

Thèse de doctorat

NNT : 2024IPPAS019



INSTITUT
POLYTECHNIQUE
DE PARIS



Generative Markov models for sequential Bayesian classification

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom SudParis

École doctorale n°574 École Doctorale de Mathématiques Hadamard (EDMH)
Spécialité de doctorat : Mathématiques appliquées

Thèse présentée et soutenue à Télécom SudParis, campus Évry, le 02/10/2024, par

KATHERINE TANIA MORALES QUINGA

Composition du Jury :

François Septier Professeur, Université Bretagne Sud	Rapporteur
Myriam Maumy Professeure, École des Hautes Études en Santé Publique	Rapporteure
Stéphane Derrode Professeur, École Centrale de Lyon	Examinateur
Sylvie Le Hégarat Professeure, Université Paris-Saclay	Examinaterice
Erwan Le-Pennec Professeur, École Polytechnique	Examinateur
Yohan Petetin Maître de conférences, HDR, Télécom SudParis	Directeur de thèse

Acknowledgements

I would like to express my deepest gratitude to everyone who contributed to the completion of this PhD.

First, I extend my heartfelt thanks to my jury for their time, expertise, and valuable insights, which have greatly enriched this work. I am especially grateful to my supervisor, Yohan Petetin, for their guidance, support, and encouragement throughout this journey. Your mentorship has been invaluable. I would also like to express my deep appreciation to Hugo Ganglof, who generously shared his knowledge and expertise. Your kindness and willingness to help throughout my PhD were truly inspiring.

A special thanks to the department director, Professor Wojciech Pieczynski, whose support was crucial not only for the advancement of my research but also for his genuine concern for my well-being.

To my lab partners, I feel fortunate to have met you all, especially during this last year. It has been a pleasure to work alongside such wonderful colleagues. I also want to thank Laura and Julie, for their constant assistance. Your help has been invaluable in navigating the administrative aspects of this process.

This thesis would not have been possible without the help and support of many people. To my family, for their unconditional love and support, especially my mother Martha, my grandfather, Diana and Christopher, my father Felipe, and my uncle Fernando, for being my source of strength and comfort.

To my friends from Ecuador, particularly Josselyn, Mateo, and my high school friends, for keeping me grounded and connected from afar. To my heart-family in France, especially Any, Eli, Alex, Majo, Sofi, Quentin, François, and Albita, for their friendship and for standing by me through this journey. And to Dani, for being my rock during difficult times and my partner in this

adventure. Lastly, I would like to thank my adoptive family in France for their warmth and hospitality, and everyone who, in one way or another, has helped me along this journey. My deepest gratitude to all of you.

Spanish

Esta tesis doctoral no habría sido posible sin la ayuda de muchas personas. Quiero expresar mis más sinceros agradecimientos a todos los que me han ayudado a lo largo de este viaje.

En primer lugar, extiendo mi más sincero agradecimiento a mi jurado por su tiempo, experiencia y valiosas aportaciones, que han enriquecido enormemente este trabajo. Estoy especialmente agradecida a mi director de tesis, Yohan Petetin, por su guía, apoyo y ánimo a lo largo de este viaje. Su mentoría ha sido invaluable.

A Hugo Ganglof, quien compartió generosamente su conocimiento y experiencia. Tu amabilidad y disposición para ayudar durante todo mi doctorado fueron realmente inspiradoras.

Un agradecimiento especial al director del departamento, el Profesor Wojciech Pieczynski, cuyo apoyo fue crucial no solo para el avance de mi investigación, sino también por su genuino interés en mi bienestar.

A mis compañeros de laboratorio, me siento afortunada de haberlos conocido, especialmente durante este último año.

A Laura y Julie por su constante ayuda. Su asistencia ha sido invaluable para navegar los aspectos administrativos de este proceso.

A mi familia, por su apoyo incondicional a la distancia, por su amor y saber que juntos podemos sobrellevar momentos difíciles. Cada momento que tuve la oportunidad de pasar con ustedes durante el desarrollo de la tesis, fue un momento de felicidad y me ayudó a seguir adelante.

A mi madre Martha, por su amor, consejos y esfuerzos para que cumpliera lo que siempre soñé. Gracias por ser mi ejemplo a seguir y por enseñarme a ser fuerte y, sobretodo, por cuidarme en los momentos difíciles. Ser mi paño de lágrimas y mi fuente de amor.

A mi abuelito Alberto, por el amor que me ha dado a la distancia y sus preguntas sobre Francia y el viejo continente, que siempre me sacaban una sonrisa.

A mis hermanos, por escucharme y hacerme reír cuando lo necesitaba.

Diana, por ser una hermana comprensiva, escucharme y darme consejos en los momentos en los cuales me sentía sola y desanimada.

Christopher, por su inocencia, por hacerme reír con sus ocurrencias, y por

darme bellas experiencias siempre que viajaba a casa.

A mi tío Fernando por cuidarme y apoyarme a lo largo de mi vida, siendo como un segundo padre para mi.

A mi padre, por sus consejos y ánimos para que continuara con mi sueño, fue algo que necesitaba en los momentos de duda y saber que usted confiaba en mi, me ayudó a seguir adelante.

A mi primer gordito, Edi por estar cerca de mi, por sus palabras de aliento y por ser mi hermanito menor de corazón.

Mayerli, Brandon, Xavier, gracias por esas conversaciones que tuvimos, el apoyo y por hacerme sentir como en casa.

A Laura, por su apoyo y por aceptarme como una hija más. A mi sobrina Mayte y a Erika.

A mis amigos de Ecuador, cada viaje que hice a casa fue una oportunidad para saber que cuento con ustedes. Gracias por esas conversaciones que me ayudaron a encontrarme de nuevo y recordar las experiencias que vivimos juntos. Gracias a mis amigos de la U, en particular a Josselyn, por su apoyo y por ser una amiga que siempre está ahí. A Mateo, por ser como un hermano mayor y por escucharme y apoyarme en diferentes momentos de mi vida. A mis amigos del colegio, Bryan, Alex, Juanito, Steven (Monito), Angel y Geovanny (Chochito), por su apoyo y las salidas que siempre disfrutamos juntos. A Edgar, por ser un amigo que siempre está ahí y el apoyo que me brindó a la distancia.

A mi familia de corazón aquí en Francia:

Any, mi mejor amiga, mi hermana, gracias por cada momento que compartimos juntas, momentos de felicidad y de tristeza. Gracias por estar siempre ahí y ayudarme a ver la luz y creer en mi, incluso cuando yo no lo hacía.

Eli, por tus consejos, por estar ahí siempre para escuchar los momentos buenos y malos que me pasaron, y por todos tus detalles.

Alex, por ser como un hermano mayor, por tus consejos y por escucharme, por compartir este camino doctoral juntos y por las risas y lágrimas que compartimos.

Majo, por ser una amiga sincera, comprensiva, por escucharme, por los consejos, y por llevarme al Crossfit.

Quentin y Francois, por ayudarme a sentirme como en casa e incursionarnos en la cultura francesa. Viva el norte y la Bretagne!

Sofi, gracias por apoyarme y escucharme, por ser una amiga incondicional, por lo bueno y malo que hemos pasado juntas.

Albita, por el apoyo y risas de nuestra experiencia doctoral, por ser una amiga que a la distancia me ha demostrado mucho amor y apoyo.

Dani, gracias por compartir este experiencia doctoral, por los momentos de estrés, tristeza y felicidad que compartimos juntos. Gracias por ser mi soporte y por el amor que me has brindado. Que esta etapa sea el inicio de una aventura juntos.

A todos mis amigos de Francia, gracias! Mona, Yannick, Nas, Belén, Vivi, Vale, Panchito, Mónica, Balthazar, Agustín, Louis (Lucho), Thimo, Geremy, David, Mica, GianKa, Cata. Con cada uno de ustedes he compartido momentos que guardo en mi corazón.

Por último, y no menos importante, a mi familia adoptiva de Francia por su amor y por hacerme sentir como en casa. Gracias Isabelle, Olivier, Valentin, Camille, Chloe, por sus consejos y hospitalidad.

Hay muchas personas que están y estuvieron presentes a lo largo de este viaje, Luz Marina, mis tíos, mis primos, mis amigos de la infancia. Algunas que ya no forman parte de mi vida, pero que en algún momento me ayudaron a seguir adelante y me brindaron su apoyo y amor. A todos ellos, gracias! Siempre estarán en mi corazón y les deseo lo mejor en sus vidas.

Contents

Acknowledgements	i
Nomenclature	vii
Acronyms	x
Introduction générale	1
General introduction	5
1 Technical introduction	13
1.1 Deep learning	14
1.1.1 Fundamental principle	14
1.1.2 Deep neural networks architectures for sequential data .	14
1.2 Bayesian estimation	16
1.2.1 Approximated Maximum Likelihood estimation with Variational Inference	17
1.2.2 Posterior distribution	22
1.2.3 Discussion	23
1.3 Sequential data modeling	24
1.3.1 Hidden Markov chains	24
1.3.2 Pairwise Markov chains	25
1.3.3 Sequential generative models for Bayesian classification	26
1.3.4 Organization of the thesis	26
2 Generative hidden Markov models	29
2.1 Introduction	30

2.2	The pairwise Markov chain as a unified model	30
2.3	Parameter estimation for general PMCs	33
2.3.1	General parameterization of PMCs	33
2.3.2	Variational Inference for PMCs	35
2.4	Experiments and results	37
2.4.1	Model description	37
2.4.2	Results	38
2.5	Generative power of PMCs	42
2.5.1	Linear and stationary Gaussian PMCs	43
2.5.2	Theoretical analysis of PMCs	44
2.6	Conclusions	48
3	Triplet Markov models for semi-supervised classification	49
3.1	Introduction	50
3.2	Semi-supervised estimation in general TMC	51
3.2.1	General parameterization of the TMC	51
3.2.2	A brief description of the semi-supervised problem	53
3.3	Semi-supervised Variational Inference for TMCs	54
3.3.1	ELBO for semi-supervised learning	54
3.3.2	Learning semi-supervised TMCs	55
3.4	Experiments	58
3.4.1	DTMC vs existing models	58
3.4.2	Binary data generation	60
3.4.3	Semi-supervised binary image segmentation	61
3.4.4	Results	62
3.5	Conclusions	64
4	Deep Markov models for unsupervised classification	65
4.1	Introduction	66
4.2	PMCs for unsupervised classification	67
4.2.1	Bayesian inference for PMCs	68
4.2.2	Deep PMCs for unsupervised classification	71
4.2.3	Simulations	73
4.3	TMCs for unsupervised classification	75
4.3.1	Variational Inference for general TMCs	76
4.3.2	Estimation algorithm for TMCs	77
4.3.3	Deep TMCs for unsupervised classification	83
4.3.4	Simulations	86
4.4	Experiments on real datasets	90
4.4.1	Unsupervised segmentation of biomedical images	91

4.4.2	Unsupervised clustering for human activity recognition .	91
4.5	Conclusions	93
5	Medical Perspectives	95
5.1	Context and motivation	96
5.2	Data and preprocessing	97
5.2.1	Data availability	97
5.2.2	Challenges	100
5.2.3	Pre-processing of the CT and micro CT images	101
5.3	Medical image segmentation	104
5.3.1	Results	105
5.4	Remaining challenges	107
5.5	Conclusions	109
Conclusions and Perspectives		111
Appendices		115
A	Additional material	117
B	Generative Pairwise Markov Models	119
C	Supervised Bayesian classification	123
D	Semi-supervised Bayesian classification	125
E	Unsupervised Bayesian classification	127
F	Medical image segmentation	137
Bibliography		152

Nomenclature

General

- a Light lower case letters represent scalars or functions
 A Upper case letters represent matrices
 \mathbb{R}^n Set of real vectors of dimension n
 $\mathbb{R}^{n \times m}$ Set of real matrices of dimension $n \times m$
 \mathbb{C} Set of complex numbers
 \Re Real part of a complex number
 $\text{diag}(\cdot)$ Diagonal matrix
 \cdot^\top Transpose operator of a matrix or a vector
 sigm Sigmoid function $\text{sigm}(x) = 1/(1 + \exp(-x))$

Probability Theory

- x (Observed) random variable and its realization. As far as the context is clear, we use the same symbol for both.
 y Random variable and its realization, which represents the label associated with x
 z Latent random variable and its realization
 x_t Random variable or realization at time t
 $x_{t:k}$ Random variable or realization from time t to k

$x_{0:T}$ Sequence of random variables or realizations from time 0 to T

$\mathcal{N}(x; \mu, \sigma)$ Gaussian density function with mean μ and covariance σ taken at point x

$\mathcal{N}(\mu, \sigma)$ Gaussian distribution with mean μ and covariance σ

$\mathcal{B}er(p)$ Bernoulli distribution of parameter p

\sim Indicates the law followed by a random variable

$p(x)$ Probability of a realization x (discrete case) and probability density function of x (continuous case)

$D_{\text{KL}}(q | p)$ Kullback-Leibler divergence between distributions q and p

$\mathbb{E}(x)$ Expectation of a random variable x

$\mathbb{E}_{p(\cdot)}(\cdot)$ Expectation under the distribution $p(\cdot)$

$\text{Var}(x)$ Variance of a random variable x

$\text{Cov}(x, y)$ Covariance between the random variables x , and y

$[\mu_\theta^x, \sigma_\theta^x]$ Mean and variance or covariance matrix of the random variable x that depends on θ

ρ_θ^x Bernoulli parameter of the random variable x depends on θ

List of Acronyms

CVAE Conditional Variational AutoEncoder.

DNN Deep Neural Network.

DPMC Deep Pairwise Markov Chain.

DPPMC Deep Partially Pairwise Markov Chain.

DTMC Deep Triplet Markov Chain.

ELBO Evidence Lower Bound.

EM Expectation-Maximization.

GANs Generative Adversarial Networks.

GEPROMED Groupe Européen de Recherche sur les Prothèses Appliquées à la Chirurgie Vasculaire.

GMMs Gaussian Mixture Models.

GRU Gated Recurrent Unit.

GUM Generative Unified Model.

HMC Hidden Markov Chain.

KLD Kullback-Leibler Divergence.

LSTM Long Short-Term Memory.

ML Maximum Likelihood.

PMC Pairwise Markov Chain.

PPMC Partially Pairwise Markov Chain.

RNN Recurrent Neural Network.

SMC Sequential Monte Carlo.

SPMC Semi Pairwise Markov Chain.

SVRNN Semi-supervised Variational Recurrent Neural Network.

TMC Triplet Markov Chain.

VAE Variational AutoEncoder.

VI Variational Inference.

VSL Variational Sequential Labeler.

Introduction générale

Contexte

Cette thèse vise à modéliser des données séquentielles à travers l'utilisation de modèles probabilistes à variables latentes et paramétrés par des architectures de type réseaux de neurones profonds. Notre objectif est de développer des modèles dynamiques capables de capturer des dynamiques temporelles complexes inhérentes aux données séquentielles tout en étant applicables dans des domaines variés tels que la classification, la prédiction et la génération de données pour n'importe quel type de données séquentielles.

Notre approche se concentre sur plusieurs problématiques liés à la modélisation de ce type de données, chacune étant détaillée dans un chapitre de ce manuscrit. Dans un premier temps, nous balayons les principes fondamentaux de l'apprentissage profond et de l'estimation bayésienne. Par la suite, nous nous focalissons sur la modélisation de données séquentielles par des modèles de Markov cachés qui constitueront le socle commun des modèles génératifs développés par la suite. Plus précisément, notre travail s'intéresse au problème de la classification (bayésienne) séquentielle de séries temporelles dans différents contextes : supervisé (les données observées sont étiquetées) ; semi-supervisé (les données sont partiellement étiquetées) ; et enfin non supervisés (aucune étiquette n'est disponible). Pour cela, la combinaison de réseaux de neurones profonds avec des modèles probabilistes markoviens vise à améliorer le pouvoir génératif des modélisations plus classiques mais pose de nombreux défis du point de vue de l'inférence bayésienne : estimation d'un grand nombre de paramètres, estimation de lois à postériori et interprétabilité de certaines

variables cachées (les labels). En plus de proposer une solution pour chacun de ces problèmes, nous nous intéressons également à des approches novatrices pour relever des défis spécifiques en imagerie médicale posés par le Groupe Européen de Recherche sur les Prothèses Appliquées à la Chirurgie Vasculaire (GEPROMED).

Plan

Notre manuscrit est organisé en 5 chapitres.

Le chapitre 1 consiste en une introduction technique dans lequel nous discutons du principe de l'apprentissage profond et de l'estimation bayésienne. Nous y introduisons également des modèles Markoviens pour le traitement des données temporelles.

Le chapitre 2 propose s'intéresse aux chaînes de Markov génératives, en se concentrant spécifiquement sur les chaînes de Markov couples (PMCs). Nous montrons que ce modèle propose un cadre unificateur pour les modèles de Markov cachés ainsi que les récentes architectures de type « réseaux de neurones récurrents stochastiques ». Nous proposons une paramétrisation de ces modèles basée sur des réseaux de neurones profonds et nous détaillons des méthodes d'estimation paramétriques basées sur l'adaptation de l'inférence variationnelle au cas séquentiel. Nous mettons en évidence le pouvoir génératif de ces nouveaux modèles, tant d'un point de vue expérimental que théorique.

Le chapitre 3 vise à utiliser les modèles précédemment développés pour le problème de la classification séquentielle de données. Dans la mesure où le cas supervisé ne présente pas de difficultés supplémentaires par rapport aux techniques mises en place dans le Chapitre 2, nous nous intéressons au cas où les étiquettes/labels associés aux données ne sont que partiellement accessibles. Cette contrainte nous amène à revoir les méthodes d'inférence variationnelle précédemment discutées et à étendre nos modèles de manière à pouvoir prendre en compte deux types de variables cachées : les variables latentes du modèles et les labels non accessibles que l'on cherche à retrouver. Pour cela, nous faisons appel aux modèles de Markov triplet. Notre approche est validée par des simulations numériques portant sur le problème de segmentation d'images binaires en contexte semi-supervisé.

Le chapitre 4 étend le problème au cas non supervisé. L'application directe des méthodes précédentes peut conduire à l'apprentissage de modèles probabilistes dans lesquels la variable étiquette/label n'est pas interprétable

physiquement (ex : classe blanc/noir associée à un pixel en niveau de gris), en particulier dans des modèles reposant sur un grand nombre de paramètres. Pour résoudre ce problème, nous proposons des méthodes d'estimation ad-hoc visant à prendre en compte cette contrainte d'interprétabilité. Pour ce faire, nous commençons avec des modèles de Markov couple visant à modéliser le couple observation/label, puis nous réintroduisons dans un second temps une troisième variable latente continue visant à complexifier la loi du couple précédent. Les apports de nos modèles couple/triplet, paramétrisés par des architectures profondes, ainsi que de nos algorithmes d'estimation paramétrique sont évalués sur différentes tâches telles que la segmentation d'images biomédicales ou la reconnaissance d'activités humaines.

Enfin, le chapitre 5 donne quelques perspectives sur les outils développés précédemment pour des problématiques relatives aux données manipulées par le GEPRMED. Nous y décrivons quelques problématiques liées aux images médicales acquises dans un cadre préopératoire, présentons des résultats préliminaires et proposons une feuille de route pour s'attaquer aux différents défis restant.

Finalement, le manuscrit s'achève par un résumé des résultats ainsi qu'une discussion sur les orientations futures pour l'exploration et l'application des résultats obtenus.

General introduction

Context

This thesis explores and models sequential data through the application of various probabilistic models with latent variables, complemented by deep neural networks. The motivation for this research is the development of dynamic models that adeptly capture the complex temporal dynamics inherent in sequential data. Designed to be versatile and adaptable, these models aim to be applicable across domains including classification, prediction, and data generation, and adaptable to diverse data types. The research focuses on several key areas, each detailed in its respective chapter. Initially, the fundamental principles of deep learning, and Bayesian estimation are introduced. Sequential data modeling is then explored, emphasizing the Markov chain models, which set the stage for the generative models discussed in subsequent chapters. In particular, the research delves into the sequential Bayesian classification of data in supervised, semi-supervised, and unsupervised contexts. The integration of deep neural networks with well-established probabilistic models is a key strategic aspect of this research, leveraging the strengths of both approaches to address complex sequential data problems more effectively. This integration leverages the capabilities of deep neural networks to capture complex nonlinear relationships, significantly improving the applicability and performance of the models.

In addition to our contributions, this thesis also proposes novel approaches to address specific challenges posed by the Groupe Européen de Recherche sur les Prothèses Appliquées à la Chirurgie Vasculaire (GEPROMED). These proposed solutions reflect the practical and possible impactful application of this research, demonstrating its potential contribution to the field of vascular surgery.

