_s}(x_{s+1}, x_s) = p(x_s|x_{s+1}, y_{0:s})$ which satisfy

$$\mathbf{B}_{\phi_s}(x_{s+1}, x_s) = \frac{f_{s+1}(x_s, x_{s+1})\phi_s(x_s)}{\int_{\mathsf{X}} f_{s+1}(x_s, x_{s+1})\phi_s(x_s)\mathrm{d}\nu(x_s)}, \quad \text{for all } s \in [0 : t-1].$$

Finally, the kernel defined by $\mathbf{T}_t(x_t, x_{0:t-1}) = p(x_{0:t-1}|x_t, y_{0:t})$ satisfies

$$\mathbf{T}_t(x_t, x_{0:t-1}) = \prod_{s=0}^{t-1} \mathbf{B}_{\phi_s}(x_{s+1}, x_s), \quad \text{for all } t \geq 1.$$

The bootstrap particle filter - This algorithm is nothing more than a particular case of Alg. 2.2 of the previous chapter in which we set $q_t(x_t|x_{t-1}) = f_t(x_{t-1}, x_t)$ (Gordon et al., 1993). In this case, the importance weights ω_t^i reduce to the conditional likelihood of a particle with the current observation y_t, $\omega_t^i = g_t(\xi_t^i)$. As we will see, it can be relevant to retrace the genealogy of each particle ξ_t^i until time $s = 0$. We note the index of this ancestor $\mathrm{I}_{t,0}^i$ (it represents the index of the particle a time $t = 0$ which has finally produces ξ_t^i) The bootstrap algorithm is recalled in Alg. 3.1 and we include the sequential computation of $\mathrm{I}_{t,0}^i$. From Alg. 3.1, we deduce the approximations

$$\eta_t^N(h) = \frac{1}{N}\sum_{i=1}^N h(\xi_t^i), \quad \phi_t^N(h) = \sum_{i=1}^N \omega_t^i h(\xi_t^i).$$

Finally, based on the identities $\eta_{t-1}(g_{t-1}) = \frac{\gamma_t(1)}{\gamma_{t-1}(1)}$, $\gamma_t(1) = \prod_{s=1}^t \frac{\gamma_s(1)}{\gamma_{s-1}(1)}$, and remembering that $\gamma_t(h) = \gamma_t(1)\eta_t(h)$, we deduce an approximation of $\gamma_t(h)$,

$$\gamma_t^N(h) = \left[\prod_{s=1}^t \eta_{s-1}^N(g_{s-1})\right] \eta_t^N(h) = \left[\prod_{s=1}^t N^{-1}\Omega_s\right] \eta_t^N(h).$$

Algorithm 3.1 Bootstrap particle filter

Require: $\{\omega_{t-1}^i, \xi_{t-1}^i\}_{i=1}^N$, y_t
  Set $\Omega_{t-1} = \sum_{i=1}^N \omega_{t-1}^i$
  for $i \in [1:N]$ **do**
    Sample $A_{t-1}^i \overset{\text{i.i.d.}}{\sim} \text{Categorical}\left(\{\mathcal{W}_{t-1}^l = \omega_{t-1}^l \Omega_{t-1}^{-1}\}_{l\in[1:N]}\right)$.
    Set $\mathrm{E}_{t,0}^i = \mathrm{E}_{t-1,0}^{A_{t-1}^i}$.
    Sample $\xi_t^i \sim q_t(x_t|\xi_{t-1}^{A_{t-1}^i})$.
    Set $\omega_t^i = \frac{f_t(\xi_t^i|\xi_{t-1}^{A_{t-1}^i}) g_t(y_t|\xi_t^i)}{q_t(\xi_t^i|\xi_{t-1}^{A_{t-1}^i})}$.
  end for
  return $\{\omega_t^i, \xi_t^i\}_{i=1}^N$

It can be shown that $\gamma_t^N(h)$ is an unbiased estimator of $\gamma_t(h)$ (Del Moral, 2004); in the particular case where $h = \mathbf{1}$, we have an unbiased estimator of the likelihood $p(y_{0:t})$.

The estimators $\gamma_t^N(h)$, $\eta_t^N(h)$ and $\phi_t^N(h)$ have asymptotic properties which ensure the validity of Alg. 3.1 and which extend the static results of the previous chapter. Let us assume that

(A1) there exists a constant $G_\infty > 0$ such that for all $t \in \mathbb{N}$ and $x \in \mathsf{X}$, $0 < g_t(x) \leq G_\infty$.

Then for any bounded functional h, the three estimators converge almost surely to $\gamma_t^N(h)$, $\eta_t^N(h)$ and $\phi_t^N(h)$, respectively, and they also satisfy a CLT,

$$\begin{aligned}
\sqrt{N}\left(\gamma_t^N(h) - \gamma_t(h)\right) &\underset{N\to\infty}{\Longrightarrow} \mathcal{N}\left(0; \mathcal{V}_{\gamma,t}^\infty(h)\right), \\
\sqrt{N}\left(\eta_t^N(h) - \eta_t(h)\right) &\underset{N\to\infty}{\Longrightarrow} \mathcal{N}\left(0, \mathcal{V}_{\eta,t}^\infty(h)\right), \\
\sqrt{N}\left(\phi_t^N(h) - \phi_t(h)\right) &\underset{N\to\infty}{\Longrightarrow} \mathcal{N}\left(0, \mathcal{V}_{\phi,t}^\infty(h)\right),
\end{aligned} \tag{3.1}$$

where

$$\mathcal{V}_{\gamma,t}^\infty(h) = \sum_{s=0}^t \left\{ \gamma_s(\mathbf{1})\gamma_s\left(\mathbf{Q}_{s+1:t}[h]^2\right) - \gamma_t(h)^2 \right\}, \tag{3.2}$$

$$\mathcal{V}_{\eta,t}^\infty(h) = \sum_{s=0}^t \frac{\gamma_s(\mathbf{1})\gamma_s\left(\mathbf{Q}_{s+1:t}[h - \eta_t(h)]^2\right)}{\gamma_t(\mathbf{1})^2}, \tag{3.3}$$

$$\mathcal{V}_{\phi,t}^\infty(h) = \sum_{s=0}^t \frac{\gamma_s(\mathbf{1})\gamma_s\left(\mathbf{Q}_{s+1:t}[g_t\{h - \phi_t(h)\}]^2\right)}{\gamma_{t+1}(\mathbf{1})^2}. \tag{3.4}$$

With the previous notations, remember that $\mathbf{Q}_{s+1:t}[h](x_s)$ is a function of x_s; so $\gamma_s(\mathbf{Q}_{s+1:t}[h]^2)$ is nothing more than the expectation of this function w.r.t. γ_s. The asymptotic variances (3.2)-(3.4) describe the reliability of the estimators but are not computable in practice. We address their approximation under the constraint that the samples produced by Alg. 3.1 should be recycled. The reason why is that these estimators are computed in an online context, so their variance should be estimated at each time step. In our work, we mainly focus on the estimation of $\mathcal{V}_{\gamma,t}^\infty(h)$. The asymptotic variances $\mathcal{V}_{\eta,t}^\infty(h)$ and $\mathcal{V}_{\phi,t}^\infty(h)$ can be deduced

from $\mathcal{V}^{\infty}_{\gamma,t}(h)$ since it is easy to check that

$$\mathcal{V}^{\infty}_{\eta,t}(h) = \frac{\mathcal{V}_{\gamma,t}(h - \eta_t(h))}{\gamma_t(1)^2}, \quad \mathcal{V}^{\infty}_{\phi,t}(h) = \frac{\mathcal{V}_{\gamma,t}\{g_t(h - \phi_t(h))\}}{\gamma_{t+1}(1)^2}. \tag{3.5}$$

A first estimator of $\mathcal{V}^{\infty}_{\gamma,t}(h)$ has been proposed by Chan and Lai (2013) and relies on the ancestors of each sample ξ^i_t. The Chan & Lai estimator (CLE) reads

$$\mathcal{V}^{N}_{\eta,t}(h) = -N^{-1} \sum_{i,j \in [N]^2} \mathbb{1}_{\mathrm{E}^i_{t,0} \neq \mathrm{E}^j_{t,0}} \left(h(\xi^i_t) - \eta^N_t(h)\right) \left(h(\xi^j_t) - \eta^N_t(h)\right). \tag{3.6}$$

It is very easy to compute since it only considers the samples at time t which do not have a common ancestor at time $t = 0$. However, due to degeneration phenomenon detailed in the previous chapter, $\mathcal{V}^{N}_{\gamma,t}(h)$ tends to be equal to 0 when t is large; in this setting, all the samples have a common ancestor. This phenomenon can be controlled by considering a fixed-lag parameter λ which truncates the genealogy of the particle system. The truncated estimator proposed by Olsson and Douc (2019) reads

$$\mathcal{V}^{N,\lambda}_{\gamma,t}(h) = -N^{-1} \sum_{i,j \in [N]^2} \mathbb{1}_{\mathrm{E}^i_{t,t-\lambda} \neq \mathrm{E}^j_{t,t-\lambda}} \left(h(\xi^i_t) - \eta^N_t(h)\right) \left(h(\xi^j_t) - \eta^N_t(h)\right), \tag{3.7}$$

where $\mathrm{E}^i_{t,t-\lambda}$ is the index ancestor of ξ^i_t at time $t - \lambda$. $\mathrm{E}^i_{t,t-\lambda}$ can be computed in the same way as $\mathrm{E}^i_{t,0}$ in Alg. 3.1. This estimator can be made stable provided λ is well chosen but its practical choice is not trivial. Finally, Lee and Whiteley (2018) proposes a consistent term by term estimator of (3.2). Its construction will be described later since the alternative estimator we propose also relies on it. It also brings an alternative point of view about the CLE estimator. Actually, (3.6) can be rewritten as a mathematical expectation,

$$\mathcal{V}^{N,\lambda}_{\gamma,t}(h) = -N\mathbb{E}\left[\prod_{s=0}^{t} \mathbb{1}_{K^1_s \neq K^2_s} \times \{h(\xi^{K^1_t}_t) - \eta^N_t(h)\}\{h(\xi^{K^2_t}_t) - \eta^N_t(h)\} \middle| \mathcal{F}^N_t\right], \tag{3.8}$$

where

$$\mathcal{F}^N_t = \sigma\left(\{\xi^i_0\}^N_{i=1}, \cdots, \{\xi^i_t\}^N_{i=1}, \{A^i_0\}^N_{i=1}, \cdots, \{A^i_t\}^N_{i=1}\right)$$

is the σ-field containing all the particles and ancestors up to time t, $(K^1_{0:t}, K^2_{0:t})$ are discrete random variables such as K^1_t and K^2_t are i.i.d. uniformly on $[N]$ and

$$K^1_s = A^{K^1_{s+1}}_s, \quad K^2_s = \mathbb{1}_{K^1_{s+1} \neq K^2_{s+1}} A^{K^2_{s+1}}_s + \mathbb{1}_{K^1_{s+1} = K^2_{s+1}} C_s,$$

where $C_s \sim \mathrm{Categorical}(\{\mathcal{W}^l_s\}_{l \in [1:N]})$.

The FFBS algorithm - By considering the complete trajectories generated by Alg. 3.1, it is possible to approximate directly $\phi_{0:t|t}(h_{0:t})$ with the bootstrap particle filter (see the rationale of sequential importance sampling in chapter 2). However, due to the degeneration phenomenon, such estimators are very poor in practice, in particular when we consider the problem of approximating $\phi_{s|t}(h_{0:t})$ for $s << t$. In this chapter, we consider additive functionals

$$h_{0:t}(x_{0:t}) = \sum_{s=0}^{t-1} h_s(x_s, x_{s+1}); \tag{3.9}$$

for such functionals, we also set $\tilde{h}_{s:r}(x_{s:r}) = \sum_{l=s}^{r-1} h_l(x_l, x_{l+1})$, if $s < r$, and $\tilde{h}_{s:r}(x_{s:r}) = 0$, otherwise. When (3.9) is satisfied, the output $\phi^N_{0:t|t}(h_{0:t})$ of the FFBS algorithm can be directly computed in a forward way and so updated when a new observation y_{t+1} is available. Let us briefly explain why. First, remark that

$$\phi_{0:t|t}\left(h_{0:t}\right) = \phi_t\left(\mathbf{T}_t[h_{0:t}]\right)$$

(remember that $\mathbf{T}_t[h_{0:t}](x_t)$ depends on x_t). On the other hand, Alg. 3.1 provides an approximation $\phi^N_t[h] = \sum_{i=1}^N \mathcal{W}^i_t h(\xi^i_t)$ for any functional h from X. So it remains to obtain an approximation of $\mathbf{T}_t[h_{0:t}](x_t)$ for any x_t. This can be done sequentially since

$$\begin{aligned}
\mathbf{T}_t[h_{0:t}](x_t) &= \int \left\{\tilde{h}_{0:t-1}(x_{0:t-1}) + h_t(x_{t-1}, x_t)\right\} \mathbf{T}_t(x_t, x_{0:t-1})\nu(\mathrm{d}x_{0:t-1}),\\
&= \int \left\{\tilde{h}_{0:t-1}(x_{0:t-1}) + h_{t-1}(x_{t-1}, x_t)\right\} \mathbf{B}_{\phi_{t-1}}(x_t, x_{t-1})\mathbf{T}_{t-1}(x_{t-1}, x_{0:t-2})\nu(\mathrm{d}x_{0:t-1}),\\
&= \mathbf{B}_{\phi_{t-1}}\left[\mathbf{T}_{t-1}[\tilde{h}_{0:t-1}] + h_{t-1}(., x_t)\right](x_t).
\end{aligned}$$

Using that $\mathbf{B}_{\phi_{t-1}}[h](x_t)$ is a ratio of integrals w.r.t. ϕ_{t-1} (see (3.1)), we obtain a Monte Carlo approximation $\mathbf{T}^N_t[h_{0:t}](x_t)$ by plugging $\mathbf{T}^N_{t-1}$ and ϕ^N_{t-1} above,

$$\mathbf{T}^N_t[h_{0:t}](x_t) = \sum_{i=1}^N \frac{\omega^i_{t-1} f_t(\xi^i_{t-1}, x_t)}{\sum_{j=1}^N \omega^j_{t-1} f_t(\xi^j_{t-1}, x_t)} \left\{\mathbf{T}^N_{t-1}[\tilde{h}_{0:t-1}](\xi^i_{t-1}) + h_{t-1}(\xi^i_{t-1}, x_t)\right\}.$$

Finally,

$$\phi^N_{0:t|t}(h_{0:t}) = \sum_{i=1}^N \mathcal{W}^i_t \mathbf{T}^N_t[h_{0:t}](\xi^i_t).$$

In other words, the computation of the FFBS algorithm for additive functionals (3.9) coincides with the bootstrap particle filter, Alg. 3.1, up to the additive computation of $\mathbf{T}^N_t[h_{0:t}](\xi^i_t)$ after the sampling step at time t.

For any bounded functionals $h_{0:t}$ (not necessarily additive) and under $\mathbf{A}1$, $\phi^N_{0:t|t}(h_{0:t})$ also satisfies a CLT (Douc et al., 2011a),

$$\sqrt{N}\left(\phi^N_{0:t|t}(h_{0:t}) - \phi_{0:t|t}(h_{0:t})\right) \Longrightarrow \mathcal{N}\left(0; \mathcal{V}^{\mathrm{FFBS}}_{0:t|t}(h_{0:t})\right),$$

where

$$\mathcal{V}^{\mathrm{FFBS}}_{0:t|t}(h_{0:t}) = \sum_{s=0}^t \frac{\eta_s\left(\mathbf{G}_{s,t}\left[g_t\left\{h - \phi_{0:t|t}(h_{0:t})\right\}\right]^2\right)}{\eta_s(\mathbf{Q}_{s+1,t}[g_t])^2}, \tag{3.10}$$

and where $\mathbf{G}_{s,t}$ is the kernel that integrates $h_{0:t}$ forward and backward starting from x_s, *i.e.*

$$\mathbf{G}_{s,t}[h_{0:t}](x_s) = \mathbf{T}_s\left[\mathbf{Q}_{s+1:t}[h_{0:t}]\right](x_s) = \int h_{0:t}(x_{0:t})\mathbf{T}_s(x_s, x_{0:s-1})\mathbf{Q}_{s+1:t}(x_s, x_{s+1:t})\nu(\mathrm{d}x_{0:s-1}, \mathrm{d}x_{s+1:t}),$$

for any $s \in [0:t]$ and $x_s \in \mathsf{X}$. However, and contrary to the bootstrap particle filter, no estimator of (3.10) has been proposed. It is the objective of our second contribution but we limit it to additive functionals in order to propose an online algorithm.

3.2 Variance estimation for filtering estimators

The starting point of our approach is the following. From (3.8), it can be observed that the degeneration of the CLE can be explained by the distribution of $(K^1_{0:t}, K^2_{0:t})$ which only considers the ancestors of $\xi_t^{K_t^1}$ and of $\xi_t^{K_t^2}$. However, once we have observed a full sequence of observation $y_{0:t}$, there may be other relevant trajectories which are not necessarily the ancestral ones; we would like to take into account such posterior trajectories. Our idea is to replace the deterministic assignment in (3.8) by a stochastic distribution related to that used in the FFBS algorithm and which allows to re-evaluate the contribution of past trajectories.

Principle - We first rediscover the rationale of the general construction of the asymptotic variance estimator of Lee and Whiteley (2018). Let $X \sim \pi(x)$, and assume that we want to compute unbiased estimators of $\mathbb{E}(f(X)g(X))$ and of $\mathbb{E}(f(X))\mathbb{E}(g(X))$. These two quantities can be seen as the same moment according to two distributions on $(\mathsf{X}^2, \mathcal{X}^{\otimes 2})$,

$$\mathcal{Q}_b(x, x') = \pi(x)\left(\mathbb{1}_{b=0}\pi(x') + \mathbb{1}_{b=1}\delta_x(x')\right).$$

For functionals $\mathbf{h}$ defined on X^2 as $\mathbf{h} = f \otimes g$ (so remember that $\mathbf{h}(x, x') = f(x)g(x')$), we have

$$\mathbb{E}(f(X))\mathbb{E}(g(X)) = \mathcal{Q}_0(f \otimes g), \quad \mathbb{E}(f(X)g(X)) = \mathcal{Q}_1(f \otimes g).$$

Using this point of view, it seems that we need to sample in augmented dimension to obtain unbiased estimators of $\mathbb{E}(f(X)g(X))$ and of $\mathbb{E}(f(X))\mathbb{E}(g(X))$. However, an unbiased estimator of $\mathbb{E}(f(X)g(X))$ is easily deduced from i.i.d. samples from π. Using the fact that an unbiased estimator of $\mathbb{C}ov(f(X), g(X))$ is also available, one of $\mathbb{E}(f(X))\mathbb{E}(g(X)) = \mathbb{E}(f(X)g(X)) - \mathbb{C}ov(f(X), g(X))$ can be deduced. In summary,

$$\mathcal{Q}_0^N(f \otimes g) = \frac{1}{N(N-1)} \sum_{i,j \in [N]^2} \mathbb{1}_{i \neq j} f(\xi^i)g(\xi^j), \quad \mathcal{Q}_1^N(f \otimes g) = \frac{1}{N}\sum_{i=1}^N f(\xi^i)g(\xi^i), \quad \{\xi^i\}_{i=1}^N \overset{\text{i.i.d}}{\sim} \pi$$

are unbiased estimators only based on i.i.d. samples according to π. The rationale of the estimator of the asymptotic variance is the same: we look for building distributions $\mathcal{Q}_{b,t}(x_{0:t}, x_{0:t'})$ such that each term $\gamma_s(\mathbf{1})\gamma_s\left(\mathbf{Q}_{s+1:t}[h]^2\right) - \gamma_t(h)^2$ in (3.2) can be seen as a moment according to a particular $\mathcal{Q}_{b,t}$; we next derive an unbiased estimator of $\mathcal{Q}_{b,t}(h)$ from the samples generated by the bootstrap particle filter. We now detail this construction and the statistical properties of our estimator.

Our estimator - Using the same construction as previously, we can rewrite $\gamma_s(\mathbf{1})\gamma_s\left(\mathbf{Q}_{s+1:t}[h]^2\right)$ as an integral in augmented dimension. For any $t \in \mathbb{N}$, we define $\mathcal{B}_t = \{0,1\}^{t+1}$ and $b = (b_0, \cdots, b_t) \in \mathcal{B}_t$. Denote by $\mathbf{0}$ the null vector in $\mathcal{B}_t$ and e_s the vector with 1 at position s and 0 elsewhere. We also consider the following HMC in an augmented space,

$$\mathcal{Q}_{b,t}\left(x_{0:t}, x'_{0:t}\right) = p^{b_0}\left(x_0, x'_0\right)\prod_{s=0}^{t-1} g_s^{\otimes 2}(x_s, x'_s)\prod_{s=1}^{t} f_s^{b_s}\left((x_{s-1}, x'_{s-1}), (x_s, x'_s)\right),$$

where

$$\begin{aligned} p^{b_0}\left(x_0, x'_0\right) &= p(x_0)(\mathbb{1}_{b_0=0}p(x'_0) + \mathbb{1}_{b_0=1}\delta_{x_0}(x'_0)), \\ f_t^{b_t}\left((x_{t-1}, x'_{t-1}), (x_t, x'_t)\right) &= f_t(x_{t-1}, x_t)(\mathbb{1}_{b_t=0}f_t(x'_{t-1}, x'_t)) + \mathbb{1}_{b_t=1}\delta_{x_t}(x'_t). \end{aligned}$$

Then the terms in (3.2) can be seen as particular moments according to $\mathcal{Q}_{b,t}$ and we have for any functional h from X

$$\begin{aligned}\mathcal{Q}_{\mathbf{0},t}(h^{\otimes 2}) &= \gamma_t(h)^2,\\ \mathcal{Q}_{e_s,t}(h^{\otimes 2}) &= \gamma_s(\mathbf{1})\gamma_s\left(\mathbf{Q}_{s+1:t}[h]^2\right).\end{aligned}$$

Consequently, our target $\mathcal{V}^N_{\gamma,t}(h)$ can be rewritten as

$$\sum_{s=0}^{t}\left(\mathcal{Q}_{e_s,t}(h^{\otimes 2}) - \mathcal{Q}_{\mathbf{0},t}(h^{\otimes 2})\right). \tag{3.11}$$

Let us now see how to obtain intuitively an unbiased estimator of $\mathcal{Q}_{b,t}(h)$ with the following example.

Example 3.1. Let $(X_0, X_1) \sim \pi(x_0, x_1)$ and let us assume that we have $(\xi_0^i, \xi_1^i) \overset{\text{i.i.d.}}{\sim} \pi(x_0, x_1)$, for $i \in [1 : N]$. The objective is to compute

$$\pi_b(\mathbf{h}) = \int \mathbf{h}(x_{0:1}, x'_{0:1})\pi_b(\mathrm{d}x_{0:1}, \mathrm{d}x'_{0:1}),$$

where $b = (b_0, b_1) \in \{0, 1\}^2$ and

$$\pi_b(x_{0:1}, x'_{0:1}) = \pi(x_0)(\mathbb{1}_{b_0=0}\pi(x'_0) + \mathbb{1}_{b_0=1}\delta_{x_0}(x'_0))\pi(x_1|x_0)(\mathbb{1}_{b_1=0}\pi(x'_1|x'_0)) + \mathbb{1}_{b_1=1}\delta_{x_1}(x'_1),$$

from $\{\xi_0^i, \xi_1^i\}_{i=1}^N$.

- Case $b = (0, 0)$: it is the direct application in augmented dimension of what we have seen above, so an unbiased estimator is given by (up to the proper constant)

$$\pi^N_{(0,0)}(\mathbf{h}) \propto \sum_{i\neq j} \mathbf{h}(\xi^i_{0:1}, \xi^j_{0:1});$$

- Case $b = (1, 1)$: it also the application in augmented dimension of what we have seen before, so an unbiased estimator is given by

$$\pi^N_{(1,1)}(\mathbf{h}) \propto \sum_{i=1}^{N} \mathbf{h}(\xi^i_{0:1}, \xi^i_{0:1});$$

- Case $b = (0, 1)$: $\pi_b(h)$ can be rewritten as

$$\pi_{(0,1)}(\mathbf{h}) = \int \left[\int \mathbf{h}(x_{0:1}, x'_0, x_1)\pi(\mathrm{d}x_1|x_0)\right]\pi(\mathrm{d}x_0)\pi(\mathrm{d}x'_0),$$

  so again one can recycle $\{\xi_0^i\}_{i=1}^N$ and compute

$$\pi^N_{(0,1)}(\mathbf{h}) \propto \sum_{i\neq j} \mathbf{h}(\xi^i_{0:1}, \xi^j_0, \xi^i_1);$$

- Case $b = (1, 0)$ is the most challenging case. $\pi_b(\mathbf{h})$ can be rewritten as

$$\pi_{(1,0)}(\mathbf{h}) = \int \left[\int \mathbf{h}(x_{0:1}, x_0, x'_1)\pi(\mathrm{d}x_1|x_0)\pi(\mathrm{d}x'_1|x_0)\right]\pi(\mathrm{d}x_0).$$

However, we cannot recycle $\{\xi_1^i\}_{i=1}^N$ because ξ_1^j is not sampled according to $\pi(x_1|\xi_0^i)$ for $j \neq i$. The trick is to resample artificially the set $\{\xi_0^i\}_{i=1}^N$ via indexes $A_0^i \sim \mathrm{Cat}(1/N)$ before extending the trajectories by sampling $\xi_1^i \sim \pi(x_1|\xi_0^{A_0^i})$. An unbiased estimator reads

$$\pi_{(1,0)}^N(\mathbf{h}) \propto \sum_{i \neq j} \mathbb{1}_{A_0^i = A_0^j} \mathbf{h}(\xi_0^{A_0^i}, \xi_1^i, \xi_0^{A_0^j}, \xi_1^j).$$

The computation of the last estimator has prompted us to review our sampling scheme by adding a resampling step. So we have to review the computation of the third first estimators in the case where we use the sampling-resampling-sampling prodedure in order to have a common scheme for any b.

- Case $b = (0,0)$: the main difference is that we now have dependent but still identically distributed samples according to $\pi(x_0, x_1)$. An unbiased estimator reads

$$\pi_{(0,0)}^N(\mathbf{h}) \propto \sum_{i,j} \mathbb{1}_{A_0^i \neq A_0^j} \mathbf{h}(\xi_0^{A_0^i}, \xi_1^i, \xi_0^{A_0^j}, \xi_1^j);$$

- Case $b = (1,1)$: this case is also trivial and the unbiased estimator becomes

$$\pi_{(1,1)}^N(\mathbf{h}) \propto \sum_{i=1}^N \mathbf{h}(\xi_0^{A_0^i}, \xi_1^i, \xi_0^{A_0^i}, \xi_1^i);$$

- Case $b = (0,1)$: this case should take into account the samples $\{\xi_0^j\}_{j=1}^N$ different from $\{\xi_0^{A_0^i}\}$. To that end, it recycles the samples $\{\xi_0^j\}_{j=1}^N$. The estimator becomes

$$\pi_{(0,1)}^N(\mathbf{h}) \propto \sum_{i,j}^N \mathbb{1}_{j \neq A_0^i} \mathbf{h}(\xi_0^{A_0^i}, \xi_1^i, \xi_0^j, \xi_1^i);$$

Actually, these estimators remain unbiased as their previous version, but their variance increases because of the resampling step.

The unbiased estimator of $\mathcal{Q}_{b,t}(\mathbf{h})$ of Lee and Whiteley (2018) can be interpreted as the generalization of Ex. 3.1 in the case where the integrals involve $\gamma_{0:t}(x_{0:t})$. However, remember that the bootstrap particle filter provides an unbiased estimator of any functional according to $\gamma_{0:t}$ (Del Moral, 2004). So the estimator of Lee and Whiteley (2018) relies on a discrete Markov chain $K_{0:t}^1$ and a conditional discrete Markov chain $K_{0:t}^2$ whose distributions satisfy

$$\begin{aligned} \Lambda_{1,t}(k_{0:t}^1) &= \frac{1}{N} \prod_{s=1}^t \overbrace{\beta_s(k_s^1, k_{s-1}^1)}^{p(k_{s-1}^1|k_s^1)}, \\ \Lambda_{2,t}(k_{0:t}^1, k_{0:t}^2) = p(k_{0:t}^2|k_{0:t}^1) &= \frac{1}{N} \prod_{s=1}^t \mathbb{1}_{k_s^1 = k_s^2} \mathcal{W}_{s-1}^{k_{s-1}^2} + \mathbb{1}_{k_s^1 \neq k_s^2} \beta_s(k_s^2; k_{s-1}^2), \end{aligned} \tag{3.12}$$

where

$$\beta_s(k,l) = \beta_s^{\mathsf{GT}}(k,l) = \mathbb{1}_{l = A_{s-1}^k}. \tag{3.13}$$

The indicator function in (3.13) enables to retrace the genealogy of a final particle, as we did in Ex. 3.1, while $\mathcal{W}_{s-1}^{k_{s-1}^2}$ in (3.12) quantifies the relevance of sample $\xi_{s-1}^{k_{s-1}^2}$ when it comes to consider the support of particles at time $s-1$ (see the analogy with the case $b=(0,1)$ in Ex. 3.1). Next, introducing the coalescence function

$$\mathrm{I}_{b,s}(k_{0:s}^1, k_{0:s}^2) = \prod_{l=0}^{s} \{\mathbb{1}_{k_l^1=k_l^2}\mathbb{1}_{b_l=1} + \mathbb{1}_{k_l^1\neq k_l^2}\mathbb{1}_{b_l=0}\}, \quad \forall s \in [0:t], \tag{3.14}$$

which is equal to 1 if vector b is in accordance with trajectories $(k_{0:s}^1, k_{0:s}^2)$, an unbiased estimator of $\mathcal{Q}_{b,t}(\mathbf{h})$ for any bounded functionals $\mathbf{h}(x_{0:t}, x'_{0:t})$ from the augmented space $\mathsf{X}^{2(t+1)}$ reads

$$\mathcal{Q}_{b,t}^{N,\mathsf{GT}}(\mathbf{h}) = \prod_{s=0}^{t} N^{b_s} \left(\frac{N}{N-1}\right)^{1-b_s} \times \gamma_t^N(\mathbf{1})^2 \mathbb{E}_{\mathsf{GT}}\left[\mathrm{I}_{b,t}\left(K_{0:t}^1, K_{0:t}^2\right) \mathbf{h}(\xi_{0:t}^{K_{0:t}^1}, \xi_{0:t}^{K_{0:t}^2}) | \mathcal{F}_t\right], \tag{3.15}$$

where $\mathbb{E}_{\mathsf{GT}}$ means for the expectation under the distribution (3.12)-(3.13). However, this estimator has not been used in practice due to its computational cost. Surprisingly, the online procedure that we propose for our refined estimator can also be applied to $\mathcal{Q}_{b,t}^{N,\mathsf{GT}}(\mathbf{h})$ but it has not been exploited in (Lee and Whiteley, 2018). Following the intuition of Ex. 3.1, we look for building an estimator which aims at considering other alternative relevant trajectories rather than only the ancestors of the final samples. In the case of Ex. 3.1, it would be equivalent to consider trajectories (ξ_0^j, ξ_1^i) where $j \neq A_0^i$ to build our estimator. To that end, we would need to quantify the relevance of associating ξ_0^j with ξ_1^i. It is what we do when we build our new estimator: in the same way as the FFBS algorithm modifies the output of the bootstrap particle filter for considering such trajectories, we replace the conditional distribution $\beta_s^{\mathsf{GT}}(k,l)$ in (3.13) by the backward kernel

$$\beta_s^{\mathsf{BS}}(k,l) = \frac{\omega_{s-1}^l f(\xi_{s-1}^l, \xi_s^k)}{\sum_{j=1}^N \omega_{s-1}^j f(\xi_{s-1}^j, \xi_s^k)}. \tag{3.16}$$

Intuitively, $\beta_s(k,l)$ aims at selecting among the samples $\{\xi_{s-1}^l\}_{l=1}^N$ those which are in accordance with ξ_s^k and the future observations. We then propose a new estimator

$$\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(\mathbf{h}) = \prod_{s=0}^{t} N^{b_s} \left(\frac{N}{N-1}\right)^{1-b_s} \times \gamma_t^N(\mathbf{1})^2 \mathbb{E}_{\mathsf{BS}}\left[\mathrm{I}_{b,t}\left(K_{0:t}^1, K_{0:t}^2\right) \mathbf{h}(\xi_{0:t}^{K_{0:t}^1}, \xi_{0:t}^{K_{0:t}^2}) | \mathcal{F}_t\right], \tag{3.17}$$

where the expectation is now taken under the distribution defined by (3.12) and (3.16). From (3.11), we also deduce an estimator of $\mathcal{V}_{\gamma,t}^{\infty}(h)$ for any functional h from X,

$$\overline{\mathcal{V}}_{\gamma,t}^{N,\mathsf{BS}}(h) = \sum_{s=0}^{t} \left\{\mathcal{Q}_{e_s,t}^{N,\mathsf{BS}}(h^{\otimes 2}) - \mathcal{Q}_{\mathbf{0},t}^{N,\mathsf{BS}}(h^{\otimes 2})\right\}. \tag{3.18}$$

Let us now discuss on the statistical properties of these estimators.