Neural networks

Neural Networks (NNs) are foundational in machine learning, used for tasks like classification, regression, and clustering. Their strength lies in representation learning, the ability to discern a data representation that simplifies model building. Structurally, NNs are composed of layers of (artificial) neurons, where each neuron generates an output that is a non-linear function of a linear combination of its inputs. This architectural feature allows NNs to model complex and non-linear relationships within data. The power of NNs is fundamentally based on universal approximation theorems (Cybenko, 1989; Hornik, 1991; Pinkus, 1999; Lu et al., 2017; Liang & Srikant, 2016), which affirms its ability to approximate any continuous multivariable function. This theoretical basis is fundamental to their versatility and adaptability in a variety of problem domains.

Deep Neural Networks (DNNs), characterized by their multiple hidden layers, further extend this capability. Unlike traditional NNs, DNNs have a significantly larger number of layers, which allows them to learn more complex data representations. DNNs are particularly well suited for understanding intricate data patterns, which has led to state-of-the-art performance in areas like speech recognition (Deng et al., 2013; Chan et al., 2016; Abdel-Hamid et al., 2013; Nassif et al., 2019), image classification (Huang, 2023), image recognition (Fu et al., 2017; Traore et al., 2018; Zheng et al., 2017), and natural language processing (Li, 2018; Collobert & Weston, 2008; Goldberg, 2017). Their depth, that is, the number of hidden layers, enables deeper learning of data features at various levels of abstraction, making (deep) NNs particularly well suited for our context, sequential data modeling.

Generative models

Generative models are designed to capture the underlying distribution of data, allowing them to generate new data points similar to those observed. These models are fundamental in fields such as image and speech recognition and natural language processing, where it is essential to understand and reproduce the complexity of natural data. The range of generative models spans from classical probabilistic models to the more recent deep generative models.

Classical generative models, such as Gaussian Mixture Models (GMMs) and Hidden Markov Chain (HMC) models, have been fundamental in statistical modeling, providing a solid foundation for understanding data distribu-

tions and dependencies (Harshvardhan et al., 2020). On the other hand, deep generative models, such as Generative Adversarial Networks (GANs) (Goodfellow et al., 2020), and Variational AutoEncoder (VAE) (Kingma & Welling, 2014), represent a more recent paradigm that integrates the power of deep learning. Both classical and deep generative models continue to evolve, driven by advancements in computational power and algorithmic innovations, further expanding their applications and capabilities in various domains.

VAE and Variational Inference

VAEs integrate probabilistic approaches with neural networks, allowing for the generation of complex data structures with variability and flexibility. They use latent variables to model complex, high-dimensional data structures in a way that classical models cannot efficiently capture. These models are characterized by their parametric nature, where parameters are usually determined by Maximum Likelihood (ML) estimation. However, in VAEs, these parameters are often derived from deep neural networks, which adds a layer of complexity to the learning process. Given the complexity of VAEs, the likelihood function of these models is often intractable. This difficulty makes direct likelihood maximization impractical or even impossible. To address this problem, Variational Inference (VI) (Jaakkola & Jordan, 2000; Blei et al., 2017) is employed. VI offers a powerful approach to approximate the intractable likelihood, allowing VAEs to be trained and used efficiently in a variety of applications (An & Cho, 2015; Pu et al., 2016; Xu et al., 2017; Chira et al., 2022).

Despite their ability to capture complex data patterns, VAEs often take a fundamentally static perspective. They typically process each data point independently, without considering the temporal or sequential dynamics characteristic of many real-world datasets. This limitation is especially evident in contexts involving time series, video, or text, where the inherent sequence aspect of the data is crucial.

Probabilistic models

Popular probabilistic models such as HMC (Rabiner, 1989), Pairwise Markov Chain (PMC) (Pieczynski, 2003; Derrode & Pieczynski, 2004), and Triplet Markov Chain (TMC) (Pieczynski, 2002; Pieczynski & Desbouvries, 2005) models, are capable of capturing temporal dependencies and latent factors in sequential data. Each of these models provides a fundamental framework for processing sequential data, offering unique advantages and posing distinct challenges.

HMC models are widely used to model sequences with both hidden and observed variables. The applications of HMC models are diverse, including natural language processing for tasks such as part-of-speech labeling; computer vision for image segmentation; bioinformatics for genetic sequence analysis (Rabiner, 1989; Gales et al., 2008; Yoon, 2009; Li et al., 2021a; Kupiec, 1992; Paul et al., 2015), etc. PMCs and TMCs extend the fundamental principles of HMCs. They aim to relax some underlying assumptions of HMCs by extending the direct dependencies between random variables or by incorporating an additional third latent process. The assumptions inherent in each of these models are fundamental. Not only do they shape the structure of the model, but they also define the nature of the relationships between variables, thus simplifying the inference and learning processes. The adaptability of these models to different types of data and their ability to capture complex dependencies make them particularly well suited for sequential data modeling.

The parameter estimation is usually performed by maximizing the likelihood function with respect to the parameters. However, when dealing with sequential data, the likelihood function can be intractable. Depending on the model structure, this estimator can be approximated by VI methods (Jaakkola & Jordan, 2000; Blei et al., 2017) or by the Expectation-Maximization (EM) algorithm (Dempster et al., 1977).

Sequential Bayesian classification

Classification is a fundamental task in machine learning, and Bayesian classification is a widely adopted approach to this challenge. In this approach, the main goal is to estimate the posterior distribution of classes (labels) given the observations. This task takes on additional complexity in the context of sequential data, where the observations are a sequence of random variables, and each observation is associated with a corresponding label. The estimation

of these labels from the observations depends on the posterior distribution, which is usually unknown and can be estimated using a parametric model. This model selection process involves choosing a suitable generative model and a learning algorithm to estimate the model parameters. Markov chain models, such as HMC, PMC and TMC models, are particularly suitable for modeling sequential data with both hidden and observed variables.

In the context of sequential classification, the learning process is influenced by the availability of labels associated with the observations. In supervised scenarios, where labels are fully observed, learning involves estimating model parameters using both observations and labels. While in semi-supervised contexts, where only a subset of labels is available, the challenge is to estimate the parameters from the observations and this partial set of labels. Finally, in unsupervised learning, no observed labels are available, then parameter estimation must be performed from the observations alone. Each of these learning contexts presents unique challenges and requires specialized methodologies to address them effectively.

Colaboration with the GEPROMED

GEPROMED¹ is a non-profit organization founded in 1993. The organization emerged from the collaborative vision of Pr. Nabil Chakfé, a vascular surgeon in Strasbourg, France, and Pr. Bernard Durand, an expert in the mechanics of flexible materials. Their primary goal was to investigate and understand the complications associated with vascular prostheses, particularly focusing on the phenomena of tearing and rupture observed post-implantation in patients. GEPROMED is dedicated to promoting specialized learning methods, and continuous quality improvement and ensuring patient safety in the field of vascular surgery. A key area of focus for the organization is the advancement of image processing techniques in vascular surgery. Previous research ([Gangloff, 2020](#)) has shown that deep learning methods, and probabilistic models are very promising for addressing these challenges. For example, on medical image segmentation tasks, probabilistic and deep learning methods have been shown good performance.

Medical image segmentation, a critical task in this field, involves identifying regions of interest within images. These identified regions are crucial for diagnosis, treatment planning and guidance of surgical procedures. The overall goal is to develop automated approaches that can be broadly applied

¹<https://gepromed.com/en/aboutUs>

to various types of medical images, thereby improving the efficiency of diagnostics, and medical interventions.

Contributions

This thesis aims at proposing innovative methodologies that bridge the gap between classical probabilistic models based on Markov Chains and deep neural networks, specifically adapted to sequential data modeling. The results obtained have been presented at different national and international conferences and published in peer-reviewed journals. Our contributions are detailed in the following sections and are based on the following publications:

- [Morales & Petetin \(2021\)](#): Variational Bayesian inference for pairwise Markov models, In *2021 IEEE Statistical Signal Processing Workshop (SSP) (pp. 251-255)*, IEEE.
- [Gangloff, Morales, & Petetin \(2021\)](#): A general parametrization framework for pairwise Markov models: An application to unsupervised image segmentation, In *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP) (pp. 1-6)*, IEEE.
- [Morales & Petetin \(2022\)](#): Pairwise Markov Chains as Generative Models, *Colloque GRETSI 2022*, (pp. 649–652).
- [Gangloff, Morales, & Petetin \(2022\)](#): Chaînes de Markov cachées à bruit généralisé, *Colloque GRETSI 2022*, (pp. 17–20).
- [Gangloff, Morales, & Petetin \(2023\)](#): Deep parameterizations of pairwise and triplet Markov models for unsupervised classification of sequential data, *Computational Statistics & Data Analysis*, *180*, 107663.
- [Morales & Petetin \(2023\)](#): A Probabilistic Semi-Supervised Approach with Triplet Markov Chains, In *2023 IEEE 33rd International Workshop on Machine Learning for Signal Processing (MLSP) (pp. 1-6)*, IEEE

This thesis also includes preliminary but promising results in the area of low-resolution medical image segmentation. This area of research, while still a work in progress, demonstrates the potential of our methodologies to make significant advances in medical image analysis. Initial results are encouraging and lay the groundwork for further exploration and refinement. These efforts are currently continuing with the goal of culminating in a future publication.

Outline

This thesis is structured to introduce and explore methodologies in sequential data modeling, particularly through deep learning and Bayesian estimation techniques. It provides a comprehensive examination of both theoretical and practical aspects of generative models and their applications in supervised, semi-supervised and unsupervised classification tasks.

The thesis comprises five chapters. Chapter 1 offers a technical introduction, discussing the principles of deep learning, Bayesian estimation, and sequential data modeling with Markov chains. This chapter sets the foundation by covering topics such as maximum likelihood estimation with VI, and posterior distribution estimation.

Chapter 2 delves into Generative Markov Chains, specifically focusing on PMCs as a unified model. It details parameter estimation methods, including general parametrization and VI for PMCs, and presents experiments and results that highlight the generative power of these models.

Chapter 3 extends the discussion to Generalized Hidden Markov Models for semi-supervised classification. It introduces the problem of semi-supervised estimation in TMCs, explores ELBO for semi-supervised learning, and describes the learning process. The chapter also includes experiments comparing deep TMCs with existing models, and semi-supervised binary image segmentation.

Chapter 4 addresses Markov Chains for unsupervised classification, detailing Bayesian inference for PMCs and deep PMCs for unsupervised classification. It further explores TMCs for unsupervised classification, including VI, and deep TMCs. This chapter also presents simulations and experiments on real datasets, such as unsupervised segmentation of biomedical images and clustering for human activity recognition.

Chapter 5 shows a workflow adapted to data provided by the GEPROMED group, and future perspectives that can merge the models presented in the first chapters. Finally, the thesis concludes with a summary of findings, a discussion on the implications of the research, and future directions for exploration and application.

CHAPTER 1

Technical introduction

Contents

1.1	Deep learning	14
1.1.1	Fundamental principle	14
1.1.2	Deep neural networks architectures for sequential data .	14
1.2	Bayesian estimation	16
1.2.1	Approximated Maximum Likelihood estimation with Variational Inference	17
1.2.2	Posterior distribution	22
1.2.3	Discussion	23
1.3	Sequential data modeling	24
1.3.1	Hidden Markov chains	24
1.3.2	Pairwise Markov chains	25
1.3.3	Sequential generative models for Bayesian classification	26
1.3.4	Organization of the thesis	26

1.1. Deep learning

1.1.1 Fundamental principle

DNNs have significantly gained popularity in recent years due to their remarkable performance in various tasks such as speech recognition (Deng et al., 2013; Chan et al., 2016; Abdel-Hamid et al., 2013), image recognition (Fu et al., 2017; Traore et al., 2018; Zheng et al., 2017), natural language processing (Collobert & Weston, 2008; Goldberg, 2017). Mathematically, a DNN is a parameterized vector-valued function $f_\theta(x)$, $x \in \mathbb{R}^{d_x}$, constructed through the sequential and alternating composition of linear and non-linear functions. If vector x' represents the input to a specific hidden layer, the scalar output of a neuron is computed as $\sigma(wx' + b)$, where wx' is the dot product of a vector of weights w and x' . Here b represents the bias, and $\sigma(\cdot)$ is the (non-linear) activation function. Common activation functions include sigmoid ($\text{sigm}(x) = \frac{1}{1+e^{-x}}$), hyperbolic tangent ($\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$), and Rectified Linear Unit ($\text{ReLU}=\max(0, x)$).

The set of parameters θ of a DNN, which includes all weights and biases, enables these networks to act as universal approximators, theoretically capable of approximating any vector-valued function $f(x)$ under some assumptions (Cybenko, 1989; Hornik, 1991; Pinkus, 1999; Lu et al., 2017; Liang & Srikant, 2016). The estimation of θ relies on the observation that the gradient of f_θ w.r.t. θ can be exactly computed with the backpropagation algorithm, a foundational technique for learning in neural networks, as described by Rumelhart et al. (1985); Hecht-Nielsen (1992). This efficiency is because the algorithm takes advantage of the chain rule of computation to decompose the global gradient computation into a series of simpler local gradient computations along the layers of the network. For example, in a classification problem of an observation x , the function $f_\theta(x)$ aims at approximating the conditional probability $P(Y = y | x)$ for all y in the set $\Omega = \{\omega_1, \dots, \omega_C\}$, where C is the number of classes. Provided that we have access to a labeled training dataset $\mathcal{D} = \{(x^i, y^i)\}_{i=1}^n$, it is possible to minimize a loss function $\mathcal{L}(\mathcal{D})$, e.g. the cross-entropy loss, with a gradient descent approach (Ruder, 2016).

1.1.2 Deep neural networks architectures for sequential data

While classic DNNs have demonstrated significant versatility and power in various domains, their conventional architectures may not be optimal for processing sequential data, such as time series, audio signals, or textual content. Different types of neural networks have been developed to address this issue, such as Recurrent Neural Network (RNN) (Fausett, 1994; Medsker & Jain,

2001; Mikolov et al., 2015). RNNs are architected to process sequential information, where dependencies exist across temporal intervals. This capability is achieved by incorporating recurrent connections within the network, allowing information to be retained across time steps. The design allows an RNN to not only process the current input, but also to use the context provided by previously received inputs. For instance, when predicting the next word in a sentence, the RNN considers the sequence of words that preceded it to make a more accurate prediction.

In contrast to classic DNNs, the parameters θ in an RNN are shared across different time steps, rather than learning a separate set of parameters for each moment in time. This sharing reduces the model’s complexity and enables the RNN to generalize across sequences of different lengths. At each time step t , the hidden state $h_t \in \mathbb{R}^{d_h}$ of the RNN is updated based on the current input x_t and the previous hidden state h_{t-1} . The RNN’s output o_t at time t is computed based on the hidden state h_t . This model can be expressed as follows:

$$h_t = f_\theta(h_{t-1}, x_t), \text{ for all } t \in \mathbb{N}, \quad (1.1)$$

$$o_t = g_\theta(h_t), \text{ for all } t \in \mathbb{N}. \quad (1.2)$$

Here f_θ and g_θ are parameterized activation functions, *e.g.* neural networks. Figure 1.1 illustrates the graphical representation of an RNN. This architectural design enables the RNN to effectively handle data where current decisions depend on past information, such as time series data, speech, or text.

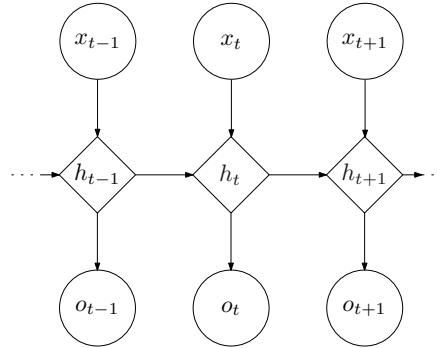


Figure 1.1: Graphical representation of a Recurrent Neural Network. The recurrent connections between the nodes highlight the network’s ability to process sequences of data by maintaining a ‘memory’ of previous inputs through the hidden states.

Remark 1.1.1. The output o_t of an RNN has a dual predictive capability. For instance, in a stock market analysis application, it could predict the label y_t categorizing market trends or forecast future stock prices x_{t+1} . This versatility makes RNNs a tool of choice for various predictive modeling tasks.

Despite their advantages, RNNs are not without challenges. They are particularly prone to issues of vanishing and exploding gradients, especially when dealing with longer sequences. To overcome these problems, architectures such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have been developed. LSTMs and GRUs incorporate mechanisms that regulate the flow of information, allowing the network to retain or forget information selectively. This capability significantly improves their performance on tasks involving long sequences or where the temporal gap between relevant information is large. While these networks are beyond the scope of this thesis, interested readers can refer to [Sherstinsky \(2020\)](#); [Hochreiter & Schmidhuber \(1997\)](#); [Chung et al. \(2014\)](#) for more details.

1.2. Bayesian estimation

In the context of deep learning, we have seen how common DNNs, including RNNs, can be used to approximate functions for various tasks. While these models are powerful, they often do not directly account for the uncertainty inherent in real-world data. Bayesian estimation extends the predictive power by incorporating a probabilistic framework capable of capturing not just the observed data but also the underlying latent structures, such as the intrinsic features of an image that are not immediately observable.

In Bayesian Estimation, we deal with the observed random variable (r.v.) $x \in \mathbb{R}^{d_x}$ and the latent (unobserved or hidden) r.v. $z \in \mathbb{R}^{d_z}$, each playing a distinct role in the modeling process. Throughout this thesis, we do not distinguish between random variables and their realizations. Our interest, which will be explained in more detail later, lies in calculating the posterior distribution

$$p(z|x) = \frac{p(z, x)}{p(x)},$$

which offers insights into the latent variables given the observed data. However, the direct computation of $p(x, z)$ is often impractical, whether due to the high-dimensional nature of the data, which leads to computational complexity, or the unknown distributional characteristics. Thus, we can start by parameterizing the joint distribution $p(x, z)$ with a set of parameters θ ,

leading to the model $p_\theta(x, z)$. Once a class of distributions p_θ has been chosen, the objective is to estimate the parameter θ from a realization x in an unsupervised way, that is to say without observing z . A common approach for parameter estimation is the Maximum-Likelihood (ML) estimator, $\hat{\theta}^{\text{ML}} = \arg \max_{\theta} p_\theta(x) = \arg \max_{\theta} \int p_\theta(z, x) dz$, due to its statistical properties (Huber, 1967; White, 1982). However, the ML estimator may not be tractable since $p_\theta(x)$ is not necessarily known in a closed form. According to the structure of $p_\theta(z, x)$, the ML estimator can be approximated with a gradient ascent method on the likelihood function, the EM algorithm (Dempster et al., 1977) or a Variational Inference algorithm (Jaakkola & Jordan, 2000; Blei et al., 2017).

In summary, Bayesian estimation offers a probabilistic approach to modeling by considering both observed and latent variables. This method provides a comprehensive framework for understanding the underlying uncertainties in data. In practice, computing the posterior distribution $p_\theta(z|x)$ directly is often infeasible due to high-dimensional data or unknown distribution characteristics. VI provides a robust alternative by approximating the true posterior with a simpler, and parameterized distribution. This approach is discussed in the following subsection.

1.2.1 Approximated Maximum Likelihood estimation with Variational Inference

Given independent and identically distributed observations $\{x^i\}_{i=1}^M$, a direct computation of $\arg \max_{\theta} p_\theta(x)$ becomes impractical except in simpler cases, such as when $p_\theta(x)$ is directly available (*e.g.* when z is discrete or in linear and Gaussian scenarios). VI offers a flexible and scalable alternative for more complex models where such straightforward calculations are not feasible. VI is introduced as a method for approximate inference in models where the computation of the posterior distribution is complex or intractable. Unlike Maximum Likelihood estimation, which focuses on finding parameter values that maximize the likelihood of the observed data, VI approaches the problem by approximating the true posterior distribution $p_\theta(z|x)$ with a simpler, parameterized distribution $q_\phi(z|x)$ (*see e.g.* Blei et al. (2017) for a detailed introduction).