Proposition 3.1. Let $t \in \mathbb{N}$. For any $b \in B_t$ and any bounded functional $\mathbf{h}$ from $\mathsf{X}^{2(t+1)}$,

(i) $\mathbb{E}\left[\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(\mathbf{h}) | \mathcal{F}_{t-1}^N\right] = \mathcal{Q}_{b,t-1}^{N,\mathsf{BS}}\left(g_{t-1}^{\otimes 2} f_t^{b_t}[\mathbf{h}]\right)$ for all $t \in \mathbb{N}^*$;

(ii) $\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(\mathbf{h})$ is an unbiased estimator of $\mathcal{Q}_{b,t}(\mathbf{h})$.

(iii) For any bounded functional h , $\overline{\mathcal{V}}_{\gamma,t}^{N,\mathsf{BS}}(h)$ is an unbiased estimator of $\mathcal{V}_{\gamma,t}^{\infty}(h)$.

The proof of (i) is based on the identity

$$\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(\mathbf{h}) = \sum_{k_{0:t}^1, k_{0:t}^2} \overline{\Lambda}_{b,t}^{1,2}(k_{0:t}^1, k_{0:t}^2)\mathbf{h}(\xi_{0:t}^{k_{0:t}^1}, \xi_{0:t}^{k_{0:t}^2}),$$

where

$$\overline{\Lambda}_{b,t}^{1,2}(k_{0:t}^1, k_{0:t}^2) = \prod_{s=0}^{t} N^{b_s} \left(\frac{N}{N-1}\right)^{1-b_s} \times \gamma_t^N(\mathbf{1})^2 \Lambda_{1,t}(k_{0:t}^1)\Lambda_{2,t}(k_{0:t}^1, k_{0:t}^2);$$

the sum on $(k_{0:t}^1, k_{0:t}^2)$ is split in a sum on $(k_{0:t-1}^1, k_{0:t-1}^2)$ and on (k_t^1, k_t^2). This last sum is next manipulated in order to obtain the result. For item (ii), we proceed by induction: the property is valid for $t = 0$ (this case coincides with the static one that we described above); we next apply (i) to obtain the property at time t. Finally, (iii) is a direct consequence of (ii) and (3.11).

We now focus on the convergence of $\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(\mathbf{h})$ (resp. $\overline{\mathcal{V}}_{\gamma,t}^{N,\mathsf{BS}}(\mathbf{h})$) for any bounded functional $\mathbf{h}$ (resp. h) under the following set of assumptions:

(**A2**) For all $t > 0$ and $(x, x') \in \mathsf{X}^2$, $f_t(x', x) > 0$.

(**A3**) There exists $\sigma_+ > 0$ such that for all $t \geq 1$, $\sup_{x,x' \in \mathsf{X}^2} f_t(x', x) \leq \sigma_+$.

(**A4**) There exists $0 < \sigma_- < \sigma_+$ such that for all $t \geq 1$, $\inf_{x,x' \in \mathsf{X}^2} f_t(x', x) \geq \sigma_-$.

Theorem 3.1. Assume that **A**1 − **A**3 hold. For any $t \in \mathbb{N}$, $b \in \mathcal{B}_t$ and for any bounded functional $\mathbf{h}$,

$$\lim_{N\to\infty} \mathbb{E}\left[\left(\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(\mathbf{h}) - \mathcal{Q}_{b,t}(\mathbf{h})\right)^2\right] = 0. \tag{3.19}$$

In addition, if **A**4 holds, the convergence rate is $\mathcal{O}(1/\sqrt{N})$.

The proof proceeds by induction. After proving the convergence is true at time $t = 0$, the main term of (3.19) is written as

$$\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(\mathbf{h}) - \mathcal{Q}_{b,t}(\mathbf{h}) = \mathcal{Q}_{b,t}^{N,\mathsf{BS}}(\mathbf{h}) - \mathcal{Q}_{b,t-1}^{N,\mathsf{BS}}\left(g_{t-1}^{\otimes 2} f_t^{b_t}[\mathbf{h}]\right) + \mathcal{Q}_{b,t-1}^{N,\mathsf{BS}}\left(g_{t-1}^{\otimes 2} f_t^{b_t}[\mathbf{h}]\right) - \mathcal{Q}_{b,t-1}\left(g_{t-1}^{\otimes 2} f_t^{b_t}[\mathbf{h}]\right).$$

Next, by the induction hypothesis

$$\lim_{N\to\infty} \mathbb{E}\left[\left(\mathcal{Q}_{b,t-1}^{N,\mathsf{BS}}\left(g_{t-1}^{\otimes 2} f_t^{b_t}[\mathbf{h}]\right) - \mathcal{Q}_{b,t-1}\left(g_{t-1}^{\otimes 2} f_t^{b_t}[\mathbf{h}]\right)\right)^2\right] = 0;$$

it remains to show that

$$\lim_{N\to\infty} \mathbb{E}\left[\left(\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(\mathbf{h}) - \mathcal{Q}_{b,t-1}^{N,\mathsf{BS}}\left(g_{t-1}^{\otimes 2} f_t^{b_t}[\mathbf{h}]\right)\right)^2\right] = \lim_{N\to\infty} \mathbb{E}\left[\left(\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(\mathbf{h})\right)^2\right] - \mathbb{E}\left[\left(\mathcal{Q}_{b,t-1}^{N,\mathsf{BS}}\left(g_{t-1}^{\otimes 2} f_t^{b_t}[\mathbf{h}]\right)\right)^2\right] = 0.$$

This is the technical part of the proof and it relies on an upper bound of $\mathbb{E}[(\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(\mathbf{h}))^2]$ in function of $\mathbb{E}[(\mathcal{Q}_{b,t-1}^{N,\mathsf{BS}}(g_{t-1}^{\otimes 2} f_t^{b_t}[\mathbf{h}]))^2]$ and next on **A**1 − **A**3. Assumption **A**4 enables us to obtain the convergence rate of this upper-bound. From (3.18), a direct consequence is that $\overline{\mathcal{V}}_{\gamma,t}^{N,\mathsf{BS}}(h)$ converges in probability to $\mathcal{V}_{\gamma,t}^{\infty}(h)$.

Computing $\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(\mathbf{h})$ and $\overline{\mathcal{V}}_{\gamma,t}^{N,\mathsf{BS}}(h)$ - While our estimator is theoretically valid, its computation is not obvious and relies on that of $\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(\mathbf{h})$ for $b = e_s$ and $b = \mathbf{0}$. This can be done sequentially by introducing

$$\mathcal{T}_t^b(K_t^1, K_t^2) = \mathbb{E}_{\mathsf{BS}}\left[\mathrm{I}_{b,t}\left(K_{0:t}^1, K_{0:t}^2\right) | \mathcal{F}_t, K_t^1, K_t^2\right]. \tag{3.20}$$

In the practical case where $\mathbf{h}(x_{0:t}, x'_{0:t}) = \mathbf{h}(x_t, x'_t)$, (3.17) can be rewritten as

$$\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(\mathbf{h}) = \prod_{s=0}^{t} N^{b_s}\left(\frac{N}{N-1}\right)^{1-b_s} \times \frac{\gamma_t^N(\mathbf{1})^2}{N^2}\sum_{k,l}\mathcal{T}_t^b(k,l)\mathbf{h}(\xi_t^k, \xi_t^l).$$

Indeed, remember that K_t^1 and K_t^2 are uniformly distributed. In addition, by introducing

$$S_t(k,l) = \sum_{s=0}^{t}\mathcal{T}_t^{e_s}(k,l), \quad \text{for all } (k,l) \in \mathbb{N}^2, \tag{3.21}$$

$\overline{\mathcal{V}}_{\gamma,t}^{N,\mathsf{BS}}(h)$ reads

$$\overline{\mathcal{V}}_{\gamma,t}^{N,\mathsf{BS}}(h) = \frac{N^{t-1}\gamma_t^N(\mathbf{1})^2}{(N-1)^t}\sum_{k,l}\left\{S_t(k,l) - \frac{t+1}{N-1}\mathcal{T}_t^{\mathbf{0}}(k,l)\right\}h(\xi_t^k)h(\xi_t^l).$$

It suffices to show that $\mathcal{T}_t^b(k,l)$ and $S_t(k,l)$ are sequentially computable to ensure the computation of our estimator. Using (3.14) and the fact that $(K_{0:t}^1, K_{0:t}^2)$ is a Markov chain, we have

$$\begin{cases} \mathcal{T}_0^b(k,l) = \mathbb{1}_{k\neq l, b_0=0} + \mathbb{1}_{k=l, b_0=1}, \\ \mathcal{T}_t^b(k,l) = \mathbb{1}_{k\neq l}\sum_{i,j\in[N]^2}\beta_t^{\mathsf{BS}}(k,i)\beta_t^{\mathsf{BS}}(l,j)\mathcal{T}_{t-1}^b(i,j) & \text{if} \quad b_t = 0, \\ \mathcal{T}_t^b(k,l) = \mathbb{1}_{k=k}\sum_{i,j\in[N]^2}\beta_t^{\mathsf{BS}}(k,i)\mathcal{W}_{t-1}^j\mathcal{T}_{t-1}^b(i,j) & \text{if} \quad b_t = 1. \end{cases} \tag{3.22}$$

In particular, for $b = \mathbf{0}$ we have

$$\mathcal{T}_t^{\mathbf{0}}(k,l) = \mathbb{1}_{k\neq l}\sum_{i,j\in[N]^2}\beta_t^{\mathsf{BS}}(k,i)\beta_t^{\mathsf{BS}}(l,j)\mathcal{T}_{t-1}^{\mathbf{0}}(i,j),$$

and for $b = e_s$,

$$\mathcal{T}_t^{e_s}(k,l) = \begin{cases} \mathbb{1}_{k\neq l}\sum_{i,j\in[N]^2}\beta_t^{\mathsf{BS}}(k,i)\beta_t^{\mathsf{BS}}(l,j)\mathcal{T}_{t-1}^{e_s}(i,j) & t > s, \\ \mathbb{1}_{k=l}\sum_{i,j\in[N]^2}\beta_t^{\mathsf{BS}}(k,i)\mathcal{W}_{t-1}^j\mathcal{T}_{t-1}^{\mathbf{0}}(i,j) & t = s, \\ \mathcal{T}_t^{\mathbf{0}}(k,l) & t < s. \end{cases} \tag{3.23}$$

Plugging (3.23) in (3.21), we obtain

$$S_t(k,l) = \mathcal{T}_t^{e_t}(k,l) + \mathbb{1}_{k\neq l}\sum_{i,j\in[N]^2}\beta_t^{\mathsf{BS}}(k,i)\beta_t^{\mathsf{BS}}(l,j)S_{t-1}(i,j).$$

Consequently, $\overline{\mathcal{V}}_{\gamma,t}^{N,\mathsf{BS}}(h)$ can be updated online through the update of $S_t(k,l)$ and of $\mathcal{T}_t^{\mathbf{0}}(k,l)$.

An alternative estimator - If we remove the unbiased constraint, it is possible to derive an altervative estimator from the following observation. Since $\gamma_t^N(h)$ is unbiased, its variance coincides with

$$\mathbb{E}\left(\left(\gamma_t^N(h)-\gamma_t(h)\right)^2\right)=\mathbb{E}\left(\gamma_t^N(h)^2\right)-\gamma_t(h)^2=\mathbb{E}\left(\gamma_t^N(h)^2\right)-\mathcal{Q}_{\mathbf{0},t}(h^{\otimes 2})\,.$$

Moreover, if $h(x_t)$ is bounded, $N(\gamma_t^N(h)-\gamma_t(h))^2$ is uniformly integrable; this can be seen by using a Hoeffding type inequality (see e.g. (Douc et al., 2014)). Consequently, using the CLT (3.1) and Theorem (25.12) of Billingsley (1986),

$$\lim_{N\to\infty} N\mathbb{E}\left(\left(\gamma_t^N(h)-\gamma_t(h)\right)^2\right)=\mathcal{V}_{\gamma,t}^{\infty}(h).$$

Because $\mathcal{Q}_{\mathbf{0},t}(h^{\otimes 2})=\gamma_t(h)^2$, a natural estimator of $\mathcal{V}_{\gamma,t}^{\infty}(h)$ is

$$\begin{aligned}\mathcal{V}_{\gamma,t}^{\mathsf{BS}}(h)&=N\left(\gamma_t^N(h)^2-\mathcal{Q}_{\mathbf{0},t}^{N,\mathsf{BS}}(h_t^{\otimes 2})\right)\\&=N\gamma_t^N(\mathbf{1})^2\left(\eta_t^N(h)^2-\frac{N^{t-1}}{(N-1)^{t+1}}\sum_{i,j}\mathcal{T}_t^{\mathbf{0}}(i,j)h(\xi_t^i)h(\xi_t^j)\right).\end{aligned}\tag{3.24}$$

Note that $\mathcal{V}_{\gamma,t}^{\mathsf{BS}}(h)$ is an unbiased estimator of a quantity which converges to the asymptotic variance, while $\overline{\mathcal{V}}_{\gamma,t}^{N,\mathsf{BS}}(h)$ is a unbiased estimator of the asymptotic variance. The following theorem ensures that $\mathcal{V}_{\gamma,t}^{\mathsf{BS}}(h)$ also converges to $\mathcal{V}_{\gamma,t}^{\infty}(h)$.

Theorem 3.2. Let $\mathbf{A}1-\mathbf{A}3$ hold. For any bounded functional h from X, $\mathcal{V}_{\gamma,t}^{N,\mathsf{BS}}(h)$ converges in probability to $\mathcal{V}_{\gamma,t}^{\infty}(h)$.

The proof consists in expressing $\gamma_t^N(h)^2$ in function of $\mathcal{Q}_{b,t}^{\mathsf{BS}}(h^{\otimes 2})$ and relies on the equality of Cérou et al. (2011),

$$\begin{aligned}&\sum_{b\in\mathcal{B}_t}\left\{\prod_{s=0}^{t}\frac{1}{N^{b_s}}\left(\frac{N-1}{N}\right)^{1-b_s}\right\}\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(h^{\otimes 2})\\&\quad=\gamma t^N(\mathbf{1})^2\mathbb{E}_{\mathsf{BS}}\left[\sum_{b\in\mathcal{B}_t}\mathrm{I}_{b,t}(K_{0:t}^1,K_{0:t}^2)h(\xi_t^{K_t^1})h(\xi_t^{K_t^2})\middle|\mathcal{F}_t\right]=\gamma t^N(\mathbf{1})^2\eta_t^N(h)^2=\gamma_t^N(h)^2.\end{aligned}\tag{3.25}$$

Using Theorem 3.1 which states that $\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(\mathbf{h})$ converges in probability to $\mathcal{Q}_{b,t}(\mathbf{h})$, the convergence of $\mathcal{V}_{\gamma,t}^{\mathsf{BS}}(h)$ is also ensured.

A major advantage of $\mathcal{V}_{\gamma,t}^{\mathsf{BS}}(h)$ w.r.t. $\overline{\mathcal{V}}_{\gamma,t}^{N,\mathsf{BS}}(h)$ is the computational cost. Indeed, the first estimator only relies on the update of $\mathcal{T}_t^{\mathbf{0}}$ contrary to the second one for which we need to compute $\mathcal{T}_t^{e_s}$

Reducing the computational cost of $\mathcal{V}_{\gamma,t}^{\mathsf{BS}}(h)$ and of $\overline{\mathcal{V}}_{\gamma,t}^{N,\mathsf{BS}}(h)$ - It is possible to reduce the computational cost of our previous estimators by interpreting (3.22) as expectations according to the discrete conditional distributions $\beta_t^{\mathsf{BS}}(k,i)\beta_t^{\mathsf{BS}}(l,j)$ or $\beta_t^{\mathsf{BS}}(k,i)$. By sampling

$$\{J_{k,t-1}^i,J_{l,t-1}^i\}_{i=1}^M\overset{\text{i.i.d}}{\sim}\beta_t^{\mathsf{BS}}(k,.)\beta_t^{\mathsf{BS}}(l,.)\quad\text{for all }(k,l),\,k\neq l$$

$\mathcal{T}_t^b(k,l)$ in (3.22) are replaced by their Monte Carlo approximation

$$\tilde{\mathcal{T}}_t^b(k,l) = \frac{\mathbb{1}_{k\neq l}}{M}\sum_{i=1}^{M}\tilde{\mathcal{T}}_{t-1}^b(J_{k,t-1}^i, J_{l,t-1}^i) \qquad \text{if} \quad b_t = 0,$$

$$\tilde{\mathcal{T}}_t^b(k,l) = \frac{\mathbb{1}_{k=l}}{M}\sum_{i=1}^{M}\sum_{j=1}^{N}\mathcal{W}_{t-1}^j\tilde{\mathcal{T}}_{t-1}^b(J_{k,t-1}^i, j) \quad \text{if} \quad b_t = 1.$$

The time complexity to compute $\tilde{\mathcal{T}}_t^b(k,l)$ becomes $\mathcal{O}(MN^2)$ and in practice M does not need to be large (it has been observed that $M=3$ is sufficient, as for the *PaRIS* smoothing algorithm of Olsson and Westerborn (2017)). An interesting result it that the convergence of the corresponding *PaRIS* estimator of $\mathcal{Q}_{b,t}(\mathbf{h})$

$$\tilde{\mathcal{Q}}_{b,t}^{N,M}(\mathbf{h}) = \prod_{s=0}^{t} N^{b_s}\left(\frac{N}{N-1}\right)^{1-b_s} \times \frac{\gamma_t^N(\mathbf{1})^2}{N^2}\sum_{k,l}\tilde{\mathcal{T}}_t^b(k,l)h(\xi_t^k,\xi_t^l). \tag{3.26}$$

remains ensured for any value of M.

Theorem 3.3. Assume that **A**1 − **A**3 hold. For any $t\in\mathbb{N}$, $b\in\mathcal{B}_t$, $M>1$ and any bounded functional $\mathbf{h}$ from $\mathsf{X}^{2(t+1)}$,

$$\lim_{N\to\infty}\mathbb{E}\left[\left(\tilde{\mathcal{Q}}_{b,t}^{N,M}(\mathbf{h}) - \mathcal{Q}_{b,t}(\mathbf{h})\right)^2\right] = 0. \tag{3.27}$$

In addition, if **A**4 holds the convergence rate is $\mathcal{O}(1/\sqrt{N})$.

Theorem 3.3 can be seen as the *PaRIS* version of Theorem 3.1; the proof follows the same steps but the additional sampling steps introduce non trivial terms that need to be handled carefully. A direct consequence is that the *PaRIS* version of $\overline{\mathcal{V}}_{\gamma,t}^{N,\mathsf{BS}}(h)$,

$$\overline{\mathsf{V}}_{\gamma,t}^{N,M}(h) = \sum_{s=0}^{t}\left\{\tilde{\mathcal{Q}}_{e_s,t}^{N,M}(h^{\otimes 2}) - \tilde{\mathcal{Q}}_{\mathbf{0},t}^{N,M}(h^{\otimes 2})\right\}, \tag{3.28}$$

also converges in probability to $\mathcal{V}_{\gamma,t}^{\infty}(h)$ Finally, the *PaRIS* version of $\mathcal{V}_{\gamma,t}^{N,\mathsf{BS}}(h)$,

$$\mathsf{V}_{\gamma,t}^{N,M}(h) = N\left(\gamma_t^N(h)^2 - \tilde{\mathcal{Q}}_{\mathbf{0},t}^{N,M}(h^{\otimes 2})\right). \tag{3.29}$$

also converges in probability to $\mathcal{V}_{\gamma,t}^{\infty}(h)$.

Theorem 3.4. Let **A**.1 − 3 hold. For all $t\in\mathbb{N}$, $M>1$, and any bounded functionnal $h(x_t)$, $\mathsf{V}_{\gamma,t}^{N,M}(h)$ converges in probability to $\mathcal{V}_{\gamma,t}^{\infty}(h)$ when N goes to infinity.

Its proof is based on the same steps as those of Theorem 3.2 but we start to show that identity (3.25) is still valid when $\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(h^{\otimes 2})$ is replaced by $\tilde{\mathcal{Q}}_{b,t}^{N,M}(h^{\otimes 2})$.

Asymptotic variance associated to the predicting and filtering distributions - Based on (3.5), we are now able to deduce estimators of $\mathcal{V}_{\eta,t}^{\infty}$ and $\mathcal{V}_{\phi,t}^{\infty}$. These estimators are obtained by replacing $\mathcal{V}_{\gamma,t}^{\infty}$ by one of the estimators obtained above, and $\gamma_t(\mathbf{1})$, $\eta_t(h)$ or $\phi_t(h)$ by their Monte Carlo estimators. The resulting estimators satisfy the same asymptotic properties as those of $\mathcal{V}_{\eta,t}^{\infty}$ because

$$\begin{aligned}\mathcal{Q}_{b,t}^{N,\mathsf{BS}}\left(\{h-\eta_t^N(h)\}^{\otimes 2}\right) &= \mathcal{Q}_{b,t}^{N,\mathsf{BS}}(h^{\otimes 2}) - \eta_t^N(h)\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(h\otimes\mathbf{1}) - \eta_t^N(h)\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(\mathbf{1}\otimes h) + \eta_t^N(h)^2\mathcal{Q}_{b,t}^{N,\mathsf{BS}}(\mathbf{1})\\ &\xrightarrow{\mathbb{P}} \mathcal{Q}_{b,t}\left(\{h-\eta_t(h)\}^{\otimes 2}\right).\end{aligned}$$

Numerical Experiments (Figs. 3.1-3.3) - Let us consider the stochastic volatility model

$$\begin{aligned} p(x_0) &= \mathcal{N}\left(x_0; 0; \sigma^2/(1-\varphi^2)\right), \\ f_t(x_{t-1}, x_t) &= \mathcal{N}\left(x_t; \phi x_{t-1}; \sigma^2\right), \\ g_t(x_t) &= \mathcal{N}\left(y_t; 0; \beta^2 \exp(x_t)\right), \end{aligned} \tag{3.30}$$

with $(\varphi, \beta, \sigma) = (.975, .641, .165)$ (Pitt and Shephard, 1999). We generate a sequence of 750 observations and we consider the estimation of the asymptotic variance of the predictor $\gamma_t(h)$ where $h(x) = x$. The true asymptotic variance is approximated by the empirical variance obtained by running 1000 particle filters with 10000 samples and multiplied by 10000. We next compute our three estimators $\overline{\mathcal{V}}_{\eta,t}^{N,\mathsf{BS}}$, $\mathcal{V}_{\eta,t}^{N,\mathsf{BS}}$ and $\mathsf{V}_{\gamma,t}^{N,M}(h)$ with $N = 3000$ and $M = 3, 50$ times. Fig. 3.1 displays the behavior of the thee estimators. They have approximately the same performances but $\overline{\mathcal{V}}_{\eta,t}^{N,\mathsf{BS}}$ tends to be biased when t is large. It appears that the most interesting estimator in terms of performances and of computational cost is $\mathsf{V}_{\gamma,t}^{N,M=3}(h)$. Consequently, we compare it to the fixed-lag estimators $\mathcal{V}_{\gamma,t}^{N,\lambda}(h)$ of Olsson and Douc (2019) (see (3.7)) with $\lambda \in \{20, 100, 200, 750\}$ for a sequence of 3000 observations. The results are displayed in Figs. 3.2 and 3.3. Our estimator performs better even if its computational cost is approximately twice larger than the fixed-lag ones. Note however that the fine tuning of λ is not obvious in practice.

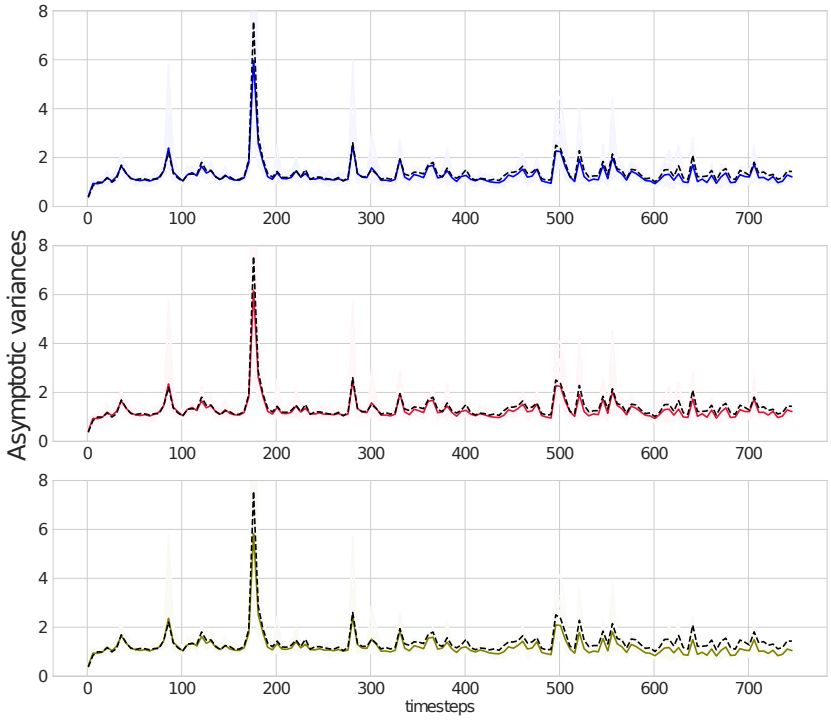


Figure 3.1: Long-term behavior of $\mathcal{V}_{\eta,t}^{\mathsf{BS}}$ (top), $\mathsf{V}_{\eta,t}^{N,M}$ with $M = 3$ (middle) and $\overline{\mathcal{V}}_{\eta,t}^{\mathsf{BS}}$ (bottom). The black dashed line is the asymptotic variance estimated using brute force. The number of particles is set to $N = 3000$.

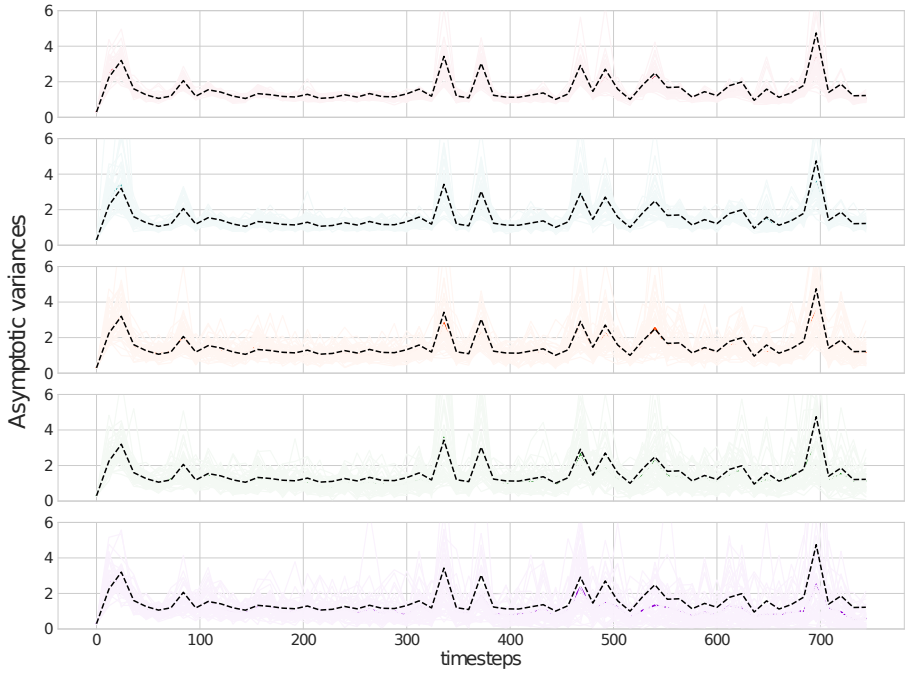


Figure 3.2: Long-term behavior of the asymptotic variance estimators up to $t = 750$. From top to bottom: *PaRIS* estimator $\mathsf{V}_{\eta,t}^{N,M}$ with $M = 3$, lagged estimators with (in order) $\lambda \in \{20, 100, 200, 750\}$. The case $\lambda = 750$ corresponds to the CLE estimator. For each estimator, the blurred colored lines represent each run out of fifty runs and solid colored lines correspond to their average. The black dashed line is the asymptotic variance obtained by brute force. The number of particles N is set to 3000.

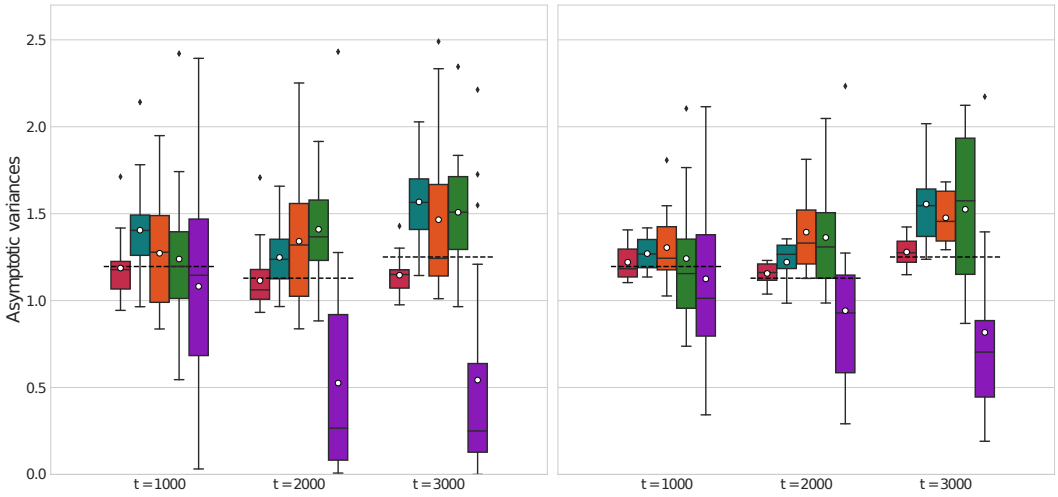


Figure 3.3: Boxplots of the long-term behavior of the asymptotic variance estimates up to $t = 3000$. VBS refers to $\mathsf{V}_{\eta,t}^{N,M}$ with $M = 3$. The white dots represent the average of the asymptotic variance estimates of each algorithm at the specified time t. The dashed black lines correspond to the asymptotic variances estimated by brute force. N is set to 5000 on the left boxplot and 10000 on the right one. The boxplots at each time step from left to right are: $\mathsf{V}_{\gamma,t}^{N,M}(h)$ with $M = 3$ and then the lagged CLEs with $\lambda \in \{20, 100, 200, 3000\}$, in order.