This method is the cornerstone of the Bayesian inference algorithms we propose in this thesis, for our (highly) parameterized models. Let us consider the general problem of computing or approximating a posterior distribution $p_\theta(z|x) \propto p_\theta(z, x)$ known up to a constant when x is observed and z is latent. VI relies on a parameterized distribution $q_\phi(z|x)$ that is optimized to fit the

posterior distribution $p(z|x)$ by minimizing the Kullback-Leibler Divergence (KLD) between $q_\phi(z|x)$ and $p_\theta(z|x)$, *i.e.*

$$\begin{aligned} \text{D}_{\text{KL}}(q_\phi, p_\theta) &= \int q_\phi(z|x) \log \left(\frac{q_\phi(z|x)}{p_\theta(z|x)} \right) dz \geq 0, \\ &= \int q_\phi(z|x) \log \left(\frac{q_\phi(z|x)}{p_\theta(z, x)} \right) dz + \log(p_\theta(x)) \end{aligned} \quad (1.3)$$

w.r.t. θ . The choice of the variational distribution $q_\phi(z|x)$ is critical, as the first term on the right-hand side of the above equation must be computable or easily approximated, and subsequently optimized with respect to ϕ . A popular choice of variational distribution is the mean-field approximation (Bishop & Nasrabadi, 2006) where the variational components of $z = (z_1, \dots, z_{d_z})$ are independent given x and one set of parameters ϕ_i is associated to each component z_i , *i.e.* $q_\phi(z|x) = \prod_{i=1}^{d_x} q_{\phi_i}(z_i|x)$ and $\phi = (\phi_1, \dots, \phi_{d_z})$.

This approach also provides a parameter estimation method when some parameters of the original model p_θ are unknown. Indeed, we deduce from (1.3) that

$$\log p_\theta(x) \geq - \int q_\phi(z|x) \log \left(\frac{q_\phi(z|x)}{p_\theta(z, x)} \right) dz = \mathcal{Q}(\theta, \phi), \quad (1.4)$$

where equality holds when $q_\phi(z|x) = p_\theta(z|x)$.

Computing the so-called Evidence Lower Bound (ELBO) $\mathcal{Q}(\theta, \phi)$ and next maximizing it w.r.t. (θ, ϕ) leads to a maximization of a lower bound of the log-likelihood $\log p_\theta(x)$. The resulting variational EM algorithm (Tzikas et al., 2008) is an alternative to the EM algorithm (Dempster et al., 1977) when the original posterior $p_\theta(z|x)$ is not available. Our objective is to maximize the ELBO $\mathcal{Q}(\theta, \phi)$ as defined in Equation (1.4), w.r.t. the parameters (θ, ϕ) .

To address scenarios whereThe, we employ Monte Carlo estimators, which provide a practical solution for obtaining unbiased gradient estimates using statistical sampling techniques. For continuous latent variables, we use the reparameterization trick (Kingma & Welling, 2014), which facilitates obtaining an unbiased estimator for the gradient of the ELBO. When z is discrete, we use the Gumbel-Softmax (G-S) trick (Maddison et al., 2017; Jang et al., 2017). These techniques are detailed below

Continuous latent variables: The idea of the reparameterization trick is to rewrite the random variable z as a deterministic differentiable function of a random variable ϵ , that is independent of ϕ (Kingma & Welling, 2014). In other words, we want to rewrite the random variable z as

$$z = g(\epsilon, \phi, x), \quad (1.5)$$

where ϵ is independent of ϕ and x . The expectations w.r.t $q_\phi(z|x)$ can be then rewritten as

$$\mathbb{E}_{q_\phi(z|x)}(f(z)) = \mathbb{E}_{p(\epsilon)}(g(\epsilon, \phi, x)),$$

and the gradients of the previous expectation w.r.t ϕ ,

$$\nabla_\phi \mathbb{E}_{q_\phi(z|x)}(f(z)) = \nabla_\phi \mathbb{E}_{p(\epsilon)} f(g(\epsilon, \phi, x)),$$

can be now estimated with a Monte Carlo estimator. We now obtain unbiased estimates of the gradient of the ELBO w.r.t ϕ and θ . The reparameterization trick is illustrated in Figure 1.2 for the case of continuous latent variables.

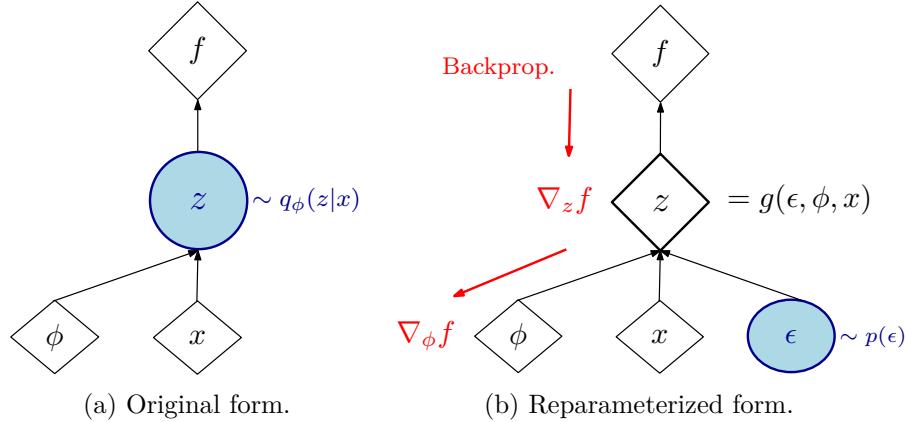


Figure 1.2: Illustration of the reparameterization trick. In the original form, we cannot compute the gradient of f w.r.t ϕ . While in the reparameterized form, gradient of f w.r.t ϕ is easily computed. Diamonds indicate no stochasticity, while blue circles highlight its presence. Figure based on ([Kingma & Welling, 2014](#)).

Example 1.2.1. We present the Variational AutoEncoder model with a continuous latent variable, where the joint distribution $p_\theta(x, z)$ is factorized into the prior distribution $p_\theta(z)$ and the conditional distribution $p_\theta(x|z)$, also called the probabilistic decoder. Here, the set of parameters θ could be the output of (deep) neural networks, which are estimated from a dataset with the assumption that the data points are *i.i.d.*. The variational distribution (probabilistic encoder) $q_\phi(z|x)$ is a multivariate Gaussian distribution with diagonal covariance matrix,

$$q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \text{diag}(\sigma_\phi(x))),$$

where $\mu_\phi(x)$ and $\sigma_\phi(x)$ are the outputs of neural networks. Next, a sample $z^{(m)}$ is drawn from $q_\phi(z|x)$ with the reparameterization trick,

$$z^{(m)} = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon^{(m)} \text{ for all } m = 1, \dots, M,$$

where $\epsilon^{(m)}$ is a sample from the standard Gaussian distribution and \odot denotes the element-wise product.

The ELBO (1.4) is then approximated with the Monte Carlo estimator,

$$\mathcal{Q}(\theta, \phi) \approx -\frac{1}{M} \sum_{m=1}^M \log \frac{q_\phi(z^{(m)}|x)}{p_\theta(x|z^{(m)})p_\theta(z^{(m)})}.$$

After this, the parameters (θ, ϕ) are estimated by maximizing the previous expression w.r.t (θ, ϕ) with a gradient ascent algorithm. Figure 1.3 illustrates the VAE model.

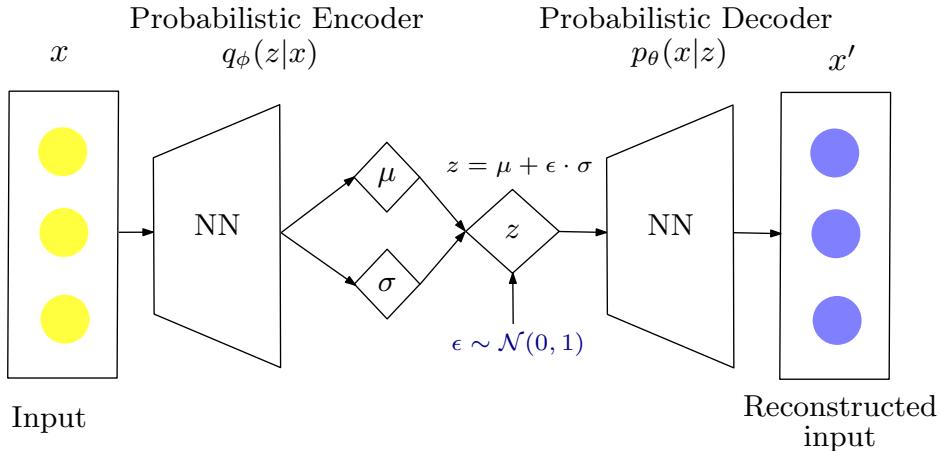


Figure 1.3: Illustration of a Gaussian-Variational AutoEncoder model.

Discrete latent variables: Let π_c denote the probability of the class c , with the condition that $\sum_{c=1}^C \pi_c = 1$. The Gumbel-Max trick (Gumbel, 1948; Maddison et al., 2014) facilitates sampling from this distribution by adding *i.i.d.* Gumbel (noise) samples to the log-probabilities $\log \pi_c$. The class corresponding to the highest resulting value is then selected as the sample, *i.e.* $k = \arg \max_{c=1, \dots, C} (\log \pi_c + G_c)$. Although the Gumbel-Max trick facilitates sampling, it does not inherently allow for gradient-based optimization because the argmax operation is not differentiable. To address this limitation,

the Gumbel-Softmax trick (Maddison et al., 2017; Jang et al., 2017) is used, which introduces a differentiable approximation to the categorical distribution. The G-S trick involves the softmax function, a continuous and differentiable approximation of the arg max operation. It begins by expressing the latent vector z as a one-hot vector, *i.e.* $z \in \{0, 1\}^C$, where C is the number of classes. This generates a C -dimensional vector z^{G-S} within the range $[0, 1]^C$, defined as

$$z_c^{G-S} = \frac{\exp((\log \pi_c + G_c)/\tau)}{\sum_j^C \exp((\log \pi_j + G_j)/\tau)}, \text{ for all } c = 1, \dots, C,$$

where τ is the temperature parameter, and G_c is a Gumbel sample drawn from $\text{Gumbel}(0, 1)$. As the softmax temperature τ approaches 0, samples from the G-S distribution become one-hot and the G-S distribution becomes identical to the categorical distribution (more details in Maddison et al. (2017)). The Gumbel-Max and Gumbel-Softmax tricks are illustrated in Figure 1.4 with $C = 3$.

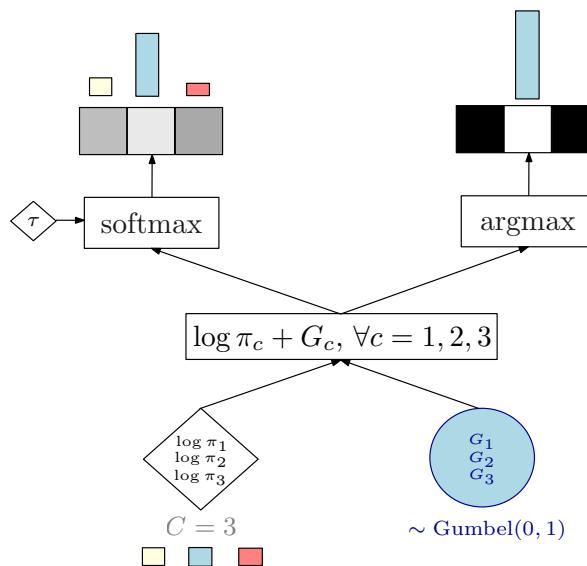


Figure 1.4: Illustration of the Gumbel-Max and Gumbel-Softmax tricks with $C = 3$. The blue circle represents the Gumbel samples drawn from $\text{Gumbel}(0, 1)$. The result of the Gumbel-Max trick is the index c of the maximum value and the result of the Gumbel-Softmax trick is a C -dimensional vector z^{G-S} with values in $[0, 1]^C$, which is a continuous, differentiable approximation of the arg max.

Remark 1.2.1. In machine learning models that require discrete decision-making it is crucial to maintain consistency between the training and evaluation phases. The Straight Through Gumbel-Softmax (Maddison et al., 2017) technique addresses this issue by using the G-S distribution for sampling during the forward step, followed immediately by an argmax operation to discretize the output into one-shot vectors. This ensures that the behavior of the model during training matches its evaluation. In the backward step, the original smooth probabilities from the G-S distribution are used to compute gradients, thus maintaining differentiability and allowing efficient backpropagation. This technique bridges the gap between the need for discrete outputs and the advantages of gradient-based optimization.

1.2.2 Posterior distribution

In Bayesian Estimation, a fundamental objective is to compute the posterior distribution $p(z|x)$, which provides insights into the hidden or latent variable z given the observed data x . However, as established earlier, direct computation of this posterior is often infeasible due to the unknown or complex nature of the joint distribution $p(z,x)$. To address this, we can use approximation techniques such as variational distributions (Blei et al., 2017) and normalized importance sampling (Doucet et al., 2001a). As we previously discussed, the variational approach is a powerful tool for approximating the posterior distribution $p_\theta(z|x)$, which involves defining a simpler, and parameterized variational distribution $q_\phi(z|x)$. This distribution, often a tractable distribution such as a Gaussian, is allows for efficient approximation and computation. The variational distribution depends on a set of parameters ϕ that can be optimized by maximizing the ELBO $\mathcal{Q}(\theta, \phi)$ in Equation (1.4) since the maximization is w.r.t (θ, ϕ) .

On the other hand, the normalized importance sampling technique involves selecting a proposal distribution that is easier to sample from, calculating weights for these samples based on the ratio of the posterior to the proposal distribution, and then normalizing these weights to ensure they sum to one, thus transforming them into proper probabilities. We can use the variational distribution as the proposal distribution because we generally choose a variational distribution from which we can sample efficiently due to the optimization of the ELBO. Both the variational distribution and normalized importance sampling offer robust solutions for approximating the posterior distribution in scenarios where direct computation is challenging. By leveraging these methods, we can gain valuable insights into the latent structures of complex models, enhancing our understanding and predictive capabilities in

various applications of Bayesian Estimation.

1.2.3 Discussion

This discussion synthesizes the concepts introduced in previous sections, highlighting the relationship between Deep Learning and Bayesian Estimation methodologies and their applications in significant areas like Generative Modeling and Unsupervised Bayesian Estimation/Classification.

Generative models: These models are mainly concerned with modeling the data distribution $p_\theta(x)$ or sampling new data points according to this distribution. The ability of generative models to learn complex distributions and generate new data is one of their main advantages. The connection with Bayesian estimation becomes evident when we consider that $p_\theta(x)$ can be expressed by the joint distribution and the posterior distribution as $p_\theta(x) = \frac{p_\theta(x,z)}{p_\theta(z|x)}$. Here, the latent variables z play a crucial role in generative models and are often used to capture the underlying structure of the data.

On the other hand, a popular generative model is the Variational AutoEncoder, which combines the principles of deep learning and Bayesian estimation to generate new data samples (see Example 1.2.1). VAEs consist of two key components, an encoder and a decoder. The encoder, a neural network, maps the input data x to a latent representation z , effectively approximating the posterior $p_\theta(z|x)$. Next, the decoder, another neural network, reconstructs the data x from the latent representation z , approximating $p_\theta(x|z)$. The integration of VAE into the broader context of deep learning highlights the compatibility and complementarity of these frameworks. VAE provides a bridge the representational capabilities of neural networks, and the probabilistic modeling capabilities of Bayesian methods. This combination allows the creation of powerful generative models that not only generate plausible and diverse data samples, but also provide information about the underlying data distribution and the latent structures present in it.

Unsupervised Bayesian classification: In the context of unsupervised learning, Bayesian estimation methods are adapted to provide insightful solutions. Here, the latent variable z can be redefined by a variable of interest y ($z \leftarrow y$). This adaptation enables the application of Bayesian inference techniques to estimate y from the observed data x , leading to the computation of the posterior distribution $p_\theta(y|x)$. This approach overcomes the limitations of point estimation. Instead of providing a single estimate of y , it provides ac-

cess to the entire posterior distribution of y . This comprehensive perspective is especially valuable in unsupervised scenarios where direct observations of y are not available. However, the performance of this approach depends on the formulation of the model $p_\theta(y, x)$. It is crucial that this model not only captures the relationship between x and y , but also facilitates the interpretability of y in an unsupervised context. We can also consider a model $p_\theta(y, x, z)$, where the latent variable z is introduced to capture the relationship between x and y . This additional latent variable z can help in capturing more complex, underlying relationships within the data that might not be directly observable from x alone.

1.3. Sequential data modeling

We now consider sequential data, building on the foundations presented in the previous sections. Sequential data present unique challenges, particularly when it comes to modeling temporal dependencies and extracting meaningful patterns over time. This discussion leads us to focus on probabilistic models specifically designed for sequential data, such as Hidden Markov Models (HMMs) and their extensions. We denote a sequence of observations as $x_{0:T} = (x_0, x_1, \dots, x_T)$, where T is the length of the sequence. Similarly, we use $z_{0:T} = (z_0, z_1, \dots, z_T)$ to denote a sequence of latent variables.

1.3.1 Hidden Markov chains

HMCs are a class of probabilistic models where the latent process is a Markov chain, and the observations are conditionally independent given the latent process and x_t depends only on z_t . The joint distribution of the sequence of observations and latent variables is given by

$$p_\theta(z_{0:T}, x_{0:T}) \stackrel{\text{HMC}}{=} \underbrace{p_\theta(z_0)}_{p_\theta(z_{0:T})} \underbrace{\prod_{t=1}^T p_\theta(z_t | z_{t-1})}_{p_\theta(z_{0:T} | z_{0:T})} \underbrace{\prod_{t=0}^T p_\theta(x_t | z_t)}_{p_\theta(x_{0:T} | z_{0:T})}, \quad \text{for all } T \in \mathbb{N}. \quad (1.6)$$

When the parameters of the HMC are unknown, they can be estimated from a set of observations that we have at our disposal. The Maximum Likelihood (ML) approach for estimating the parameters of the HMC has been widely theoretically studied in [Douc et al. \(2004\)](#); [Douc & Moulines \(2012\)](#). However, the distribution of the observations $p_\theta(x_{0:T})$ is intractable in general. Therefore, the ML approach is not applicable. The distribution $p_\theta(x_{0:T})$

can be approximated with Sequential Monte Carlo (SMC) methods (Doucet et al., 2001b; Chopin et al., 2020). Nonetheless, the SMC methods are computationally expensive and differentiable approximations to use gradient-based optimization methods could be a problem. This is due to the resampling steps of such algorithms (Kantas et al., 2015). The EM algorithm (Dempster et al., 1977) is also an alternative approach for estimating the parameters of the HMC (1.6) (see Algorithm 9). When the parameters of the HMC are estimated, the predictive distribution $p_\theta(x_{T+1}|x_{0:T})$ can be sequentially computed or approximated. SMC methods can be used to approximate this distribution.

1.3.2 Pairwise Markov chains

In HMCs, the latent process is Markovian, *i.e.* the latent variable z_t depends only on z_{t-1} . It can be relevant to consider latent variables that depend on more than one previous latent variable. It is also valid for the observations x_t , which can depend on more than one previous latent variable. For example, in the case of time series, the observations x_t can depend on the previous observation x_{t-1} and the latent variable z_{t-1} . We introduce the Pairwise Markov Chain (PMC) (Pieczynski, 2003; Derrode & Pieczynski, 2004; Le Cam et al., 2008) model that relax the Markovianity assumption of the HMC. PMCs generalize the HMC by considering the joint process of $\{z_t, x_t\}_{t \in \mathbb{N}}$ as a Markov chain. The joint distribution of the sequence of observations and latent variables is given by

$$p_\theta(z_{0:T}, x_{0:T}) \stackrel{\text{PMC}}{=} p_\theta(z_0, x_0) \prod_{t=1}^T p_\theta(z_t, x_t | z_{t-1}, x_{t-1}), \quad \text{for all } T \in \mathbb{N}. \quad (1.7)$$

The use of such models has been proposed in past contributions, in simpler contexts where the sequence $z_{0:T} \leftarrow y_{0:T}$ represents a series of labels for the sequence of observations $x_{0:T}$. It has been shown that when the PMC model is stationary, it is possible to propose an unsupervised estimation method to estimate jointly θ and y_t from $x_{0:T}$ provided that the distribution of the observation given the hidden states is restricted to a set of classical distributions such as the Gaussian one (Gorynin et al., 2018). Several questions then arise: how can we use the structure of PMCs as generative models for modeling $p_\theta(x_{0:T})$? Can these models be adapted to unsupervised classification scenarios where $p_\theta(y_{0:T}, x_{0:T})$ is parameterized by deep neural networks? In next chapters we will focus on these questions.