3.3 Asymptotic variance estimation for smoothing estimators

We now turn back to the smoothing problem. As we recalled before, no estimator of

$$\mathcal{V}^{\mathrm{FFBS}}_{0:t|t}(h_{0:t}) = \sum_{s=0}^{t} \frac{\eta_s\left(\mathbf{G}_{s,t}\left[g_t\left\{h_{0:t} - \phi_{0:t|t}(h_{0:t})\right\}\right]^2\right)}{\eta_s(\mathbf{Q}_{s+1,t}[g_t])^2} = \sum_{s=0}^{t} \frac{\gamma_s(\mathbf{1})\gamma_s\left(\mathbf{G}_{s,t}\left[g_t\left\{h_{0:t} - \phi_{0:t|t}(h_{0:t})\right\}\right]^2\right)}{\gamma_{t+1}(\mathbf{1})^2}$$

have been proposed. The principle used for the construction of $\overline{\mathcal{V}}^{N,\mathsf{BS}}_{\eta,t}$ can be applied to estimate $\mathcal{V}^{\mathrm{FFBS}}_{0:t|t}(h_{0:t})$. It relies on the following Proposition.

Proposition 3.2. For any $s \in [0:t]$ and any additive functional $h_{0:t}$,

$$\gamma_s(\mathbf{1})\gamma_s\left(\mathbf{G}_{s,t}[h_{0:t}]^2\right) = \mathcal{Q}_{e_s,t}\left(\left[\mathbf{T}_s[\tilde{h}_{0:s}] + \tilde{h}_{s:t}\right]^{\otimes 2}\right).$$

Indeed, first note that $\gamma_s(\mathbf{1}) = \int \gamma_{0:s-1}(x'_{0:s-1})g_{s-1}(x'_{s-1})\nu(\mathrm{d}x'_{0:s-1})$; next, $\gamma_s(\mathbf{G}_{s,t}[h_{0:t}]^2)$ can be expressed as an integral of $\mathbf{Q}_{s+1,t}[\mathbf{T}_s[\tilde{h}_{0:s}] + \tilde{h}_{s:t}](x_s)$ w.r.t. $\gamma_{0:s}$ if $h_{0:t}$ is indeed additive. Finally, the product of integrals can be rewritten as a unique integral w.r.t. $\mathcal{Q}_{e_s,t}$.

We can now deduce a natural estimator of $\mathcal{V}^{\mathrm{FFBS}}_{0:t|t}(h_{0:t})$ which takes into account that $\mathbf{T}_s[\tilde{h}_{0:s}]$ and $\phi_{0:t|t}(h)$ are not computable but can be approximated sequentially. This estimator reads

$$\mathcal{V}^{N,\mathsf{BS}}_{0:t|t}(h_{0:t}) = \sum_{s=0}^{t} \frac{\mathcal{Q}^N_{e_s,t}\left(\left[g_t\left\{\mathbf{T}^N_s[\tilde{h}_{0:s}] + \tilde{h}_{s:t} - \phi^N_{0:t|t}(h_{0:t})\right\}\right]^{\otimes 2}\right)}{\gamma^N_{t+1}(\mathbf{1})^2}. \tag{3.31}$$

It remains to show that $\mathcal{V}^{N,\mathsf{BS}}_{0:t|t}$ converges in probability to $\mathcal{V}^{\mathrm{FFBS}}_{0:t|t}(h_{0:t})$. It is not a direct consequence of Theorem 3.1 because $\mathbf{T}^N_s[\tilde{h}_{0:s}]$ is a function of x_s which depends on N. We have the following theorem.

Theorem 3.5. Let $h_{0:t}(x_{0:t})$ be an additive functional such that $h_s(x_s, x_{s+1})$ is bounded for all $s \in [0:t-1]$. Then

$$\mathcal{V}^{N,\mathsf{BS}}_{0:t|t}(h_{0:t}) \xrightarrow{\mathbb{P}} \mathcal{V}^{\mathrm{FFBS}}_{0:t|t}(h_{0:t}).$$

The proof consists in showing that

$$\lim_{N\to\infty} \mathbb{E}\left[\left(\mathcal{Q}^N_{e_s,t}\left(\left[g_t\left\{\mathbf{T}^N_s[\tilde{h}_{0:s}] + \tilde{h}_{s:t}\right\}\right]^{\otimes 2}\right) - \mathcal{Q}_{e_s,t}\left(\left[g_t\left\{\mathbf{T}_s[\tilde{h}_{0:s}] + \tilde{h}_{s:t}\right\}\right]^{\otimes 2}\right)\right)^2\right] = 0$$

(remember that $\phi^N_{0:t|t}(h_{0:t})$ is a constant w.r.t. $x_{0:t}$ and converges to $\phi_{0:t|t}(h_{0:t})$, so we do not need to consider it). Using the triangle inequality and Theorem 3.1, it suffices to show

$$\lim_{N\to\infty} \mathbb{E}\left[\left(\mathcal{Q}^N_{e_s,t}\left(g_t\left\{\mathbf{T}^N_s[\tilde{h}_{0:s}] + \tilde{h}_{s:t}\right\}^{\otimes 2}\right) - \mathcal{Q}^N_{e_s,t}\left(\left[g_t\left\{\mathbf{T}_s[\tilde{h}_{0:s}] + \tilde{h}_{s:t}\right\}\right]^{\otimes 2}\right)\right)^2\right] = 0.$$

The proof of this last result consists of three steps. First, for any x_s, $\mathbf{T}^N_s[\tilde{h}_{0:s}](x_s)$ converges almost surely to $\mathbf{T}_s[\tilde{h}_{0:s}](x_s)$; next, we show that the result is satisfied for $t = s$; finally, it remains valid by induction, in the same spirit of Theorem 3.1.

Practical computation of $\mathcal{V}_{0:t|t}^{N,\mathsf{BS}}(h)$ - We finally propose the computation of our estimator for the marginal smoothing problem, *i.e.* $h_{0:t}(x_{0:t}) = h_l(x_l)$, in the spirit of what we have done for computing $\mathcal{V}_{\gamma,t}^{N,\mathsf{BS}}(h)$. For marginal functionals $h_l(x_l)$, (3.31) becomes

$$\mathcal{V}_{l|t}^{N,\mathsf{BS}}(h_l) = \frac{1}{\gamma_{t+1}^N(\mathbf{1})^2}\left\{\sum_{s=0}^{l}\mathcal{Q}_{e_s,t}^{N,\mathsf{BS}}\left(\left[g_t\{h_l - \phi_{l|t}^N(h_l)\}\right]^{\otimes 2}\right) + \sum_{s=l+1}^{t}\mathcal{Q}_{e_s,t}^{N,\mathsf{BS}}\left(\left[g_t\{\mathbf{T}_s^N[h_l] - \phi_{l|t}^N(h_l)\}\right]^{\otimes 2}\right)\right\}. \tag{3.32}$$

and can be rewritten as

$$\mathcal{V}_{l|t}^{N,\mathsf{BS}}(h_l) = R_{1,t}^l - \phi_{l|t}^N(h_l)R_{2,t}^l + \phi_{l|t}^N(h_l)^2 R_t,$$

where

$$\begin{cases} R_{l,t}^1 &= \sum_{s=0}^{l}\mathcal{Q}_{e_s,t}^{N,\mathsf{BS}}\left([g_t h_l]^{\otimes 2}\right) + \sum_{s=l+1}^{t}\mathcal{Q}_{e_s,t}^{N,\mathsf{BS}}\left(\left[g_t\mathbf{T}_s^N[h_l]\right]^{\otimes 2}\right), \\ R_{l,t}^2 &= \left\{\sum_{s=0}^{l}\mathcal{Q}_{e_s,t}^{N,\mathsf{BS}}(g_t h_l \otimes g_t) + \mathcal{Q}_{e_s,t}^{N,\mathsf{BS}}(g_t \otimes g_t h_l)\right\} \\ &\qquad + \left\{\sum_{s=l+1}^{t}\mathcal{Q}_{e_s,t}^{N,\mathsf{BS}}(g_t\mathbf{T}_s^N[h_l] \otimes g_t) + \mathcal{Q}_{e_s,t}^{N,\mathsf{BS}}(g_t \otimes g_t\mathbf{T}_s^N[h_l])\right\}, \\ R_t &= \sum_{s=0}^{t}\mathcal{Q}_{e_s,t}^{N,\mathsf{BS}}(g_t^{\otimes 2}). \end{cases}$$

As previously, the objective consists in computing sequentially the sums. However, we have to take into account that they rely on $\mathcal{Q}_{e_s,t}^{N,\mathsf{BS}}([g_t h_l]^{\otimes 2})$ and involve a function at time $l \leq t$. To that end, we need to generalize the definition of $\mathcal{T}_t^b(K_t^1, K_t^2)$ in (3.20) and we introduce for a functional $\mathbf{h}_l$ from X^2

$$\mathcal{T}_t^{e_s}[\mathbf{h}_l](K_t^1, K_t^2) = \mathbb{E}_{\mathsf{BS}}\left[\mathrm{I}_{e_s,t}\left(K_{0:t}^1, K_{0:t}^2\right)\mathbf{h}_l(\xi_l^{K_l^1}, \xi_l^{K_l^2})\middle|\mathcal{F}_{t-1}, K_t^1, K_t^2\right]. \tag{3.33}$$

For example, from (3.17), we have

$$\mathcal{Q}_{e_s,t}^{N,\mathsf{BS}}\left([g_t h_l]^{\otimes 2}\right) = N\left(\frac{N}{N-1}\right)^t \gamma_t^N(\mathbf{1})^2\frac{1}{N^2}\sum_{i,j}\mathcal{T}_t^{e_s}[h_l^{\otimes 2}](i,j)g_t(\xi_t^i)g_t(\xi_t^j).$$

For this particular term, it remains to propagate $\sum_{i,j}\mathcal{T}_t^{e_s}[h_l^{\otimes 2}](i,j)g_t(\xi_t^i)g_t(\xi_t^j)$. Turning back to the general case, we also generalize the expression of $S_t(i,j)$ in (3.21) by introducing

$$S_{l,t}^1(K_t^1, K_t^2) = \begin{cases} \sum_{s=0}^{l}\mathcal{T}_t^{e_s}[h_l^{\otimes 2}](K_t^1, K_t^2) + \sum_{s=l+1}^{t}\mathcal{T}_t^{e_s}\left[\mathbf{T}_s^N[h_l]^{\otimes 2}\right](K_t^1, K_t^2) & \text{if} \quad t > l, \\ S_l(K_l^1, K_l^2)h_l^{\otimes 2}(\xi_l^{K_l^1}, \xi_l^{K_l^2}) & \text{if} \quad t = l, \end{cases}$$

$$S_{l,t}^2(K_t^1, K_t^2) = \begin{cases} \sum_{s=0}^{l}\mathcal{T}_t^{e_s}[h_l^{\oplus 2}](K_t^1, K_t^2) + \sum_{s=l+1}^{t}\mathcal{T}_t^{e_s}\left[\mathbf{T}_s^N[h_l]^{\oplus 2}\right](K_t^1, K_t^2) & \text{if} \quad t > l, \\ S_l(K_l^1, K_l^2)h_l^{\oplus 2}(\xi_l^{K_l^1}, \xi_l^{K_l^2}) & \text{if} \quad t = l, \end{cases}$$

where for any functional h_l, $h_l^{\oplus^2}(x_l) = h_l(x_l) + h_l(x_l')$. With these definitions, $\mathcal{V}_{l|t}^{N,\mathsf{BS}}(h_l)$ reads

$$\mathcal{V}_{l|t}^{N,\mathsf{BS}}(h_l) = N\left(\frac{N}{N-1}\right)^t \sum_{i,j\in[N]^2}\mathcal{W}_t^i\mathcal{W}_t^j\left\{S_{l,t}^1(i,j) - \phi_{l|t}^N(h_l)S_{l,t}^2(i,j) + \phi_{l|t}^N(h_l)^2 S_t(i,j)\right\};$$

Using the same rationale as for the update of $S_t(i,j)$ in (3.23), $S^1_{l,t}(i,j)$ and $S^2_{l,t}(i,j)$ can be updated as

$$S^1_{l,t+1}(i,j) = \mathcal{T}^{e_{t+1}}_{t+1}(i,j)\mathbf{T}^N_{t+1}[h_l](\xi^i_{t+1})\mathbf{T}^N_{t+1}[h_l](\xi^j_{t+1}) + \mathbb{1}_{i\neq j} \sum_{m,n\in[N]^2} \beta^{\mathsf{BS}}_{t+1}(i,m)\beta^{\mathsf{BS}}_{t+1}(j,n)S^1_{l,t}(m,n), \quad (3.34)$$

and

$$S^2_{l,t+1}(i,j) = \mathcal{T}^{e_{t+1}}_{t+1}(i,j)\{\mathbf{T}^N_{t+1}[h_l](\xi^i_{t+1}) + \mathbf{T}^N_{t+1}[h_l](\xi^j_{t+1})\} + \mathbb{1}_{i\neq j} \sum_{m,n\in[N]^2} \beta^{\mathsf{BS}}_{t+1}(i,m)\beta^{\mathsf{BS}}_{t+1}(j,n)S^2_{l,t}(m,n), \quad (3.35)$$

which enables us to compute sequentially $\mathcal{V}^{N,\mathsf{BS}}_{l|t}(h_l)$.

Numerical experiments (Fig. 3.4) - We go on with model (3.30) for which we generate 4 sequences of 160 observations. We next compute $\mathcal{V}^{N,\mathsf{BS}}_{l|t}(h_l)$ for $h_l(x_l) = x_l$, $l = 100$, $t \in [100:160]$ and $N = 5000$. The results are displayed in Fig. 3.4. When t increases the variance tends to be stable. This is not surprising since future observations do not affect the estimator $\phi^N_{l|t}(x_l)$ when t becomes large. Our estimator is close to the true variance that we estimated in the same way as our previous numerical experiments.

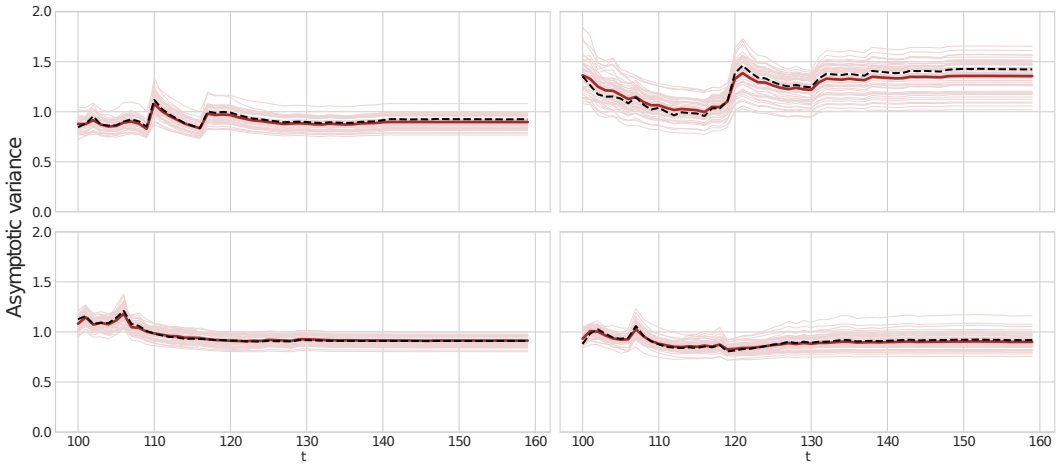


Figure 3.4: Asymptotic variance estimates for four different observation records of the marginal mean $\phi^N_{100|t}(\mathrm{Id})$ where $t \in [100, 160]$. The blurred brown lines on the left plot represent 50 runs and the solid brown line their average. The black dashed line is the crude variance estimator. The number of particles N is set to 5000.

Pseudo codes for computing $\mathcal{V}_{\gamma,t}^{N,\mathsf{BS}}(h)$, $\mathsf{V}_{\gamma,t}^{N,M}(h)$ and $\mathcal{V}_{l|t}^{N,\mathsf{BS}}(h_l)$

We give the algorithms for computing our main estimators. If $A, B \in \mathbb{R}^{M\times N}$ are two matrices, then the Hadamard product $A \odot B$ is the element-wise product, i.e for all $1 \leq i \leq M$ and $1 \leq j \leq N$, $(A \odot B)_{i,j} = a_{i,j}b_{i,j}$. If $A \in \mathbb{R}^{N\times N}$, then $\mathrm{Diag}(A)$ is the $N \times N$ diagonal matrix such that for all $1 \leq i \leq N$, $\mathrm{Diag}(A)_{i,i} = A_{i,i}$ and if $\boldsymbol{x} \in \mathbb{R}^{N\times 1}$ then $\mathrm{Diag}(\boldsymbol{x})$ is the $N \times N$ diagonal matrix such that $\mathrm{Diag}(\boldsymbol{x})_{i,i} = \boldsymbol{x}_i$. If f is a mapping from $[N]^2$ to $\mathbb{R}$, we denote by $\boldsymbol{f}$ the associated $N \times N$ matrix such that $\boldsymbol{f}_{i,j} = f(i,j)$.

Algorithm 3.2 Variance estimators $\mathcal{V}_{\gamma,t}^{N,\mathsf{BS}}(h)$ (see (3.24)) and $\mathsf{V}_{\gamma,t}^{N,M}(h)$ (see (3.29)) associated to $\gamma_t^N(h)$

Require: $M, \omega_t^{1:N}, \xi_t^{1:N}, \xi_{t+1}^{1:N}, \boldsymbol{\mathcal{T}}_t^{\mathbf{0}}$ and $\gamma_t^N(\mathbf{1})$
  Compute $\boldsymbol{\beta}_{t+1}^{\mathsf{BS}}$
  if *PaRis* **then**
    for $k \in [1:N]$ **do**
      Sample $J_{k,t}^{1:M} \overset{\text{i.i.d}}{\sim} \beta_{t+1}^{\mathsf{BS}}(k,.)$
    end for
    for $(k,l) \in [1:N]^2$ **do**
      Set $\mathcal{T}_{t+1}^{\mathbf{0}}(k,l) = \mathbb{1}_{k\neq l}\sum_{i=1}^M \mathcal{T}_t^{\mathbf{0}}(J_{k,t}^i, J_{l,t}^i)/M$
    end for
  else
    Compute $\overline{\boldsymbol{\mathcal{T}}}_{t+1}^{\mathbf{0}} = \boldsymbol{\beta}_{t+1}^{\mathsf{BS}}\boldsymbol{\mathcal{T}}_t^{\mathbf{0}}\boldsymbol{\beta}_{t+1}^{\mathsf{BS}\prime}$.
    Set $\boldsymbol{\mathcal{T}}_{t+1}^{\mathbf{0}} = \overline{\boldsymbol{\mathcal{T}}}_{t+1}^{\mathbf{0}} - \mathrm{Diag}(\overline{\boldsymbol{\mathcal{T}}}_{t+1}^{\mathbf{0}})$.
  end if
  Compute $\boldsymbol{\mathcal{Q}} = \boldsymbol{\mathcal{T}}_{t+1}^{\mathbf{0}} \odot \left[h(\xi_{t+1}^{1:N})h(\xi_{t+1}^{1:N})^\top\right]$. ▷ h is applied elementwise
  return $N\gamma_t^N(\mathbf{1})^2\left\{\eta_t^N(h)^2 - N^{t-2}\sum_{i,j\in[N]^2}\boldsymbol{\mathcal{Q}}_{i,j}/(N-1)^{t+1}\right\}$, $\boldsymbol{\mathcal{T}}_{t+1}^{\mathbf{0}}$.

Algorithm 3.3 Variance estimator $\mathcal{V}_{l|t}^{N,\mathsf{BS}}(h_l)$ (see (3.32)) associated to $\phi_{l|t}^N(h_l)$

Require: $\boldsymbol{\mathcal{W}}_{t+1}^{1:N}, \boldsymbol{\mathcal{W}}_t^{1:N}, \boldsymbol{\beta}_{t+1}^{\mathsf{BS}}, \mathbf{T}t+1^N[h_l], \boldsymbol{\mathcal{T}}_t^{\mathbf{0}}, \boldsymbol{S}_{l,t}^1, \boldsymbol{S}_{l,t}^2, \phi_{l|t+1}^N(h_l)$
  Compute $\overline{\boldsymbol{\mathcal{T}}}_{t+1}^{e_{t+1}} = \boldsymbol{\beta}_{t+1}^{\mathsf{BS}}\boldsymbol{\mathcal{T}}_t^{\mathbf{0}}\boldsymbol{\mathcal{W}}_t$, $\overline{\boldsymbol{\mathcal{T}}}_{t+1}^{\mathbf{0}} = \boldsymbol{\beta}_{t+1}^{\mathsf{BS}}\boldsymbol{\mathcal{T}}_t^{\mathbf{0}}\boldsymbol{\beta}_{t+1}^{\mathsf{BS}\top}$, $\widetilde{\boldsymbol{S}}_{t+1} = \boldsymbol{\beta}_{t+1}^{\mathsf{BS}}\boldsymbol{S}_t\boldsymbol{\beta}_{t+1}^{\mathsf{BS}\top}$
  Set $\boldsymbol{\mathcal{T}}_{t+1}^{e_{t+1}} = \mathrm{Diag}(\overline{\boldsymbol{\mathcal{T}}}_{t+1}^{e_{t+1}})$, $\boldsymbol{\mathcal{T}}_{t+1}^{\mathbf{0}} = \overline{\boldsymbol{\mathcal{T}}}_{t+1}^{\mathbf{0}} - \mathrm{Diag}(\overline{\boldsymbol{\mathcal{T}}}_{t+1}^{\mathbf{0}})$, $\boldsymbol{S}_{t+1} = \widetilde{\boldsymbol{S}}_{t+1} - \mathrm{Diag}(\widetilde{\boldsymbol{S}}_{t+1}) + \mathcal{T}_{t+1}^{e_{t+1}}$
  for i $\in \{1,2\}$ **do**
    Compute $\widetilde{\boldsymbol{S}}_{l,t+1}^i = \boldsymbol{\beta}_{t+1}^{\mathsf{BS}}\boldsymbol{S}_{l,t}^i\boldsymbol{\beta}_{t+1}^{\mathsf{BS}\top}$.
    Set $\widetilde{\boldsymbol{S}}_{l,t+1}^i = \widetilde{\boldsymbol{S}}_{t+1}^i - \mathrm{Diag}(\widetilde{\boldsymbol{S}}_{t+1}^i)$
  end for
  Set $\boldsymbol{S}_{l,t+1}^1 = \widetilde{\boldsymbol{S}}_{l,t+1}^1 + \boldsymbol{\mathcal{T}}_{t+1}^{e_{t+1}} \odot \left[\mathbf{T}_{t+1}[h_l]\mathbf{T}_{t+1}[h_l]^\top\right]$
  Set $\boldsymbol{S}_{l,t+1}^2 = \widetilde{\boldsymbol{S}}_{l,t+1}^2 + \boldsymbol{\mathcal{T}}_{t+1}^{e_{t+1}} \odot \left[\mathbf{T}_{t+1}[h_l] + \mathbf{T}_{t+1}[h_l]^\top\right]$
  Set $\overline{\boldsymbol{S}}_{l,t+1} = \boldsymbol{S}_{l,t+1}^1 - \phi_{l|t+1}^N(h_l)\boldsymbol{S}_{l,t+1}^2 + \phi_{l|t+1}^N(x_l)^2\boldsymbol{S}_{t+1}$
  return $N^{t+2}/(N-1)^{t+1}\sum_{i,j\in[N]^2}\mathcal{W}_{t+1}^i\mathcal{W}_{t+1}^j\overline{S}_{l,t+1}(i,j), \mathcal{T}_{t+1}^{\mathbf{0}}, \boldsymbol{S}_{l,t+1}^1, \boldsymbol{S}_{l,t+1}^2, \boldsymbol{S}_{t+1}$.

CHAPTER 4

About the expressivity of latent variable models

The previous chapters were devoted to problem $\mathbf{P}.2$ (and a part of $\mathbf{P}.3$) in the HMC model. We now focus on $\mathbf{P}.1$ in the generative models introduced in chapter 1. Remember that when we deal with time series, our initial objective consists in choosing a proper generative model for building the distribution of $\{Y_t\}_{t\in\mathbb{N}}$ (here, we will assume that $Y_t \in \mathbb{R}$). Probabilistic models based on a latent process which is now denoted as $\{H_t\}_{t\in\mathbb{N}}$ has been used in many applications. The reason why we use another notation is that we emphasize that $\{H_t\}_{t\in\mathbb{N}}$ does not necessarily represent a physical process of interest. In this chapter, we address the comparison of these generative models without considering their associated computational challenges. More precisely, we compare the models on the basis of their expressivity, *i.e.* we want to compare the distributions $p_\theta(y_{0:t})$ induced by each model, so θ represents the set of parameters associated to a given model among the HMC, the RNN or the PMC.

The motivation of this work has emerged from some problems addressed during the CIFRE thesis of Achille Salaun (2017-2021) that I supervised with F. Desbouvries (Télécom SudParis), A. Bouillard (Nokia Bell-labs, then Huawei) and M-O. Buob (Nokia Bell-labs). A part of this work was devoted to the prediction of alarms in telecommunication networks. Such alarms can be considered as time series and so we wonder what model fits the data well. From a practical point of view, a direct solution would have consisted in comparing the models directly on a dataset. In this work, we propose a different approach since our main motivation is to understand the impact of the structural differences between the HMC, the RNN and their direct generalization [17], in the linear case. Because these models can be seen as particular instance of the (generative) PMC, we also propose an extension of our initial analysis to such models. This extension has has been studied during the beginning of the thesis of Katherine Morales (2020-2023) [14].

4.1 Background

HMC and RNN generative models - Let us start by reintroducing the generative models that we want to compare. As stated in chapter 1, the prediction of a future observation in an HMC can be done from the computation of the predictive likelihood

$$p(y_t|y_{0:t-1}) = \int p(y_t|h_t)p(h_t|y_{0:t-1})\nu(\mathrm{d}h_t) = \eta_t(g_t).$$

When the parameters of the models are unknown (so p becomes p_θ), θ can be estimated from the EM-algorithm which relies on the computation of

$$Q(\theta,\theta^{(i)}) = \mathbb{E}_{\theta^{(i)}}\left(\log\left(p_\theta(H_{0:t},Y_{0:t})\right)|Y_{0:t}=y_{0:t}\right);$$

$Q(\theta,\theta^{(i)})$ is nothing more than the expectation of a particular additive functional $h_{0:t,\theta}$,

$$Q(\theta,\theta^{(i)}) = \phi_{0:t|t,\theta^{(i)}}(h_{0:t,\theta}),$$

and the problem of approximating such moments have been addressed in the previous chapter. Consequently, in the context of time series forecasting, using an HMC requires to run a smoothing algorithm to estimate the parameters of the model (Kantas et al., 2015) and next a particle filter to compute sequentially $\eta_t^N(g_t)$.

By contrast, the construction of the RNN avoids to resort to such approximations. Indeed, a generative RNN is defined from

$$p_\theta(y_{0:t}) = p_\theta(y_0) = \prod_{s=1}^{t} p_\theta(y_s|y_{0:s-1}) \tag{4.1}$$

where $p_\theta(y_t|y_{0:t-1})$ is deduced from a parameterized distribution $p_\theta(y_t|h_t)$ and a latent variable h_t computed from a given parameterized function f_θ,

$$\begin{aligned} p_\theta(y_t|y_{0:t-1}) &= p_\theta(y_t|h_{t-1}), \\ h_{-1} &= 0, \\ h_{t-1} &= f_\theta(h_{t-2}, y_{t-1}). \end{aligned} \tag{4.2}$$

θ represents the parameters of the function f_θ and of the conditional distribution $p_\theta(y_t|h_t)$. By construction, the likelihood is directly computable, so the estimation of θ can be done by running an ascent gradient method on $p_\theta(y_{0:t})$ w.r.t. θ. Due to the sequential structure of the model, the gradient can be computed with the backpropagation algorithm (Hochreiter and Schmidhuber, 1997).

In summary, these two models can be used for the common objective of time series forecasting and both are generative models $p_\theta(y_{0:t})$ based on latent variables. For the HMC, this distribution is implicit and relies on *stochastic* latent variables; for the RNN, the distribution is explicit and relies on *deterministic* (given the past observations) latent variables. For each model, inference algorithms are available and have been well studied. Now, from a practical point of view, if the objective is to design a generative model which aims at modelling a times series, should we use an HMC or an RNN? It may be to possible to answer experimentally but the conclusions will depend on the dataset considered. As far as we are aware, such comparisons have indeed been done experimentally for several problems (Deshmukh, 2020; Bikmukhamedov et al., 2020). We also note that the HMCs considered in these comparisons consist of models with discrete hidden states. However, nothing prevents from using the general HMC we described before as a generative model, in particular HMCs with continuous latent variables.

If we put aside the computational aspects recalled above, the question boils down to comparing the distributions $p_\theta(y_{0:t})$ resulting of each construction. Of course, this is a thorny issue because the nature of the distribution $p_\theta(y_{0:t})$ is unknown in both cases; even if we have its expression for the RNN, its characteristics such as its covariance matrix are not computable. Under linear assumptions, our comparison relies on an original tool borrowed from the stochastic realization theory (Faurre, 1976; Gevers and Wouters, 1978; Faurre, 1979; Gevers, 2006; Caines, 2018) that we next recall and which is a part of systems theory (Chen, 1970; Kailath, 1980; Chui and Chen, 2012).

Deterministic realization theory - Let us consider a linear discrete time and deterministic system with state h_t,

$$\begin{aligned} h_{t+1} &= Fh_t + Nu_t, \\ y_t &= Hh_t, \end{aligned} \tag{4.3}$$

where F (resp. N, H) are $n \times n$ (resp. $n \times 1$, $1 \times n$) matrices (we only deal here with the case where observation y_t and known input u_t are one-dimensional). The mapping between input u_t and output y_t is

given by the convolution equation $y_t = \sum_{k=1}^{+\infty} H_k u_{t-k}$, where the lags H_k of the impulse response (the so-called Markov parameters of the system) satisfy

$$H_k = HF^{k-1}N, \text{ for all } k \geq 1. \tag{4.4}$$

Equivalently, the strictly causal transfer function $H(z) = \sum_{k=1}^{+\infty} H_k z^{-k}$ can be written as $H(z) = H(zI - F)^{-1}N$.

The deterministic realization problem consists in building three matrices H, F, N, with F of minimal dimension, from the impulse response of the system, *i.e.* move from the infinite representation $(H_k)_{k\in\mathbb{N}^*}$ to the finite representation (H, F, N), with F of minimal dimension. The key tool for this problem is the infinite Hankel matrix

$$\mathcal{H}_\infty = \begin{bmatrix} H_1 & H_2 & H_3 & \dots \\ H_2 & H_3 & & \\ H_3 & & & \\ \vdots & & & \end{bmatrix}.$$

In model (4.3), it can be seen from (4.4) that $\mathcal{H}_\infty$ factorizes as

$$\mathcal{H}_\infty = \begin{bmatrix} H \\ HF \\ HF^2 \\ \vdots \end{bmatrix} .[N, FN, F^2N, ...], \tag{4.5}$$

and so has finite rank, which moreover is equal to n (the dimension of F) if and only if each factor is itself full rank n. Besides, by introducing between these two factors the matrix $I = TT^{-1}$ (where T is any invertible matrix), we get an equivalent factorization. Conversely, if $\mathcal{H}_\infty$ has finite rank n, then it can be factorized as a product of two factors of dimensions $(\infty \times n)$ and $(n \times \infty)$, both of them being of full rank n, and due to the Hankel structure, there exists $F_{n\times n}, N_{n\times 1}, H_{1\times n}$ so that (4.5) (and so (4.4)) is satisfied. Moreover, from the proposition below, all minimal realizations of $H(z)$ are isomorphic:

Proposition 4.1. (Ho and Kalman, 1966, Proposition 3) (H_1, F_1, N_1) and (H_2, F_2, N_2) are two minimal realizations of $H(z)$ if and only if there exists T invertible such that $F_2 = TF_1T^{-1}$, $N_2 = TN_1$ and $H_2 = H_1T^{-1}$.

Numerically efficient deterministic realization algorithms have been proposed by De Jong (1975, 1978).

Stochastic realization theory - Let us now consider the state space system

$$\begin{aligned} H_{t+1} &= FH_t + \tilde{U}_t, \\ Y_t &= Hh_t + \tilde{V}_t, \end{aligned} \tag{4.6}$$

(so we now deal with random variables) where H_0 is zero-mean and uncorrelated with $(\tilde{U}_t, \tilde{V}_t)$, and where $(\tilde{U}_t, \tilde{V}_t)$ is a zero-mean, uncorrelated, stationary random process with

$$\mathbb{E}\left[\begin{bmatrix} \tilde{U}_t \\ \tilde{V}_t \end{bmatrix} . \begin{bmatrix} \tilde{U}_{t'}^T & \tilde{V}_{t'}^T \end{bmatrix}\right] = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \delta_{t,t'},$$

where $\delta_{t,t'}$ is the Kronecker symbol. In addition, we focus on stationnary latent process $\{H_t\}_{t\in\mathbb{N}}$ Consequently, the covariance matrix $P = \mathbb{E}[H_t H_t^T]$ does not depend on t and satisfies

$$P = FPF^T + Q.$$

In this case, $\{Y_t\}_{t\in\mathbb{N}}$ is stationary as well and its covariance function is given by

$$\begin{aligned} r_0 &= \mathbb{E}[Y_t^2] = R + HPH^T, \\ r_k &= \mathbb{E}[Y_t Y_{t+k}] = HF^{k-1}\underbrace{(FPH^T + S)}_{N}, \quad \text{for all } k \in \mathbb{N}^*. \end{aligned} \tag{4.7}$$

Starting from a covariance series $\{r_k\}_{k\in\mathbb{N}}$ which satisfies (4.7), the stochastic realization problem consists of building a minimal "Markovian representation" of $(Y_t)_{t\in\mathbb{N}}$, *i.e.* a stationnary state-space system (4.6) with F of minimal dimension. It consists of two steps.

- First, thanks to the structure of function $\{r_k\}_{k\in\mathbb{N}^*}$, we can build a Hankel matrix

$$\mathcal{H}_\infty = \begin{bmatrix} r_1 & r_2 & r_3 & \dots \\ r_2 & r_3 & & \\ r_3 & & & \\ \vdots & & & \end{bmatrix} = \begin{bmatrix} H \\ HF \\ HF^2 \\ \vdots \end{bmatrix} \begin{bmatrix} N & FN & F^2N & \dots \end{bmatrix} \tag{4.8}$$

  which should be compared to factorization (4.5). The first (and, in fact, "deterministic") step of a stochastic realization algorithm consists in building a minimal realization (H, F, N) of $(r_k)_{k\in\mathbb{N}^*}$ (unique up to an invertible matrix).

- At this point, we dispose of (H, F, N) but N remains a function of P and S, and it remains to identify Q and R. This second step is more delicate and the problem must be solved under positivity constraints: P and $\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}$ are covariance matrices and so must be semi-definite positive (≥ 0). If these constraints were not satisfied, the solution would be meaningless. Finally, the problem is as follows: knowing (H, F, N, r_0), we look for (P, Q, R, S) such that

$$\begin{aligned} \begin{bmatrix} P & N \\ N^T & r_0 \end{bmatrix} - \begin{bmatrix} F \\ H \end{bmatrix} P \begin{bmatrix} F^T & H^T \end{bmatrix} &= \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}, \\ \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} &\geq 0, \\ P &> 0, \end{aligned} \tag{4.9}$$

  in which > 0 stands for definite positive (the constraint on P should be, *a priori*, that P is *semi-definite* positive, but indeed it happens that any solution P must be *definite* positive Faurre (1979), see theorem 4.1 below).

  System (4.9) can be seen as a system with three equations and four unknowns (P, Q, R and S), or rather as a system with three equations and three unknowns (Q, R and S) parameterized by P. Finally, P parameterizes solutions of the constrained system (4.9) and we denote as $\mathcal{P}$ the set of parameters

$$\mathcal{P} = \{P \text{ s.t. } (4.9) \text{ is satisfied}\}. \tag{4.10}$$

A result known as the *positive real lemma* connects the positivity of the series $\{r_k\}_{k\in\mathbb{N}}$ (in other words, whether $(r_k)_{k\in\mathbb{N}}$ is a *covariance* series) to the existence of at least one solution to the constrained system (4.9). Let us recall that the infinite series $\{r_k\}_{k\in\mathbb{N}}$ is a covariance function iff. the Toeplitz form $\sum_{i,j=0}^{m} u_i u_j r_{|j-i|}$ is positive or null for all m, *i.e.* iff. the associated Toeplitz matrix is semi-definite positive for all m.

Lemma 4.1 (Positive real lemma Faurre (1979))**.** The (factorizable) series $(r_k)_{k\in\mathbb{N}}$ is a covariance function iff. $\mathcal{P}$ is non void.

In addition, the structure of $\mathcal{P}$ can be described by the following theorem.

Theorem 4.1 (Faurre (1979))**.** The set $\mathcal{P}$ is closed, convex, bounded and definite positive; it admits (for the usual order relation between symmetric matrices) a maximum P^* and a minimum P_*.

Let us finally notice that there exists efficient algorithms for building elements of $\mathcal{P}$ (see Faurre (1979); Caines (2018)).

4.2 HMC vs. RNN from stochastic realization theory

We have now the necessary tools to compare some kind of RNNs and HMCs. We start by interpreting these models as particular instances of a more general model, and we next discuss on the generative properties of this model.

A unified framework - We cast the HMC and RNN generative models into a more general model. The HMC

$$p_\theta(h_{0:t}, y_{0:t}) \overset{\text{HMC}}{=} p(h_0) \prod_{s=1}^{t} p_\theta(h_s|h_{s-1}) \prod_{s=0}^{t} p_\theta(y_s|h_s), \quad \text{for all } t \in \mathbb{N},$$

and the RNN (up to the transformation $h_{t-1} \leftarrow h_t$ in (4.2))

$$p_\theta(y_{0:t}) \overset{\text{RNN}}{=} p_\theta(y_0) \prod_{s=1}^{t} p_\theta(y_s|h_s), \quad \text{with } h_t = f_\theta(h_{t-1}, y_{t-1}) \text{ and } h_0 = 0, \quad \text{for all } t \in \mathbb{N},$$

can be seen as a particular instance of a generative model that we call Generative Unified Model (GUM); it satisfies

$$p_\theta(h_{0:t}, y_{0:t}) \overset{\text{GUM}}{=} p(h_0) \prod_{s=1}^{t} p_\theta(h_s|h_{s-1}, y_{s-1}) \prod_{s=0}^{t} p_\theta(y_s|h_s), \quad \text{for all } t \in \mathbb{N}. \tag{4.11}$$

Indeed, starting from (4.11), we obtain the HMC by setting

$$p_\theta(h_s|h_{s-1}, y_{s-1}) \overset{\text{HMC}}{=} p_\theta(h_s|h_{s-1})$$

or the RNN by setting

$$p_\theta(h_s|h_{s-1}, y_{s-1}) \overset{\text{RNN}}{=} \delta_{f_\theta(h_{s-1}, y_s)}(h_s), \quad h_0 \overset{\text{RNN}}{=} 0, \quad p_\theta(y_0|h_0) \overset{\text{RNN}}{=} p_\theta(y_0).$$

These models share common properties. First, an observation Y_t only depends on its associated hidden state H_t given the past; next, the latent process $\{H_t\}_{t\in\mathbb{N}}$ is a Markov chain, even in the GUM. Indeed, in this case H_t also depends on the observation Y_{t-1} given the past, but

$$p_\theta(h_t|h_{0:t-1}) = \int p_\theta(\mathrm{d}y_{t-1}|h_{t-1})p_\theta(h_t|h_{t-1}, y_{t-1}) = p_\theta(h_t|h_{t-1}).$$

So the main difference between the models comes from the conditional distribution of the observations given the latent process, $p_\theta(y_{0:t}|h_{0:t})$, which does not necessarily factorize as $\prod_{s=0}^t p_\theta(y_s|h_s)$. In terms of comparison between the RNN and the HMC, we can state that:

- in both models, the latent process is Markovian and the distribution of an observation at time t given the past only depends on the latent variable at the same time;
- in an HMC, given (h_{t-1}, y_{t-1}), H_t only depends on h_{t-1} but is stochastic, so $p_\theta(y_{0:t}|h_{0:t}) = \prod_{s=0}^t p_\theta\ (y_s|h_s)$ but $p_\theta(y_{0:t})$ is known in a closed-form expression;
- in an RNN, given (h_{t-1}, y_{t-1}), h_t also depends on y_{t-1} but is deterministic, so $p_\theta(y_{0:t}|h_{0:t})$ is more complex but we have a closed-form expression of $p_\theta(y_{0:t})$.

As we see, comparing the distribution $p_\theta(y_{0:t})$ induced by an RNN or an HMM is equivalent to study that induced by the GUM and next considering the two particular instances. In the GUM, the study of such a distribution is difficult since we do not have a closed-form expression of $p_\theta(y_{0:t}) = \int p_\theta(x_{0:t}, y_{0:t})\nu(\mathrm{d}x_{0:t})$.

Linear and stationary GUM - In order to address a comparison, we consider linear GUMs in which

$$\begin{cases} \mathbb{E}(H_0) & = 0 \\ \mathbb{E}(H_t|h_{t-1}, y_{t-1}) & = ah_{t-1} + cy_{t-1}\ , \\ \mathbb{E}(Y_t|h_t) & = bh_t \end{cases} \quad \begin{cases} \mathbb{V}\mathrm{ar}(H_0) & = \eta \\ \mathbb{V}\mathrm{ar}(H_t|h_{t-1}, y_{t-1}) & = \alpha\ , \\ \mathbb{V}\mathrm{ar}(Y_t|h_t) & = \beta \end{cases}$$

where H_t is an n-dimensional random variable and Y_t is scalar; so the dimensions of a, b and c are $n \times n$, $1 \times n$, $n \times 1$, respectively, η and α are n square covariance matrices, and $\beta \geq 0$. So $(a, b, c, \alpha, \beta, \eta)$ is included in θ. The GUM can be rewritten as a state-space model with additive noise,

$$\begin{aligned} H_t &= aH_{t-1} + cY_{t-1} + U_t, \\ Y_t &= bH_t + V_t, \end{aligned} \tag{4.12}$$

where $\{U_t\}_{t\in\mathbb{N}}$ and $\{V_t\}_{t\in\mathbb{N}}$ are i.i.d., U_i is independent of V_j for all $(i, j) \in \mathbb{N}^2$, $\mathbb{E}(U_t) = 0$, $\mathbb{E}(V_t) = 0$, $\mathbb{V}\mathrm{ar}(U_t) = \alpha$ and $\mathbb{V}\mathrm{ar}(V_t) = \beta$. It is easy to check that $p_\theta(y_{0:t})$ is a zero mean multivariate distribution for all $t \in \mathbb{N}$. If we note $\eta_t = \mathbb{V}\mathrm{ar}(X_t) = \mathbb{E}(X_t X_t^T)$, the covariance function of $\{Y_t\}_{t\in\mathbb{N}}$ is described by

$$\begin{aligned} \mathbb{V}\mathrm{ar}(Y_t) &= \beta + b\eta_t b^T, \quad \text{for all } t \in \mathbb{N}, \\ \mathbb{C}\mathrm{ov}(Y_t, Y_{t+k}) &= b + (a + cb)^{k-1}(a\eta_t b^T + c(\beta + b\eta_t b^T)), \quad \text{for all } (t, k) \in \mathbb{N} \times \mathbb{N}^*. \end{aligned} \tag{4.13}$$

Our comparison is limited to the comparison of the covariance function associated to each model. Note that in the Gaussian case (*i.e.* the conditional distributions in (4.11) are Gaussian or the noises $\{U_t\}_{t\in\mathbb{N}}$ and $\{V_t\}_{t\in\mathbb{N}}$ in (4.12) are Gaussian), then $p_\theta(y_{0:t})$ is also a centered Gaussian and is fully described by (4.13).

Since (4.13) depends on time, we also introduce simple sufficient conditions yielding stationarity of $\{Y_t\}_{t\in\mathbb{N}}$. First, remark that (4.13) depends on t via η_t which satisfies

$$\begin{aligned}\eta_0 &= \eta,\\ \eta_{t+1} &= (\alpha + c\beta c^T) + (a+cb)\eta_t(a+cb)^T,\end{aligned} \tag{4.14}$$

and becomes constant if $\eta_1 = \eta_0$, *i.e.*

$$\eta = (\alpha + cbc^T) + (a+cb)\eta(a+cb)^T. \tag{4.15}$$

This equation admits a semi-definite positive solution if (Gevers and Wouters, 1978; Brockett, 2015)

$$(a+cb) \text{ has all its eigenvalues in } \{z\in\mathbb{C}; |z|<1\}.$$

Under these assumptions, $\{Y_t\}_{t\in\mathbb{N}}$ becomes a stationary process and its associated covariance series $\{r_k = \mathbb{C}ov(Y_t, Y_{t+k})\}_{k\in\mathbb{N}}$ reads

$$\begin{aligned}r_0 &= \beta + b\eta b^T,\\ r_k &= \underbrace{b}_{H}\underbrace{(a+cb)}_{F}^{k-1}\underbrace{(a\eta b^T + c(\beta + b\eta b^T))}_{N}, \quad \text{for all } k\in\mathbb{N}^*.\end{aligned} \tag{4.16}$$

Remark 4.1. The covariance function $\{r_k\}_{k\in\mathbb{N}}$ associated to a stationary HMC or RNN coincides with two particular cases of (4.16):

- setting $c=0$ (HMC), we have $r_k = ba^{k-1}a\eta b^T$;
- in an RNN, the transition between (h_{t-1}, y_{t-1}) and h_t is deterministic so $\alpha = 0$; in, addition remember that $h_0 = 0$ and Y_0 is independent of h_0. So if $\mathbb{V}ar(Y_0) = r_0 = \mathbb{V}ar(Y_t) = \beta + b\eta b^T$, the constraint $\eta = c(\beta + b\eta b^T)c^T$ should also be satisfied to ensure that $\eta_t = \eta$ and $\mathbb{V}ar(Y_t) = r_0$ for all $t\in\mathbb{N}$.

In conclusion, in a linear and stationary GUM, the covariance function $\{r_k\}_{k\in\mathbb{N}}$ can be factorized as

$$r_k = HF^{k-1}N, \quad \text{for all } k\in\mathbb{N}^*.$$

This particular factorization gives rise to the following questions:

1. Let $\{r_k\}_{k\in\mathbb{N}}$ be a given real series; what are the conditions to factorize it as $r_k = HF^{k-1}N$?;
2. if $\{r_k\}_{k\in\mathbb{N}}$ satisfies such a factorization, is it a covariance series?
3. finally, if $\{r_k = HF^{k-1}N\}_{k\in\mathbb{N}^*}$ is indeed a covariance series, which ones can be realized by a GUM, an RNN and an HMC?

The first point has been addressed in the background section and is a direct application of the deterministic realization theory. In order to address the two other points, we could first identify the parameters H, F and N such that $r_k = HF^{k-1}N$ is indeed a covariance matrix (it can be characterized either by the constraint that for all $k\in\mathbb{N}$, the Toeplitz matrix with first row $[r_0, \cdots, r_k]$ is positive semi definite or, equivalently by the Carathéodory-Toeplitz theorem which states that $C(z) = r_0 + 2\sum_{k=1}^{\infty} r_k z^k$ is a Carathéodory function, *i.e.* has positive real part in the open unit disk $\{z\in\mathbb{C}; |z|<1\}$ (Akhiezer and Kemmer, 1965)), and next looking for the parameters $(a,b,c,\alpha,\beta,\eta)$ satisfying (4.16) under positivity constraints. However, the analysis of $C(z)$ may be difficult so we resort to the stochastic realization recalled above and which enables us to address these points simultaneously.

Resorting to stochastic realization theory for the linear GUM and its particular instances - Starting from the state-space representation (4.12) and plugging the observation equation in that describing the latent process, a GUM admits an alternative state-space representation given by (4.6) where

$$\begin{cases} F &= a + cb \\ H &= b \\ R &= \beta \\ Q &= c\beta c^T + \alpha \\ S &= c\beta \end{cases} \Leftrightarrow \begin{cases} a &= F - SR^{-1}H \\ b &= H \\ c &= SR^{-1} \\ \alpha &= Q - SR^{-1}S^T \\ \beta &= R \end{cases}. \tag{4.17}$$

In other words, there is a unique correspondence between the linear and stationary GUMs and the stationary state-space models (4.12) considered in the stochastic realization theory.

Consequently, we are now able to address some properties related to the GUM and its particular instances. Let $\{r_k\}_{k\in\mathbb{N}}$ be a real valued series (not necessarily a covariance series). Then we have seen that it is factorizable (*i.e.* there exists a triplet (H, F, N) such that $r_k = HF^{k-1}N$ for all $k \in \mathbb{N}^*$) iff. the Hankel matrix $\mathcal{H}_\infty$ defined in (4.8) is finite rank; the rank n of $\mathcal{H}_\infty$ is also the minimal dimension of any realization of $\{r_k\}_{k\in\mathbb{N}}$. In addition, this series is a covariance function iff. there exists a matrix P satisfying (4.9). Due to the correspondence between a GUM and a state-space model (4.12), the following Proposition can be seen as a direct consequence of the positive real Lemma 4.1.

Proposition 4.2. Let $\{r_k\}_{k\in\mathbb{N}}$ be a real valued series satisfying $r_k = HF^{k-1}N$, for all $k > 0$, for some triplet (H, F, N). Then $\{r_k\}_{k\in\mathbb{N}}$ is a covariance function if and only if there exists a GUM which produces a stationary observation process described by covariance function $\{r_k\}_{k\in\mathbb{N}}$. In other words, there exists $P > 0$ such that

$$\begin{aligned} Q &= P - FPF^T, \\ R &= r_0 - HPH^T, \\ S &= N - FPH^T, \end{aligned} \tag{4.18}$$

defines a semi-positive definite matrix

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \geq 0. \tag{4.19}$$

Equivalently, $\{r_k\}_{k\in\mathbb{N}}$ is a covariance series iff. there exists a set of parameters $(a, b, c, \alpha, \beta, \eta)$ satisfying (4.15)-(4.16).

This result can be illustrated by Fig. 4.1. Among all the real valued series $\{r_k\}_{k\in\mathbb{N}}$, a GUM can produce any (covariance) function of the South-West quarter.

Among all the covariance series which can be produced by a GUM, we now describe the subset of those which can be produced by an HMC or by an RNN of same dimension (*i.e.* $\dim(h_t) = n$). For the HMC, we have the following (implicit) characterization.

Proposition 4.3. Let a triplet (H, F, N) such that $\{r_k = HF^{k-1}N\}_{k\in\mathbb{N}^*}$ defines a covariance series. Let $\mathcal{P}$ be the set of solutions P of (4.18)-(4.19). Then if there exists P in $\mathcal{P}$ such that

$$N - HFP = 0,$$

the covariance series can be produced by an HMC.

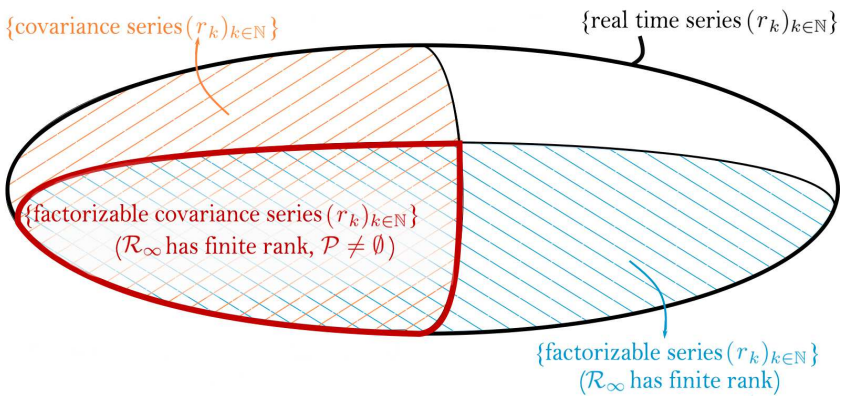


Figure 4.1: This figure represents the set of all real times series. The series $\{r_k\}_{k\in\mathbb{N}}$ which are factorizable covariance series is the South-West quarter of the figure (orange and blue lines). Computing $\mathcal{H}_\infty$ enables to move from the full set to the Southern part, whereas the positive real lemma enables to move from the Southern part to the South-West quarter.

For example, this condition cannot be satisfied if $N \notin \text{Span}(F)$

Remark 4.2. Let us observe that if no solutions can be found in a set $\mathcal{P}_1$ associated to a triplet (H_1, F_1, N_1), then we cannot found any solution in the set $\mathcal{P}_2$ associated to the triplet $(H_2 = H_1T_{12}^{-}1, T_{12}F_2T_{12}^{-}1, T_{12}N_2)$ which produces the same covariance series $\{r_k\}_{k\in\mathbb{N}}$. The reason why is that any solution P_2 of $\mathcal{P}_2$ can be written as $P_2 = T_{12}P_1T_{12}^T$ where $P_1 \in \mathcal{P}$. Consequently, if there is a solution P_1 in $\mathcal{P}_1$ which satisfies $N_1 = H_1F_1P_1$ (so P_1 can be associated to an HMC), then

$$N_2 = T_{12}N_1 = T_{12}FT_{12}^{-1}T_{12}P_1T_{12}^TT_{12}^{-T}H_1^T = F_2P_2H_2^T$$

and so P_2 in $\mathcal{P}_2$ can also be associated to an HMC.

We finally give a description of the deterministic GUM (*i.e.* the transition distribution between $p_\theta(h_t|h_{t-1}, y_{t-1})$ is deterministic) and of the RNN in which an additional constraint on c has to be satisfied (see Remark 4.1).

Proposition 4.4. Let a triplet (H, F, N) such that $\{r_k = HF^{k-1}N\}_{k\in\mathbb{N}^*}$ is a covariance matrix. Let $\mathcal{P}$ be the set of solutions P satisfying (4.18)-(4.19). Then if there exists P in $\mathcal{P}$ such that

$$Q - SR^{-1}S^T = P - FPF^T - (N - FPH^T)(r_0 - HPH^T)^{-1}(N - FPH^T)^T = 0,$$

the covariance series can be produced by a GUM with a deterministic transition. If in addition P satisfies

$$P = SR^{-1}r_0R^{-T}S^T = r_0(r_0 - HPH^T)^{-2}(N - FPH^T)(N - FPH^T)^T,$$

the covariance series can be produced by a traditional RNN initialized to $h_0 = 0$ with a linear activation function.

By construction, if P satisfies the RNN condition above, then P is a rank 1 $n \times n$ semi-definite positive matrix and is positive definite only if $n = 1$. In other words, a factorizable covariance series can be realized by an RNN if the latent vector is monodimensional and the condition above holds but can never be realized by an RNN if $n > 1$.

An illustration in the scalar case (Fig. 4.2) - In this paragraph, we set $\dim(h_t) = n = 1$. For this particular case, (4.18)-(4.19) becomes

$$\begin{cases} Q & = P(1-F^2) \geq 0 \\ R & = r_0 - PH^2 \geq 0 \\ S & = N - HFP \\ QR - S^2 & \geq 0 \end{cases}. \tag{4.20}$$

The first constraint is satisfied if $-1 \leq F \leq 1$. We also show that the last constraint is satisfied if

$$-H^2P^2 + \left[r_0(1-F^2) + 2HFN\right] P - N^2 \geq 0.$$

As a quadratic function of P, it admits a solution provided

$$\Delta = (1-F^2)(r_0(1+F) - 2HN)(r_0(1-F) + 2HN) \geq 0,$$

i.e.

$$\frac{r_0(F-1)}{2} \leq HN \leq \frac{r_0(F+1)}{2},$$

when F satisfies the first constraint. If the pair (F, HN) is valid, $\mathcal{P}$ coincides with the interval $[P_1, P_2]$ where

$$P_i = \frac{2FHN + r_0\left(1-F^2\right) + (-1)^i\sqrt{\Delta}}{2F^2}.$$

Finally, we remark that $[P_1, P_2]$ is included in $[0, r_0/H^2]$ so the constraint $r_0 - PH^2 \geq 0$ is satisfied. In conclusion, in the scalar case, $\mathcal{P}$ is non void iff.

$$\begin{cases} -1 & \leq F \leq 1 \\ \frac{r_0(F-1)}{2} & \leq HN \leq \frac{r_0(F+1)}{2} \end{cases},$$

and coincides with $\mathcal{P} = [P_1, P_2]$. According to the real-positive lemma, we deduce that series defined by r_0 and $\{r_k = HNF^{k-1}\}_{k\in\mathbb{N}^*}$ are covariance series if and only if F and HN satisfy the conditions above.

Such covariance series can be produced by an HMC if P satisfies

$$P = \frac{N}{HF}, \quad P \in \left[0, \frac{r_0}{H^2}\right].$$

Such a P exists provided

$$\begin{cases} 0 \leq HN \leq r_0F, & \text{if } F \geq 0 \\ r_0F \leq HN \leq 0, & \text{if } F \leq 0 \end{cases}.$$

In other words, the HMC cannot produce the covariance series $\{r_k = HNF^{k-1}\}_{k\in\mathbb{N}^*}$ which do not respect the conditions above, contrary to the GUM.

Finally, GUMs with a deterministic transition can describe the same covariance series as the GUM. Indeed, the deterministic condition $QR - S^2 = 0$ is satisfied if $P = P_1$ or $P = P_2$. Even if the set of solutions is reduced to $\{P_1, P_2\}$, it is sufficient to produce any factorizable covariance series of degree n and it requires less parameters than the GUM (since $\alpha = 0$). If we add the RNN constraint $P = c^2r_0$, we show by solving

$$Qr_0 = P_iR \Leftrightarrow P_i(1-F^2)r_0 = P_i(r_0 - P_iH^2), \quad \text{for } i \in \{1, 2\}$$

that there exists a solution if

$$HN = r_0 F$$

or

$$HN = r_0 F(2F^2 - 1).$$

Consequently, an original RNN with a linear activation function produces a restricted set of covariance series but can be easily transformed into a deterministic GUM (or an observation driven model) by considering a random (and not deterministic) X_0.

Fig. 4.2 summarizes this discussion and describes the covariance series described by r_0 and by $\{r_k = HNF^{k-1}\}_{k\geq 0}$ which can be produced by each model in function of (F, HN).

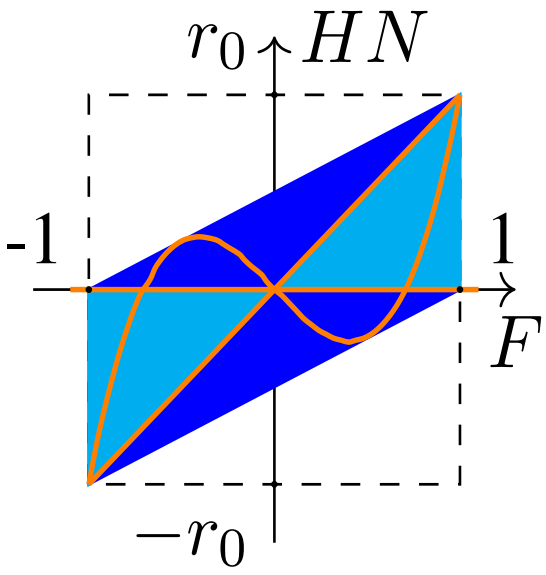


Figure 4.2: Modeling powers of RNN, HMM and GUM with regards to F and HN. The parallelogram (blue+cyan) coincides with the distributions with a covariance function $\mathrm{Cov}(Y_t, Y_{t+k}) = F^{k-1}HN$. Such distributions can be modeled by a GUM or a deterministic GUM. The cyan (resp. orange) areas (resp. curves) coincides with the value of F and HN which can be taken by the HMC (resp. the RNN).

4.3 About the generative power of PMCs

We have seen that a linear and stationary GUM with a latent random variable of dimension n produces a factorizable covariance series $\{r_k\}_{k\in\mathbb{N}}$,

$$r_k = HF^{k-1}N, \qquad \text{for all } k \in \mathbb{N}^*,$$

where F is an $n \times n$ matrix. On the other hand, a series $\{r_k\}_{k\in\mathbb{N}}$ satisfying this factorization is a covariance function if there exists an n-dimensional GUM which can produce it. Actually, a GUM can be seen as a

particular instance of the PMC. In a PMC, we only assume that the pair $\{H_t, Y_t\}_{t\in\mathbb{N}}$ is Markovian,

$$\begin{aligned} p_\theta(h_{0:t}, y_{0:t}) &\overset{\text{PMC}}{=} p_\theta(h_0, y_0)\prod_{s=1}^{t} p_\theta(h_s, y_s|h_{s-1}, y_{s-1}) \\ &= p_\theta(h_0, y_0)\prod_{s=1}^{t} p_\theta(h_s|h_{s-1}, y_{s-1})p_\theta(y_s|h_{s-1:s}, y_{s-1}). \end{aligned}$$

Actually, this property ensures that all the computing Bayesian inference tools (particle filters, EM algorithm,...) developed for the HMC are adaptable for the PMC. Consequently, using PMCs as generative models could be relevant, and we will see some practical examples of these models in the next chapter. At this point, a natural problem is to quantify the gain of this model w.r.t. the previous GUM.