1.3.3 Sequential generative models for Bayesian classification

We can introduce an additional level of complexity with Triplet Markov Chains (TMCs) (Pieczynski, 2002; Pieczynski & Desbouvries, 2005) for classification tasks. TMCs provide a refined framework in which we can model not only the sequence of observations $x_{0:T}$ and their associated labels $y_{0:T}$, but also incorporate an auxiliary sequence $z_{0:T}$, enriching the relationships within the data. TMC have been mainly used with a discrete auxiliary sequence $z_{0:T}$ (Gorynin et al., 2018; Lanchantin et al., 2008; Pieczynski, 2007). In this thesis, we will focus on the case where the sequence $z_{0:T}$ is continuous. Thus, we consider the joint distribution $p_\theta(y_{0:T}, x_{0:T}, z_{0:T})$, for all $T \in \mathbb{N}$, given by

$$p_\theta(y_{0:T}, z_{0:T}, x_{0:T}) \stackrel{\text{TMC}}{=} p_\theta(y_0, z_0, x_0) \prod_{t=1}^T p_\theta(y_t, z_t, x_t | y_{t-1}, z_{t-1}, x_{t-1}). \quad (1.8)$$

In a supervised context, the sequence of labels $y_{0:T}$ is known, the TMC can be seen as a PMC with an augmented representation of latent variables, *i.e.* $x_t \leftarrow (y_t, z_t)$, for all $t \in \mathbb{N}$. However, the TMC model becomes more interesting when the sequence $\{y_t\}_{t \in \mathbb{N}}$ corresponds to an unobserved physical process of interest, and $\{z_t\}_{t \in \mathbb{N}}$ is treated as a separate, distinct process. However, the TMC model can be used for semi-supervised and unsupervised classification tasks, where the labels $y_{0:T}$ are unobserved or partially observed. In an unsupervised learning (Lanchantin & Pieczynski, 2004), we have to estimate the parameters of the model which takes into account the interpretability of $y_{0:T}$ and also the different roles of $y_{0:T}$ and $z_{0:T}$. While in a semi-supervised context, the labels $y_{0:T}$ are partially observed, and we look for estimating the missing labels associated to each sequence. In both semi-supervised and unsupervised context, TMCs combined with DNN can provide a more refined approach to Bayesian classification, which is one of the main objectives of this thesis.

1.3.4 Organization of the thesis

In this thesis, we address several fundamental questions that arise from the concepts and models presented. These questions guide the research and structure of the thesis, ensuring a comprehensive exploration of the topics. The key questions include:

- How can we build powerful generative models from PMC models (1.7), and what guarantees do we have on their modeling power?

- In sequential Bayesian classification, when only a subset of labels is observed, how can we estimate the unobserved labels from the observations and the observed labels using a general TMC model (1.8)? Does the supervised case coincide with the previous question?
- How can PMC and TMC models be applied to unsupervised classification tasks, and what are the challenges of ensuring interpretability of hidden random variables in unsupervised classification?
- Can these models be adapted to different scenarios where the distributions (1.7) and (1.8) are parameterized by deep neural networks?
- What adaptations to the VI algorithm are necessary for general parameter estimation of these models?
- What are the challenges in AI for vascular surgery that could be addressed with the techniques presented in this thesis (applied perspective)?

To systematically address these questions, the thesis is structured to first introduce the fundamental principles and challenges of deep learning, followed by a detailed exploration of Bayesian estimation and its application in understanding complex data structures. Later sections focus on models for sequential data such as the PMC and TMC models, underscoring their theoretical foundations and practical implications.

Chapter 2 introduces the PMC model as a generative model that can model complex dependencies between observations and latent variables. We discuss how it can serve as a general model from which different models such as the HMC and the RNN, among others, can be derived as specific cases. Additionally, we present a general parameterization of the PMC model, which includes deep parameterization (DNNs). In the second part, we develop an adapted VI algorithm for general parameter estimation of this model. We also explore the linear and stationary Gaussian PMC model for a theoretical analysis of its generative power.

Chapter 3 is dedicated to the semi-supervised learning problem. We propose a probabilistic approach to deal with sequential Bayesian classification when only a subset of labels is observed. The goal is to estimate the unobserved labels from the observations and the observed labels using a general TMC model (which includes a deep parameterization). We introduce a new adaptation of VI, enabling us to estimate the parameters of the model and the unobserved labels.

Chapter 4 focuses on the unsupervised classification task. We propose PMC and TMC models for estimating unobserved labels associated with a sequence of observations. For each introduced model, an original unsupervised Bayesian estimation method is proposed. In particular, it considers the interpretability of the hidden random variables in terms of classification.

Finally, chapter 5 presents a workflow adapted to the data provided by the GEPFROMED group, and future perspectives that integrate the classical neural network models with a probabilistic approach. We also discuss the potential of the proposed models for the segmentation of medical data.

CHAPTER 2

Generative hidden Markov models

Contents

2.1	Introduction	30
2.2	The pairwise Markov chain as a unified model	30
2.3	Parameter estimation for general PMCs	33
2.3.1	General parameterization of PMCs	33
2.3.2	Variational Inference for PMCs	35
2.4	Experiments and results	37
2.4.1	Model description	37
2.4.2	Results	38
2.5	Generative power of PMCs	42
2.5.1	Linear and stationary Gaussian PMCs	43
2.5.2	Theoretical analysis of PMCs	44
2.6	Conclusions	48

2.1. Introduction

This chapter introduces the PMC as a general generative model that can be used to model complex dependencies between the observations and the latent variables. First, we recall the PMC model (Pieczynski, 2003), and introduce how it can be used as a general model for generative modeling, from which the HMC (Rabiner, 1989), the RNN (Medsker & Jain, 2001), and the Generative Unified Model (GUM) (Salaün et al., 2019) can be derived as particular cases. Moreover, we present a general parameterization of the PMC model, which includes a deep parameterization (DNNs). In the second part, we develop an adapted VI algorithm (Jaakkola & Jordan, 2000; Blei et al., 2017) for a general parameter estimation of this model, which can be applied to any PMC model, linear or not, Gaussian or not. We provide some experimental results, demonstrate PMC as a generative model, and see how it compares to other popular models that use latent variables and DNNs. To conclude, we focus on a particular instance of the PMC model, the linear, and stationary Gaussian PMC, for a theoretical analysis of the generative power of the PMC model. This analysis is based on the expressivity of the PMC, *i.e.* the distribution of the observations generated by the model.

2.2. The pairwise Markov chain as a unified model

We recall the notation introduced in Chapter 1, $x_{0:T} = (x_0, \dots, x_T)$, $x_t \in \mathbb{R}^{d_x}$, and $z_{0:T} = (z_0, \dots, z_T)$, $z_t \in \mathbb{R}^{d_z}$ which are two sequences of observed and latent random variables (r.v.), respectively, of length $T + 1$. This chapter focuses on the case where the latent variables are not interpretable as the labels of the observations. In other words, our interest is a generative model, where the latent variables are just an intermediate step to create a complex distribution of the observations. For that, we recall the joint distribution of the observed and latent variables in the PMC model (1.7), which is given by

$$p_\theta(z_{0:T}, x_{0:T}) = p_\theta(z_0, x_0) \prod_{t=1}^T p_\theta(z_t, x_t | z_{t-1}, x_{t-1}), \quad \text{for all } T \in \mathbb{N}.$$

From a modeling point of view, the choice of the transition distribution $p_\theta(z_t, x_t | z_{t-1}, x_{t-1})$ is a thorny problem. The transition distribution can be factorized in different ways. The choice of the factorization depends on the specific problem and the underlying assumptions about the dependencies between variables. In practice, we need to choose a transition distribution that

has an impact on the relevance of the model $p_\theta(x_{0:T})$ and is able to fit the data. To that end, we consider the following factorization for the transition distribution:

$$p_\theta(z_t, x_t | z_{t-1}, x_{t-1}) = p_\theta(z_t | z_{t-1}, x_{t-1})p_\theta(x_t | z_{t-1:t}, x_{t-1}). \quad (2.1)$$

The joint distribution of $(x_{0:T}, z_{0:T})$ reads

$$p_\theta(z_{0:T}, x_{0:T}) = p_\theta(z_0, x_0) \prod_{t=1}^T p_\theta(z_t | z_{t-1}, x_{t-1})p_\theta(x_t | z_{t-1:t}, x_{t-1}), \text{ for all } T \in \mathbb{N}. \quad (2.2)$$

This factorization assumes that the observation x_t depends on the previous latent variables z_{t-1} and z_t , and not only on the current latent variable z_t and the previous observation x_{t-1} . While the distribution of latent variable z_t is determined by the previous observation x_{t-1} and the previous latent variable z_{t-1} . The choice of this factorization is motivated by the fact that some popular generative models based on latent variables can be derived from it, such as the HMC, the RNN and the GUM (Salaün et al., 2019).

Our objective is to cast the HMC, RNN and GUM generative models into a more general one, the PMC. To this end, we recall the HMC (1.6) where the joint distribution of the observed and latent variables reads

$$p_\theta(x_{0:T}, z_{0:T}) \stackrel{\text{HMC}}{=} p_\theta(z_0, x_0) \prod_{t=1}^T p_\theta(z_t | z_{t-1})p_\theta(x_t | z_t), \text{ for all } T \in \mathbb{N}.$$

Here (x_t, z_t) becomes conditionally independent of x_{t-1} given z_{t-1} , and x_t , in addition, does not depend on z_{t-1} . Similarly, the GUM is a particular case of the PMC defined as follows

$$p_\theta(x_{0:T}, z_{0:T}) \stackrel{\text{GUM}}{=} p_\theta(z_0, x_0) \prod_{t=1}^T p_\theta(z_t | z_{t-1}, x_{t-1})p_\theta(x_t | z_t), \text{ for all } T \in \mathbb{N},$$

where x_t becomes conditionally independent of (x_{t-1}, z_{t-1}) . In the case of predicting future observations with RNNs, a probabilistic approach seems more appropriate when we want to quantify the uncertainty associated with our prediction. To do this, we simply replace g_θ by a parametric distribution p_θ and o_t by x_{t+1} in Equation (1.2). In addition, we use the transformation $z_t \leftarrow h_{t-1}$ in equations (1.1)-(1.2) to obtain the following model:

$$\begin{aligned} p_\theta(x_{0:T}) &\stackrel{\text{RNN}}{=} p_\theta(x_0) \prod_{t=1}^T p_\theta(x_t | z_t), \text{ for all } T \in \mathbb{N}, \\ p_\theta(z_t | z_{t-1}, x_{t-1}) &\stackrel{\text{RNN}}{=} \delta_{f_\theta(z_{t-1}, x_{t-1})}(z_t), \quad z_0 \stackrel{\text{RNN}}{=} 0, \quad p_\theta(x_0 | z_0) \stackrel{\text{RNN}}{=} p_\theta(x_0). \end{aligned}$$

Contrary to the HMC and the GUM, the RNN follows a different approach. In the RNN, the latent variable z_t is deterministically determined, given the previous observation x_{t-1} and the previous latent variable z_{t-1} . With a slight abuse of notation, $p_\theta(z_t|z_{t-1}, x_{t-1})$ coincides with the Dirac measure and is not a probability density function. The expression of z_t relies on an activation function f_θ . Similar to the HMC and the GUM, in the RNN x_t depends on z_t given the past. We can see a common underlying framework that captures the joint probability distributions of the observed and latent variables in these models. The graphical structures of the models are summarized in Figure 2.1.

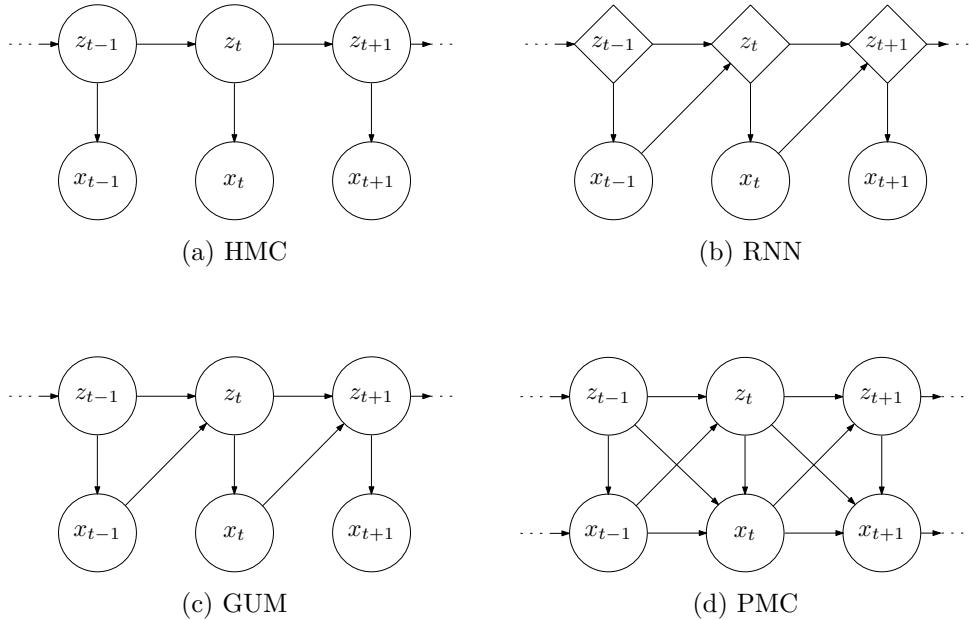


Figure 2.1: Conditional dependencies of the HMC, RNN, GUM, and PMC. In the RNN, the hidden states z_t are shown as diamonds to stress that they are no source of stochasticity. The HMC, RNN, and GUM are particular cases of the PMC.

Remark 2.2.1. Salaün et al. (2019) have proposed the GUM as a unified framework to compare the expressivity of generative models based on latent variables. The GUM can be seen as a stochastic version of the RNN which includes popular generative models such as the Variational RNN (Chung et al., 2015) and the Stochastic RNN (Fraccaro et al., 2016) as particular cases, with a latent variable $z_t \leftarrow (h_t, z_t)$.

2.3. Parameter estimation for general PMCs

In this section, we propose a VI approach to estimate the parameters θ of general PMC models. This new approach can be applied to any sequence $x_{0:T}$ of varying length T and does not require the knowledge of the latent variables $z_{0:T}$. In addition, it is suitable for high dimensional models (Blei et al., 2017). First, we introduce a general parameterization of the PMC model which includes a deep parameterization (via DNNs). Next, we adapt the (static) VI framework described in Subsection 1.2.1 for the sequential case with PMCs.

2.3.1 General parameterization of PMCs

We propose a general parameterization of the PMC model that can be applied to any PMC. Without loss of generality, we consider the transition distribution $p_\theta(z_t, x_t | z_{t-1}, x_{t-1})$ given in (2.1). A general parameterization allows us to consider different any (conditional) distributions $p_\theta(z_t | z_{t-1}, x_{t-1})$ and $p_\theta(x_t | z_{t-1:t}, x_{t-1})$, e.g. Gaussian distributions. Thus, for fixed distributions, the parameters are learned based on functions of the conditional variables. This parameterization extends beyond linear functions and also includes the application of deep neural networks due to the universal approximation property (see Section 1.1).

Let $\psi_\theta^z(z_{t-1}, x_{t-1})$ and $\psi_\theta^x(z_{t-1:t}, x_{t-1})$ be two vector-valued functions of (z_{t-1}, x_{t-1}) and of $(z_{t-1:t}, x_{t-1})$, respectively, that are assumed to be differentiable w.r.t. θ . Let also $\eta(z; w)$ and $\zeta(x; w')$ be probability density functions (pdf) on \mathbb{R}^{d_z} and \mathbb{R}^{d_x} , respectively, whose parameters are given by the vectors w and w' , respectively. η and ζ are assumed to be differentiable w.r.t. w and w' , respectively. Then, we parameterize the conditional distributions in (2.2) as

$$p_\theta(z_t | z_{t-1}, x_{t-1}) = \eta(z_t; \psi_\theta^z(z_{t-1}, x_{t-1})), \quad (2.3)$$

$$p_\theta(x_t | z_{t-1:t}, x_{t-1}) = \zeta(x_t; \psi_\theta^x(z_{t-1:t}, x_{t-1})). \quad (2.4)$$

In other words, ψ_θ^z (resp. ψ_θ^x) describes the parameters of the (conditional) distribution η (resp. ζ).

Example 2.3.1. As an illustration, we consider η as a multivariate Gaussian distribution. ψ_θ^z is the vector which contains the mean and the covariance matrix of η . In this case, $p_\theta(z_t | z_{t-1}, x_{t-1})$ reads

$$p_\theta(z_t | z_{t-1}, x_{t-1}) = \mathcal{N}(z_t; \mu_\theta^z, \sigma_\theta^z), \text{ where } [\mu_\theta^z, \sigma_\theta^z] = \psi_\theta^z(z_{t-1}, x_{t-1}),$$

It shows how the mean and covariance matrix of this Gaussian distribution are derived from the values given by the function ψ_θ^z , which is assumed to be differentiable w.r.t. θ .

Deep pairwise Markov chain - A particular case of the general parameterization of the PMC model is the Deep Pairwise Markov Chain (DPMC), where the parameterization of the transition distribution $p_\theta(z_t, x_t | z_{t-1}, x_{t-1})$ presented in (2.3) and (2.4) is given by DNNs. Since DNNs can theoretically approximate any function which satisfies reasonable assumptions (see Section 1.1.2), our objective is to use them to approximate any parameterization of η and ζ of the distributions $p_\theta(z_t | z_{t-1}, x_{t-1})$ and $p_\theta(x_t | z_{t-1:t}, x_{t-1})$, respectively. In other words, ψ_θ^z and ψ_θ^x are the outputs of two Deep Neural Network (DNN). For example, with (z_{t-1}, x_{t-1}) and $(z_{t-1:t}, x_{t-1})$ as inputs, respectively in (2.3) and (2.4). The set of parameters θ now consists of the parameters of these DNN (weights and biases). In this case, their gradients are computable from the backpropagation algorithm (Rumelhart et al., 1985; Hecht-Nielsen, 1992) since ψ_θ^z and ψ_θ^x are differentiable w.r.t. θ (see Section 1.1.1).

Example 2.3.2. We recall the previous example 2.3.1, where η is a Gaussian distribution. In this case, the mean μ_θ^z and the covariance matrix $\text{diag}(\sigma_\theta^z)$ are the output of a neural network as illustrated in Figure 2.2. For example, μ_θ^z and σ_θ^z can be the output of a linear layer.

$$p_\theta(z_t | z_{t-1}, x_{t-1}) = \mathcal{N}(z_t; \mu_\theta^z, \text{diag}(\sigma_\theta^z)), \text{ where } [\mu_\theta^z, \sigma_\theta^z] = \psi_\theta^z(z_{t-1}, x_{t-1}).$$

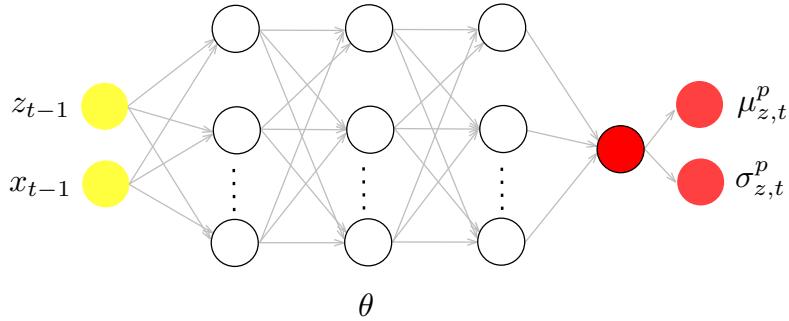


Figure 2.2: Illustration of a deep parameterization of the distribution $p_\theta(z_t | z_{t-1}, x_{t-1})$, where the parameters μ_θ^z and σ_θ^z of the Gaussian distribution are the output of a DNN.