Linear and stationary PMCs - In order to extend our previous study, we include the same linear assumptions; so we consider that the first and second order moments of $p_\theta(h_t|h_{t-1}, y_{t-1})$ and of $p_\theta(y_t|h_{t-1:t}, y_{t-1})$ read

$$\begin{cases} \mathbb{E}([H_0, Y_0]^T) &= \begin{bmatrix} 0 & 0 \end{bmatrix}^T \\ \mathbb{E}(H_t|h_{t-1}, y_{t-1}) &= ah_{t-1} + cy_{t-1} \\ \mathbb{E}(Y_t|h_{t-1:t}, y_{t-1}) &= bh_t + eh_{t-1} + fy_{t-1} \end{cases}, \quad \begin{cases} \mathbb{V}\text{ar}([H_0, Y_0]^T) &= \Sigma_0 = \begin{bmatrix} \eta & \tilde{\gamma}^T \\ \tilde{\gamma} & r_0 \end{bmatrix} \\ \mathbb{V}\text{ar}(H_t|h_{t-1}, y_{t-1}) &= \alpha \\ \mathbb{V}\text{ar}(Y_t|h_t) &= \beta \end{cases}.$$

This generalization involves three new parameters e, f and $\tilde{\gamma}$ of dimension $1\times n$, 1×1 and $1\times n$, respectively. When $e = 0$, $f = 0$ and $\tilde{\gamma} = b\eta$, the model coincides with the GUM of the previous section. An equivalent representation is obtained by considering the first and second order moments of the pair (H_t, Y_t) given (h_{t-1}, y_{t-1}),

$$\mathbb{E}\left(\begin{bmatrix} H_t & Y_t \end{bmatrix}^T |h_{t-1}, y_{t-1}\right) = M\begin{bmatrix} h_{t-1} \\ y_{t-1} \end{bmatrix}, \quad \mathbb{V}\text{ar}\left(\begin{bmatrix} H_t & Y_t \end{bmatrix}^T |h_{t-1}, y_{t-1}\right) = \Sigma_{t|t-1},$$

where

$$M = \begin{bmatrix} a & c \\ ba + e & bc + f \end{bmatrix}, \quad \Sigma_{t|t-1} = \begin{bmatrix} \alpha & \alpha b^T \\ b\alpha & \beta + b\alpha b^T \end{bmatrix}. \tag{4.21}$$

In the previous section, the stationarity of $\{H_t\}_{t\in\mathbb{N}}$ involved that of $\{Y_t\}_{t\in\mathbb{N}}$ due to the structure of the model. By extension, we consider directly that the process $\{H_t, Y_t\}_{t\in\mathbb{N}}$ is stationary. Consequently, $\mathbb{V}\text{ar}([H_0, Y_0]^T) = \Sigma_0$ should satisfy

$$\Sigma_0 = M\Sigma_0 M^T + \Sigma_{t|t-1}. \tag{4.22}$$

This matrix equation describes a set of three constraints which generalizes the constraints (4.15)-(4.16) : in the GUM case, $e = 0$, $f = 0$ and $\mathbb{C}ov(H_0, Y_0) = \eta b^T$, the global constraint (4.22) is reduced to the two constraints described by (4.15) and the first line of (4.16).

Stochastic realization theory for linear PMCs - We are now tempted to resort again to stochastic realization theory to describe the covariance series which can be produced by these models. However, the main difficulty is that they do not admit a state-space model representation (4.6). The reason why is that the

introduction of the new dependencies through parameters (e, f) cancels the Markoviannity of $\{H_t\}_{t\in\mathbb{N}}$ of the previous models. This can be seen if we use a state-space representation of the same form as (4.12) for the PMC,

$$\begin{aligned} H_t &= aH_{t-1} + cY_{t-1} + U_t, \\ Y_t &= bH_t + eH_{t-1} + fY_{t-1} + V_t. \end{aligned} \tag{4.23}$$

If we now plug the second equation into the first one, we are not able to obtain the state-space representation (4.6). The trick is to interpret the PMC as a particular HMC in augmented dimension. Let us set $\tilde{H}_t = (H_t, Y_t)$. Then the linear PMC can be seen as a particular HMC with $n+1$ latent random variables,

$$\begin{aligned} \tilde{H}_t &= MH_{t-1} + U_t, \\ Y_t &= \begin{bmatrix} 0_{1\times n} & 1 \end{bmatrix} \tilde{H}_t, \end{aligned} \tag{4.24}$$

where $\{U_t\}_{t\in\mathbb{N}}$ is a sequence of uncorrelated noises, $\mathbb{E}(U_t) = 0$, $\mathbb{E}(U_t U_t^T) = \Sigma_{t|t-1}$. In other words, our linear PMC can now be seen as a particular state-space model (4.12) of dimension $n+1$, with $H = [0_{1\times n}, 1]$, $S = 0$ and $R = 0$. As a direct consequence, we have the following Proposition.

Proposition 4.5. Let $\tilde{F}$ (resp. $\tilde{N}$) be an $(n+1)\times(n+1)$ (resp. $(n+1)\times 1$) matrix such that the series $\{r_k\}_{k\in\mathbb{N}}$ satisfies $r_k = \begin{bmatrix} 0_{1\times n} & 1 \end{bmatrix} \tilde{F}^{k-1}\tilde{N}$ for all $k \in \mathbb{N}^*$. If there exists $\tilde{P} > 0$ such that

$$\begin{cases} \tilde{Q} &= \tilde{P} - \tilde{F}\tilde{P}\tilde{F}^T \geq 0 \\ \tilde{R} &= r_0 - \begin{bmatrix} 0_{1\times n} & 1 \end{bmatrix} \tilde{P} \begin{bmatrix} 0_{1\times n} & 1 \end{bmatrix}^T = 0 \\ \tilde{S} &= \tilde{N} - \tilde{F}\tilde{P} \begin{bmatrix} 0_{1\times n} & 1 \end{bmatrix}^T = 0 \end{cases}, \tag{4.25}$$

then $\{r_k\}_{k\in\mathbb{N}}$ is a covariance function and can be produced by a PMC.

This proposition shows that it is theoretically possible to aim a subset of the factorizable covariance series of degree $n+1$ with a PMC with n latent random variables, contrary to the GUM.

Remark 4.3. From the previous section, we already know that a GUM of dimension n can only produces covariance series of degree n, at most. However, we could use the same argument as that used for the PMC and interpret the GUM of degree n as a particular HMC of degree $n+1$. However, note that contrary to the PMC, the transition matrix of the GUM in augmented dimension is not free and reads (see (4.21) with $e = 0$, $f = 0$)

$$M = \begin{bmatrix} a & c \\ ba & bc \end{bmatrix} = \begin{bmatrix} I_{n\times n} \\ b \end{bmatrix} \begin{bmatrix} a & c \end{bmatrix}.$$

In this case, the rank of M is lower than $n+1$.

The scalar case - Proposition 4.5 gives an implicit characterization of covariance functions of degree $n+1$ which can be produced by a PMC with n latent variables. We would like to illustrate the result with $n = 1$ and to compare the covariance functions which can be produced by a PMC w.r.t. that produced by the GUM. However, we would first need to describe explicitly the set $\mathcal{P}$ associated to covariances functions of degree $n+1 = 2$ and next looking for those which satisfy the constraints of Proposition 4.5. An explicit characterization of $\mathcal{P}$ is difficult in this case and we use an alternative path. We start again from the (scalar)

stationary PMC described by (4.21) which satisfies (4.22); for clarity, we set $r_0 = 1$ and we parameterize $\tilde{\gamma} = \gamma\eta$.

In order to extend the scalar case of the GUM, we assume that M is diagonalizable, *i.e.* $M = PDP^{-1}$ with

$$\begin{aligned} P &= \begin{bmatrix} \frac{-a+bc+f+K}{2(ab+e)} & \frac{a-bc-f+K}{2(ab+e)} \\ 1 & 1 \end{bmatrix}, \\ D &= \begin{bmatrix} \frac{1}{2}(a+bc+f-K) & 0 \\ 0 & \frac{1}{2}(a+bc+f+K) \end{bmatrix}, \\ P^{-1} &= \begin{bmatrix} -\frac{ab+e}{K} & \frac{a-bc-f+K}{2K} \\ \frac{ab+e}{K} & \frac{-a+bc+f+K}{2K} \end{bmatrix}, \end{aligned}$$

where

$$K = \sqrt{(a+bc+f)^2 - 4(af-ce)}.$$

So it is assumed that $(a+bc+f)^2 - 4(af-ce) \geq 0$, and note that this condition is always satisfied in the GUM ($e = f = 0$). The covariance function $\{r_k\}_{k\in\mathbb{N}}$ is then deduced from that of the joint process $\{H_t, Y_t\}_{t\in\mathbb{N}}$ which reads $\Sigma_0 \times (M^k)^T$. We have the following result.

Proposition 4.6. Let a linear and stationnary (scalar) PMC defined by the transition and the conditional covariance matrices M and Σ in (4.21) and the initial covariance matrix

$$\Sigma_0 = \begin{bmatrix} \eta & \gamma\eta \\ \gamma\eta & 1 \end{bmatrix}.$$

If M is diagonalizable, the covariance function of $\{Y_t\}_{t\in\mathbb{N}}$ reads

$$\mathbb{C}ov(Y_t, Y_{t+k}) = \overline{A}^k(\overline{B} + \frac{1}{2}) - \overline{C}^k(\overline{B} - \frac{1}{2}), \tag{4.26}$$

where

$$\begin{aligned} \overline{A} &= \frac{a+bc+f-K}{2}, \\ \overline{B} &= \frac{a-bc-f-2\gamma\eta(ab+e)}{2K}, \\ \overline{C} &= \frac{a+bc+f+K}{2}, \\ K &= \sqrt{(a+bc+f)^2 - 4(af-ce)} \end{aligned}$$

and where the following stationnarity constraints are satisfied :

$$\begin{aligned} b\eta + (ae + af\gamma + ce\gamma) + fc &= \gamma\eta, \\ (1 - a^2 - 2ac\gamma)\eta - c^2 &\geq 0, \\ 1 - b^2\eta - 2b\eta(\gamma - b) - e\eta(e + 2f\gamma) - f^2 &\geq 0. \end{aligned}$$

If we set $e = f = 0$, then necessarily $\gamma = b$ and this covariance series reduces to (4.13). While the form of the covariance function associated to a scalar linear PMC is more general, it remains difficult to identify if any covariance series of the form (4.26) can be produced by a PMC because identifying $\overline{A}$, $\overline{B}$ and $\overline{C}$ such that (4.26) is indeed a covariance series is a thorny issue. However, we can exhibit some particular covariance functions which can be produced by a PMC but not by a GUM.

Proposition 4.7. Let $\tilde{A}$ and $\tilde{B}$ be two scalars, $r_0 = 1$ and

$$r_k = \begin{cases} \tilde{A}^k & \text{if } k \text{ is even} \\ \tilde{A}^{k-1}\tilde{B} & \text{otherwise.} \end{cases}. \tag{4.27}$$

Then $\{r_k\}_{k\in\mathbb{N}}$ is a covariance function if and only if

$$-1 \leq \tilde{A} \leq 1 \quad \text{and} \quad -\frac{\tilde{A}^2+1}{2} \leq \tilde{B} \leq \frac{\tilde{A}^2+1}{2}, \tag{4.28}$$

and can be realized by a PMC.

The proof relies on the Caratheodory theorem of Akhiezer and Kemmer (1965) which enables us to describe the values of $\tilde{A}$ and of $\tilde{B}$ such that $\{r_k\}_{k\in\mathbb{N}}$ is covariance function. Next, setting $\gamma = b$, and $f = 0$, we show that (4.26) coincides with (4.27) with

$$\tilde{A} = \sqrt{ce} \quad \text{and} \quad \tilde{B} = b(c(1-b^2\eta)+e\eta).$$

Finally, for any $(\tilde{A}, \tilde{B})$ satisfying (4.28), we show that it is possible to find a set of parameters $(a, b, c, e, \eta, \alpha, \beta)$ which satisfies the previous system and the stationnarity constraints (4.22). This result should be compared with that of the GUM in the scalar case. Remember that the GUM of dimension 1 can produce any covariance function

$$r_k = A^{k-1}B.$$

Proposition 4.7 shows that it is possible to produce a covariance function

$$r_k = A^{k-1}B(k).$$

with a switching $B(k)$ satisfying $B(k) = A$ if k is even and $B(k) = B$, otherwise.

CHAPTER 5

Cross benefits of hidden Markov models and recurrent neural networks architectures

This chapter is the result of a cross fertilization between the previous chapters of this manuscript. Each chapter was devoted to a particular problem among $\mathbf{P}.1 - \mathbf{P}.3$. Here, we synthesize three contributions in which the full chain of processing $\mathbf{P}.1$-$\mathbf{P}.3$ is addressed for three different objectives: (i) designing powerful generative models; (ii) designing powerful and interpretable models for Bayesian classification; (iii) designing fast approximations for hidden Markov models with discrete jumps. For each contribution, we propose a probabilistic model in accordance with our objective and we next address the specific Bayesian inference tools for these models.

This work results from the supervising of 3 Ph.D students. Section 5.2 coincides with the first work of K. Morales (2020-2023), a Ph.D. student I supervise with E. Monfrini (Télécom SudParis) [14]. The second contribution has also been led with K. Morales (2020-2023) and H. Gangloff (IRISA, Université de Bretagne Sud), a post-doctoral student [16,19]. Finally, the third contribution coincides with the first project of Y. Janati (2020-2023) during his last year at Télécom SudParis and the beginning of its thesis [15].

5.1 Background

Markovian models - As we have seen in chapters 1 and 4, a PMC

$$p_\theta(h_{0:t}, y_{0:t}) \overset{\text{PMC}}{=} p_\theta(h_0, y_0) \prod_{s=1}^{t} p_\theta(h_s, y_s | h_{s-1}, y_{s-1}), \qquad \text{for all } t \in \mathbb{N}, \tag{5.1}$$

may be more adapted to model observed data $\{Y_t\}_{t\in\mathbb{N}}$ via a latent process $\{H_t\}_{t\in\mathbb{N}}$. However, the choice of the PMC model (*i.e.* of $p_\theta(h_t, y_t|h_{t-1}, y_{t-1})$) is a critical problem. In the case where we want to build a generative model (which means that $\{H_t\}_{t\in\mathbb{N}}$ does not need to be interpretable), the choice of the nature of the transition distribution and its parameterization is not obvious. In the case where the latent process is interpretable, so $H_t \leftarrow X_t$ (*e.g.* X_t coincides with the position of a target), sliding from the intuitive HMC, where $p_\theta(x_t|x_{t-1})$ describes the evolution of the hidden process and $p_\theta(y_t|x_t)$ the relationship between the observation and the hidden random variable at the same time, to the PMC in which we should model the relationship between the hidden state X_t with X_{t-1} but also with Y_{t-1} is a difficult problem. Our objective is thus to introduce "universal" parameterizations based on neural network architectures. In the first application, where the objective is to design a relevant distribution $p_\theta(y_{0:t})$, PMCs parameterized by neural networks aim at estimating this implicit distribution. In the second application, they are introduced to propose a parameterization of the joint distribution $p_\theta(x_{0:t}, y_{0:t})$ in problems where we look for estimating a discrete random variable X_s from $Y_{0:t}$. If we want also to estimate the nature of this distribution, we resort

to TMCs where a third non observed process $\{Z_t\}_{t\in\mathbb{N}}$ is introduced and where

$$p_\theta(z_{0:t}, x_{0:t}, y_{0:t}) = p_\theta(z_0, x_0, y_0) \prod_{s=1}^{t} p_\theta(z_s, x_s, y_s | z_{s-1}, x_{s-1}, y_{s-1}).$$

The same kind of parameterizations are used but we have to take into account that $\{X_t\}_{t\in\mathbb{N}}$ is interpretable, contrary to $\{Z_t\}_{t\in\mathbb{N}}$.

Variational Bayesian inference - This method is the cornerstone of the Bayesian inference algorithms we propose for our highly parameterized models. It provides an alternative when the EM algorithm is not computable or the Monte Carlo approximations proposed in the previous chapters are unreliable (*e.g.* in high dimensional problems). Let us consider the generals problem of computing or approximating a posterior distribution $p_\theta(x|y) \propto p_\theta(x, y)$ known up to a constant when y is observed and X is hidden. Variational Bayesian inference (see *e.g.* (Blei et al., 2017) for a detailed introduction) relies on a parameterized distribution $q_\phi(x|y)$ that is optimized to fit the posterior distribution $p(x|y)$ by minimizing the Kullback-Leibler Divergence (KLD)

$$\begin{aligned} \mathrm{D_{KL}}(q_\phi, p_\theta) &= \int q_\phi(x|y) \log\left(\frac{q_\phi(x|y)}{p_\theta(x|y)}\right) \mathrm{d}x \geq 0, \\ &= \int q_\phi(x|y) \log\left(\frac{q_\phi(x|y)}{p_\theta(x, y)}\right) \mathrm{d}x + \log\left(p_\theta(y)\right) \end{aligned} \tag{5.2}$$

w.r.t. θ . Of course, the choice of the variational distribution $q_\phi(x|y)$ is critical since the first term of the r.h.s. of (5.2) has to be computed or easily approximated, and next optimized w.r.t. ϕ. A popular choice of variational distribution is the mean-field approximation (Bishop, 2006) where the variational components of $x = (x_1, \cdots, x_{d_x})$ are independent given y and one set of parameters ϕ_i is associated to each component x_i, *i.e.* $q_\phi(x|y) = \prod_{i=1}^{d_x} q_{\phi_i}(x_i|y)$ and $\phi = (\phi_1, \cdots, \phi_{d_x})$.

This approach also provides a parameter estimation method when some parameters of the original model p_θ are unknown. Indeed, we deduce from (5.2) that

$$\log p_\theta(y) \geq -\int q_\phi(x|y) \log\left(\frac{q_\phi(x|y)}{p_\theta(x, y)}\right) \mathrm{d}x = F(\theta, \phi), \tag{5.3}$$

where equality holds when $q_\phi(x|y) = p_\theta(x|y)$. Computing the so-called Evidence Lower Bound (ELBO) $F(\theta, \phi)$ and next maximizing it w.r.t. (θ, ϕ) leads to a maximization of a lower bound of the log-likelihood $\log p_\theta(y)$. The resulting variational EM algorithm (Tzikas et al., 2008) is an alternative to the EM algorithm (Dempster et al., 1977) when the original posterior $p_\theta(x|y)$ is not available. This tool will be extended for our sequential models.

5.2 Generative models based on Variational PMCs

Variational Bayesian Inference for PMCs - We start with (5.1) in which we assume that H_t is a continuous random variable. Before introducing particular parameterizations, we consider the general case in which we only assume that

$$p_\theta(h_0, y_0) \quad \text{and} \quad p_\theta(h_t, y_t | h_{t-1}, y_{t-1})$$

are differentiable w.r.t. θ, for all $t \in \mathbb{N}$. To estimate θ, we want to maximize the (uncomputable) log-likelihood $\log(p_\theta(y_{0:t}))$. Since the dimension of H_t may be large, we adapt the variational Bayesian inference framework described previously. In our case, the ELBO becomes

$$F(\theta, \phi) = -\int \log\left(\frac{q_\phi(h_{0:t}|y_{0:t})}{p_\theta(h_{0:t}, y_{0:t})}\right) q_\phi(\mathrm{d}h_{0:t}|y_{0:t}), \tag{5.4}$$

where $q_\phi(h_{0:t}|y_{0:t})$ is a variational distribution depending on a set of parameters ϕ. In the sequential case, the variational distribution

$$q_\phi(h_{0:t}|y_{0:t}) = q_\phi(h_0|y_{0:t}) \prod_{s=1}^{t} q_\phi(h_s|h_{0:s-1}, y_{0:t})$$

should respect the following constraints:

- $q_\phi(h_s|h_{0:s-1}, y_{0:t})$ should be parameterized in such a way that it can be used with any sequence $y_{0:t}$, so ϕ cannot depend on t;
- q_ϕ should be chosen in such a way that $F(\theta, \phi)$ is computable or can be approximated but is also differentiable. Typically, for general PMCs, $F(\theta, \phi)$ is not computable. Interpreting it as an expectation w.r.t. q_ϕ, an unbiased Monte Carlo approximation can be obtained if q_ϕ is a distribution from which we can easily sample. However, the samples depend on ϕ; in order to ensure that the Monte Carlo approximation remains differentiable w.r.t. ϕ, a sample $h^i_{0:t}$ has to be reparamatrized as a differentiable function of ϕ. This is the reparametrization trick (Kingma and Welling, 2014). The final approximation reads

$$\hat{F}(\theta, \phi) = -\frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{q_\phi(h^i_{0:t}|y_{0:t})}{p_\theta(h^i_{0:t}, y_{0:t})}\right), \quad h^i_s \sim q_\phi(h^i_s|h^i_{0:s-1}, y_{0:t}), \text{ for } i \in [1:N],$$

  and where h^i_s is a differentiable function of ϕ, for $i \in [1:N]$ and $s \in [0:t]$.

Example 5.1. A variational distribution satisfying the previous constraint is

$$q_\phi(h_s|h_{0:s-1}, y_{0:t}) = q_\phi(h_s|h_{s-1}, y_s) = \mathcal{N}\left(h_s; f_\phi(h_{s-1}, y_s); \mathrm{diag}(g_\phi(h_{s-1}, y_{s-1:s}))\right),$$

where f_ϕ and g_ϕ are differentiable functions w.r.t. ϕ (*e.g.* they represent the output of neural networks) and diag(.) denotes the diagonal matrix deduced from the values of g_ϕ. In this case, a sample h^i_s can be reparametrized as

$$h^i_s = f_\phi(h^i_{s-1}, y_s) + \mathrm{diag}(g_\phi(h_{s-1}, y_s))^{\frac{1}{2}} \times \epsilon^i_s, \quad \epsilon^i_s \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I), \quad \text{for } i \in [1:N], \quad \text{for } s \in [1:t].$$

The Monte Carlo approximation of the ELBO can be rewritten as

$$\hat{F}(\theta, \phi) = -\frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{q_\phi(h^i_0)}{p_\theta(h^i_0, y_0)}\right) - \frac{1}{N}\sum_{i=1}^{N}\sum_{s=1}^{t} \log\left(\frac{q_\phi(h^i_s|h^i_{s-1}, y_s)}{p_\theta(h^i_s, y_s|h^i_{s-1}, y_{s-1})}\right)$$

and is next optimized w.r.t. (θ, ϕ).

Deep Generative PMCs and experiments - We now propose particular PMC architectures for modelling times series $\{Y_t\}_{t\in\mathbb{N}}$. It is possible to generalize the Stochastic RNN architectures described in Bayer and Osendorfer (2014); Chung et al. (2015) and which are actually particular instances of the GUM (see section 4.2 of the previous chapter) and so of our generative PMCs. Stochastic RNN architectures have provided good experimental results (Chung et al., 2015) so it is natural to compare them with their PMC extension. Our generative PMC consists of a latent process in augmented dimension, $H_t \leftarrow (H_t, Z_t)$,

$$p_\theta(z_t, h_t, y_t|z_{t-1}, h_{t-1}, y_{t-1}) = p_\theta(z_t|z_{t-1}, h_{t-1}, y_{t-1})p_\theta(h_t|z_{t-1:t}, h_{t-1}, y_{t-1})p_\theta(y_t|z_{t-1:t}, h_{t-1:t}, y_{t-1}), \tag{5.5}$$

and so is nothing more than a TMC with transition (5.5). We now need to parameterize (5.5). Let ς, λ and μ be distributions on Z, H and Y, respectively, and parameterized by differentiable (w.r.t. θ) and vector valued functions denoted as s_θ, f_θ and g_θ and which can depend on $(z_{t-1}, h_{t-1}, y_{t-1})$, $(z_{t-1:t}, h_{t-1}, y_{t-1})$ and on $(z_{t-1:t}, h_{t-1:t}, y_{t-1})$, respectively. The transition (5.5) is then parameterized as

$$\begin{aligned} p_\theta(z_t|z_{t-1}, h_{t-1}, y_{t-1}) &= \varsigma\left(z_t; s_\theta(z_{t-1}, h_{t-1}, y_{t-1})\right), \\ p_\theta(h_t|z_{t-1:t}, h_{t-1}, y_{t-1}) &= \lambda\left(h_t; f_\theta(z_{t-1:t}, h_{t-1}, y_{t-1})\right), \\ p_\theta(y_t|z_{t-1:t}, h_{t-1:t}, y_{t-1}) &= \mu\left(y_t; g_\theta(z_{t-1:t}, h_{t-1:t}, y_{t-1})\right). \end{aligned} \tag{5.6}$$

In the context of Stochastic RNN architectures, the variable Z_t is a deterministic summary of the past until time $t-1$ while H_t is a noisy version of Z_t (it is why we have split the latent process in two). Keeping this rationale for the two latent processes (so with a slight abuse, ς coincides with the Dirac measure and is not a pdf), it is possible to include several degrees of generalization of the classical RNN and of the variational RNN (VRNN) of Chung et al. (2015). Our different models are defined in Table 5.1 through the particular dependencies of the involved random variables.

Model \ Parameterized function	s_θ	f_θ	g_θ
RNN	(h_{t-1}, y_{t-1})	X	z_t
VRNN	$(z_{t-1}, h_{t-1}, y_{t-1})$	z_t	(z_t, h_t)
PMC-I	$(z_{t-1}, h_{t-1}, y_{t-1})$	z_t	(z_t, h_t, y_{t-1})
PMC-II	$(z_{t-1}, h_{t-1}, y_{t-1})$	z_t	$(z_t, h_{t-1:t}, y_{t-1})$
PMC-III	$(z_{t-1}, h_{t-1}, y_{t-1})$	z_t	$(z_{t-1:t}, h_{t-1:t}, y_{t-1})$
PMC-IV	$(z_{t-1}, h_{t-1}, y_{t-1})$	(z_t, y_{t-1})	$(z_t, h_{t-1:t}, y_{t-1})$

Table 5.1: Configuration of the dependencies of different deep generative PMCs. For each model, $\{Z_t\}_{t\in\mathbb{N}}$ is deterministic variable given the observations, so ς coincides with the Dirac measure. λ is generally chosen as the Gaussian distribution, while μ depends on the nature of the observations. Remember that in a classical RNN, $\{H_t\}_{t\in\mathbb{N}}$ is not considered.

In our experiments, $\{Y_t\}_{t\in\mathbb{N}}$ are discrete random variables where $Y_t \in \{0,1\}^{d_y}$, so μ coincides with the product of Bernoulli distribution and the output of g_θ with its parameters. For λ, we choose the Gaussian distribution so the output of f_θ coincides with mean and covariance matrix parameters. The variational distribution q_ϕ is chosen as

$$q_\phi(h_t|z_t, h_{t-1}, y_t) = \mathcal{N}\left(h_t; \nu_\phi(z_t, y_t)\right), \tag{5.7}$$

where the output of ν_ϕ is a mean and a diagonal covariance matrix. Note that since $Z_{0:t}$ is deterministic given $(h_{0:t}, y_{0:t})$, its posterior distribution is trivial and we do not need to consider a variational one. Finally, s_θ, f_θ, g_θ and ν_ϕ are neural networks with two hidden layers, we use the ReLu activation function and the outputs of the neural networks are adapted according to their role (*e.g.* the output of g_θ is a layer of d_y sigmoid functions due to the nature of the observations).

We first work on the MNIST dataset (LeCun, 1998) which contains 60000 (resp. 10000) train (resp. test) 28×28 binary images. An observation Y_t consists of a column of the image ($\dim(y_t) = d_y = 28$), and the length of a sequence is $t + 1 = 28$. Each model was trained with a stochastic gradient ascent method on the approximated ELBO with the Adam optimizer (Kingma and Ba, 2015) using a learning rate of 0.001 and a batch size of 512 images. The number of hidden units of each neural network coincides with the dimension d_h of H_t. For d_h, we consider two configurations (see Table 5.2). In the first one, we set $d_h = 100$ for each model; in the second one, we take into account that each model should be compared with the same number of parameters, so we set $d_h = 100$, $d_h = 95$, $d_h = 79$, $d_h = 78$, $d_h = 74$ and $d_h = 162$ for the VRNN, the PMC-I, the PMC-II, the PMC-III, the PMC-IV and the RNN, respectively. The performance of the models is evaluated in terms of the approximated ELBO and log-likelihood of the observations on the test data set; we use a particle filter with the estimated variational distribution as importance distribution and $N = 100$ particles, see Alg. 2.2. In Table 5.3, we report the averaged ELBO and the averaged approximated log-likelihood on the test set assigned by our models. The results with the Config.1 (resp. Config. 2) show that PMC-IV (resp. PMC-II) has the higher averaged ELBO and averaged approximated log-likelihood. As we see, PMCs perform better than VRNN and RNN.

Model \ Data set	MNIST 1		MNIST 2		MIDI	
	d_z	d_h	d_z	d_h	d_z	d_h
RNN	3	100	3	162	300	562
VRNN	3	100	3	100	300	300
PMC-I	3	100	3	95	300	294
PMC-II	3	100	3	79	300	278
PMC-III	3	100	3	78	300	260
PMC-IV	3	100	3	74	300	272

Table 5.2: Dimensions of latent variables for each Deep PMC. s_θ, f_θ, g_θ and ν_ϕ are neural networks with two hidden layers. The number of neurons on each layer coincide with d_h.

Model \ Data set	MNIST, config. 1		MNIST, config. 2	
	ELBO	approx. log-likelihood	ELBO	approx. log-likelihood
RNN	-65,976	-65,976	-65,700	-65,700
VRNN	-67,248	-64,760	-67,222	-64,762
PMC-I	-66,544	-64,076	-67,322	-64,698
PMC-II	-66,784	-64,201	**-66,815**	**-64,255**
PMC-III	-66,518	-63,876	-67,513	-64,876
PMC-IV	**-66,150**	**-63,60318**	-67,648	-64,924

Table 5.3: Averaged ELBO and approximated log-likelihood of the observations on the test set with two different configurations. For the RNN, the ELBO coincides with the (exact) log-likelihood.

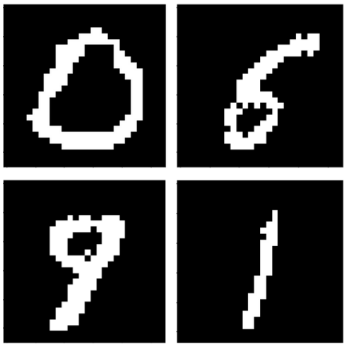


Figure 5.1: Examples of images generated from estimated $p_\theta(y_{0:t})$ of the PMC-II.

We finally consider the three polyphonic music data sets, classical piano music (Piano), folk tunes (Nottingham) and the four-part chorales by J.S. Bach (JSB). Here $y_t \in \{0,1\}^{88}$ consists of a MIDI note than span the whole range of piano from A0 to C8 and we have compared the models in terms of approximated log-likelihood for the same number of parameters (again, by adjusting d_h, see Table 5.2). For each data set, d_z is fixed and is set to 300. The results are presented in Table 5.4 where $d_h = 300$, $d_h = 294$, $d_h = 278$, $d_h = 260$, $d_h = 272$ and $d_h = 562$ for the VRNN, the PMC-I, the PMC-II, the PMC-III, the PMC-IV and the RNN respectively.

Model \ Data set	Piano	Nottingham	JSB
RNN	-10,52	-23,89	-10,77
VRNN	-9,4011	-13,2982	-10,2739
PMC-I	-9,3077	-11,3856	-10,3126
PMC-II	**-8,8265**	-14,8485	-10,2409
PMC-III	-9,2285	-13,3900	-10,1103
PMC-IV	-9,4134	**-10,6323**	**-9,2372**

Table 5.4: Approximated likelihoods on the MIDI data sets. For the RNN, the exact log-likelihood is computed.

5.3 Deep and interpretable hidden Markov models

Our objective is to build powerful generative models with the same rationale as before, but we take into account that the latent process $\{X_t\}_{t\in\mathbb{N}}$ is of physical interest and needs to be estimated. We address this problem in the case where X_t is discrete and represents an interpretable class related to Y_t. In order to illustrate our models, we consider the problem of unsupervised image segmentation which consists in estimating the class of a pixel X_s (*e.g.* black or white) of an image $X_{0:t}$ from a noisy image $Y_{0:t} = y_{0:t}$. So we have

$$X_t \in \Omega = \{\omega_1, \cdots, \omega_K\} \quad \text{and} \quad Y_t \in \mathbb{R}^{d_y}, \quad \text{for all } t \in \mathbb{N}.$$

As experimented in some preliminary simulations, maximizing directly the log-likelihood of highly parameterized models can produce poor classifications compared to a simple models such as a discrete HMC with Gaussian noise. The reason why is that the estimation method aims at maximizing the likelihood of the model but the hidden variables coinciding with the estimated model do not correspond to the desired interpretation.