2.3.2 Variational Inference for PMCs

For PMCs, we can extend the ELBO given in (1.4), which was formulated for static models to the sequential case. We now define $x \leftarrow x_{0:T}$, and $z \leftarrow z_{0:T}$. Then the following inequality holds for any variational distribution $q_\phi(z_{0:T}|x_{0:T})$,

$$\log(p_\theta(x_{0:T})) \geq - \int q_\phi(z_{0:T}|x_{0:T}) \log \left(\frac{q_\phi(z_{0:T}|x_{0:T})}{p_\theta(x_{0:T}, z_{0:T})} \right) dz_{0:T} = \mathcal{Q}_{\text{gen}}(\theta, \phi), \quad (2.5)$$

where q_ϕ depends on a set of parameters ϕ . In our sequential case, we choose the following variational distribution

$$q_\phi(z_{0:T}|x_{0:T}) = q_\phi(z_0|x_{0:T}) \prod_{t=1}^T q_\phi(z_t|z_{0:t-1}, x_{0:T}). \quad (2.6)$$

This general factorization, that is based on transitions $q_\phi(z_t|z_{0:t-1}, x_{0:T})$, captures the temporal dependencies inherent in sequential data. This form involves a choice of a variational distribution $q_\phi(z_t|z_{0:t-1}, x_{0:T})$, and the parameters that govern this distribution remain constant across time. The variational distribution q_ϕ should respect the differentiability and computational tractability constraints. For efficient optimization, $q_\phi(z_t|z_{0:t-1}, x_{0:T})$ should be differentiable w.r.t. ϕ and should be chosen in a way that $\mathcal{Q}_{\text{gen}}(\theta, \phi)$ is computable or can be approximated (see Subsection 1.2.1). Thus, the factorization of $p_\theta(z_{0:T}, x_{0:T})$ and $q_\phi(z_{0:T}|x_{0:T})$ given by (2.2) and (2.6), respectively, allows us to rewrite the ELBO (2.5) as follows

$$\mathcal{Q}_{\text{gen}}(\theta, \phi) = \mathcal{L}_1(\theta, \phi) + \mathcal{L}_2(\theta, \phi) \quad (2.7)$$

where

$$\begin{aligned} \mathcal{L}_1(\theta, \phi) &= \mathbb{E}_{q_\phi(z_0|x_{0:T})} (\log p_\theta(x_0|z_0)) \\ &\quad + \sum_{t=1}^T \mathbb{E}_{q_\phi(z_t|z_{0:t-1}, x_{0:T})} (\log p_\theta(x_t|z_{t-1:t}, x_{t-1})), \end{aligned} \quad (2.8)$$

$$\begin{aligned} \mathcal{L}_2(\theta, \phi) &= - D_{\text{KL}}(q_\phi(z_0|x_{0:T}) || p_\theta(z_0)) \\ &\quad - \sum_{t=1}^T D_{\text{KL}}(q_\phi(z_t|z_{0:t-1}, x_{0:T}) || p_\theta(z_t|z_{t-1}, x_{t-1})). \end{aligned} \quad (2.9)$$

$\mathcal{Q}_{\text{gen}}(\theta, \phi)$ involves the sum of two terms. The first term $\mathcal{L}_1(\theta, \phi)$ represents a reconstruction term which measures the ability to reconstruct observations

according to the conditional likelihood p_θ from the latent variables distributed according to q_ϕ . The second term $\mathcal{L}_2(\theta, \phi)$ involves a KLD term between the variational q_ϕ and the conditional prior p_θ distributions, which encourages q_ϕ to be close to p_θ (Kingma & Welling, 2014). It remains to compute and optimize the ELBO (2.7) w.r.t. (θ, ϕ) in order to estimate the parameters of the PMC model. On one hand, the term $\mathcal{L}_2(\theta, \phi)$ (2.9) involves the KLD between q_ϕ and p_θ .

Algorithm 1 General parameter estimation for generative PMCs

Input: $x_{0:T}$, the data; ϱ , the learning rate; M the number of samples

Output: (θ^*, ϕ^*) , sets of estimated parameters

- 1: Initialize the parameters θ^0 and ϕ^0
 - 2: $j \leftarrow 0$
 - 3: **while** convergence is not attained **do**
 - 4: Sample $z_0^{(m)} \sim q_{\phi^j}(z_0 | x_{0:T})$, for all $1 \leq m \leq M$
 - 5: Sample $z_t^{(m)} \sim q_{\phi^j}(z_t | z_{0:t-1}^{(m)}, x_{0:T})$, for all $1 \leq m \leq M$, for all $1 \leq t \leq T$
 - 6: Evaluate the loss $\widehat{\mathcal{Q}}_{\text{gen}}(\theta^j, \phi^j)$ from (2.7), (2.9), and (2.12).
 - 7: Compute the derivative of the loss function $\nabla_{(\theta, \phi)} \widehat{\mathcal{Q}}_{\text{gen}}(\theta, \phi)$.
 - 8: Update the parameters with gradient ascent
- $$\begin{pmatrix} \theta^{(j+1)} \\ \phi^{(j+1)} \end{pmatrix} = \begin{pmatrix} \theta^j \\ \phi^j \end{pmatrix} + \varrho \nabla_{(\theta, \phi)} \widehat{\mathcal{Q}}_{\text{gen}}(\theta, \phi) \Big|_{(\theta^j, \phi^j)} \quad (2.10)$$
- 9: $j \leftarrow j + 1$
 - 10: **end while**
 - 11: $\theta^* \leftarrow \theta^j$
 - 12: $\phi^* \leftarrow \phi^j$
-

On the other hand, the term $\mathcal{L}_1(\theta, \phi)$ (2.8) coincides with expectations w.r.t. q_ϕ and can be approximated by Monte Carlo estimation. For this, we use the reparameterization trick presented in Subsection 1.2.1, which can be extended to the sequential case by considering Equation (1.5) for each time step t , as follows:

$$z_{0:T}^{(m)} = g(\phi, \epsilon_{0:T}^{(m)}), \text{ for } m \in [1 : M]. \quad (2.11)$$

Thus, $\mathcal{L}_1(\theta, \phi)$ (2.8) can be approximated by

$$\hat{\mathcal{L}}_1(\theta, \phi) = \frac{1}{M} \sum_{m=1}^M \log p_\theta(x_0 | z_0^{(m)}) + \frac{1}{M} \sum_{m=1}^M \sum_{t=1}^T \log p_\theta(x_t | z_{t-1:t}^{(m)}, x_{t-1}), \quad (2.12)$$

where $z_t^{(m)}$ is a differentiable function of ϕ that is sampled from $q_\phi(z_t|z_{0:t-1}, x_{0:T})$, for $m \in [1 : M]$ and $t \in [0 : T]$. Algorithm 1 summarizes the general estimation algorithm for general PMCs. Here, we learn the generative model by maximizing the ELBO \mathcal{Q}_{gen} with respect to their parameters θ and ϕ .

2.4. Experiments and results

In this section, we first introduce a particular instance of the PMC model which combines the deep PMC model (see Section 2.3.1) and the stochastic RNN model (SRNN) (Bayer & Osendorfer, 2014; Chung et al., 2015). From this instance, we derive different generative models for sequential data with specific dependencies between latent and observed variables. Finally, we compare their performance with the stochastic RNN model on two datasets.

2.4.1 Model description

SRNN architectures are specific instances of the PMCs, which have demonstrated good experimental results (Bayer & Osendorfer, 2014; Chung et al., 2015), making it natural to compare them with their PMC extension. We introduce a model that combines the DPMC, and the SRNN models. This generative (deep) PMC model consists of a latent process in an augmented dimension, $z_t \leftarrow (h_t, z_t)$, the transition distribution now reads

$$p_\theta(h_t, z_t, x_t | h_{t-1}, z_{t-1}, x_{t-1}) = p_\theta(h_t | h_{t-1}, z_{t-1}, x_{t-1}) p_\theta(z_t | h_{t-1:t}, z_{t-1}, x_{t-1}) \times p_\theta(x_t | h_{t-1:t}, z_{t-1:t}, x_{t-1}). \quad (2.13)$$

Remark 2.4.1. Note that the previous equation is nothing more than a particular case of the TMC model with transition (2.13). However, we consider it as a particular instance of the PMC model since h_t and z_t , for all $t \in [0 : T]$, are considered as latent variables with no physical interpretation.

On the other hand, the variational distribution q_ϕ defined in (2.6) is factorized as follows

$$q_\phi(z_t, h_t | z_{0:t-1}, h_{0:t-1}, x_{0:T}) = q_\phi(z_t | z_{0:t-1}, h_{0:t}, x_{0:T}) q_\phi(h_t | z_{0:t-1}, h_{0:t-1}, x_{0:T}).$$

We consider the general parameterization presented in Subsection 2.2. However, we now have an additional distribution because of the new variable h_t . Let λ be a distribution on h_t parameterized by a differentiable (w.r.t. θ) and vector valued function denoted as ψ_θ^h and which can depend

on $(h_{t-1}, z_{t-1}, x_{t-1})$. We recall that ψ_θ^z and ψ_θ^x are defined in (2.3) and (2.4), respectively. Thus, the parameterized transition (2.13) reads

$$\begin{aligned} p_\theta(h_t|h_{t-1}, z_{t-1}, x_{t-1}) &= \lambda\left(h_t; \psi_\theta^h(h_{t-1}, z_{t-1}, x_{t-1})\right), \\ p_\theta(z_t|h_{t-1:t}, z_{t-1}, x_{t-1}) &= \eta(z_t; \psi_\theta^z(h_{t-1:t}, z_{t-1}, x_{t-1})), \\ p_\theta(x_t|h_{t-1:t}, z_{t-1:t}, x_{t-1}) &= \zeta(x_t; \psi_\theta^x(h_{t-1:t}, z_{t-1:t}, x_{t-1})). \end{aligned}$$

In the context of SRNN architectures, the variable h_t represents a deterministic summary of the past until time $t - 1$, *i.e.* $h_t = \psi_\theta^h(h_{t-1}, z_{t-1}, x_{t-1})$. While z_t corresponds to a noisy version of h_t (it is why we have split the latent process in two). Note that since $h_{0:T}$ is deterministic given $(z_{0:T}, x_{0:T})$, its posterior distribution becomes trivial, and thus there is no need to consider a variational distribution for it. The variational distribution q_ϕ is then parameterized as

$$q_\phi(z_t|z_{0:t-1}, h_{0:t}, x_{0:T}) = q_\phi(z_t|h_t, x_t) = \tau(z_t; \psi_\phi^z(h_t, x_t)), \quad (2.14)$$

where $\tau(z; \psi_\phi^z)$ is a probability density function on \mathbb{R}^{d_z} whose parameters are given by ψ_ϕ^z , which is differentiable w.r.t. ϕ . Following this reasoning and with a slight abuse of notation (where λ coincides with the Dirac measure), we can incorporate several degrees of generalization of the classical RNN and of the SRNN of Chung et al. (2015). The different deep PMC models we consider are defined in Table 2.1 and are based on the specific dependencies of the involved random variables. Note that ψ_θ^x , ψ_θ^h , ψ_θ^z and ψ_ϕ^z are now neural networks.

2.4.2 Results

Model configuration - In our experiments, the observed random variables are discrete, and each x_t takes values in a binary space $\{0, 1\}^{d_x}$. As a consequence, the distribution ζ coincides with the Bernoulli distribution, and the output of ψ_θ^x with its parameter. For λ , we choose the Gaussian distribution and the output of ψ_θ^z corresponds to the mean and to the diagonal covariance matrix of the Gaussian distribution, which is summarized as follows

$$\begin{aligned} h_t &= \psi_\theta^h(\cdot), \\ p_\theta(z_t|\cdot) &= \mathcal{N}(z_t; \mu_\theta^z, \text{diag}(\sigma_\theta^z)), \text{ where } [\mu_\theta^z, \sigma_\theta^z] = \psi_\theta^z(\cdot), \\ p_\theta(x_t|\cdot) &= \text{Ber}(x_t; \rho_\theta^x), \text{ where } \rho_\theta^x = \psi_\theta^x(\cdot). \end{aligned}$$

Here the notation (\cdot) is used to avoid presenting a specific dependence between variables. These dependencies are specified for each model and are presented in Table 2.1. The variational distribution q_ϕ given in (2.14) is chosen

Models	Parameterized function		
	ψ_θ^h	ψ_θ^z	ψ_θ^x
RNN	(h_{t-1}, x_{t-1})	×	h_t
SRNN	$(h_{t-1}, z_{t-1}, x_{t-1})$	h_t	(h_t, z_t)
PMC-I	$(h_{t-1}, z_{t-1}, x_{t-1})$	h_t	(h_t, z_t, x_{t-1})
PMC-II	$(h_{t-1}, z_{t-1}, x_{t-1})$	h_t	$(h_{t-1:t}, z_t, x_{t-1})$
PMC-III	$(h_{t-1}, z_{t-1}, x_{t-1})$	h_t	$(h_{t-1:t}, z_{t-1:t}, x_{t-1})$
PMC-IV	$(h_{t-1}, z_{t-1}, x_{t-1})$	(h_t, x_{t-1})	$(h_{t-1:t}, z_t, x_{t-1})$

Table 2.1: Configuration of the dependencies for different deep generative PMCs. In each model, the sequence of latent variables $\{h_t\}_{t \in \mathbb{N}}$ is treated as a deterministic variable given the observations. As a result, η coincides with the Dirac measure. The distribution λ is typically chosen to be Gaussian, while ζ depends on the nature of the observations. Remember that in a classical RNN, $\{z_t\}_{t \in \mathbb{N}}$ is not considered.

as Gaussian, which satisfies

$$q_\phi(z_t | h_t, x_t) = \mathcal{N}(z_t; \mu_z^q, \text{diag}(\sigma_z^q)), \text{ where } [\mu_z^q, \sigma_z^q] = \psi_\phi^z(h_t, x_t).$$

The functions ψ_θ^x , ψ_θ^z , ψ_θ^h and ψ_ϕ^z are implemented as neural networks consisting of two hidden layers. The rectified linear unit (ReLU) activation function is used for the hidden layers, and the outputs of the neural networks are adapted according to their role. For example, the output of ψ_θ^x is a layer of d_x sigmoid functions due to the nature of the observations (binary values). Additionally, the number of hidden units of each neural network coincides with the dimension d_z of z_t and is different for each model and data set, which is specified in the next part.

Training - Each model was trained with stochastic gradient descent on the negative evidence lower bound using the Adam optimizer (Kingma & Ba, 2015) with a learning rate of 0.001. The number of epochs was set to 100 for both data sets. The number of hidden units of the neural networks (d_z) can be fixed for all the models, or can be chosen by considering the number of parameters of the models to be compared (*i.e.* the number of parameters are the same or close to).

Evaluation - The performance of the models is evaluated in terms of the approximated ELBO and log-likelihood of the observations on the test data

Model	MNIST data set				Music data sets	
	Config. 1		Config. 2		Config. 2	
	d_z	d_h	d_z	d_h	d_z	d_h
RNN	3	100	3	162	300	562
SRNN	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 2.2: Dimensions of latent variables for each Deep PMC. ψ_θ^h , ψ_θ^z , ψ_θ^x and ψ_ϕ^z are implemented as neural networks with two hidden layers. The number of neurons on each layer coincide with d_h .

set; we use a particle filter with the estimated variational distribution as importance distribution and $N = 100$ particles.

Image generation - The MNIST dataset ([LeCun, 1998](#)) contains 60000 (resp. 10000) train (resp. test) 28×28 binary images. In this case, an observation x_t consists of a column of the image and its dimensionality is $d_x = 28$. The length of each sequence is $T + 1 = 28$. For this data set, we consider two configurations for the training of the models and are summarized in Table 2.2. Config. 1 corresponds to the configuration in which the number of hidden units of the neural networks is fixed $d_h = 100$ for all the models. In Config. 2, the number of hidden units of the neural networks is chosen by considering the number of parameters of the models to be compared. We set $d_h = 162$, $d_h = 100$, $d_h = 95$, $d_h = 79$, $d_h = 78$ and $d_h = 74$ for the RNN, SRNN, the PMC-I, the PMC-II, the PMC-III, and the PMC-IV, respectively. We also set $d_z = 3$ for each model and both configurations.

Table 2.3 presents 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 compared to other models. This indicates that the performance of the PMCs is better than of SRNN and RNN models. An example of images generated from the estimated $p_\theta(x_{0:t})$ of the PMC-II is shown in Figure 2.3.

Model	MNIST, config. 1		MNIST, config. 2	
	ELBO	approx. LL	ELBO	approx. LL
RNN	-65,976	-65,976	-65,700	-65,700
SRNN	-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	-63,603	-67,648	-64,924

Table 2.3: Averaged ELBO and approximated log-likelihood (approx. LL) of the observations on the test set with two different configurations. For the RNN, the ELBO coincides with the (exact) log-likelihood.



Figure 2.3: Examples of generated images from estimated $p_\theta(x_{0:t})$ for the MNIST data set with the PMC-II model.

Polyphonic music generation - We also consider the polyphonic music data sets (Bengio et al., 2013), where three polyphonic music data sets are available, the classical piano music (Piano), the folk tunes (Nottingham) and the four-part chorales by J.S. Bach (JSB). The input consists of 88 binary visible units that span the whole range of piano from A0 to C8 (*i.e.* $x_t \in \{0, 1\}^{88}$). In this case, we consider the Config. 2 for the training of the models (see Table 2.2) since it is a fairer comparison between the models.

We set $d_z = 300$ for each model, and $d_h = 562$, $d_h = 300$, $d_h = 294$, $d_h = 278$, $d_h = 260$ and $d_h = 272$ for the RNN, the SRNN, the PMC-I, the PMC-II, the PMC-III and the PMC-IV respectively. Table 2.4 presents the results of the averaged ELBO and the averaged approximated log-likelihood where the PMC-II (resp. PMC-IV) has the best performance compared to other models on the Piano (resp. Nottingham and JSB) data set.

Model	Polyphonic music data sets		
	Piano	Nottingham	JSB
RNN	-10,52	-23,89	-10,77
SRNN	-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 2.4: Approximated likelihoods on the polyphonic music data sets. For the RNN, the exact log-likelihood is computed.

2.5. Generative power of PMCs

In this section, our objective is to analyze the previous models from a theoretical point of view. We consider a linear and stationary Gaussian PMC, with $d_x = 1$. In a stationary Gaussian process, the statistical properties, like the mean and covariance of the observations, do not change over time. This stationarity implies that the covariance between two observations depends only on the time difference k between them. This analysis is then based on the associated covariance function $r_k = \text{Cov}(x_t, x_{t+k})$, for all $k \in \mathbb{N}$, which characterize the distribution $p_\theta(x_{0:T})$ induced by each model.

2.5.1 Linear and stationary Gaussian PMCs

Linear PMC - We consider the case where ψ_θ^z and ψ_θ^x in equations (2.3)-(2.4) are vectorial linear functions. We have the following linear parameterization of the PMC

$$p_\theta(z_0, x_0) = \varsigma((z_0, x_0); [0; \Sigma_0]), \quad (2.15)$$

$$p_\theta(z_t|z_{t-1}, x_{t-1}) = \eta(z_t; [az_{t-1} + cx_{t-1}; \alpha]), \quad (2.16)$$

$$p_\theta(x_t|z_{t-1:t}, x_{t-1}) = \zeta(x_t; [bz_t + ez_{t-1} + fx_{t-1}; \beta]), \quad (2.17)$$

where the notation $[\cdot ; \cdot]$ considers the first and second order of the initial distribution $p_\theta(z_0, x_0)$, of $p_\theta(x_t|z_{t-1:t}, x_{t-1})$, and of $p_\theta(z_t|z_{t-1}, x_{t-1})$. The dimensions of the parameters a, b, c, e and f are $d_z \times d_z$, $1 \times d_z$, $d_z \times 1$, $1 \times d_z$, 1×1 , respectively. The covariance matrix α is a square matrix and $\beta \geq 0$. In the initial distribution $p_\theta(z_0, x_0)$, 0 is a (d_z+1) zero vector and Σ_0 is a (d_z+1) square covariance matrix given by

$$\Sigma_0 = \begin{bmatrix} \eta & \tilde{\gamma}^\top \\ \tilde{\gamma} & r_0 \end{bmatrix}.$$

The dimensions of η and $\tilde{\gamma}$ are $d_z \times d_z$ and $1 \times d_z$, respectively; and r_0 is scalar. Thus, the set of parameters now is $\theta = (a, b, c, e, f, \alpha, \beta, \eta, \tilde{\gamma}, r_0)$.

Gaussian PMC - We now consider ς , η and ζ as Gaussian distributions so the distribution $p_\theta(x_{0:T})$ is a multivariate Gaussian distribution due to the linear structure of the model. However, it is important to note that this assumption does not result in any loss of generality; we employ it here for the sake of clarity. The covariance function r_k associated to $p_\theta(x_{0:T})$ can be deduced from the covariance matrix Σ_t associated to the distribution $p_\theta(z_t, x_t)$. First, an equivalent representation of (2.15) -(2.17) is obtained by considering the first and second order moments of the pair (z_t, x_t) given (z_{t-1}, x_{t-1}) . Since this distribution involves the product of two Gaussian distributions, one being linear in the other and with results on conditional Gaussian distributions, we obtain:

$$\begin{aligned} \mathbb{E} \left([z_t, x_t]^\top | z_{t-1}, x_{t-1} \right) &= M \begin{bmatrix} z_{t-1} \\ x_{t-1} \end{bmatrix}, \\ \text{Var} \left([z_t, x_t]^\top | z_{t-1}, x_{t-1} \right) &= \Sigma_{t|t-1}, \end{aligned}$$

where

$$M = \begin{bmatrix} a & c \\ ba + e & bc + f \end{bmatrix}, \quad \Sigma_{t|t-1} = \begin{bmatrix} \alpha & \alpha b^\top \\ b\alpha & \beta + b\alpha b^\top \end{bmatrix}. \quad (2.18)$$

The covariance of the pair (z_t, x_t) is given by $\Sigma_0 \times (M^k)^\top$, which can be easily deduced from the previous representation. We also obtain the following expression for the covariance matrix associated to the distribution $p_\theta(z_t, x_t)$ Σ_t ,

$$\Sigma_t = M\Sigma_{t-1}M^\top + \Sigma_{t|t-1}, \quad (2.19)$$

which is an immediate consequence of the Lemma A.0.1 in Appendix A.