A general Parameterization of PMCs - First, we only consider the hidden and observed processes; the general parameterization (5.5) becomes

$$\begin{aligned} p_\theta(x_t|x_{t-1}, y_{t-1}) &= \lambda\left(x_t; f_\theta(x_{t-1}, y_{t-1})\right), \\ p_\theta(y_t|x_{t-1:t}, y_{t-1}) &= \mu\left(y_t; g_\theta(x_{t-1:t}, y_{t-1})\right) \end{aligned} \tag{5.8}$$

and parameterizes a PMC

$$p_\theta(x_{0:t}, y_{0:t}) = p_\theta(x_0, y_0) \prod_{s=1}^{t} p_\theta(x_s, y_s|x_{s-1}, y_{s-1}), \quad \text{for all } t \in \mathbb{N}.$$

Example 5.2. This parameterization includes the HMC with discrete hidden states and Gaussian noise (Rabiner, 1989). For clarity, let us assume that $\Omega = \{\omega_1, \omega_2\}$, and we note $\mathrm{sigm}(z) = 1/(1+\exp(-z)) \in [0,1]$ the sigmoid function, $\mathrm{Ber}(x, v)$ the Bernoulli distribution of parameter v evaluated in x. Then this model can be described by

$$\begin{aligned} f_\theta(x_{t-1}, y_{t-1}) &= \mathrm{sigm}\left(b_{x_{t-1}}\right), \\ g_\theta(x_{t-1:t}, y_{t-1}) &= \begin{bmatrix} d_{x_t} & \sigma_{x_t} \end{bmatrix}, \\ \lambda(x; v) &= \mathrm{Ber}(x, v), \\ \mu\left(x; v' = \begin{bmatrix} v'_1; v'_2 \end{bmatrix}\right) &= \mathcal{N}\left(x; v'_1; (v'_2)^2\right), \end{aligned} \tag{5.9}$$

and $\theta = (b_{\omega_i}, d_{\omega_j}, \sigma_{\omega_j} | (\omega_i, \omega_j) \in \Omega \times \Omega)$.

Example 5.3. A direct extension of the previous HMC is the linear and Gaussian PMC in which

$$f_\theta(x_{t-1}, y_{t-1}) = \mathrm{sigm}\left(a_{x_{t-1}} y_{t-1} + b_{x_{t-1}}\right), \tag{5.10}$$

$$g_\theta(x_{t-1:t}, y_{t-1}) = \left[c_{x_{t-1},x_t} y_{t-1} + d_{x_{t-1},x_t};\ \sigma_{x_{t-1},x_t}\right]. \tag{5.11}$$

Actually, whatever the parameterization, the likelihood $p_\theta(y_{0:t})$ and the posterior probabilities $p_\theta(x_s|y_{0:t})$ can be computed for $s \in [0:t]$. The reason why in that in discrete PMCs, these probabilities can be deduced from

$$\alpha_{\theta,s}(x_s) = p_\theta(x_s, y_{0:s}), \quad \beta_{\theta,s}(x_s) = p_\theta(y_{s+1:t}|x_s, y_s), \quad \beta_{\theta,t}(x_t) = 1,$$

which are sequentially computable (Pieczynski, 2003). Indeed,

$$\alpha_{\theta,s}(x_s) = \sum_{x_{s-1} \in \Omega} \alpha_{\theta,s-1}(x_{s-1}) \lambda(x_s; f_\theta(x_{s-1}, y_{s-1})) \mu(y_s; g_\theta(x_{s-1:s}, y_{s-1})), \tag{5.12}$$

$$\beta_{\theta,s-1}(x_{s-1}) = \sum_{x_s \in \Omega} \beta_{\theta,s}(x_s) \lambda(x_s; f_\theta(x_{s-1}, y_{s-1})) \mu(y_s; g_\theta(x_{s-1:s}, y_{s-1})); \tag{5.13}$$

finally

$$p_\theta(y_{0:t}) = \sum_{x_t \in \Omega} \alpha_{\theta,t}(x_t), \tag{5.14}$$

$$p_\theta(x_{s-1:s}|y_{0:t}) \propto \alpha_{\theta,s-1}(x_{s-1}) \beta_{\theta,s}(x_s) \lambda(x_s; f_\theta(x_{s-1}, y_{s-1})) \mu(y_s; g_\theta(x_{s-1:s}, y_{s-1})), \tag{5.15}$$

$$p_\theta(x_s|y_{0:t}) = \sum_{x_{s-1} \in \Omega} p_\theta(x_{s-1:s}|y_{0:t}). \tag{5.16}$$

For a PMC (5.8), the estimation of θ (from a gradient ascent method) and the computation of the posterior distributions are recalled in Algs. 5.1 and 5.2, respectively.

Algorithm 5.1 Estimation of θ in general PMC models.

Require: Observations $y_{0:t}$, a learning rate ϵ, an initial parameter $\theta^{(0)}$

$j = 0$
while convergence of $\log p_{\theta^{(j)}}(y_{0:t})$ in (5.14) is not attained **do**
  Compute $\log \alpha_{\theta^{(j)},s}(x_s)$ and $\nabla_\theta \log \alpha_{\theta^{(j)},s}(x_s)\Big|_{\theta=\theta^{(j)}}$, for $x_s \in \Omega$, for $s \in [0:s]$, with (5.12)
  Compute $\log p_{\theta^{(j)}}(y_{0:t})$ and $\nabla_\theta \log p_\theta(y_{0:t})\Big|_{\theta=\theta^{(j)}}$ with (5.14)
  Set $\theta^{(j+1)} = \theta^{(j)} + \epsilon \nabla_\theta \log p_\theta(y_{0:t})\Big|_{\theta=\theta^{(j)}}$
$j \leftarrow j + 1$
end while
return $\theta^* = \theta^{(j)}$

Algorithm 5.2 Estimation of X_s in general PMC models.

Require: A realization $y_{0:t}$, a given parameter θ

Compute $\alpha_{\theta,s}(x_s)$, for $x_s \in \Omega$, for $s \in [0:t]$, with (5.12)
Compute $\beta_{\theta,s}(x_s)$, for $x_s \in \Omega$, for $s \in [0:t]$, with (5.13)
Compute $p_\theta(x_{k-1:k}|y_{0:t})$, for $x_{s-1:s} \in \Omega \times \Omega$, for $s \in [0:t]$, with (5.15)
Compute $\hat{x}_s = \arg\max_{x_s} p_\theta(x_s|y_{0:t})$, for $s \in [0:t]$, with (5.16)
return $\hat{x}_{0:t}$, the estimated hidden process

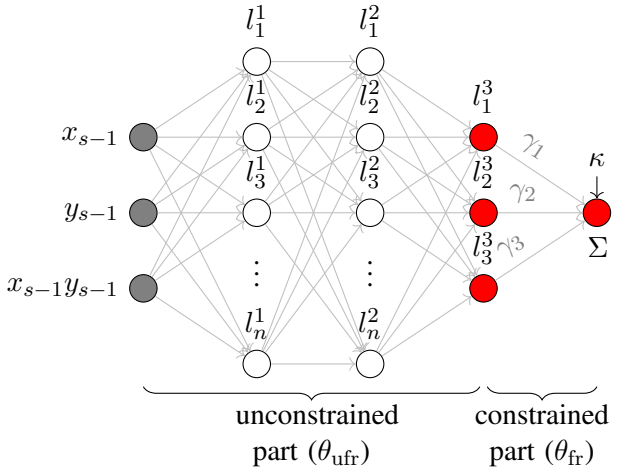
Pretraining for deep PMCs (Fig. 5.2, Alg. 5.3) - As we did for our generative models of the previous section, our objective is to parameterize f_θ and g_θ by neural networks. However, running directly Alg. 5.1 for estimating the parameters of the resulting PMC may give poor results in practice because the estimated sequence $\hat{x}_{0:t}$ does not necessarily have the desired interpretation obtained with a simple model. For these models, we propose a particular pre-training step which aims at keeping the interpretability of a simple model such as those of Exs. 5.2-5.3 This pretraining step consists of two steps illustrated in Fig. 5.2 and Alg. 5.3 and aims at initializing the neural architectures in such a way that our initial highly parameterized model coincides with the simple model:

1. we estimate the parameters of a basic model such as a linear PMC of Ex. 5.3 with Alg. 5.1. The estimated linear functions can be seen as the output of a neural network with no hidden layer and their associated parameters θ_{fr} are now frozen in the sense that we will not further tune them. Using Alg. 5.2, this step also returns a pre-classification $\hat{x}^{\text{pre}}_{0:t}$

2. we next consider the linear layers as the output of general DNNs where the other parameters are denoted as θ_{ufr}. θ_{ufr} is initialized in such a way that the DNNs produce the same output as the previous linear functions. To that end, we introduce cost functions $\mathcal{C}_{f_\theta}$ and $\mathcal{C}_{g_\theta}$. $\mathcal{C}_{f_\theta}$ is typically the averaged overtime cross-entropy between the output of f_θ and $\hat{x}^{\text{pre}}_s$, for all $s \in [0:t]$, while $\mathcal{C}_{g_\theta}$ is the mean square error between the output of g_θ and the output of the linear model where $\hat{x}^{\text{pre}}_{s-1:s}$ is used as input, for all $s \in [1:t]$. These cost functions can be optimized with backpropagation algorithms. Finally, after this pretraining step, the unfrozen parameters are fine-tuned with Alg. 5.1.

This pretraining step can be interpreted as a reverse approach w.r.t. those proposed at the beginning of 2010s to help supervised learning in deep neural networks (Erhan et al., 2010). In such architectures, Mohamed et al. (2012); Glorot and Bengio (2010); Hinton et al. (2012) have suggested to first pretrain in

an unsupervised way a deep neural network from a generative probabilistic model which shares common parameters with the original architecture (*e.g.* a Deep Belief Network). The backpropagation algorithm for supervised estimation is next initialized with the (approximated) maximum likelihood estimator of this probabilistic model. Here, we have started to pretrain our architecture in a supervised way thanks to a pre-classification and next embedded it in our original probabilistic model in which we compute an approximation of the maximum likelihood estimator.



$\Sigma = f_\theta(x_{s-1}, y_{s-1}, x_{s-1}y_{s-1}) = \text{sigm}(\gamma_1 l_1^3 + \gamma_2 l_2^3 + \gamma_3 l_3^3 + \kappa)$ where the last layer parameters $\{\gamma_1, \gamma_2, \gamma_3, \kappa\}$ are frozen to $\gamma_1 = b_{\omega_2} - b_{\omega_1}$, $\gamma_2 = a_{\omega_2} - a_{\omega_1}$, $\gamma_3 = a_{\omega_1}$ and $\kappa = b_{\omega_1}$

Figure 5.2: DNN architecture with constrained output layer for f_θ with two hidden layers. The parameters θ_{fr} are related to the output layer which computes the function f_θ of the linear PMC model (5.10). Due to the one-hot encoding of the discrete random variable x_{s-1} ($x_{s-1} = \omega_1 \leftrightarrow x_{s-1} = 0$ and $x_{s-1} = \omega_2 \leftrightarrow x_{s-1} = 1$), this parameterization is equivalent to that of (5.10) up to the given correspondence between $\theta_{\text{fr}} = (\gamma_1, \gamma_2, \gamma_3, \kappa)$ and $(a_{\omega_1}, a_{\omega_2}, b_{\omega_1}, b_{\omega_2})$. Linear activation functions are used in the last hidden layer in red.

Algorithm 5.3 A general estimation algorithm for deep parameterization of PMC models.

Require: Observations $y_{0:t}$

/* Linear model: initialization of the ouput layer of f_θ and g_θ

1: Initialize randomly $\theta_{\text{fr}}^{(0)}$
2: Estimate θ_{fr}^* using Alg. 5.1 with $\theta_{\text{fr}}^{(0)}$
3: Estimate $\hat{x}_{0:t}^{\text{pre}}$ using Alg. 5.2 with θ_{fr}^*

/* Pretraining of θ_{ufr} */

4: $\theta_{\text{ufr}}^{(0)} \leftarrow \text{Backprop}(\hat{x}_{0:t}^{\text{pre}}, y_{0:t}, \theta_{\text{fr}}^*, \mathcal{C}_{f_\theta}, \mathcal{C}_{g_\theta})$

/* Complete deep model: fine-tuning */

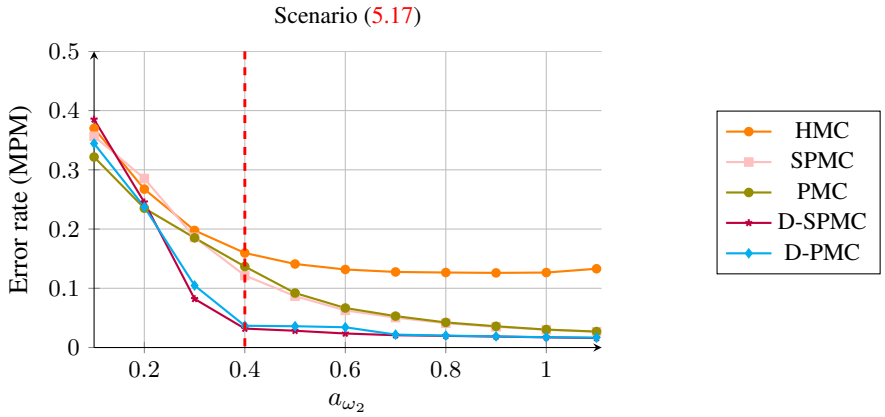
5: Compute θ_{ufr}^* using Alg. 5.1 with $(\theta_{\text{fr}}^*, \theta_{\text{ufr}}^{(0)})$ (θ_{fr}^* is not updated)
6: Compute $\hat{x}_{0:t}$ using Alg. 5.2 with $(\theta_{\text{fr}}^*, \theta_{\text{ufr}}^*)$
7: **return** $\hat{x}_{0:t}$, the final classification

PMCs versus Deep PMCs (Fig. 5.3) - We illustrate the gain obtained with these models by considering the binary image segmentation problem. We consider the cattle-type images of the Binary Shape Database (http://vision.lems.brown.edu/content/available-software-and-databases). These

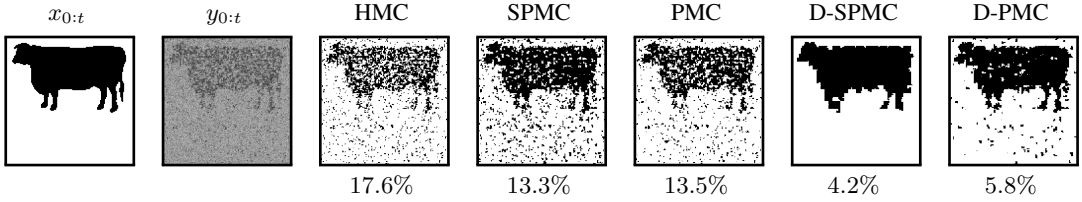
images are transformed into a 1-D signal $y_{0:t}$ with a Hilbert-Peano filling curve (Sagan, 2012). We next generate an artificial noise according to

$$Y_s|x_s, y_{s-1} \sim \mathcal{N}\Big(\sin(a_{x_s} + y_{s-1}); \sigma^2\Big), \tag{5.17}$$

where $a_{\omega_1} = 0$, $\sigma^2 = 0.25$ and a_{ω_2} is a varying parameter. We consider several models: the HMC with Gaussian noise (ex. 5.9), two PMCs (SPMC and PMC) based on the linear parameterizations of Ex. 5.3 and two Deep PMCs (D-SPMC and D-PMC) based on deep neural networks with one hidden layer, 100 neurons and the ReLU activation function. λ and μ coincides with the Bernoulli and Gaussian distributions, respectively. In the SPMC and D-SPMC, g_θ does not depend on h_{s-1}. The results are displayed in Fig. 5.3 where it can be observed that the deep versions of PMC models outperform their non-deep counterpart.



(a) Error rate from the unsupervised segmentations with a noise described by (5.17). Results are averaged on all the *cattle*-type images from the database.



(b) Selected classifications for $a_{\omega_2} = 0.4$ (signaled by the red vertical line in Fig. 5.3a). Error rates appear below the images.

Figure 5.3: Unsupervised image segmentation with PMC models.

Variational inference for interpretable TMCs - We now have at our disposal a robust algorithm to estimate highly parameterized PMCs. Through an additional parameterization, we want to complexify the nature of the joint distribution associated to $\{X_t, Y_t\}_{t\in\mathbb{N}}$. To that end, we add an additional latent process $\{Z_t\}_{t\in\mathbb{N}}$ such that $\{Z_t, X_t, Y_t\}_{t\in\mathbb{N}}$ is described by a TMC; the joint distribution of interest is now a marginal of the TMC,

$$p_\theta(x_{0:t}, y_{0:t}) = \int p_\theta(\mathrm{d}z_{0:t}, x_{0:t}, y_{0:t}), \quad \text{for all } t \in \mathbb{N}.$$

However, we have to take into account that $\{Z_t\}_{t\in\mathbb{N}}$ does not need to be interpretable contrary to $\{X_t\}_{t\in\mathbb{N}}$. Moreover, in the case where Z_t is continuous, Algs. 5.1 and 5.2 cannot be computed contrary to the case where Z_t is also discrete (Gorynin et al., 2018) (it suffices to consider the previous algorithms in augmented

dimension). Consequently, we resort to variational Bayesian inference but we modify the objective function in order to strengthen the interpretability of $\{X_t\}_{t\in\mathbb{N}}$.

Let $\{Z_t, X_t, Y_t\}_{t\in\mathbb{N}}$ be a TMC satisfying the parameterization (5.6). The ELBO (5.4) becomes

$$F(\theta,\phi) = -\int \log\left(\frac{q_\phi(z_{0:t}, x_{0:t}|y_{0:t})}{p_\theta(z_{0:t}, x_{0:t}, y_{0:t})}\right) q_\phi(\mathrm{d}x_{0:t}|y_{0:t}) \tag{5.18}$$

and we have the following Proposition.

Proposition 5.1. Let us denote $F^{\mathrm{opt}}(\theta,\phi)$ the ELBO resulting of (5.18) with

$$q_\phi^{\mathrm{opt}}(z_{0:t}, x_{0:t}|y_{0:t}) = q_\phi(z_{0:t}|y_{0:t})p_\theta(x_{0:t}|y_{0:t}, z_{0:t}).$$

Then for any (θ,ϕ),

$$\log p_\theta(y_{0:t}) \geq F^{\mathrm{opt}}(\theta,\phi) \geq F(\theta,\phi).$$

Since the conditional posterior

$$p_\theta(x_{0:t}|y_{0:t}, z_{0:t}) = p_\theta(x_t|y_{0:t}, z_{0:t}) \prod_{s=1}^{T} p_\theta(x_{s-1}|x_s, y_{0:t}, z_{0:t})$$

is computable in the discrete case through a direct extension of the definitions of α_s and β_s in (5.12)-(5.13) (even if Z_t is continuous), this Proposition illustrates that we only need to choose a variational distribution $q_\phi(z_{0:t}|y_{0:t})$.

We now modify the ELBO $F^{\mathrm{opt}}(\theta,\phi)$ in order to enforce the interpretability of $\{X_t\}_{t\in\mathbb{N}}$. The following Corollary provides an alternative expression of $F^{\mathrm{opt}}(\theta,\phi)$

Corollary 5.1. Let us factorize $p_\theta(z_{0:t}, x_{0:t}, y_{0:t}) = \overline{p}_\theta(z_{0:t}, x_{0:t}|y_{0:t})\tilde{p}_\theta(y_{0:t}|x_{0:t}, z_{0:t})$ with

$$\begin{aligned}
\tilde{p}_\theta(y_{0:t}|x_{0:t}, z_{0:t}) &= p_\theta(y_0|x_0, z_0) \prod_{s=1}^{t} \mu\left(y_s; g_\theta(z_{s-1:s}, x_{s-1:s}, y_{s-1})\right), \\
\overline{p}_\theta(z_{0:t}, x_{0:t}|y_{0:t}) &= p_\theta(z_0, x_0) \prod_{s=1}^{t} \varsigma\left(z_s; s_\theta(z_{s-1}, h_{s-1}, y_{s-1})\right) \lambda\left(h_s; f_\theta(z_{s-1:s}, x_{s-1}, y_{s-1})\right).
\end{aligned} \tag{5.19}$$

Then

$$F^{\mathrm{opt}}(\theta,\phi) = \mathcal{L}_1(\theta,\phi) + \mathcal{L}_2(\theta,\phi), \tag{5.20}$$

where

$$\begin{aligned}
\mathcal{L}_1(\theta,\phi) &= \mathbb{E}_{q_\phi^{\mathrm{opt}}(z_{0:t}, x_{0:t}|y_{0:t})}\left(\log \tilde{p}_\theta(y_{0:t}|x_{0:t}, z_{0:t})\right), \\
\mathcal{L}_2(\theta,\phi) &= -\mathrm{D}_{\mathrm{KL}}\left(q_\phi^{\mathrm{opt}}(z_{0:t}, x_{0:t}|y_{0:t})||\overline{p}_\theta(z_{0:t}, x_{0:t}|y_{0:t})\right).
\end{aligned} \tag{5.21}$$

This result enables to adapt the concept of β-ELBO of Higgins et al. (2017); the idea is to penalize $\mathcal{L}_2(\theta,\phi)$ through the introduction of a coefficient β_1. Regarding $\mathcal{L}_1(\theta,\phi)$, its maximization guides the model to interpret the latent process $\{Z_t, X_t\}_{t\in\mathbb{N}}$ as that which explains the best the observations given the past. On the other hand, the maximization of $\mathcal{L}_2(\theta,\phi)$ tends to push the conditional variational distribution $q_\phi^{\mathrm{opt}}(z_s, x_s|z_{0:s-1}, x_{0:s-1}, y_{0:t})$ at each time step to be close to the conditional prior distribution $p_\theta(z_s, x_s|z_{s-1}, x_{s-1}, y_{s-1})$. The interest of this term is to boost the posterior distribution q_ϕ^{opt} to take into account the (conditional) prior terms at each time step and so aims at limiting the impact of the observations

on the interpretability of the hidden process, particularly in problems where the observations are a very noisy version of $\{X_t\}_{t\in\mathbb{N}}$.

Finally, we complete our objective function in order to guide the estimation process to distinguish the role of $\{X_t\}_{t\in\mathbb{N}}$ to that of $\{Z_t\}_{t\in\mathbb{N}}$. We assume that we have a pre-classification $\hat{x}_{0:t}^{\text{pre}}$ and we introduce the KLD between the empirical distribution deduced from this pre-classification, $p^{\text{emp}}(x_{0:t}) = \delta_{\hat{x}_{0:t}^{\text{pre}}}(x_{0:t})$, and the marginal variational distribution $q_\phi(x_{0:t}|y_{0:t}) = \int q_\phi^{\text{opt}}(\mathrm{d}z_{0:t}, x_{0:t}|y_{0:t})$ which aims itself at approximating the true posterior distribution $p_\theta(x_{0:t}|y_{0:t})$. Thus, the objective is to push the variational distribution q_ϕ to take into account the labels obtained from an already interpretable pre-classification through the negative cross-entropy

$$\mathcal{L}_3(\theta,\phi) = \mathbb{E}_{p^{\text{emp}}(x_{0:t})}\left(\log q_\phi(x_{0:t}|y_{0:t})\right) = \log q_\phi(\hat{x}_{0:t}^{\text{pre}}|y_{0:t}), \tag{5.22}$$

see for example (Kingma et al., 2014; Klys et al., 2018; Kumar et al., 2021). This additional term is next penalized by a scalar β_2 which controls the proximity of the pre-classification with the variational posterior distribution.

The final objective function reads

$$\mathcal{L}(\theta,\phi) = \mathcal{L}_1(\theta,\phi) + \beta_1\mathcal{L}_2(\theta,\phi) + \beta_2\mathcal{L}_3(\theta,\phi) \tag{5.23}$$

and can be approximated with the same Monte Carlo technique as that described in Section 5.2, and based on the reparametrization trick,

$$\widehat{\mathcal{L}}(\theta,\phi) = \widehat{\mathcal{L}}_1(\theta,\phi) + \widehat{\mathcal{L}}_2(\theta,\phi) + \widehat{\mathcal{L}}_3(\theta,\phi),$$

where

$$\widehat{\mathcal{L}}_1(\theta,\phi) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{p_\theta(x_{0:t}|z_{0:t}^i,y_{0:t})}\left(\log\tilde{p}_\theta(y_{0:t}|z_{0:t}^i,x_{0:t})\right),$$

$$\widehat{\mathcal{L}}_2(\theta,\phi) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{p_\theta(x_{0:t}|z_{0:t}^i,y_{0:t})}\left(\log\left(\frac{\overline{p}_\theta(z_{0:t}^i,x_{0:t}|y_{0:t})}{p_\theta(x_{0:t}|z_{0:t}^i,y_{0:t})q_\phi(z_{0:t}^i|y_{0:t})}\right)\right),$$

$$\widehat{\mathcal{L}}_3(\theta,\phi) = \log\left(\frac{1}{N}\sum_{i=1}^{N}p_\theta(\hat{x}_{0:t}^{\text{pre}}|z_{0:t}^i,y_{0:t})\prod_{s=1}^{t}p_\theta(\hat{x}_{s-1}^{\text{pre}}|z_{0:t}^i,\hat{x}_s^{\text{pre}},y_{0:t})\right),$$

and where $\{z_{0:t}^i\}_{i=1}^N$ are i.i.d. and differentiable samples from $q_\phi(z_{0:t}|y_{0:t})$. Once the model has been estimated, it remains to estimate X_s for $s\in[0:t]$. Since

$$p_\theta(x_s|y_{0:t}) = \mathbb{E}_{p_\theta(z_{0:t}|y_{0:t})}\left(p_\theta(x_s|z_{0:t},y_{0:t})\right),$$

where $p_\theta(z_{0:t}|y_{0:t})$ is known up to a constant and $p_\theta(x_s|z_{0:t},y_{0:t})$ is computable, $p_\theta(x_s|y_{0:t})$ can be approximated with any particle filter/smoothing algorithm of the previous chapters or directly from the corresponding variational distribution.

Deep TMCs (Fig. 5.4 and Alg. 5.4) - As for PMCs, we include a pretraining step when the TMCs are parameterized by DNNs. The main difference with the PMC is that the input of such architectures can also depend on the unobserved latent process $\{Z_t\}_{t\in\mathbb{N}}$ and that the parameters of the conditional posterior distribution $q_\phi(z_s|z_{0:s-1},y_{0:t})$ can also be represented by a DNN.

1. We first start by estimating the parameters of a linear TMC from the modified variational framework developed above. Since $\{Z_t\}_{t\in\mathbb{N}}$ does not need to be interpretable, $q_\phi(z_s|z_{0:s-1}, y_{0:t})$ is already parameterized by a DNN. The DNNs s_θ, f_θ and g_θ in (5.6) are next built in the same way as those in Fig. 5.2;

2. we next mimic the pretraining of PMCs based on $\hat{x}^{\text{pre}}_{0:t}$ but we take into account that $Z_{0:t}$ is not observed. Since it encodes the observations, we sample $z_{0:t} \sim q_\phi(z_{0:t}|y_{0:t})$, where q_ϕ is the estimated variational distribution estimated from the previous step, and we use the components $z_{s-1:s}$ as inputs of the neural networks s_θ, f_θ and g_θ, for $s \in [1:t]$. Finally, from the backpropagation algorithm, the neural networks are pretrained through adapted cost functions to reproduce the results of the linear TMC.

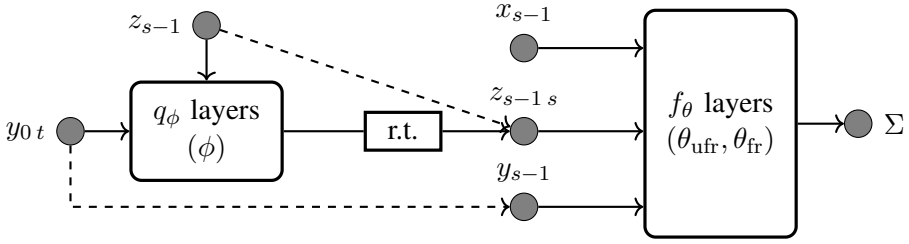


Figure 5.4: Graphical and condensed representation of the parameterization of f_θ in Deep TMCs. *r.t.* stands for reparameterization trick. The dashed arrows represent the fact that some variables are copied. For clarity, we do not represent the block f_θ which is similar to Fig. 5.2, up to the introduction of $z_{s-1:s}$.

Algorithm 5.4 A general estimation algorithm for deep parameterizations of TMC models

Require: Observations $y_{0:t}$, a parameterized variational distribution q_ϕ, β_1, β_2

/* Initialization of the output layer of s_θ, f_θ and g_θ */

Estimate $(\theta^*_{\text{fr}}, \tilde{\phi})$ and $\hat{x}^{\text{pre}}_{0:t}$ related to a non deep TMC with modified variational inference

/* Pretraining of θ_{ufr} */

$\theta^{(0)}_{\text{fr}} \leftarrow \text{Backprop}(\hat{x}^{\text{pre}}_{0:t}, y_{0:t}, \theta^*_{\text{fr}}, \tilde{\phi}, \mathcal{C}_{s_\theta}, \mathcal{C}_{f_\theta}, \mathcal{C}_{g_\theta})$

/* Fine-tuning of the complete model */

Compute $(\theta^*_{\text{ufr}}, \phi^*)$ with modified variational inference

Compute $\hat{x}_{0:t}$ a particle smoother based on q_{ϕ^*} as importance distribution

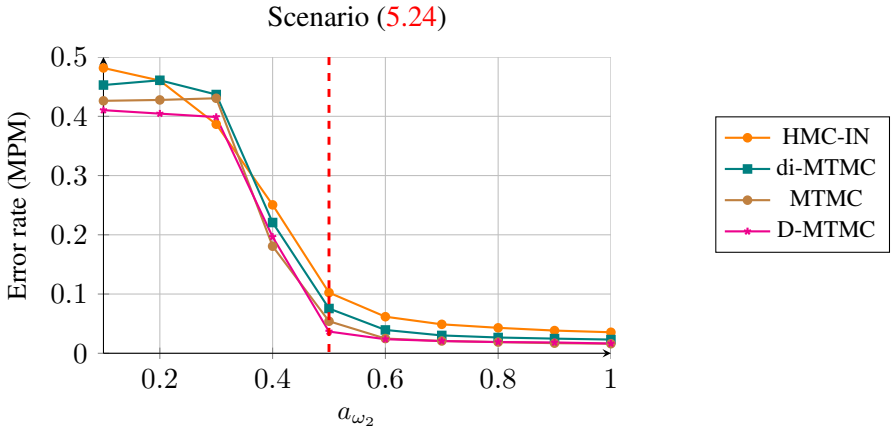
return A classification $\hat{x}_{0:t}$

Some Deep TMC architectures (Figs. 5.5-5.6) - We consider two applications of the third latent process.

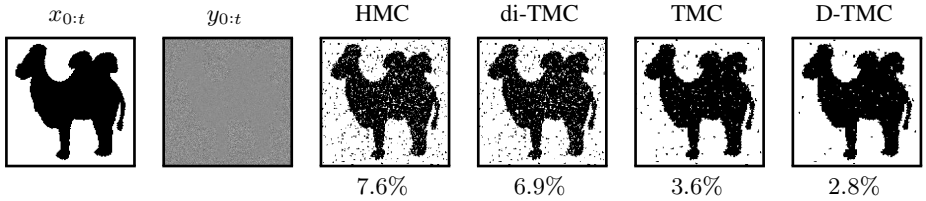
In the first one, the role of $\{Z_t\}_{t\in\mathbb{N}}$ is to model the unknown distribution of the noise $p_\theta(y_t|x_t)$ in an HMC. So we consider a TMC satisfying

$$p_\theta(z_{0:t}, x_{0:t}, y_{0:t}) = \underbrace{\prod_{s=0}^{t} \varsigma(z_s; s_\theta)}_{p_\theta(z_{0:t})} \underbrace{p_\theta(x_0) \prod_{s=1}^{t} \lambda(x_s; f_\theta(x_{s-1}))}_{p_\theta(x_{0:t}|z_{0:t})=p_\theta(x_{0:t})} \underbrace{\prod_{s=0}^{t} \mu(x_s; g_\theta(z_s, x_s))}_{p_\theta(y_{0:t}|z_{0:t},x_{0:t})} .$$

We consider two versions of this model. In both versions, Z_s is a univariate centered Gaussian distribution (so ς is the Gaussian distribution and $s_\theta = [0; 1]$). The linear version (TMC) coincides with (5.9), while



(a) Error rate from the unsupervised segmentations of Scenario (5.24). Results are averaged on all the *camel*-type images from the database.



(b) Selected illustrations for $a_{\omega_2} = 0.5$ (signaled by the red vertical line on Fig. 5.5a). Error rates appear below the images.

Figure 5.5: Unsupervised image segmentation with General Triplet Markov Chains (Scenario (5.24)).

in the deep one (D-TMC), f_θ and g_θ are neural networks with one hidden layer of 100 neurons. For both models, we use the variational distribution

$$q_\phi(z_{0:t}|y_{0:t}) = q_\phi(z_0|y_0)\prod_{s=1}^{t} q_\phi(z_s|z_{s-1}, y_s) = \prod_{s=1}^{t}\mathcal{N}(z_s; \nu_\phi(z_{s-1}, y_s)),$$

where $\nu_\phi(z_{s-1}, y_s)$ is a neural network with one hidden layer of 100 neurons and a ReLU activation function. The models are tested on the camel-type images of the Binary Shape Database and are corrupted with a stationary multiplicative noise,

$$Y_s|z_s, x_s \sim \mathcal{N}\left(a_{h_k}; b^2_{h_k}\right) z_s, \quad \text{for } s \in [0:t], \tag{5.24}$$

where $z_s \sim \mathcal{N}(0,1)$, $a_{\omega_1} = 0, a_{\omega_2}$ is a varying parameter and $b_{\omega_1} = b_{\omega_2} = 0.2$. Our models are compared with the classical HMC and a TMC with a discrete latent process (di-TMC). The results are displayed on Fig. 5.5 for $\beta_1 = 5$ and $\beta_2 = 1$.

A second application of the third process consists in transforming the PMC into a model where the pair (X_t, Y_t) depends on $Y_{0:t-1}$ in an explicit way but in which Algs. 5.1 and 5.2 remain valid. To that end, it suffices that the third latent process $\{Z_t\}_{t\in\mathbb{N}}$ becomes deterministic given the past (*i.e.* ς coincides with the Dirac measure δ) and satisfies

$$z_s = s_\theta(z_{s-1}, y_{s-1}).$$

This model is nothing more than a particular deep variational PMC of section 5.2 but with the objective of estimating a physical hidden process. In this particular case, the pair $\{X_t, Y_t\}_{t\in\mathbb{N}}$ is a partially PMC since

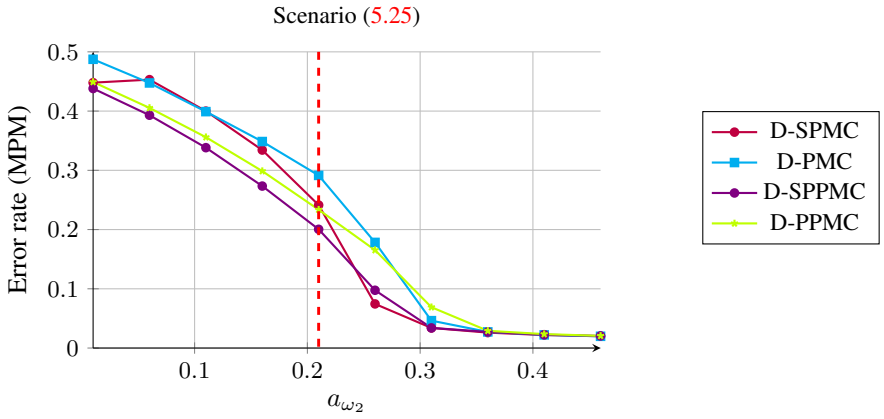
its distribution reads

$$p_\theta(x_{0:t}, y_{0:t}) = p_\theta(x_0, y_0) \prod_{s=1}^{t} \underbrace{\lambda(x_s; f_\theta(z_{s-1:s}, x_{s-1}, y_{s-1}))}_{p_\theta(x_s|x_{s-1}, y_{0:s-1})} \underbrace{\mu(y_s; g_\theta(z_{s-1:s}, x_{s-1:s}, y_{s-1}))}_{p_\theta(y_s|x_{s-1:s}, y_{0:s-1})}, \quad \text{for all } t \in \mathbb{N}.$$

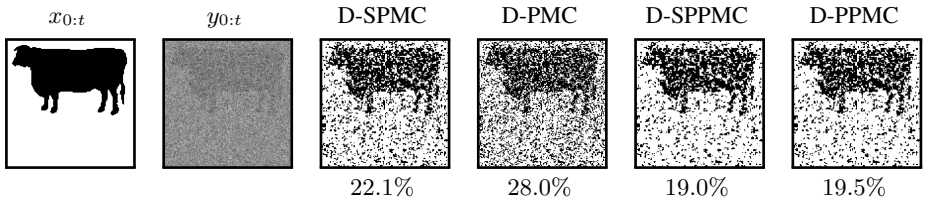
This model can also be interpreted as a particular TMC in which the optimal importance variational distribution $q_\phi^{\text{opt}}(z_s|z_{0:s-1}, y_{0:t}) = \delta_{s_\theta(z_{s-1}, y_{s-1})}(z_s)$ is now computable. Consequently, (5.23) can be exactly computed. In particular, when $\beta_1 = 1$ and $\beta_2 = 0$, the objective function coincides with the log-likelihood. In order to evaluate the interest of such a latent process w.r.t. PMCs, we compare the counterpart versions of the D-SPMC and the D-PMC of previous paragraph; our models are denoted D-SPPMC and D-PPMC. We go on with the cattle-type images of the Binary Shape Database. They are corrupted by a noise which satisfies

$$Y_s|x_s, Y_{s-2:s-1} \sim \mathcal{N}\Big(\sin(a_{x_s} + 0.2(y_{s-1} + y_{s-2})); \sigma^2\Big). \tag{5.25}$$

where $a_{\omega_1} = 0$, $\sigma^2 = 0.25$ and a_{ω_2} is a varying parameter. The results are displayed on Fig. 5.6, where z_s is a 10-dimensional vector.



(a) Error rate from the unsupervised segmentations of Scenario (5.25). Results are averaged on all the *cattle*-type images from the database.



(b) Selected illustrations for $a_{\omega_2} = 0.21$ (signaled by the red vertical line on Figure 5.6a). Error rates appear below the images.

Figure 5.6: Unsupervised image segmentation with Partially Pairwise Markov Chains.

5.4 Variational Inference in linear and Gaussian TMC

We finally consider another application of our parameterized TMC models. Our goal is to provide a fast Bayesian inference algorithms for the linear and Gaussian JMSS and its generalizations. This model is a particular TMC in which $\{X_t\}_{t\in\mathbb{N}}$ is continuous and $\{Z_t\}_{t\in\mathbb{N}}$ is discrete. Our approximation consists of two

steps. First, we show that there exists a particular parameterization (5.5) in which it is possible to compute sequentially the first and second order moments of the posterior distribution in a linear cost in the number of observations; next, we show that the ELBO associated to our (variational) TMC and the original JMSS can be computed exactly.

General linear and Gaussian Jump Markov state-space models - Let $\{Z_t\}_{t\in\mathbb{N}}$ be a discrete Markov Chain, $Z_t \in \Omega = \{\omega_1, \cdots, \omega_K\}$ and

$$\begin{aligned} p_\theta(z_{0:t}, x_{0:t}, y_{0:t}) &= p_\theta(z_0, x_0, y_0) \prod_{s=1}^{t} p_\theta(z_s|z_{s-1}) p_\theta(x_s, y_s|z_s, x_{s-1}, y_{0:s-1}), \\ p_\theta(z_t = \omega_j | z_{t-1} = \omega_i) &= \mu_{i,j}, \\ p_\theta(x_t, y_t|z_t, x_{t-1}, y_{0:t-1}) &= \mathcal{N}\left(\begin{pmatrix} x_t \\ y_t \end{pmatrix}; B_\theta(z_t, y_{0:t-1}) x_{t-1} + g_\theta(z_t, y_{0:t-1}); \Xi_\theta(z_t, y_{0:t-1})\right), \end{aligned} \tag{5.26}$$

where B_θ, g_θ and Ξ_θ are parameterized vector valued functions with appropriate dimensions. This model is nothing more than a generalization of the popular linear and Gaussian JMSS,

$$\begin{aligned} p_\theta(x_t, y_t|z_t, x_{t-1}, y_{0:t-1}) &= p_\theta(x_t|z_t, x_{t-1}) p_\theta(y_t|z_t, x_t), \\ p_\theta(x_t|z_t, x_{t-1}) &= \mathcal{N}\left(x_t; F(z_t) x_{t-1}; Q(z_t)\right), \\ p_\theta(y_t|z_t, x_t) &= \mathcal{N}\left(y_t; H(z_t) x_t; R(z_t)\right), \end{aligned} \tag{5.27}$$

where

$$\theta = \left(\mu_{i,j}, F(\omega_j), Q(\omega_j), H(\omega_j), R(\omega_j)/(\omega_i, \omega_j) \in \Omega \times \Omega\right).$$

The introduction of a discrete latent process in such models transforms the computation of the first and second order moments of the filtering distribution into an NP-hard problem. The particle filters of the previous chapters and their Rao-Blackwellized versions (Schön et al., 2005) can be used to approximate such moments. However, we present an alternative and faster solution which does not require any Monte Carlo approximation. Our solution is based on variational Bayesian inference and also enables us to the estimate the parameters θ of the original model. For sake of clarity, we only focus on the linear and Gaussian JMSS (5.27) and we use its alternative representation given by

$$\begin{aligned} p_\theta(x_t, y_t|z_t, x_{t-1}, y_{t-1}) &= p_\theta(y_t|z_t, x_{t-1}) p_\theta(x_t|z_t, x_{t-1}, y_t), \\ p_\theta(y_t|x_{t-1}, z_t) &= \mathcal{N}(y_t; \tilde{H}(z_t) x_{t-1}; \tilde{R}(z_t)), \\ p_\theta(x_t|z_t, x_{t-1}, y_t) &= \mathcal{N}(x_t; \tilde{F}(z_t) x_{t-1} + \tilde{D}(z_t) y_t; \tilde{Q}(z_t)), \end{aligned} \tag{5.28}$$

where $\tilde{F}$, $\tilde{D}$, $\tilde{Q}$, $\tilde{H}$ and $\tilde{R}$ are deduced from (5.27) with conditioning results on Gaussian distributions.

An alternative probabilistic model with jumps - Let us now consider a distribution q_ϕ which satisfies for all $t \in \mathbb{N}$

$$\begin{aligned} q_\phi(z_{0:t}, x_{0:t}, y_{0:t}) &= q_\phi(z_0, x_0, y_0) \prod_{s=1}^{t} q_\phi(z_s, y_s|z_{s-1}, y_{0:s-1}) q_\phi(x_s|z_s, x_{s-1}, y_{0:t}), \\ q_\phi(z_t, y_t|z_{t-1}, y_{0:t-1}) &= \varsigma\left(z_t; s_\phi(z_{t-1}, y_{0:t-1})\right) \times \mu\left(y_t; g_\phi(z_{t-1:t}, y_{0:t-1})\right), \\ q_\phi(x_t|z_t, x_{t-1}, y_{0:t}) &= \mathcal{N}\left(x_t; C_\phi(z_t, y_{0:t}) x_{t-1} + h_\phi(z_t, y_{0:t}); \Sigma_\phi(z_t, y_{0:t})\right). \end{aligned} \tag{5.29}$$

Strictly, this model is not a TMC but rather a partially TMC since the parameters can depend on all the past observations. As we did for the partially PMC of Section 5.2, these dependencies can be modelled by an RNN and so a fourth deterministic latent process. In this model, it is possible to compute $\mathbb{E}(X_s|y_{0:t})$ and $\mathbb{V}ar(X_s|y_{0:t})$ at a linear cost in the number of observations. As an extension of Pieczynski (2002); Derrode and Pieczynski (2013), model (5.29) satisfies,

$$\mathbb{E}_{q_\phi}(X_s|y_{0:t}) = \sum_{z_s\in\Omega} q_\phi(z_s|y_{0:t}) \underbrace{\mathbb{E}_{q_\phi}(X_s|z_s,y_{0:s})}_{m_{\phi,s}(z_s)}, \quad \mathbb{E}_{q_\phi}(X_sX_s^T|y_{0:t}) = \sum_{z_s\in\Omega} q_\phi(z_s|y_{0:t}) \underbrace{\mathbb{E}_{q_\phi}(X_sX_s^T|z_s,y_{0:s})}_{v_{\phi,s}(z_s)}.$$

Next, observing that $\{Z_t, X_t\}_{t\in\mathbb{N}}$ is a partially PMC, the sequential computation of $q_\phi(z_{s-1:s}|y_{0:t})$ is similar to (5.15) where the products $\lambda(.)\mu(.)$ are replaced by $\varsigma(.)\mu(.)$. Finally, $m_{\phi,s}(z_s)$ and $v_{\phi,s}(z_s)$ can be computed as

$$m_{\phi,s}(z_s) = \sum_{z_{s-1}} q_\phi(z_{s-1}|z_s,y_{0:s}) \left(C_\phi(z_s,y_{0:s})m_{\phi,s-1}(z_{s-1}) + h_\phi(z_s,y_{0:s})\right), \tag{5.30}$$

$$\begin{aligned} v_{\phi,s}(z_s) = \sum_{z_{s-1}} q_\phi(z_{s-1}|z_s,y_{0:s}) \Big(&\Sigma_\phi(z_s,y_{0:s}) + C_\phi(z_s,y_{0:s})v_{\phi,s-1}(z_{s-1})C_\phi(z_s,y_{0:s})^T + \\ &h_\phi(z_s,y_{0:s})m_{\phi,s-1}(z_{s-1})^T C_\phi(z_s,y_{0:s})^T + C_\phi(z_s,y_{0:s})m_{\phi,s-1}(z_{s-1})h_\phi(z_s,y_{0:s})^T + \\ &h_\phi(z_s,y_{0:s})h_\phi(z_s,y_{0:s})^T\Big). \end{aligned} \tag{5.31}$$

Since q_ϕ has interesting computational properties, our objective is to use it to approximate the first and second order moments of the filtering distribution in the model p_θ. Even if θ is known, we first need to estimate ϕ. This can be done by minimizing the KLD between the posterior distributions of the two models, *i.e.* by maximizing the ELBO. As a bonus, we also have a new estimation method of θ in such models. Note that it is not a direct application of the variational Bayesian framework because we have not parameterized a posterior distribution $q_\phi(z_{0:t}, x_{0:t}|y_{0:t})$ (which is unknown in the case of model (5.29)) but rather a variational generative model which aims at mimic the original one.

Computing the ELBO from model (5.29) - It is possible to compute the ELBO

$$F(\theta,\phi) = \mathbb{E}_{q_\phi(z_{0:t},x_{0:t}|y_{0:t})}\left[\log\left(\frac{q_\phi(z_{0:t},x_{0:t}|y_{0:t})}{p_\theta(z_{0:t},x_{0:t},y_{0:t})}\right)\right]$$

from the two following lemmas (Petersen and Pedersen, 2008; Mathai and Provost, 1992), even if the variational distribution $q_\phi(z_{0:t}, x_{0:t}|y_{0:t})$ is unkonwn.

Lemma 5.1. Let $x \in \mathbb{R}^{d_x}$ and $p_1(x)$ and $p_2(x)$ be two Gaussian distributions, $p_1(x) = \mathcal{N}(x; m_1; P_1)$ and $p_2(x) = \mathcal{N}(x; m_2; P_2)$. Then the KLD between p_1 and p_2 reads

$$\mathrm{D}_{\mathrm{KL}}(p_1,p_2) = \mathrm{Tr}(P_2^{-1}P_1) + (m_2-m_1)^T P_2^{-1}(m_2-m_1) - d_x + \log\left(\frac{\det P_2}{\det P_1}\right).$$

Lemma 5.2. Let X be a random variable with pdf $p(x)$ such that $\mathbb{E}_p(X) = \mu$ and $\mathbb{V}ar_p(X) = P$. Then for any covariance matrix Q,

$$\mathbb{E}_p\left((HX-b)^TQ(HX-b)\right) = \mathrm{Tr}(QHPH^T) + (H\mu-b)^TQ(H\mu-b).$$

We next have the following Proposition.

Proposition 5.2. Let $q_\phi(z_{0:s}, x_{0:s}|y_{0:t})$ and $p_\theta(z_{0:s}, x_{0:s}|y_{0:s})$ be the posterior distributions associated to (5.29) and to the JMSS (5.27), respectively. We denote as d_x (resp. d_y) the dimension of x (resp. of y). Once $q_\phi(z_{s-1:s}|y_{0:t})$, $m_{\theta,s}(z_s)$ and $P_{\theta,s}(z_s) = \mathbb{V}\mathrm{ar}(X_s|z_s, y_{0:s})$ have been computed for $s \in [0:t]$ (see (5.30)-(5.31)), then the ELBO $F(\theta,\phi)$ is available for free and reads

$$F(\theta,\phi) = \sum_{z_0} q_\phi(z_0|y_{0:s})\mathcal{D}_{\theta,\phi,0}(z_0) + \sum_{s=1}^{t}\sum_{z_{s-1:s}} q_\phi(z_{s-1:s}|y_{0:s})\mathcal{D}_{\theta,\phi,s}(z_{s-1:s}),$$

$$\mathcal{D}_{\theta,\phi,0}(z_0) = \log\left(\frac{q_\phi(z_0|y_{0:s})}{p_\theta(z_0)}\right) - \log(p_\theta(y_0|z_0)) + \mathrm{D}_{\mathrm{KL}}(q_\phi(x_0|z_0,y_0), p_\theta(x_0|z_0,y_0)),$$

$$\mathcal{D}_{\theta,\phi,s}(z_{s-1:s}) = \tilde{\gamma}_{\theta,\phi,s}(z_{s-1:s}) + \tilde{\alpha}_{\theta,\phi,s}(z_{s-1:s}) + \tilde{\beta}_{\theta,\phi,s}(z_{s-1:s}),$$

where

$$\begin{aligned}
\tilde{\gamma}_{\theta,\phi,s}(z_{s-1:s}) &= \log\left(\frac{q_\phi(z_s|z_{s-1},y_{0:s})}{p_\theta(z_s|z_{s-1})}\right),\\
\tilde{\alpha}_{\theta,\phi,t}(z_{s-1:s}) &= \frac{1}{2}\Big[G_{\theta,\phi,s}(z_s) + \mathrm{Tr}\left(\tilde{Q}(z_s)^{-1}A_{\theta,\phi,s}(z_s)P_{\phi,s-1}(z_{s-1})A_{\theta,\phi,t}(z_s)^T\right) +\\
&\left(A_{\theta,\phi,s}(z_s)m_{\phi,s-1}(z_{s-1}) + D_{\theta,\phi,t}(z_s)\right)^T \times \tilde{Q}(z_s)^{-1} \times \left(A_{\theta,\phi,t}(z_s)m_{\phi,s-1}(z_{s-1}) + D_{\theta,\phi,s}(z_s)\right)\Big],\\
\tilde{\beta}_{\theta,\phi,s}(z_{s-1:s}) &= \frac{1}{2}\Big[\log(\det(\tilde{R}(z_s))) + \mathrm{Tr}\left(\tilde{R}(z_s)^{-1}\tilde{H}(z_s)P_{\phi,t-1}(z_{s-1})\tilde{H}(z_s)^T\right) +\\
&\left(\tilde{H}(z_s)m_{\phi,s-1}(z_{s-1}) - y_s\right)^T \times \tilde{R}(z_s)^{-1} \times \left(\tilde{H}(z_s)m_{\phi,s-1}(z_{s-1}) - y_s\right) + d_y\log(2\pi)\Big],
\end{aligned}$$

where

$$\begin{aligned}
G_{\theta,\phi,s}(z_s) &= \mathrm{Tr}(\tilde{Q}(z_s)^{-1}\Sigma_\phi(z_s)) + \log\left(\frac{\det(\tilde{Q}(z_s))}{\det(\Sigma_\phi(z_s))}\right) - d_x,\\
A_{\theta,\phi,t}(z_s) &= \tilde{F}(z_s) - C_\phi(z_s, y_{0:s}),\\
D_{\theta,\phi,s}(z_s) &= \tilde{D}(z_s)y_s - h_\phi(y_{0:s}, z_s),
\end{aligned}$$

and where $\tilde{F}(z_s)$, $\tilde{D}(z_s)$, $\tilde{Q}(z_s)$, $\tilde{H}(z_s)$ and $\tilde{R}(z_s)$ define the alternate representation (5.28) of p_θ.

The ELBO is next optimized from an ascent gradient approach where the all the quantities that depend on (θ,ϕ) can be differentiated sequentially. In our experiments, we have used the Adam optimizer (Kingma and Ba, 2015) and computed the gradients by auto-differentiation (Paszke et al., 2019).

Experiments (Fig. 5.7) - Let us now evaluate the relevance of model q_ϕ to perform Bayesian inference in model p_θ. We use the following parameterization of q_ϕ,

$$\begin{aligned}
\varsigma\left(z_t = \omega_j; s_\phi(z_{t-1} = \omega_i, y_{0:t-1})\right) &= \frac{\exp(\lambda_{i,j})}{\sum_{j=1}^K \exp(\lambda_{i,j})},\\
\mu\left(y_t; g_\phi(z_{t-1} = \omega_i, z_t = \omega_j, y_{0:t-1})\right) &= \mathcal{N}(y_t; B(\omega_j)y_{t-1}; P(\omega_j)),\\
C_\phi(\omega_j, y_{0:t}) &= C(\omega_j),\\
h_\phi(\omega_j, y_{0:t}) &= D(\omega_j)y_t + D'(\omega_j)y_{t-1},\\
\Sigma_\phi(\omega_j, y_{0:t}) &= \Sigma(\omega_j),
\end{aligned}$$

which shares common properties with the linear and Gaussian JMSS. We proceed as follows. First, we generate a sequence $y_{0:t}$ from a known JMSS (5.28) of length $t = 300$; we next estimate the parameters of q_ϕ and possibly those of p_θ when they are assumed to be unknown. Once the parameters have been estimated, we generate 200 new sequences $\{x_{0:t',p}, y_{0:t',p}\}_{p=1}^{200}$ of length $t' = 100$ and we compute, for all s, the Mean Square Errors (MSE) $\frac{1}{200}\sum_{p=1}^{200}[(\hat{x}_{s,p} - x_{s,p})^T(\hat{x}_{s,p} - x_{s,p})]$, where $\hat{x}_{s,p}$ represents a Rao-Blackwellized particle filter estimator of $x_{t,p}$ with $N = 50$ particles (Fearnhead and Clifford, 2003) or our exact estimator $\mathbb{E}_{q_{\hat{\phi}}}(X_{s,p}|y_{s,p})$ in the estimated variational model $q_{\hat{\phi}}$.

We first consider a scalar model where $Z_t \in \{1, 2\}$ The true parameters of model (5.28) are set to $F(z_t) \in \{-0.95, 0.95\}$, $Q(z_t) = 10$, $H(z_t) = 1$, $R(z_t) = 1$, $p(z_t|z_{t-1}) = 0.8$ if $z_t = z_{t-1}$ and $p(z_t|z_{t-1}) = 0.2$ otherwise. We estimate these parameters, except $H(r_t)$ which is assumed to be known. The KLD converges after 500 iterations, with a learning rate of 10^{-2}. The parameters estimated for p_θ are $\hat{F}(r_t) \in \{-0.9488, 0.9469\}$, $\hat{Q}(r_t) \in \{11.9170, 11.9602\}$, $\hat{R}(r_t) = 1.0637$; for the transition matrix we have $\hat{p}(r_t = 1|r_{t-1} = 1) = 0.8252$, $\hat{p}(r_t = 2|r_{t-1} = 2) = 0.7967$. Fig. 5.7 displays the MSE on new trajectories estimated with the particle filter in true model p_θ, the particle filter in estimated model $p_{\hat{\theta}}$ and our fast estimator in model $q_{\hat{\phi}}$. No difference is obseved and the computation of our variational estimate is fast since we do not generate samples (0.4ms vs. 14ms for the particle filter for one iteration step). We have also computed the averaged MSE over time of the estimators of X_t^2 (resp. of the variance of the posterior distribution approximated by a particle filter with 10^5 particles) which can be computed exactly with our method. Again, performances are similar and the difference is not significant since averaged over time the MSEs are $\mathbf{322.45}$ (resp. $\mathbf{1.5 \times 10^-4}$) for our method, $\mathbf{322.58}$ (resp. $\mathbf{1.4 \times 10^-4}$) for the particle filter and $\mathbf{322.60}$ (resp. $\mathbf{4.5 \times 10^-3}$) for the particle filter with estimated parameters, respectively.

We finally consider a maneuvering target tracking scenario with 3 jumps (straight, turn left, turn right). Here, $H(z_t) = I_4$, $R(z_t) = 3I_4$,

$$F(z_t) = \begin{bmatrix} 1 & \frac{\sin(\omega(z_t)T_e)}{\omega(z_t)} & 0 & -\frac{1-\cos(\omega(z_t)T_e)}{\omega(z_t)} \\ 0 & \cos(\omega(z_t)T_e) & 0 & -\sin(\omega(z_t)T_e) \\ 0 & \frac{1-\cos(\omega(z_t)T_e)}{\omega(z_t)} & 1 & \frac{\sin(\omega(z_t)T_e)}{\omega(z_t)} \\ 0 & \sin(\omega(z_t)T_e) & 0 & \cos(\omega(z_t)T_e) \end{bmatrix}, \quad Q(z_t) = \sigma_v^2(z_t) \begin{bmatrix} \frac{T_e^3}{3} & \frac{T_e^2}{2} \\ \frac{T_e^2}{2} & T_e \end{bmatrix} \otimes I_2,$$

$T_e = 2$, $\omega(z_t) \in \{0, 6\pi/180, -6\pi/180\}$ and $\sigma_v(z_t) \in \{7, 10, 10\}$. We set $p(z_t|z_{t-1}) = 0.8$ if $z_t = z_{t-1}$ and 0.1 otherwise. Here, we assume that $p_\theta = p$ is known. Estimating q_ϕ is challenging because the variational model consists of 3×3 (transition probabilities) $+3 \times 2 \times 10$ (symmetric covariance matrices) $+3 \times 4 \times 4 = 117$ parameters. The model is initialized such that $q_{\phi^{(0)}}(x_t|z_t, x_{t-1}, y_{0:t}) = p(x_t|z_t, x_{t-1}, y_t)$ for all z_t; we also set $B^{(0)}(z_t) = F(1)$ and $P^{(0)}(z_t) = R + HQ(1)H^T$, for all z_t. The KLD converges after 30000 iteration steps with a learning rate of 10^{-5} for the first 10000-th iterations and next 10^{-6}. The initial KLD is $\mathbf{2474.08}$ while the final value is $\mathbf{2177.14}$. The MSE as a function of time presents the same profile as in Fig. 5.7, so we give directly the averaged MSE over time: $\mathbf{3.78}$ (estimated variational model), $\mathbf{6.69}$ (initial variational model $q_{\theta^{(0)}}$) and $\mathbf{3.76}$ (particle filter). Again, our estimated variational distribution $q_{\hat{\phi}}$ offers the same performance as the particle filter but only requires 0.8ms vs. 17ms for computing one time step.

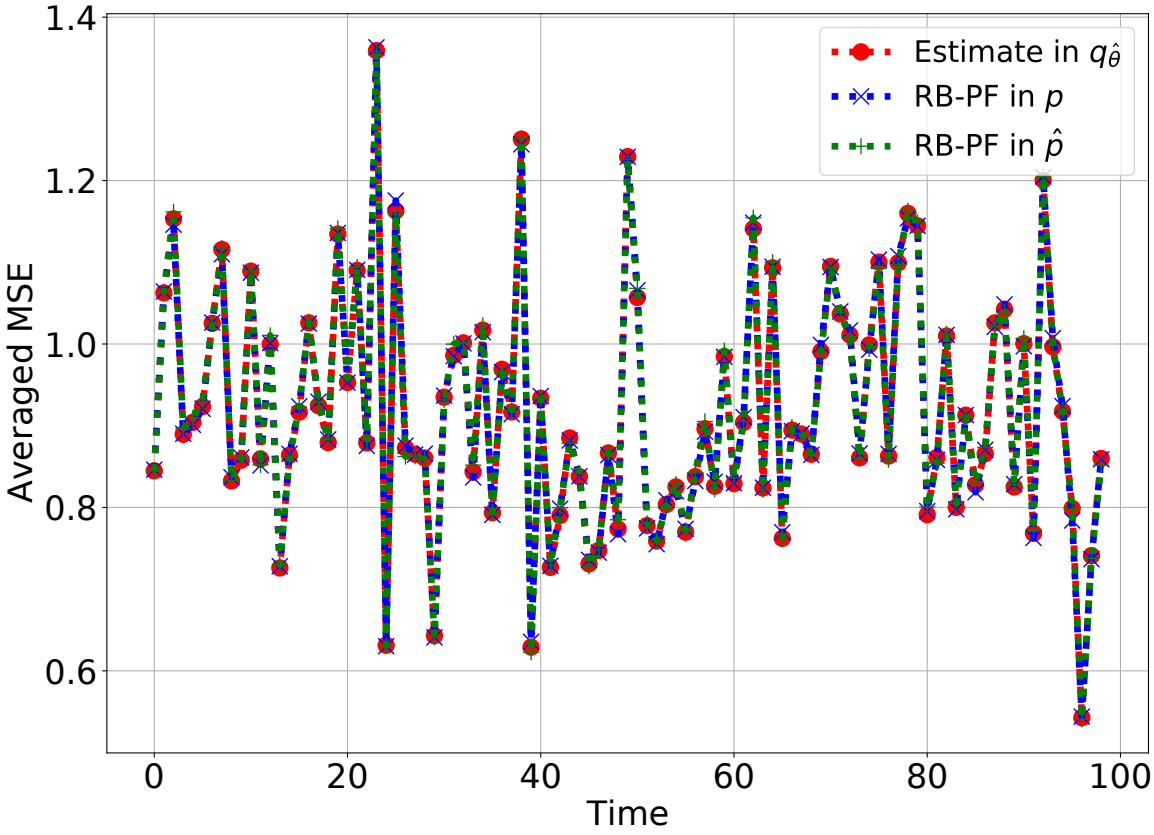


Figure 5.7: MSE of different estimators for the scalar scenario: Rao-Blackwellized particle filter (RB-PF) in the true model, RB-PF in the estimated model and exact estimator in the variational model.

Perspectives

Through this synthesis, we have revisited some problems related to Bayesian inference in models with latent variables and we have presented some contributions to address some limitations of the current tools. We end this synthesis by highlighting some unanswered questions or the new problems arose by the presented solutions. We also give some axis and projects in which I am or I will be involved and which focus on topics not discussed in the manuscript.