Stationary PMC - In order to assure the stationarity of $\{x_t\}_{t \in \mathbb{N}}$, we consider directly that the process $\{z_t, x_t\}_{t \in \mathbb{N}}$ is stationary. Consequently, Σ_0 and Σ_t (2.19) should satisfy the following equivalence

$$\Sigma_0 = M\Sigma_0M^\top + \Sigma_{t|t-1}. \quad (2.20)$$

This matrix equation describes a set of constraints on the parameters of the PMC model, which ensures the stationarity of the distributions $p_\theta(z_t, x_t)$ and $p_\theta(x_t)$.

2.5.2 Theoretical analysis of PMCs

We are interested in the modeling power of the PMC. For that, we characterize the covariance function of the distribution $p_\theta(x_{0:T})$ induced by the PMC, and compare it with the one of the GUM presented in [Salaün et al. \(2019\)](#). We focus on the case where the latent and observed variables are both scalar ($d_z = 1$ and $d_x = 1$). The scalar case is interesting because it allows for a direct deduction of the covariance function derived from the PMC.

For clarity, we set $r_0 = 1$, which means $p_\theta(x_t) = \mathcal{N}(x_t; 0; 1)$, for all $t \in \mathbb{N}$. We also parameterize $\tilde{\gamma} = \gamma\eta$, then the set of parameters is now given by $\theta = (a, b, c, e, f, \alpha, \beta, \eta, \gamma)$. We start by presenting the covariance function of the GUM, where the matrix M is diagonalizable. By plugging in $e = f = 0$, and $\gamma = b$, the covariance function of the GUM is given by

$$r_k \stackrel{\text{GUM}}{=} A^{k-1}B, \text{ for all } k \in \mathbb{N}^*, \quad (2.21)$$

where $A = a + cb$, $B = a\eta b^2 + bc(\beta + \eta b^2)$, and $\text{Var}(x_t) = \beta + \eta b^2 = 1$, for all $t \in \mathbb{N}$. The stationarity constraints are simplified to two constraints:

$$\begin{aligned} \beta &= 1 - b^2\eta, \\ \alpha &= (1 - a^2 - 2abc)\eta - c^2, \end{aligned}$$

In addition to the settings of the GUM, the covariance functions associated to a linear and stationary Gaussian HMC and RNN are also derived.

- **HMC** - with $c = 0$ and $r_k = a^k \eta b^2$.
- **RNN** - the transition between (z_{t-1}, x_{t-1}) and z_t is deterministic then $\alpha = 0$. Moreover, $z_0 = 0$ and x_0 is independent of z_0 . Since $\text{Var}(x_t) = b^2 \beta + \eta = 1$, the constraint $\eta = c^2$ should also be satisfied to ensure that $\text{Var}(x_t) = r_0 = 1$ for all $t \in \mathbb{N}$.

In order to extend this study for PMCs, we assume that M is diagonalizable in the PMC, 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

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

Note that the condition $(a+bc+f)^2 - 4(af-ce) \geq 0$ is satisfied since M is diagonalizable. As a result, we can deduce r_k for the PMC, which is summarized in the following proposition.

Proposition 2.5.1. Let a linear and stationary (scalar) Gaussian PMC be defined by the transition and the conditional covariance matrices M and $\Sigma_{t|t-1}$ in (2.18) 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 $\{x_t\}_{t \in \mathbb{N}}$ reads

$$r_k = \bar{A}^k (\bar{B} + \frac{1}{2}) - \bar{C}^k (\bar{B} - \frac{1}{2}), \quad (2.22)$$

where

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

and where the following stationarity constraints are satisfied:

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

Proof. The proof relies on the assumption that M is diagonalizable, which enables us to derive an explicit expression for the covariances of the pair (z_t, x_t) . Then r_k can directly be deduced from this expression that reads

$$\begin{aligned} \Sigma_0 \times (M^k)^\top &= \text{Cov}([z_t, x_t]^\top, [z_{t+k}, x_{t+k}]^\top) \\ &= \begin{bmatrix} \text{Cov}(z_t, z_{t+k}) & \text{Cov}(z_t, x_{t+k}) \\ \text{Cov}(x_t, z_{t+k}) & \text{Cov}(x_t, x_{t+k}) \end{bmatrix}. \end{aligned}$$

On the other hand, the stationarity constraints are given by (2.20). We set $r_0 = 1$ and $\tilde{\gamma} = \gamma\eta$, so the following stationary relation holds

$$\begin{bmatrix} \eta & \gamma\eta \\ \gamma\eta & 1 \end{bmatrix} = \begin{bmatrix} \alpha & b\alpha \\ b\alpha & \beta + b^2\alpha \end{bmatrix} + \begin{bmatrix} a & c \\ ab + e & bc + f \end{bmatrix} \begin{bmatrix} \eta & \gamma\eta \\ \gamma\eta & 1 \end{bmatrix} \begin{bmatrix} a & ab + e \\ c & bc + f \end{bmatrix}.$$

Since the covariance matrix is symmetric and the diagonal elements are positive because they are variances, the set of 3 constraints are deduced directly from the previous relation. \square

Remark 2.5.1. The stationarity of the distribution $p_\theta(x_{0:T})$ implies that its associated variance-covariance matrix Σ_T^x is a Toeplitz matrix (*i.e.* the coefficients on each diagonal are equal) fully determined by its first row given by the covariance sequence $[r_0, r_1, \dots, r_T]$, where r_k is given by (2.22), for all $k \in \mathbb{N}^*$. Thus, Σ_T^x reads as, for all $T \in \mathbb{N}^*$,

$$\Sigma_T^x = \begin{bmatrix} 1 & r_1 & r_2 & r_3 & \dots & r_T \\ r_1 & 1 & r_1 & r_2 & \dots & \vdots \\ r_2 & r_1 & 1 & r_1 & \ddots & \vdots \\ r_3 & r_2 & r_1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ r_T & \dots & r_3 & r_2 & r_1 & 1 \end{bmatrix}.$$

Proposition 2.5.1 shows that the PMC generalizes the form of the covariance matrices of the GUM, HMC and RNN by introducing the parameters e and f . However, it remains challenging to determine whether any covariance series in the form (2.22) can be generated by a PMC because identifying \bar{A} , \bar{B} and \bar{C} , in order to ensure that (2.22) represents a valid covariance series, is a complex problem. Nonetheless, we can exhibit some particular covariance functions that can be generated by a (particular) PMC but not by a GUM, HMC or RNN as shown in the next proposition.

Proposition 2.5.2. Let \tilde{A} and \tilde{B} be two scalars, $r_0 = 1$ and

$$r_k = \begin{cases} \tilde{A}^k & \text{if } k \text{ is even,} \\ \tilde{A}^{k-1}\tilde{B} & \text{otherwise.} \end{cases} \quad (2.23)$$

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

$$-1 \leq \tilde{A} \leq 1 \quad \text{and} \quad -\frac{\tilde{A}^2 + 1}{2} \leq \tilde{B} \leq \frac{\tilde{A}^2 + 1}{2}, \quad (2.24)$$

and can be realized by a linear and stationary Gaussian PMC.

Proof. The proof relies on the Carathéodory-Toeplitz theorem (Akhiezer, 1965) since Σ_T^x is defined by a Toeplitz matrix with first row

$$[1, \tilde{B}, \tilde{A}^2, \tilde{A}^2\tilde{B}, \tilde{A}^4, \dots].$$

We analyze the series expansion of the covariance function to establish the necessary conditions for the positive semi-definiteness of Σ_T^x . This theorem allows us to determine the values of \tilde{A} and of \tilde{B} in (2.23) such that Σ_T^x is a valid covariance matrix.

We deduce the constraints $-1 \leq \tilde{A} \leq 1$, $-\frac{\tilde{A}^2+1}{2} \leq \tilde{B} \leq \frac{\tilde{A}^2+1}{2}$. Next, setting $\gamma = b$, and f either as 0 or $-a - bc$ (two particular cases of the PMC), we show that (2.22) coincides with (2.23), with

$$\begin{cases} \tilde{A} = \sqrt{ce} & \text{and} & \tilde{B} = b(c(1 - b^2\eta) + e\eta) & \text{if } f = 0, \\ \tilde{A} = \sqrt{e^2\eta + a^2(1 - b^2\eta)} & \text{and} & \tilde{B} = be\eta - a(1 - b^2\eta) & \text{if } f = -a - bc. \end{cases}$$

Finally, for any (\tilde{A}, \tilde{B}) satisfying (2.24), 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 stationarity constraints (2.20) for both cases $f = 0$ and $f = -a - bc$. For a detailed step-by-step proof, please refer to the Appendix B. \square

Proposition 2.5.2 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. The constraints on A and B with this switching are $-1 \leq A \leq 1$ and $-\frac{A^2+1}{2} \leq B \leq \frac{A^2+1}{2}$ since $B(k)$ is as expression of A and B .

This result can be compared with that of the GUM in the scalar case (Salaün et al., 2019), that can produce any covariance function given by (2.21), $r_k = A^{k-1}B$, with the constraints $-1 \leq A \leq 1$ and $\frac{A-1}{2} \leq B \leq \frac{A+1}{2}$. In other words, this proposition shows that the linear and stationary Gaussian PMC can model some Gaussian distributions which cannot be modeled by the previous linear and stationary Gaussian GUM.

2.6. Conclusions

This chapter was devoted to the development, study, comparison and application of a general generative model for sequential data based on the PMC model. Our approach combined the advantages of the HMM, RNN and GUM models and encapsulated them in a single framework. A new parameter estimation method based on the variational inference framework was also presented for the general PMC model, which is computationally efficient and easy to implement. Moreover, we presented a particular instance of the variational PMC model, combining the PMC model and deep parameterizations. This model has been compared with the RNN and SRNN models on the MNIST and polyphonic music data sets. The results show that the performance of the deep PMCs is better than of SRNN and RNN models. We have also shown that the linear and stationary Gaussian PMC can model some Gaussian distributions which cannot be modeled by the previous linear and stationary Gaussian HMC, RNN and GUM.

CHAPTER 3

Triplet Markov models for semi-supervised classification

Contents

3.1	Introduction	50
3.2	Semi-supervised estimation in general TMC	51
3.2.1	General parameterization of the TMC	51
3.2.2	A brief description of the semi-supervised problem	53
3.3	Semi-supervised Variational Inference for TMCs	54
3.3.1	ELBO for semi-supervised learning	54
3.3.2	Learning semi-supervised TMCs	55
3.4	Experiments	58
3.4.1	DTMC vs existing models	58
3.4.2	Binary data generation	60
3.4.3	Semi-supervised binary image segmentation	61
3.4.4	Results	62
3.5	Conclusions	64

3.1. Introduction

In this chapter, we want to extend the study we have done in the previous chapters to the case where we have labels associated with each observation. Let us recall the sequence of random variables $x_{0:T} = (x_0, \dots, x_T)$ and the sequence of labels $y_{0:T} = (y_0, \dots, y_T)$ associated to the previous sequence $x_{0:T}$, where $x_t \in \mathbb{R}^{d_x}$, and $y_t \in \Omega = \{\omega_1, \dots, \omega_C\}$, with C the number of classes. We also consider a sequence of latent variables $z_{0:T} = (z_0, \dots, z_T)$, where $z_t \in \mathbb{R}^{d_z}$.

The objective associated to Bayesian classification consists in computing, for all t , the posterior distributions $p(y_t|x_{0:T})$. The difficulty of the problem depends on the availability of the labels associated with the observations. When the labels are observed, the problem is referred to as supervised learning. Chapter 2 was dedicated to a general generative model based on PMCs, which can be used for supervised learning. This adaptation involves taking as observed variable the pair of observations and labels $x_t \leftarrow (x_t, y_t)$, and applying the adapted variational Algorithm 1 discussed in the chapter (see Appendix C for more details). However, in many real-world applications, it is expensive or impossible to obtain labels for the entire sequence $x_{0:T}$ due to various reasons, such as the high cost of labeling, the lack of expertise, or the lack of time, etc. The labels can be partially observed or not observed at all, which leads to the semi-supervised and unsupervised learning problems, respectively. Two main challenges arise in this context:

- How to effectively design generative models that not only generate observations x_t , but also generate labels y_t ?
- How to perform effective Bayesian inference under these conditions?

This chapter is devoted to the semi-supervised learning problem, and the unsupervised learning problem will be addressed in the next chapter. Here, the objective is to estimate the unobserved labels from the observations and the observed labels. To that end, the TMC model is considered (see Subsection 1.3.3) in which we can model not only the sequence of observations $x_{0:T}$, and their associated labels $y_{0:T}$, but also incorporate an auxiliary sequence $z_{0:T}$, which can provide additional information about the relationship between the observations and the labels. We propose a new adaptation of the VI algorithm presented in the previous chapter, which enables us to estimate the parameters of a general TMC model, and the unobserved labels. This general semi-supervised learning algorithm enables us to derive a variety of (deep) generative models which have been applied to sequential Bayesian classification

problems. Finally, we consider the problem of image segmentation, where the observations $x_{0:T}$ represent a noisy grayscale image while $y_{0:T}$ represent the original black and white image. The goal is to recover the original image from a noisy version of it. We show that our approach outperforms the state-of-the-art semi-supervised learning algorithms such as the Variational Sequential Labeler (VSL) (Chen et al., 2018), and the Semi-supervised Variational Recurrent Neural Network (SVRNN) (Butepage et al., 2019).

3.2. Semi-supervised estimation in general TMC

3.2.1 General parameterization of the TMC

We recall the TMC model given in Equation (1.8):

$$p_\theta(z_{0:T}, y_{0:T}, x_{0:T}) = p_\theta(z_0, y_0, x_0) \prod_{t=1}^T p_\theta(v_t | v_{t-1}),$$

where the triplet $v_t = (z_t, y_t, x_t)$. Here, it is possible to have different factorizations of the transition distribution $p_\theta(v_t | v_{t-1})$.

Example 3.2.1. The following factorizations are the possible choices for the transition distribution $p_\theta(v_t | v_{t-1})$:

$$\begin{aligned} p_\theta(v_t | v_{t-1}) &= p_\theta(x_t | v_{t-1}) p_\theta(y_t | x_t, v_{t-1}) p_\theta(z_t | x_t, y_t, v_{t-1}), \\ p_\theta(v_t | v_{t-1}) &= p_\theta(x_t | y_t, v_{t-1}) p_\theta(y_t | v_{t-1}) p_\theta(z_t | x_t, z_t, v_{t-1}), \\ p_\theta(v_t | v_{t-1}) &= p_\theta(x_t | x_t, y_t, v_{t-1}) p_\theta(y_t | z_t, v_{t-1}) p_\theta(z_t | v_{t-1}). \end{aligned}$$

In the first example, x_t depends on the triplet v_{t-1} , the label y_t depends on the observation x_t and the triplet v_{t-1} , and the latent variable z_t depends on the observation x_t , the label y_t and the triplet v_{t-1} .

The choice of the factorization of the transition distribution depends on the specific application and the underlying model. Thus, we use a general notation for the associated conditional distributions $p_\theta(x_t | \cdot)$, $p_\theta(z_t | \cdot)$ and $p_\theta(y_t | \cdot)$ in order to avoid presenting a specific dependence between variables.

In Chapter 2, we have introduced the probability density functions on \mathbb{R}^{d_x} , \mathbb{R}^{d_z} , as ζ , and η , respectively (Equations (2.3), and (2.4)). They are introduced to describe a general parameterization of the PMC model. As we extend these ideas to the TMC model in this chapter, we continue to use the functions to parameterize the transition distribution in the TMC model. We also define ϑ

as a probability distribution on Ω , which is used to parameterize the transition distribution for the labels and is differentiable w.r.t. their parameters. The general parameterized model is described by:

$$p_\theta(v_t|v_{t-1}) = p_\theta(x_t|\cdot) p_\theta(z_t|\cdot) p_\theta(y_t|\cdot), \quad (3.1)$$

$$p_\theta(z_t|\cdot) = \eta(z_t; \psi_\theta^z(\cdot)), \quad (3.2)$$

$$p_\theta(y_t|\cdot) = \vartheta(y_t; \psi_\theta^y(\cdot)), \quad (3.3)$$

$$p_\theta(x_t|\cdot) = \zeta(x_t; \psi_\theta^x(\cdot)), \quad (3.4)$$

where ψ_θ^y , ψ_θ^x and ψ_θ^z are vector-valued functions that are assumed to be differentiable w.r.t. θ .

Example 3.2.2. For the sake of clarity, let us explore a specific application of the TMC model where the labels y_t are binary ($\Omega = \{\omega_1, \omega_2\}$). The observations satisfy $x_t \in \mathbb{R}$, and $z_t \in \mathbb{R}$, for all t . This setup is particularly useful in image processing tasks such as noise reduction and classification, where y_t represents a pixel's classification (e.g. object vs. background), x_t is the observed noisy pixel value, and z_t models the latent variables that influence the observation's noise characteristics. Thus, we can extend the TMC model proposed in (Pieczynski & Desbouvries, 2005), by incorporating a continuous latent variable z_t . In particular, the model can be described as

$$\begin{aligned} \psi_\theta^y(y_{t-1}, x_{t-1}, z_t) &= \text{sigm}(a_{y_{t-1}} x_{t-1} + b_{y_{t-1}} z_t + c_{y_{t-1}}), \\ \psi_\theta^x(x_t) &= [d_{y_t}, \sigma_{y_t}], \\ \psi_\theta^z(x_{t-1}, y_{t-1}) &= [e_{y_{t-1}} x_{t-1}, \sigma'_{y_{t-1}}], \\ \vartheta(y_t; \rho) &= \mathcal{B}er(y_t; \rho), \\ \zeta(x_t; s = [\mu, \sigma]) &= \mathcal{N}(x_t; \mu, \sigma^2), \\ \eta(z_t; s' = [\mu', \sigma']) &= \mathcal{N}(z_t; \mu', \sigma'^2), \end{aligned}$$

where $\text{sigm}(v) = 1/(1 + \exp(-v)) \in [0, 1]$ is the sigmoid function. Note that, for example, the notation d_{y_t} means that the parameter d depends on the label y_t , i.e. $p_\theta(x_t|y_t = \omega_j) = \mathcal{N}(x_t; d_{\omega_j}, \sigma_{\omega_j}^x)$. The set of parameters is then given by:

$$\theta = (a_{\omega_i}, b_{\omega_i}, c_{\omega_i}, d_{\omega_j}, \sigma_{\omega_i}, e_{\omega_i}, \sigma'_{\omega_i} | (\omega_i, \omega_j) \in \Omega^2).$$

This parameterization can be easily extended to the multi-class cases with $C > 2$ by replacing ψ_θ^y by a vector of the softmax function, and $\vartheta(y_t; \rho)$ by the categorical distribution described by the C components of a vector ρ .

Remark 3.2.1. The parameterization of the TMC model is very general and can be used to derive a variety of models. Similarly to the PMC model, the functions ψ_θ^y , ψ_θ^x and ψ_θ^z can also be parameterized by deep neural networks, where the parameters θ encompass the weights and biases of the neural networks. We will refer to this model as the Deep Triplet Markov Chain (DTMC) model.

3.2.2 A brief description of the semi-supervised problem

In many practical scenarios, obtaining complete label information for all data points is often infeasible. Consequently, we frequently encounter situations where only a subset of the labels is observed. This incomplete labeling poses significant challenges for effective model training and inference. To clarify our approach, we decompose the sequence of labels $y_{0:T}$ into observed and hidden components:

$$y_{0:T} = (y_T^{\mathbf{O}}, y_T^{\mathbf{H}}),$$

where $y_T^{\mathbf{O}} = \{y_t\}_{t \in \mathbf{O}}$ (resp. $y_T^{\mathbf{H}} = \{y_t\}_{t \in \mathbf{H}}$) is the set of observed (resp. hidden) labels. Here, \mathbf{O} (resp. \mathbf{H}) denotes the set of time indices where labels are observed (resp. hidden). We assume that $\mathbf{O} \cap \mathbf{H} = \emptyset$ and $\mathbf{O} \cup \mathbf{H} = \{0, \dots, T\}$. For example, if $T = 5$, and labels are observed at time steps $0, 1, 2$, then $\mathbf{O} = \{0, 1, 2\}$ and $\mathbf{H} = \{3, 4, 5\}$. Thus, our observed data is $(x_{0:5}, y_0, y_1, y_2)$, and the hidden labels are (y_3, y_4, y_5) .