About this work

Particle filters - In chapter 2, we have proposed two alternate importance sampling mechanisms. The first one is based on the introduction of two importance distributions and aims at reducing the variance of Monte Carlo estimators by tuning the parameters of each importance distribution. In order to keep the rationale of traditional normalized importance sampling, the samples according to these distributions have been obtained by applying two deterministic transformations of initial samples drawn from any importance distribution. While this scheme is interesting from a computational point of view (it only requires a common set of samples), it may be not optimal from a statistical point of view. Indeed, let us address the following example in which we want to compute $\int x^2 \mathcal{N}(x; 0; 1)\mathrm{d}x = 1$ with

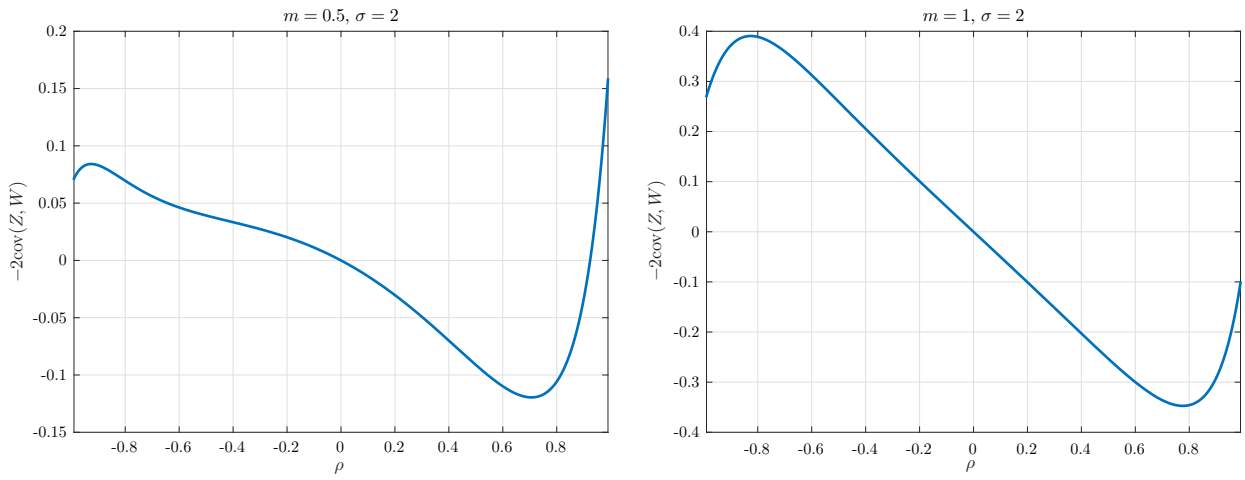
$$q_{1,2}(x_1, x_2) = \mathcal{N}\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}; \begin{bmatrix} m \\ m \end{bmatrix}; \begin{pmatrix} \sigma^2 & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 \end{pmatrix}\right).$$

Remember that the variance of the DPIS estimator can be rewritten as (see (2.12))

$$\mathcal{V}^{\infty}_{q_{1,2}}(h) = \mathbb{V}\mathrm{ar}\left(Z\right) - \mathbb{C}\mathrm{ov}\left(Z, W\right) + \mathbb{V}\mathrm{ar}\left(W\right),$$

where Z depends on $X_1 \sim q_1$ and W on $X_2 \sim q_2$. So the joint distribution acts on the opposite covariance term which should be as small as possible. Fig. 5.8 displays the behaviour of this covariance term for different values of m and σ^2, in function of ρ. Consequently, even if we have introduced a new scheme for normalized importance sampling, the tuning of a joint distribution such that all marginals are computable and with a limited extra computational cost compared to traditional normalized importance sampling remains an open problem.

In the same chapter, a new sampling-weighting-resampling mechanism has been proposed and aims at producing independent samples according to the same marginal distribution as that of the traditional mechanism while avoiding the support shrinkage due to the resampling step. In pathological models such as informative or high dimensional HMCs our technique can improve the results of classical particle filters for



(a) $m = 0.5, \sigma = 2$.

(b) $m = 1, \sigma = 2$.

Figure 5.8: Behaviour of the covariance term in the variance of the DPIS estimator.

a fixed computational cost and it involves a complexity in $\mathcal{O}(N^2)$. In the case where we would have at our disposal a procedure to sample directly N samples according to $\tilde{q}^N$ (static case) or to $\tilde{q}_t^N$ (sequential case), it would be possible to replace the three steps of particle filtering algorithm by a unique global sampling step. Some paths could be explored to reduce the computational cost in $\mathcal{O}(N^2)$. First, we could further consider the semi-independent resampling method proposed at the end of the chapter. Remember that our iterative solution is based on an uniform choice of the particles we sample again from their (conditional) importance distribution. However, an alternative solution which takes into account the current importance weights may improve the variance reduction w.r.t. our proposed strategy. Next, a different strategy would rely on a rejection sampling method based on samples according to the importance distribution q used in the classical mechanism; in this case, the averaged computational cost involved by the rejection step should be evaluated as it has been done for the *PaRIS* algorithm (Olsson and Westerborn, 2017; Douc et al., 2011b), for example. Next, in the spirit of differentiable particle filters (Corenflos et al., 2021), a solution could be to look for a transformation of independent samples according to q in independent samples according to $\tilde{q}^N$ through optimal transport solution. Finally, reducing the computational cost is not the only perspective associated to this method. Since our algorithm struggles against the degeneration phenomenon, it would be interesting to exploit it for smoothing problems. As a preliminary study, we have first measured this diversity by considering the experiment of Fig. 2.4, in which we have considered a sequence of observation of length $t = 10$ and computed the number of different ancestors from time $t - t'$, $t' \in [0, t]$. It can be observed in Fig. 5.9 that our sampling scheme limits the degeneration phenomenon not only for the filtering time ($t = 10$) but also for the past times. Note that in this figure, our independent-SIR solution only keeps N samples among the N^2 generated at each time step (so there is at most N different ancestors). In a smoothing perspective, it could be interesting to build algorithms which exploit the N^2 samples of each time step to select relevant trajectories (this is also the cost of the FFBS algorithm for additive functionals). Finally, in chapter 3 we have proposed some estimators of the asymptotic variance of a particular filtering and smoothing algorithms. A natural extension would be to consider the variance estimation associated to our independent SIR particle filter, but also to alternative smoothers (Briers et al., 2010).

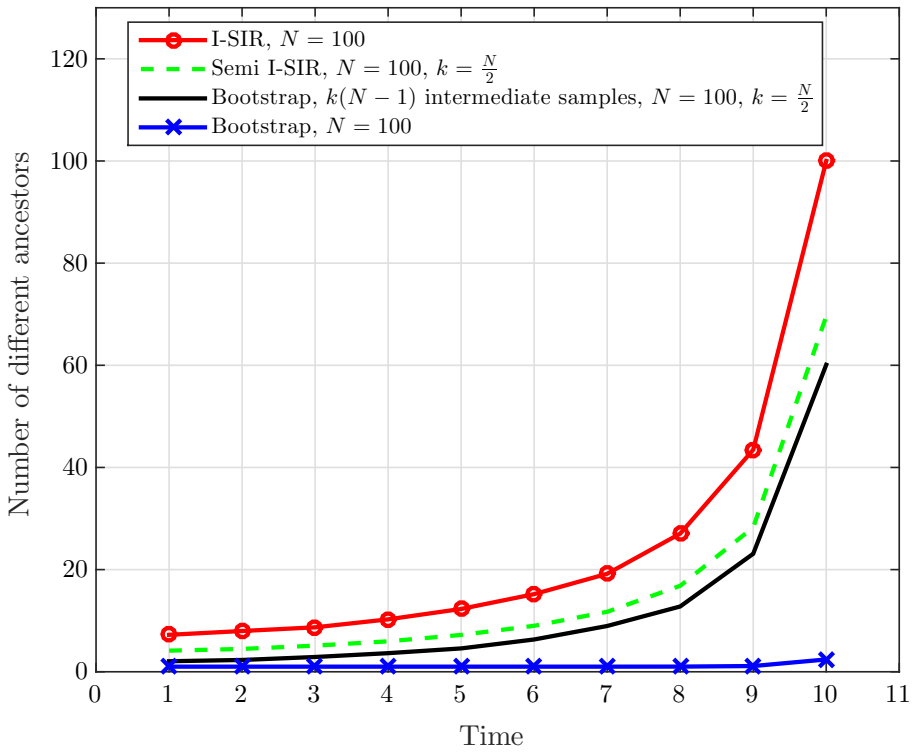


Figure 5.9: Number of different ancestors from time $10 - t'$ for a range bearing tracking scenario with $\sigma_\rho = 0.1$ et $\sigma_\theta = \frac{\pi}{1800}$.

Expressivity of generative models - In chapter 4, we have addressed the expressivity of the HMC and the RNN models from a system theory point of view. Our comparison is at the crossroad of several research field and involves statistical signal processing, machine learning and system theory. We compared the auto-correlation function of a stationnary observed process describing by one of the generative model of this manuscript. We have considered the linear case, so a natural perspective is to extend our result since RNNs are used with non linear activation function in practice. To that end, our study could be generalized by considering piecewise linear activation functions and tools relying on Extended/Unscented Kalman filter for approximating the computation of covariance function. Another perspective consists of clarifying the role of the dimension of the latent variables. We have seen that this dimension is directly related to the complexity of the covariance function produced by a model (through the rank of the Hankel matrix which enables to factorize the covariance function) but it would be interesting to obtain an explicit characterization as we did for the unidimensional case. In particular, it could be the first step to better understand the interest of generative models based on an LSTM parameterization. Indeed, LSTM architectures are sophisticated parameterizations of the deterministic function which computes the latent variable of a RNN,

$$h_t = f_\theta(h_{t-1}, y_t);$$

the parameterization of an LSTM involves a latent variable in augmented dimension,

$$c_t = \tilde{f}_\theta(h_{t-1}, c_{t-1}, y_t), \quad h_t = f_\theta(h_{t-1}, c_t, y_t),$$

where c_t is called the memory cell and aims at addressing the problem of vanishing gradient. Now, let us consider the problem where we want to build a generative model such as the core distributions of the GUM or the PMC are parameterized by such an architecture; what is the impact of this parameterization on the covariance function of the observed process? Note that since it involves a latent variable in augmented dimension, such a generative model could be interpreted as a TMC (so the latent process $\{H_t\}_{t\in\mathbb{N}}$ would coincide with a joint process including the "memory" process) and so a generative model in augmented dimension, whence the importance of understanding the effect of the dimension of latent variables. Finally, the last part of the chapter is also related to the previous question. We have seen that relaxing the Markoviannity of the latent process through a PMC enables us to describe more general covariance functions for the observed process. It however remains to fully characterize such covariance functions in order to clarify the role of the general PMC w.r.t. the GUM but also to understand the impact of the dimension since we have interpreted the PMC as a particular HMC in augmented dimension.

Deep parameterized Markovian models - In chapter 5, we have proposed hidden Markov models based on neural networks parameterizations for several Bayesian problem. In particular, we have proposed a step by step solution to build deep PMC models from simple and interpretable HMCs for classification problems. It remains to adapt this methodology for prediction problems in the continuous case such as target tracking, estimation of volatility,..., *i.e.* moving from an already parameterized HMC model to a deep PMC or TMC.

From an applicative point of view, the reliability of our construction for real data and in particular for biomedical problems is a work in progress in the context of the Ph.D of K. Morales with the Gepromed. Preliminary experiments in which we have compared some of our models have been done for the segmentation of micro-computed tomography X-ray scans of human arteries containing a metallic stent biomaterial, see

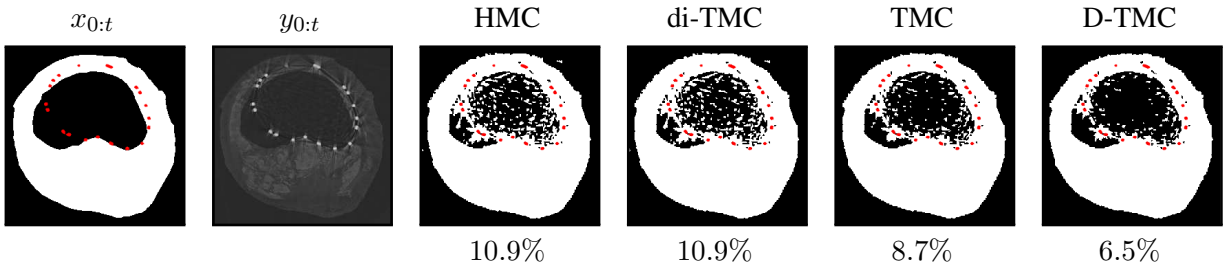


Figure 5.10: Illustration of the unsupervised segmentation of a slice. Our D-TMC appears to better fit the non-stationary noise, offering a 4%-point improvement in the error rate. The stent components appearing in red are segmented beforehand with a thresholding technique and are considered as image borders during the segmentation using the probabilistic models.

Fig. 5.10. The next step will be to decline such models to realize in a unique step two critical tasks, super resolution image reconstruction and segmentation.

Finally, the construction of our deep Markovian models has been done for time series data models. The extension for spatial models such as hidden Markov random field parameterized by convolutional neural networks is a natural perspective.

Other perspectives

Importance Sampling for high dimensional problems - Currently, in the context of the Ph.D thesis of Y. Janati, we are reviewing some adaptive sampling strategies for high dimensional distributions. It has been observed that even recent sampling strategies such as those proposed in Gabrie et al. (2022); Thin et al. (2021) tend to fail when we consider an anisotropic target distribution

$$\pi(x) = \frac{1}{4}\sum_{i=1}^{4}\mathcal{N}(x; \mathrm{m}_i, \Sigma),$$

where $\mathrm{m}_1 = -15 \cdot \mathbf{1}_{d_x}$, $\mathrm{m}_2 = 15 \cdot \mathbf{1}_{d_x}$, m_3 is such that $\mathrm{m}_{3,2i} = 10$ and $\mathrm{m}_{3,2i+1} = -10$ and m_4 such that $\mathrm{m}_{3,2i} = -10$ and $\mathrm{m}_{3,2i+1} = 10$ ($\mathbf{1}_{d_x}$ is a d_x dimensional vector where each coordinate takes the value 1). Indeed, when we estimate the normalizing constant of π (so 1), these strategies are not reliable when the dimension increases:

Method	$d = 2$	$d = 10$
NEO (50.10^3 samples)	0.98 ± 0.21	10^{-6}
Gabrié's algo (2.10^3 samples)	0.98 ± 0.11	0.90 ± 0.26

It has been observed that the main drawback of such methods is that the adaptive strategy fails to discover some modes of the target distribution when the dimension is large. We are thus building an adaptive strategy with the following rationale:

1. we first look for building a sequence of distribution $\{q_t\}_{t\in\mathbb{N}}$ such that q_{t+1} puts mass in the modes of π which have not been explored until time t;

2. we next analyse the convergence properties of this sequence of importance distributions;

3. using a combination of sampling tools (importance sampling and Monte Carlo Markov Chain methods) and variational inference, we are able to sample approximately given this sequence of distribution and to propose a parameterized distribution close to π, in a given sense.

This work is still in progress but the first results are positive; for the previous example, an estimation of the normalization constant gives 1.00 ± 10^{-2} for $d_x = 2$ and $1.01 \pm 3 \cdot 10^{-2}$ for $d_x = 10$.

Multi-target surveillance - Some parts of my Ph.D thesis were devoted to the multi-target filtering problem where contrary to the filtering problem addressed in this manuscript, we look for estimating the hidden states of an unknown number of targets from measurements which consist of observations and false alarms. I have not worked on this topic since 2013. In the context of a collaboration between IP Paris and the Direction générale de l'armement, I am involved in a three years CIEDS (Centre Interdisciplinaire d'Etudes pour la Défense et la Sécurité) project which will start at the end of 2022. The objective of this project is to develop an approach to adaptively allocate sensing resources in multisensor multi-target tracking surveillance networks based on fundamental concepts in network information theory and decision-theoretic criteria. A great challenge of this project will consist in re-evaluating the key tools in information theory applied to the challenges of multi-target surveillance based on point process theory (Clark, 2022), which is designed to accommodate uncertainty in the states of individual targets and the target number. The information-theoretic methods developed will be applied to multi-sensor problems to enable decisions to be made on how to allocate sensor resources in addition to refining the knowledge of the scene. Some applications related to defense security are aimed. With the recent advances in autonomous systems, such as Autonomous Underwater Vehicles, Unmanned Aerial Vehicles and Unmanned Ground Vehicles, the need for a mathematically coherent framework for fusing data from different vehicles is necessary if the best understanding of the environment is to be achieved. In conventional systems, a human operator typically makes the decision about which sensors to operate based on their assessment of the scenario which is infeasible for large scale sensor networks with many potential threats. This project will develop methods for autonomous sensor management for heterogeneous sensor networks to enhance situational awareness without the need for costly and sub-optimal human selection of the sensors. I will manage the supervision of postdoctoral research associate and a PhD student.

Bibliography

Abbassi, N., Benboudjema, D., and Pieczynski, W. (2011). Kalman filtering approximations in triplet Markov Gaussian switching models. In *IEEE Workshop on Statistical Signal Processing*, Nice, France. 24

Ait-El-Fquih, B. and Desbouvries, F. (2006). Kalman filtering in triplet Markov chains. *IEEE Transactions on Signal Processing*, 54(8):2957–63. 24

Akhiezer, N. I. and Kemmer, N. (1965). *The classical moment problem and some related questions in analysis*, volume 5. Oliver & Boyd Edinburgh. 71, 79

Balakrishnan, S., Wainwright, M. J., Yu, B., et al. (2017). Statistical guarantees for the EM algorithm: From population to sample-based analysis. *Annals of Statistics*, 45(1):77–120. 22

Bayer, J. and Osendorfer, C. (2014). Learning Stochastic Recurrent networks. *preprint arXiv:1411.7610*. 84

Bikmukhamedov, R., Nadeev, A., Maione, G., and Striccoli, D. (2020). Comparison of HMM and RNN models for network traffic modeling. *Internet Technology Letters*, 3. 66

Billingsley, P. (1986). *Probability and Measure*. John Wiley and Sons, second edition. 56

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer. 17, 82

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877. 16, 82

Bousquet, O., Boucheron, S., and Lugosi, G. (2003). Introduction to statistical learning theory. In Bousquet, O., von Luxburg, U., and Ratsch, G., editors, *Advanced Lectures on Machine Learning*, volume 3176 of *Lecture Notes in Computer Science*, pages 169–207. Springer. 18

Briers, M., Doucet, A., and Maskell, S. (2010). Smoothing algorithms for state–space models. *Annals of the Institute of Statistical Mathematics*, 62(1):61. 102

Brockett, R. W. (2015). *Finite dimensional linear systems*. SIAM. 71

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167. 17

Caines, P. E. (2018). *Linear stochastic systems*, volume 77. SIAM. 66, 69

Cappé, O., Godsill, S. J., and Moulines, E. (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *Proc. of the IEEE*, 95(5):899–924. 32

Cappé, O., Moulines, É., and Rydén, T. (2005). *Inference in Hidden Markov Models*. Springer Series in Statistics. Springer-Verlag. 22, 32

Cérou, F., Moral, P. D., and Guyader, A. (2011). A nonasymptotic theorem for unnormalized Feynman-Kac particle models. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 47(3):629 – 649. 56

Chan, H. P. and Lai, T. L. (2013). A general theory of particle filters in hidden Markov models and some applications. *The Annals of Statistics*, 41(6):2877 – 2904. 48

Chen, C.-T. (1970). *Introduction to linear system theory*. Holt, Rinehart and Winston. 27, 66

Chopin, N. and Papaspiliopoulos, O. (2020). *An introduction to sequential Monte Carlo*. Springer. 32

Chui, C. K. and Chen, G. (2012). *Signal processing and systems theory: selected topics*, volume 26. Springer Science & Business Media. 66

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Advances in Neural Information Processing Systems (NIPS 2006*. 26

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., and Bengio, Y. (2015). A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pages 2980–2988. 84

Clark, D. E. (2022). A Cramér Rao bound for point processes. *IEEE Transactions on Information Theory*, 68(4):2147–2155. 106

Corenflos, A., Thornton, J., Deligiannidis, G., and Doucet, A. (2021). Differentiable particle filtering via entropy-regularized optimal transport. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2100–2111. PMLR. 102

Cox, D. R., Gudmundsson, G., Lindgren, G., Bondesson, L., Harsaae, E., Laake, P., Juselius, K., and Lauritzen, S. L. (1981). Statistical analysis of time series: Some recent developments [with discussion and reply]. *Scandinavian Journal of Statistics*, 8(2):93–115. 25

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314. 17

De Jong, L. S. (1975). *Numerical aspects of realization algorithms in linear systems theory*. PhD thesis, Department of Mathematics and Computer Science. 67

De Jong, L. S. (1978). Numerical aspects of recursive realization algorithms. *SIAM Journal on Control and optimization*, 16(4):646–659. 67

Del Moral, P. (2004). Feynman-kac formulae. In *Feynman-Kac Formulae*, pages 47–93. Springer. 45, 47, 52

Del Moral, P., Doucet, A., and Sumeetpal, S. (2010). Forward smoothing using sequential monte carlo. *ArXiv:1012.5390*. 22

Dempster, A. P., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society (B)*, 39(1):1–38. 16, 22, 82

Derrode, S. and Pieczynski, W. (2004). Signal and image segmentation using pairwise Markov chains. *IEEE Transactions on Signal Processing*, 52(9):2477–89. 20, 23

Derrode, S. and Pieczynski, W. (2013). Exact fast computation of optimal filter in gaussian switching linear systems. *IEEE Signal Processing Letters*, 20(7):701–704. 24, 97

Desbouvries, F. and Pieczynski, W. (2003a). Modèles de Markov triplet et filtrage de Kalman. *Comptes Rendus de l'Académie des Sciences - Mathématiques*, 336(8):667–670. in French. 24

Desbouvries, F. and Pieczynski, W. (2003b). Particle filtering in pairwise and triplet Markov chains. In *Proc. IEEE - EURASIP Workshop on Nonlinear Signal and Image Processing*, Grado-Gorizia, Italy. 24

Deshmukh, A. M. (2020). Comparison of hidden markov model and recurrent neural network in automatic speech recognition. *European Journal of Engineering and Technology Research*, 5(8):958–965. 66

Douc, R., Cappé, O., and Moulines, É. (2005). Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis (ISPA)*, Zagreb, Croatia. 31

Douc, R., Garivier, A., Moulines, E., and Olsson, J. (2011a). Sequential Monte Carlo smoothing for general state space hidden Markov models. *The Annals of Applied Probability*, 21(6):2109 – 2145. 49

Douc, R., Garivier, A., Moulines, E., and Olsson, J. (2011b). Sequential monte carlo smoothing for general state space hidden markov models. *Annals of Applied Probability*, 21(6):2109–2145. 102

Douc, R. and Moulines, E. (2012). Asymptotic properties of the maximum likelihood estimation in misspecified hidden Markov models. *Annals of Statistics*, 40(5):2697–2732. 21

Douc, R., Moulines, E., and Ryden, T. (2004). Asymptotic properties of the maximum likelihood estimator in autoregressive models with markov regime. *Annals of Statistics*, 32(5):2254–2304. 21

Douc, R., Moulines, E., and Stoffer, D. (2014). *Nonlinear time series: Theory, methods and applications with R examples*. CRC press. 56

Doucet, A., de Freitas, N., and Gordon, N. (2001a). *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer-Verlag. 31

Doucet, A., Godsill, S. J., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208. 22, 27

Doucet, A., Gordon, N. J., and Krishnamurthy, V. (2001b). Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–24. 24

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. (2010). Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(19):625–660. 88

Faurre, P. (1979). *Opérateurs rationnels positifs*. Dunod. 66, 68, 69

Faurre, P. L. (1976). Stochastic realization algorithms. In *Mathematics in Science and Engineering*, volume 126, pages 1–25. Elsevier. 66

Fearnhead, P. and Clifford, P. (2003). On-line inference for hidden markov models via particle filters. *Journal of The Royal Statistical Society Series B-statistical Methodology*, 65:887–899. 99

Gabrie, M., Rotskoff, G., and Vanden-Eijnden, E. (2022). Adaptive monte carlo augmented with normalizing flows. *Proceedings of the National Academy of Sciences*, 119. 105

Gevers, M. (2006). A personal view of the development of system identification: A 30-year journey through an exciting field. *IEEE Control systems magazine*, 26(6):93–105. 66

Gevers, M. and Wouters, W. (1978). An innovations approach to the discrete-time stochastic realization problem. *Journal A*, 19(2):90–110. 66, 71

Geweke, J. (1989). Bayesian Inference in Econometric Models Using Monte Carlo Integration. *Econometrica*, 57(6):1317–1339. 30

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS*, volume 9 of *JMLR Proceedings*, pages 249–256. JMLR.org. 88

Goldberg, Y. (2017). Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1):1–309. 26

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge. 17, 26

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/ non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113. 27, 46

Gorynin, I., Gangloff, H., Monfrini, E., and Pieczynski, W. (2018). Assessing the segmentation performance of pairwise and triplet Markov Models. *Signal Processing*, 145:183–192. 23, 90

Hecht-Nielsen, R. (1992). Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier. 17

Hesterberg, T. (1988). *Advances in Importance Sampling*. PhD thesis, Stanford University. 16

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). Beta-VAE: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR*. OpenReview.net. 91

Hinton, G. E., Deng, L., Yu, D., Dahl, G. E., A-R., M., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97. 88

Ho, B. and Kalman, R. E. (1966). Effective construction of linear state-variable models from input/output functions. *at-Automatisierungstechnik*, 14(1-12):545–548. 67

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780. 26, 66

Hol, J. D., Schön, T. B., and Gustafsson, F. (2006). On resampling algorithms for particle filtering. In *Proc. IEEE NSSPW*, Cambridge, UK. 31

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257. 17

Hu, W., Liao, Y., and Vemuri, V. R. (2003). Robust anomaly detection using support vector machines. In *Proceedings of the international conference on machine learning*, pages 282–289. Citeseer. 17

Huber, P. J. (1967). The behavior of maximum likelihood estimates under nonstandard condition. In LeCam, N. and Neyman, J., editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, CA, USA. University of California Press. 16

Jain, A. K., Mao, J., and Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, 29(3):31–44. 17

Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory*, volume 64 of *Mathematics in Science and Engineering*. Academic Press, San Diego. 20

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233. 16

Kailath, T. (1980). *Linear systems*, volume 156. Prentice-Hall Englewood Cliffs, NJ. 66

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., and Chopin, N. (2015). On particle methods for parameter estimation in state-space models. *Statist. Sci.*, 30(3):328–351. 22, 66

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 85, 98

Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*, pages 3581–3589. 92

Kingma, D. P. and Welling, M. (2014). Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR*. 18, 35, 83

Klys, J., Snell, J., and Zemel, R. (2018). Learning latent subspaces in variational autoencoders. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems*, pages 6445–6455. 92

Koopman, S. J., Lucas, A., and Scharth, M. (2016). Predicting Time-Varying Parameters with Parameter-Driven and Observation-Driven Models. *The Review of Economics and Statistics*, 98(1):97–110. 25

Kumar, S., Pradeep, J., and Zaidi, H. (2021). Learning robust latent representations for controllable speech synthesis. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3562–3575. Association for Computational Linguistics. 92

LeCun, Y. (1998). The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*. 85

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444. 17

Lee, A. and Whiteley, N. (2018). Variance estimation in the particle filter. *Biometrika*, 105(3):609–625. 48, 50, 52, 53

Li, T., Bolić, M., and Djuric, P. M. (2015). Resampling Methods for Particle Filtering: Classification, implementation, and strategies. *IEEE Signal Processing Magazine*, 32(3):70–86. 31

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. (2017). The expressive power of neural networks: A view from the width. In *Advances in neural information processing systems*, pages 6231–6239. 17

Manton, J. H. and Amblard, P.-O. (2015). A primer on Reproducing Kernel Hilbert Spaces. *Found. Trends Signal Process.*, 8(1-2):1–126. 17

Mathai, A. and Provost, S. (1992). *Quadratic Forms in Random Variables*. Statistics: A Series of Textbooks and Monographs. Taylor & Francis. 97

Mohamed, A., Dahl, G. E., and Hinton, G. (2012). Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech and Langage Processing*, 20(1):14–22. 88

Mozer, M. C. (1995). A focused backpropagation algorithm for temporal. *Backpropagation: Theory, architectures, and applications*, 137. 26

Olsson, J. and Douc, R. (2019). Numerically stable online estimation of variance in particle filters. *Bernoulli*, 25(2):1504 – 1535. 48, 58

Olsson, J. and Westerborn, J. (2017). Efficient particle-based online smoothing in general hidden Markov models: The PaRIS algorithm. *Bernoulli*, 23(3):1951 – 1996. 57, 102

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. 26

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*. 98

Paulsen, V. I. and Raghupathi, M. (2016). *An introduction to the theory of reproducing kernel Hilbert spaces*, volume 152. Cambridge University Press. 17

Petersen, K. B. and Pedersen, M. S. (2008). The matrix cookbook. Version 20081110. 97

Pieczynski, W. (2002). Chaînes de Markov triplet. *Comptes Rendus de l'Académie des Sciences - Mathématiques*, 335:275–278. in French. 23, 97

Pieczynski, W. (2003). Pairwise Markov chains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):634–39. 23, 87

Pieczynski, W. (2011a). Exact filtering in conditionally Markov switching hidden linear models. *Comptes Rendus Mathematique*, 349(9-10):587–590. 24

Pieczynski, W. (2011b). Exact smoothing in hidden conditionally Markov switching linear models. *Communications in Statistics - Theory and Methods*, 40(16):2823–2829. 24

Pinkus, A. (1999). Approximation theory of the MLP model in neural networks. *Acta numerica*, 8(1):143–195. 17

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation : Auxiliary particle filter. *Journal of the American Statistical Association*, 94:590–99. 20, 32, 58

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286. 20, 22, 87

Robinson, A. and Fallside, F. (1987). *The utility driven dynamic error propagation network*. University of Cambridge Department of Engineering Cambridge, MA. 26

Rubin, D. B. (1988). Using the SIR algorithm to simulate posterior distributions. In *Bayesian Statistics III*. Oxford University Press. 21, 26, 29

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science. 17

Sagan, H. (2012). *Space-filling curves*. Springer. 90

Schön, T., Gustafsson, F., and Nordlund, P.-J. (2005). Marginalized particle filters for mixed linear nonlinear state-space models. *IEEE Trans. on Signal Processing*, 53:2279–2289. 96

Smith, A. F. M. and Gelfand, A. E. (1992). Bayesian statistics without tears : a sampling-resampling perspective. *The American Statistician*, 46(2):84–87. 21, 26

Tanizaki, H. and Mariano, R. (1994). Prediction, filtering and smoothing in non-linear and non-normal cases using Monte Carlo integration. *Journal of Applied Econometrics*, 9(2):163–79. 22, 27

Thin, A., Janati El Idrissi, Y., Le Corff, S., Ollion, C., Moulines, E., Doucet, A., Durmus, A., and Robert, C. X. (2021). Neo: Non equilibrium sampling on the orbits of a deterministic transform. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 17060–17071. Curran Associates, Inc. 105

Tugnait, J. K. (1982). Adaptive estimation and identification for discrete systems with Markov jump parameters. *IEEE Transactions on Automatic Control*, 27(5):1054–65. 24, 27

Tzikas, D. G., Likas, A. C., and Galatsanos, N. P. (2008). The variational approximation for Bayesian inference. *IEEE Signal Processing Magazine*, 25(6):131–146. 82

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media. 17

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience. 18

Vergé, C., Dubarry, C., Del Moral, P., and Moulines, É. (2015). On parallel implementation of sequential Monte Carlo methods: the island particle model. *Statistics and Computing*, 25(2):243–260. 40

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560. 26

White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica*, 50(1):1–25. 16