Here, our goal is to estimate the parameters θ of the TMC model from $(x_{0:T}, y_T^{\mathbf{O}})$, and compute the posterior distribution of the hidden labels y_t , for all $t \in \mathbf{H}$. The likelihood of the observed data $(x_{0:T}, y_T^{\mathbf{O}})$ reads

$$p_\theta(x_{0:T}, y_T^{\mathbf{O}}) = \sum_{y_s, s \in \mathbf{H}} \int p_\theta(z_{0:T}, y_{0:T}, x_{0:T}) dz_{0:T}, \quad (3.5)$$

and the posterior distributions, for all $t \in \mathbf{H}$, are defined as

$$p(y_t | x_{0:T}, y_T^{\mathbf{O}}) = \frac{\sum_{y_s, s \in \mathbf{H} \setminus \{t\}} \int p(z_{0:T}, y_{0:T}, x_{0:T}) dz_{0:T}}{\sum_{y_s, s \in \mathbf{H}} \int p(z_{0:T}, y_{0:T}, x_{0:T}) dz_{0:T}}. \quad (3.6)$$

Equations (3.5) and (3.6) involve integrals w.r.t. the latent variables. Consequently, they are not exactly computable in general. To that end, we have proposed a VI approach presented in Chapters 1 and 2. However, the algorithms cannot be applied directly to this case since partial observations of the sequence $y_{0:T}$ result in hidden labels. As consequence, the variational distribution has to be adapted to the case where the observed variables are $(x_{0:T}, y_T^{\mathbf{O}})$

and the latent variables are $(z_{0:T}, y_T^{\mathbf{H}})$. We deal with both discrete and continuous latent variables, which is a challenging problem since the variational distribution has to be factorized in order to be tractable, and has to be independent of the time step t in order to have a general model able to be applied to different contexts.

3.3. Semi-supervised Variational Inference for TMCs

In this section, we explore the semi-supervised variational inference method applied to general TMCs. We start by the ELBO, and the formulation of the variational distribution. Finally, we propose an algorithm to estimate the parameters of the TMC model in the semi-supervised context.

3.3.1 ELBO for semi-supervised learning

We consider the variational distribution $q_{\phi}(z_{0:T}, y_T^{\mathbf{H}}|x_{0:T}, y_T^{\mathbf{O}})$. The ELBO of the log-likelihood (3.5) reads

$$\mathcal{Q}_{\text{semi}}(\theta, \phi) = - \sum_{\substack{y_s, \\ s \in \mathbf{H}}} \int q_{\phi}(z_{0:T}, y_T^{\mathbf{H}}|x_{0:T}, y_T^{\mathbf{O}}) \log \left(\frac{q_{\phi}(z_{0:T}, y_T^{\mathbf{H}}|x_{0:T}, y_T^{\mathbf{O}})}{p_{\theta}(z_{0:T}, y_{0:T}, x_{0:T})} \right) dz_{0:T}. \quad (3.7)$$

Let us now discuss on the computation of (3.7). First, it is worthwhile to remark that it does not depend on the choice of the generative model. Any parameterized TMC model (3.1)-(3.3) can be used since $p_{\theta}(z_{0:T}, x_{0:T}, y_{0:T})$ is defined by the transition distribution $p_{\theta}(v_t|v_{t-1})$ and the initial distribution $p_{\theta}(v_0)$. Thus, its computation only depends on the choice of the variational distribution $q_{\phi}(z_{0:T}, y_T^{\mathbf{H}}|x_{0:T}, y_T^{\mathbf{O}})$, which can be factorized in two different ways.

The first factorization is given by

$$\begin{aligned} q_{\phi}(z_{0:T}, y_T^{\mathbf{H}}|x_{0:T}, y_T^{\mathbf{O}}) &= q_{\phi}^0 \times \prod_{t=1}^T q_{\phi}(z_t|z_{0:t-1}, y_{0:t-1}, x_{0:T}, y_{t+1:T}^{\mathbf{O}}) \times \\ &\quad \prod_{\substack{t \geq 1 \\ t \in \mathbf{H}}} q_{\phi}(y_t|y_{0:t-1}, z_{0:t}, x_{0:T}, y_{t+1:T}^{\mathbf{O}}), \end{aligned} \quad (3.8)$$

where $y_{0:t-1} = (y_{0:t-1}^{\mathbf{H}}, y_{0:t-1}^{\mathbf{O}})$, and

$$q_{\phi}^0 = \begin{cases} q(z_0|x_{0:T}, y_T^{\mathbf{O}}) & \text{if } t = 0 \in \mathbf{O}, \\ q_{\phi}(z_0|x_{0:T}, y_T^{\mathbf{O}})q_{\phi}(y_0|z_0, x_{0:T}, y_T^{\mathbf{O}}) & \text{otherwise.} \end{cases}$$

While the second one coincides with

$$\begin{aligned} q_\phi(z_{0:T}, y_T^{\mathbf{H}} | x_{0:T}, y_T^{\mathbf{O}}) = & q_\phi^0 \times \prod_{t=1}^T q_\phi(z_t | z_{0:t-1}, y_{0:t}, x_{0:T}, y_{t+1:T}^{\mathbf{O}}) \times \\ & \prod_{\substack{t \geq 1 \\ t \in \mathbf{H}}}^T q_\phi(y_t | y_{0:t-1}, z_{0:t-1}, x_{0:T}, y_{t+1:T}^{\mathbf{O}}), \end{aligned} \quad (3.9)$$

and

$$q_\phi^0 = \begin{cases} q(z_0 | x_{0:T}, y_T^{\mathbf{O}}) & \text{if } t = 0 \in \mathbf{O}, \\ q_\phi(z_0 | y_0, x_{0:T}, y_T^{\mathbf{O}}) q_\phi(y_0 | x_{0:T}, y_T^{\mathbf{O}}) & \text{otherwise.} \end{cases}$$

Once the variational distribution is chosen, and the generative model is fixed (*i.e.* the factorization of the transition distribution is fixed), the ELBO $\mathcal{Q}_{\text{semi}}(\theta, \phi)$ in (3.7) can be rewritten as

$$\mathcal{Q}_{\text{semi}}(\theta, \phi) = \mathcal{L}^{\mathbf{O}}(\theta, \phi) + \mathcal{L}^{\mathbf{H}}(\theta, \phi), \quad (3.10)$$

where

$$\mathcal{L}^{\mathbf{O}}(\theta, \phi) = \sum_{t \in \mathbf{O}} \left[\mathbb{E}_{q_\phi(z_t | \cdot)} (\log p(x_t | \cdot) + \log p(y_t | \cdot)) - D_{\text{KL}}(q_\phi(z_t | \cdot) || p_\theta(z_t | \cdot)) \right], \quad (3.11)$$

$$\begin{aligned} \mathcal{L}^{\mathbf{H}}(\theta, \phi) = & \sum_{t \in \mathbf{H}} \left[\mathbb{E}_{q_\phi(z_t, y_t | \cdot)} \log p(x_t | \cdot) - D_{\text{KL}}(q_\phi(z_t | \cdot) || p_\theta(z_t | \cdot)) \right. \\ & \left. - D_{\text{KL}}(q_\phi(y_t | \cdot) || p_\theta(y_t | \cdot)) \right]. \end{aligned} \quad (3.12)$$

$\mathcal{L}^{\mathbf{O}}$ and $\mathcal{L}^{\mathbf{H}}$ can be seen as the ELBOs associated to the observed and hidden labels, respectively.

3.3.2 Learning semi-supervised TMCs

Now, it remains to compute the ELBO $\mathcal{Q}_{\text{semi}}(\theta, \phi)$ (3.10), , which is not tractable in general. Moreover, we also deal with both discrete and continuous latent variables. We now present how to easily approximate the ELBO in this case.

Continuous latent variables: The ELBO $\mathcal{Q}_{\text{semi}}(\theta, \phi)$ involves computation of the expectation according to the variational distribution, $q_\phi(z_t | \cdot)$,

which is often intractable. We thus propose to use a Monte-Carlo approximation based on the reparameterization trick for continuous latent variables (see Section 1.2.1) similar to the one used in the PMC model. This technique allows us to sample from the variational distribution $q_\phi(z_t | \cdot)$. By sampling in this way, we obtain M differentiable samples $z_{0:T}^{(m)}$. The M samples $z_{0:T}^{(m)}$ are used to approximate the expectations. After this, an optimization algorithm can be used to estimate the parameters (See example 1.2.1).

Discrete latent variables: A static semi-supervised model with discrete latent variables has been proposed in (Kingma et al., 2014) and solves this problem by marginalizing out y over all the labels. However, this approach is not tractable when numerous labels are involved. In Chapter 1, we have presented the use of discrete variables in a VI framework (see Subsection 1.2.1) The Straight-Through Gumbel-Softmax estimator provides a way to relax discrete variables, making them differentiable and amenable to gradient-based optimization. In addition, the expectation with respect to the variational distribution $q_\phi(y_t | \cdot)$ is evaluated with a single relaxed sample (Andriyash et al., 2018; Jang et al., 2017).

In summary, this approach combines the (classical) reparameterization trick for continuous latent variables and the G-S trick for discrete latent variables, in order to obtain differentiable samples from the variational distributions $q_\phi(z_t | \cdot)$ and $q_\phi(y_t | \cdot)$, respectively. These samples are used to approximate the ELBO (3.10), making it computationally feasible for optimization. In addition, the D_{KL} terms in (3.11) and (3.12) can be computed analytically since the variational distribution is assumed to be tractable. Algorithm 2 summarizes the proposed approach, where we represent the hidden labels as a stochastic vector, and the observed labels as a one-hot vector. In the case of S-T Gumbel-Softmax, in the forward pass, line 6 is followed by an argmax operation to discretize the samples (see Remark 1.2.1).

Estimation of y_t , for all $t \in \mathbf{H}$: Once we have an estimate ϕ^* of ϕ of the model with Algorithm 2, we can approximate the hidden labels $y_{0:T}^{\mathcal{H}}$, for all $t \in \mathcal{H}$. This can be done by using either the variational approximation $q_{\phi^*}(y_t | \cdot)$ or an importance sampling approach with weighting. In the variational approximation method, we sample from the variational distribution $q_\phi(y_t | \cdot)$, for all $t \in \mathcal{H}$, and obtain a complete sequence of labels $\hat{y}_{0:T}$. Alternatively, using the importance sampling approach, we would sample from the proposal

Algorithm 2 General parameter estimation for TMCs in semi-supervised classification context

Input: $(x_{0:T}, y_T^{\mathbf{O}})$, the data where y_t is one-hot encoded, for all $t \in \mathbf{O}$; ϱ , the learning rate; M the number of samples, τ the temperature parameter

Output: (θ^*, ϕ^*) , sets of estimated parameters

- 1: Initialize the parameters θ^0 and ϕ^0
- 2: $j \leftarrow 0$
- 3: **while** convergence is not attained **do**
- 4: Sample $z_0^{(m)} \sim q_{\phi^j}(z_0 | \cdot)$, for all $1 \leq m \leq M$.
- 5: Sample $z_t^{(m)} \sim q_{\phi^j}(z_t | z_{0:t}^{(m)}, \dots)$, for all $1 \leq m \leq M$, for all $1 \leq t \leq T$.
- 6: Sample $y_t^{G-S} \sim q_{\phi^j}(y_t | y_{0:t}^{G-M}, \dots)$, using the Gumbel-Softmax trick, for all $t \in \mathbf{H}$, with temperature τ .
- 7: Evaluate the (approximated) loss $\widehat{\mathcal{Q}_{\text{semi}}}(\theta^j, \phi^j)$ with the samples $z_{0:T}^{(m)}$ and y_t^{G-S} , for all $t \in \mathbf{H}$.
- 8: Compute the derivative of the loss function $\nabla_{(\theta, \phi)} \widehat{\mathcal{Q}_{\text{semi}}}(\theta, \phi)$ with the samples $z_{0:T}^{(m)}$ and y_t^{G-S} , for all $t \in \mathbf{H}$.
- 9: Update the parameters with gradient ascent

$$\begin{pmatrix} \theta^{(j+1)} \\ \phi^{(j+1)} \end{pmatrix} = \begin{pmatrix} \theta^j \\ \phi^j \end{pmatrix} + \varrho \nabla_{(\theta, \phi)} \widehat{\mathcal{Q}_{\text{semi}}}(\theta, \phi) \Big|_{(\theta^j, \phi^j)} \quad (3.13)$$

- 10: $j \leftarrow j + 1$
 - 11: **end while**
 - 12: $\theta^* \leftarrow \theta^j$
 - 13: $\phi^* \leftarrow \phi^j$
-

distribution and weight the samples to obtain an approximation of the hidden labels. This method can provide a more accurate estimation, especially when the variational approximation is not sufficiently close to the true posterior.

3.4. Experiments

In this section, we present the practical applications and effectiveness of the TMC model in a semi-supervised learning framework. We start by comparing our deep TMC models with existing probabilistic and deep learning models to highlight their advantages in terms of flexibility. Next, we detail binary data generation experiments that will be used for model comparison. Finally, we discuss the implementation of the semi-supervised classification task and present the results obtained with each model variant.

3.4.1 DTMC vs existing models

We have presented a general framework for semi-supervised learning with TMCs. It depends on the choice of the generative model, which is described by the transition distribution $p_\theta(v_t|v_{t-1})$. It has an impact on the performance of the model for a specific task (classification, prediction, detection, or generation). The choice of the variational distribution $q_\phi(z_{0:T}, y_T^H|x_{0:T}, y_T^O)$ is also crucial since it has an impact on the computational complexity of the model. Different models can be obtained by choosing different factorizations of the variational distribution. A general factorization is given by

$$\begin{aligned} q_\phi(z_t|\cdot) &= \tau(z_t; \psi_\phi^z(\cdot)), \\ q_\phi(y_t|\cdot) &= \varsigma(y_t; \psi_\phi^y(\cdot)), \text{ for } t \in \mathbf{H}, \end{aligned}$$

where $\varsigma(y_t; \cdot)$ (resp. $\tau(z_t; \cdot)$) is a probability distribution on Ω (resp. probability density function on \mathbb{R}^{d_z}). ψ_ϕ^y and ψ_ϕ^z are assumed to be differentiable functions w.r.t. ϕ (remember that (\cdot) denotes a non-specified dependence between the variables of the model). In the Deep TMC model, the set of parameters (θ, ϕ) of the generating and the variational distributions can be described by deep neural networks.

Now, we present two popular models in the literature, which have been proposed for semi-supervised classification tasks. First, we present a variation of the Variational Sequential Labeler (Chen et al., 2018) model based on our general model; and then we present the Semi-supervised Variational Recurrent Neural Network model proposed by Butepage et al. (2019). Both models are

considered as particular cases of the proposed framework and will be used in the experimental section to compare the performance of the proposed model.

On one hand, the VSL is a semi-supervised learning model for sequential data which has originally been proposed for the sequence labeling tasks in natural language processing, that is based on conditional VAEs (Pagnoni et al., 2018). We propose a variation of this model by considering a modified version where the context depends on the previous observation x_{t-1} and the current latent variable z_t (more details are given in the Appendix D). We refer to it as the modified Variational Sequential Labeler (mVSL) and the associated generative model is given by

$$p_\theta(v_t|v_{t-1}) \stackrel{\text{mVSL}}{=} p_\theta(x_t|z_t)p_\theta(y_t|z_t)p_\theta(z_t|x_{t-1}, z_{t-1}).$$

While the associated variational distribution satisfies factorization (3.8) with

$$q_\phi(z_t|z_{t-1}, y_{t-1}, x_{0:T}, y_{t+1:T}^O) = q_\phi(z_t|x_{0:T}), \quad (3.14)$$

$$q_\phi(y_t|y_{t-1}, z_t, x_{0:T}, y_{t+1:T}^O) = p_\theta(y_t|z_t), \text{ for all } t \in \mathbf{H}. \quad (3.15)$$

In this case, the ELBO (3.10) reduces to

$$\begin{aligned} \mathcal{Q}_{\text{semi}}(\theta, \phi) &\stackrel{\text{mVSL}}{=} \sum_{t \in \mathbf{O}} \mathbb{E}_{q_\phi(z_t|x_{0:T})} (\log p_\theta(y_t|z_t)) + \\ &\quad \sum_{t=0}^T \left[\mathbb{E}_{q_\phi(z_t|x_{0:T})} \log p_\theta(x_t|z_t) \right. \\ &\quad \left. - D_{\text{KL}}(q_\phi(z_t|x_{0:T})||p_\theta(z_t|x_{t-1}, z_{t-1})) \right], \end{aligned}$$

where $x_{-1} = z_{-1} = \emptyset$.

On the other hand, the generative model used in the SVRNN model is a particular case of the TMC model where the latent variable z_t consists of the pair $z_t = (z'_t, h_t)$. The associated transition distribution reads:

$$p_\theta(v_t|v_{t-1}) \stackrel{\text{SVRNN}}{=} p_\theta(y_t|v_{t-1})p_\theta(z_t|y_t, v_{t-1})p_\theta(x_t|y_t, z_t, v_{t-1}),$$

where

$$\begin{aligned} p_\theta(y_t|v_{t-1}) &= p_\theta(y_t|h_{t-1}), \\ p_\theta(z_t|y_t, v_{t-1}) &= \delta_{f_\theta(z'_t, y_t, x_t, h_{t-1})}(h_t) \times p_\theta(z'_t|y_t, h_{t-1}), \\ p_\theta(x_t|y_t, z_t, v_{t-1}) &= p_\theta(x_t|y_t, z'_t, h_{t-1}), \end{aligned}$$

and where f_θ is a deterministic, *i.e.* the variable z'_t is a stochastic latent variable and h_t is deterministically given by $h_t = f_\theta(z'_t, x_t, y_t, h_{t-1})$, where f_θ is a function parameterized by a RNN, for example. The variational distribution $q_\phi(z_{0:T}, y_T^{\mathbf{H}} | x_{0:T}, y_T^{\mathbf{O}})$ satisfies the factorization (3.9) with

$$\begin{aligned} q(z'_t | z_{t-1}, y_t, x_{0:T}, y_{t+1:T}^{\mathbf{O}}) &= q_\phi(z'_t | x_t, y_t, h_{t-1}), \\ q(y_t | y_{t-1}, z'_{t-1}, x_{0:T}, y_{t+1:T}^{\mathbf{O}}) &= q_\phi(y_t | x_t, h_{t-1}). \end{aligned}$$

The ELBO of the SVRNN model is given by

$$\mathcal{Q}_{\text{semi}}(\theta, \phi) \stackrel{\text{SVRNN}}{=} \mathcal{L}^{\mathbf{O}}(\theta, \phi) + \mathcal{L}^{\mathbf{H}}(\theta, \phi) + J^{\mathbf{O}}(\theta, \phi),$$

where

$$\begin{aligned} \mathcal{L}^{\mathbf{O}}(\theta, \varphi) &= \sum_{t \in \mathbf{O}} \mathbb{E}_{q_\phi(z'_t | x_t, y_t, h_{t-1})} \log p_\theta(x_t | y_t, z'_t, h_{t-1}) + \log(p_\theta(y_t | h_{t-1})) \\ &\quad - D_{\text{KL}}(q_\phi(z'_t | x_t, y_t, h_{t-1}) || p(z'_t | y_t, h_{t-1})), \end{aligned} \quad (3.16)$$

$$\begin{aligned} \mathcal{L}^{\mathbf{H}}(\theta, \varphi) &= \sum_{t \in \mathbf{H}} \mathbb{E}_{q_\phi(z'_t, y_t | x_t, h_{t-1})} \log p_\theta(x_t | y_t, z'_t, h_{t-1}) \\ &\quad - D_{\text{KL}}(q_\phi(z'_t | x_t, y_t, h_{t-1}) || p_\theta(y_t | h_{t-1})) \\ &\quad - D_{\text{KL}}(q_\phi(y_t | x_t, h_{t-1}) || p_\theta(y_t | h_{t-1})), \end{aligned} \quad (3.17)$$

$$J^{\mathbf{O}}(\theta, \phi) = \sum_{t \in \mathbf{O}} \mathbb{E}_{\tilde{p}(y_t, x_t)} \log(p_\theta(y_t | h_{t-1}) q_\phi(y_t | x_t, h_{t-1})), \quad (3.18)$$

where $\tilde{p}(y_t, x_t)$, for $t \in \mathbf{O}$, denotes the empirical distribution of the data. Their final ELBO does not coincide with (3.10). The reason why is that they derive it from the static case (Jang et al., 2017) and add a penalization term $J^{\mathbf{O}}(\theta, \phi)$ that encourages $p_\theta(y_t | h_{t-1})$ and $q_\phi(y_t | x_t, h_{t-1})$ to be close to the empirical distribution of the data. Since h_t is deterministic given $(z'_t, x_t, y_t, h_{t-1})$, its posterior distribution becomes trivial, and thus there is no need to consider a variational distribution for it.

3.4.2 Binary data generation

We used the Binary Shape Database¹. and focused on both *cattle*-type and *camel*-type images. To transform these images into a 1-D signal ($x_{0:T}$), we used a Hilbert-Peano filling curve (Sagan, 2012). To evaluate the models presented in Section 3.4.1, we introduced non-linear blurring to highlight their

¹<http://vision.lems.brown.edu/content/available-software-and-databases>

ability to learn and correct for signal corruption. More precisely, we generated an artificial noise for the *cattle*-type by generating x_t according to

$$x_t|y_t, x_{t-1} \sim \mathcal{N}\left(\sin(a_{\omega_1} + x_{t-1}); \sigma^2\right), \quad (3.19)$$

where $a_{\omega_1} = 0$, $a_{\omega_2} = 0.4$ and $\sigma^2 = 0.25$. We now consider the *camel*-type image which is corrupted with a stationary multiplicative noise (non-elementary noise) given by

$$x_t|y_t, z_t \sim \mathcal{N}\left(a_{y_t}; b_{y_t}^2\right) * z_t, \quad (3.20)$$

where $z_t \sim \mathcal{N}(0, 1)$, $a_{\omega_1} = 0$, $a_{\omega_2} = 0.5$ and $b_{\omega_1} = b_{\omega_2} = 0.2$.

The generated images are presented in Figure 3.1(a) and Figure 3.2(a), respectively. Additionally, we randomly selected pixels $y_t \in y_T^O$, with a percentage of the pixels being labeled, and the rest considered unobserved or hidden (*e.g.* Figure 3.1(c) and Figure 3.2(c)).

3.4.3 Semi-supervised binary image segmentation

Our goal is to recover the segmentation of a binary image ($\Omega = \{\omega_1, \omega_2\}$) from the noisy observations $x_{0:T}$ when a partial segmentation y_T^O is available. In particular, $\vartheta(y_t; \cdot)$ (resp. $\varsigma(y_t; \cdot)$) is set as a Bernoulli distribution with parameters ρ_y^p (resp. ρ_y^q). As for the distribution $\zeta(x_t; \cdot)$ (resp. $\eta(z_t; \cdot)$) and $\tau(z_t; \cdot)$, we set it as a Gaussian distribution with parameters $[\mu_\theta^x, \text{diag}(\sigma_\theta^x)]$ (resp. $[\mu_\theta^z, \text{diag}(\sigma_\theta^z)]$ and $[\mu_z^q, \text{diag}(\sigma_z^q)]$), where $\text{diag}(\cdot)$ denotes the diagonal matrix deduced from the values of $\sigma_{\cdot,t}$.

In our simulations, we consider three particular cases of this deep TMC model which read as follows:

$$p_\theta(v_t|v_{t-1}) \stackrel{\text{TMC-I}}{=} p_\theta(y_t|y_{t-1})p_\theta(z_t|z_{t-1})p_\theta(x_t|y_t, z_t), \quad (3.21)$$

$$p_\theta(v_t|v_{t-1}) \stackrel{\text{TMC-II}}{=} p_\theta(y_t|y_{t-1}, x_{t-1})p_\theta(z_t|z_{t-1})p_\theta(x_t|y_t, z_t), \quad (3.22)$$

$$p_\theta(v_t|v_{t-1}) \stackrel{\text{TMC-III}}{=} p_\theta(y_t|y_{t-1}, x_{t-1})p_\theta(z_t|z_{t-1})p_\theta(x_t|y_t, z_t, x_{t-1}). \quad (3.23)$$

The TMC-I (3.21) model assumes a Markovian distribution for the labels and the latent variables aim at learning the distribution of the noise given the label and the latent variable. In the TMC-II (3.22), and TMC-III (3.23) models the Markovianity assumption for the labels is relaxed. The TMC-III model also considers the previous observation x_{t-1} as an additional input to the distribution of the observation x_t .

In order to capture temporal dependencies in the input data and to have an efficient computation of the variational distribution for the DTMC models, we use a deterministic function to generate \tilde{h}_t which takes as input $(x_t, y_t, z_t, \tilde{h}_{t-1})$. After this, the variational distribution $q_\phi(z_{0:T}, y_T^{\mathbf{H}} | x_{0:T}, y_T^{\mathbf{O}})$ satisfies the factorization (3.9) with $q_\phi(z_t | x_t, y_t, \tilde{h}_{t-1})$ and $q_\phi(y_t | x_t, \tilde{h}_{t-1})$. In the TMC-I case, the parameters are given by:

$$\begin{aligned} [\mu_\theta^x, \sigma_\theta^x] &= \psi_\theta^x(y_t, z_t), \\ [\mu_\theta^z, \sigma_\theta^z] &= \psi_\theta^z(z_{t-1}), \\ \rho_y^p &= \psi_\theta^y(y_{t-1}), \\ [\mu_z^q, \sigma_z^q] &= \psi_\phi^z(x_t, y_t, \tilde{h}_{t-1}), \\ \rho_y^q &= \psi_\phi^y(x_t, \tilde{h}_{t-1}). \end{aligned}$$

In the TMC-II and TMC-III cases, the parameters are given in the same way, except that ψ_θ^x and ψ_ϕ^y take x_{t-1} as an additional input.

3.4.4 Results

Each model was trained using stochastic gradient descent to optimize the negative associated ELBO, with the Adam optimizer Kingma & Ba (2015). The neural networks $\psi_{(\cdot)}^{(\cdot)}$ were designed with two hidden layers using rectified linear units and appropriate outputs, such as linear, softplus, and sigmoid. To ensure a fair comparison, we matched the total number of parameters of all models to be approximately equal. As a result, the number of hidden units for each hidden layer differs for each model. In fact, the SVRNN, TMC-I, TMC-II, TMC-III, and VLS models have 14, 25, 25, 24, and 30 hidden units, respectively. We used an RNN cell to generate \tilde{h}_t (resp. h_t) for the DTMC (resp. SVRNN) models. In the VLS model, we used the parameterization approach for $q_\phi(z_t | x_{0:T})$ presented in Chen et al. (2018), which involves using an RNN cell and with a regularization term equal to 0, 1. We also added a penalization term used in the SVRNN to the ELBO of the TMC-I, TMC-II, and TMC-III models that was presented in Section 3.4.1.

The performance of the models is evaluated in terms of the error rate (ER) of the reconstruction of the unobserved pixels, which are estimated by using the variational approximation approach. Table 3.1 presents the average of the error rates obtained for reconstructing unobserved pixels on all the *name*-type images. The notation *name* % is used to indicate the specific image set and the percentage of unobserved labels in the image. As shown in the table, the deep TMC models consistently outperform the VSL and the SVRNN, achieving a lower average error rate for each image set.

Model	Data sets and % of unlabeled pixels		
	Cattle 40%	Cattle 60%	Cammel 60%
VSL	20,59%	22,38%	18,82%
SVRNN	14,92%	20,12%	16,80%
DTMC-I	3,50%	6,44%	4,50%
DTMC-II	2,95%	5,53%	4,25%
DTMC-III	3,21%	6,09%	4,59%

Table 3.1: Average error rates of the reconstruction of the unobserved pixels on different sets of images with different percentages of unobserved pixels.

Moreover, our algorithm achieves superior performance for both noises. Figure 3.1 (resp. Figure 3.2) displays the performance of our proposed algorithms compared to the VSL and the SVRNN on a *cattle*-type (resp. *camel*-type) image with 60% (resp. 60%) of unobserved labels. In particular, we observe that in the VSL model, the error is mainly due to the misclassification of the black pixels (Figure 3.1(d) and Figure 3.2(d)). While for the SVRNN, the error results from the misclassification of the two classes (Figure 3.1(e) and Figure 3.2(e)).

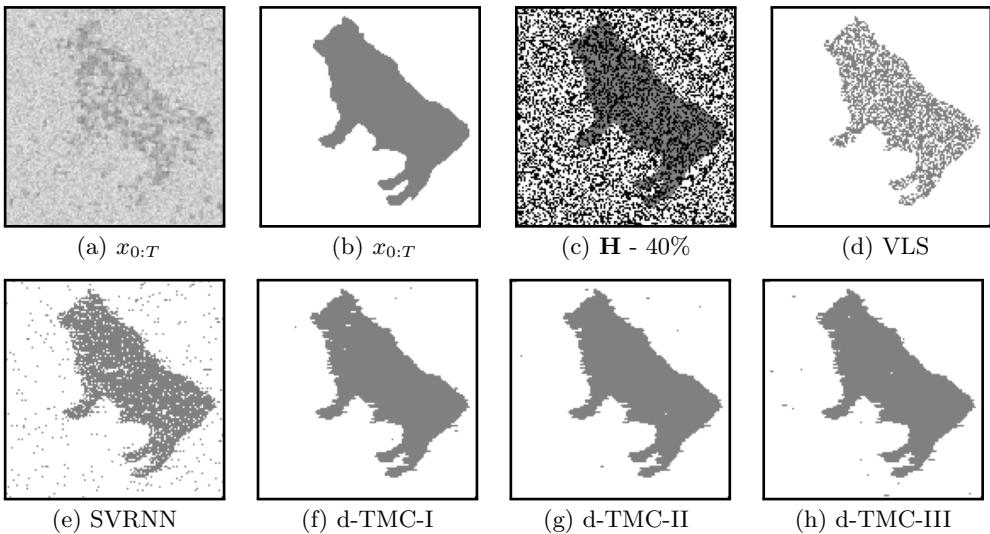


Figure 3.1: Semi-supervised image segmentation with d-TMC models with 40% of unlabeled pixels.

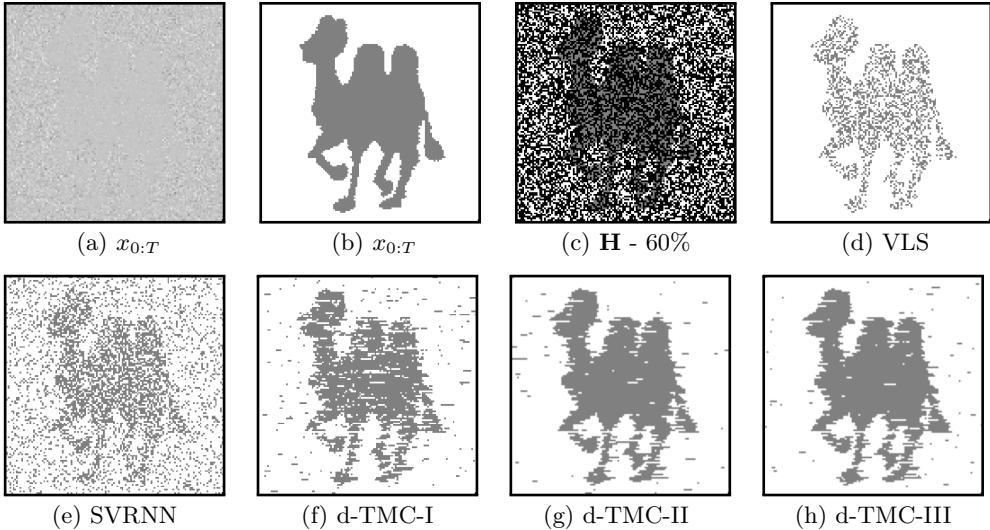


Figure 3.2: Semi-supervised image segmentation with d-TMC models with 60% of unlabeled pixels.

3.5. Conclusions

In this chapter, we presented a semi-supervised latent variable generative model. By exploring the TMC model, we have illustrated the feasibility of creating a diverse set of generative models based on the VI. This approach is particularly advantageous when dealing with data sets in which only a subset of the observations are labeled. The model we propose is capable of learning and representing a wide range of data features. It can effectively handle discrete labels and continuous feature observations over time, providing capabilities to classify, predict labels, and generate new feature sequences. This versatility makes the model particularly suitable for complex temporal data scenarios. The results of our experiments support the effectiveness of our approach in achieving good performance in the task of binary image segmentation.

CHAPTER 4

Deep Markov models for unsupervised classification

Contents

4.1	Introduction	66
4.2	PMCs for unsupervised classification	67
4.2.1	Bayesian inference for PMCs	68
4.2.2	Deep PMCs for unsupervised classification	71
4.2.3	Simulations	73
4.3	TMCs for unsupervised classification	75
4.3.1	Variational Inference for general TMCs	76
4.3.2	Estimation algorithm for TMCs	77
4.3.3	Deep TMCs for unsupervised classification	83
4.3.4	Simulations	86
4.4	Experiments on real datasets	90
4.4.1	Unsupervised segmentation of biomedical images	91
4.4.2	Unsupervised clustering for human activity recognition .	91
4.5	Conclusions	93

4.1. Introduction

In the previous Chapters 2 and 3, we have introduced the PMC and TMC models as frameworks for generative models, supervised and semi-supervised classification. In this chapter, we consider the problem of unsupervised classification where only the sequence of observations $x_{0:T}$ is observed, and that we want to estimate the sequence of hidden labels $y_{0:T}$. We recall that the estimation of y_t from $x_{0:T}$, for all t , $0 \leq t \leq T$, relies on the unknown posterior distribution $p(y_t|x_{0:T})$,

$$p_\theta(y_t|x_{0:T}) = \frac{\sum_{y_{0:t-1}, y_{t+1:T}} p_\theta(x_{0:T}, y_{0:T})}{\sum_{y_{0:T}} p_\theta(x_{0:T}, y_{0:T})},$$

which can be derived from the distribution $p_\theta(y_{0:T}, x_{0:T})$ or $p_\theta(z_{0:T}, y_{0:T}, x_{0:T})$ since $p_\theta(y_{0:T}, x_{0:T}) = \int p_\theta(z_{0:T}, y_{0:T}, x_{0:T}) dz_{0:T}$. Thus, we continue to consider the PMC and TMC models, where their associated conditional distributions can be parameterized by universal approximators (DNNs) under the constraint that $y_{0:T}$ is an interpretable hidden process. As we will see, this particular constraint requires us to review previous techniques to include the learning of an interpretable label.

This chapter is organized in three parts. First, we give up the latent variable z_t and consider a PMC model (4.1) without any latent variable. We directly parameterize the joint distribution $p_\theta(y_{0:T}, x_{0:T})$ of a PMC. We continue considering a DNN parameterization and an ad hoc procedure based on a pretraining of DNNs which aims at transforming a simple and interpretable model such as (1.6) into a complex probabilistic architecture while keeping this interpretability constraint. We show that it is possible to adapt existing Bayesian inference algorithms to our models and the VI framework is not necessary in the PMC case.

Next, we reintroduce the continuous latent variables $z_{0:T}$, and propose a modified VI framework to estimate the parameters of the model which takes into account the interpretability of $y_{0:T}$ and also the different roles of $y_{0:T}$ and $z_{0:T}$. We also propose a Sequential Monte Carlo algorithm (Doucet & Johansen, 2009) based on the previous variational framework to obtain the final estimates of y_t . For each model, we perform simulations to evaluate to what extent our generalized models lead to a better estimation of the hidden states y_t . Most of the simulations on synthetic and real data are run in the context of unsupervised image segmentation (as in Chapter 3).

4.2. PMCs for unsupervised classification

In this section, we do not consider the latent variable z_t in order to build a solution on a model without latent variables, which is already challenging due to the absence of the labels y_t associated to the observations x_t . We adapt the PMC model discussed in Chapter 2 to the unsupervised classification problem, where the pair (z_t, x_t) is replaced by (y_t, x_t) , where y_t is a discrete r.v. This modification addresses the need for interpretable models.

The PMC model reads

$$p_{\theta}(y_{0:T}, x_{0:T}) = p_{\theta}(y_0) \prod_{t=1}^T p_{\theta}(y_t, x_t | y_{t-1}, x_{t-1}), \quad (4.1)$$

where the factorization of the transition distribution is given by

$$p_{\theta}(y_t, x_t | y_{t-1}, x_{t-1}) = p_{\theta}(y_t | y_{t-1}, x_{t-1}) p_{\theta}(x_t | y_{t-1:t}, x_{t-1}). \quad (4.2)$$

We also define the Semi Pairwise Markov Chain (SPMC), a particular instance of the PMC model, where the observation x_t does not depend on y_{t-1} , given (y_t, x_{t-1}) , *i.e.*

$$p_{\theta}(y_t, x_t | y_{t-1}, x_{t-1}) = p_{\theta}(y_t | y_{t-1}, x_{t-1}) p_{\theta}(x_t | y_t, x_{t-1}). \quad (4.3)$$

This model is particularly interesting in the context of unsupervised classification, where the interpretability problem may be easier. Figure 4.1 illustrates the graphical representation of the PMC model and its particular instances that we consider in this section, *i.e.* the SPMC, and the HMC models.

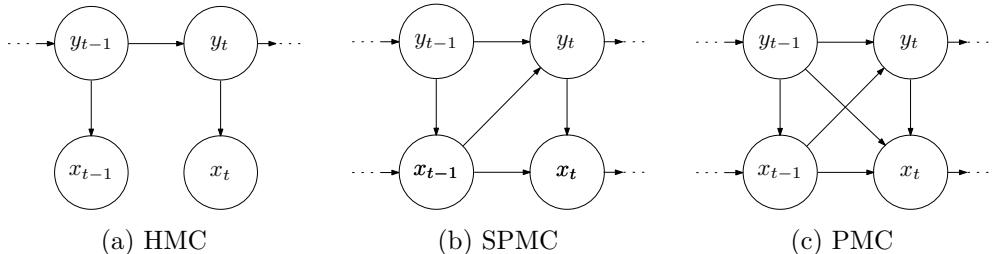


Figure 4.1: Graphical representations of the HMC, SPMC, and PMC models.

We revisit the general parameterization of the PMC model introduced in Chapter 2 to adapt it to the unsupervised classification problem. We parameterize the conditional distributions in (4.2) as

$$p_{\theta}(y_t | y_{t-1}, x_{t-1}) = \vartheta(y_t; \psi_{\theta}^y(y_{t-1}, x_{t-1})), \quad (4.4)$$

$$p_{\theta}(x_t | y_{t-1:t}, x_{t-1}) = \zeta(x_t; \psi_{\theta}^x(y_{t-1:t}, x_{t-1})). \quad (4.5)$$

Example 4.2.1. Let us show that this general parameterization includes the classical HMC with independent Gaussian noise (HMC-IN). Let us assume that $\Omega = \{\omega_1, \omega_2\}$ and $x_t \in \mathbb{R}$. In this case, the HMC-IN model can be described as

$$\psi_\theta^y(y_{t-1}, x_{t-1}, z_t) = \text{sigm}(b_{y_{t-1}}), \quad (4.6)$$

$$\psi_\theta^x(y_t) = [d_{y_t}, \sigma_{y_t}], \quad (4.7)$$

$$\vartheta(y_t; \rho) = \mathcal{B}er(y_t; \rho), \quad (4.8)$$

$$\zeta(x_t; s = [\mu, \sigma]) = \mathcal{N}(x_t; \mu, \sigma^2), \quad (4.9)$$

Indeed, (4.6)- (4.7) only depend on y_{t-1} and on y_t , respectively. Thus, we have $p_\theta(y_t = \omega_1 | y_{t-1} = \omega_i) = \text{sigm}(b_{\omega_i})$ and $p_\theta(x_t | y_t = \omega_j) = \mathcal{N}(x_t; d_{\omega_j}; \sigma_{\omega_j}^2)$. Finally, the set of parameters is given by $\theta = (b_{\omega_i}, d_{\omega_j}, \sigma_{\omega_j} | (\omega_i, \omega_j) \in \Omega \times \Omega)$. As a further illustrative example in the binary case, it is possible to start from this particular parameterization of HMCs to derive a linear and Gaussian PMC model in which we introduce dependencies on x_{t-1} and y_{t-1} . In this case, ϑ and ζ are unchanged but ψ_θ^y and ψ_θ^x now read as

$$\psi_\theta^y(y_{t-1}, x_{t-1}) = \text{sigm}(a_{y_{t-1}} x_{t-1} + b_{y_{t-1}}), \quad (4.10)$$

$$\psi_\theta^x(y_{t-1:t}, x_{t-1}) = [c_{y_{t-1}, y_t} x_{t-1} + d_{y_{t-1}, y_t}; \sigma_{y_t, y_{t-1}}]. \quad (4.11)$$

The set of parameters is now given by $\theta = (a_{\omega_i}, b_{\omega_i}, c_{\omega_j, \omega_i}, d_{\omega_j, \omega_i}, \sigma_{\omega_j, \omega_i} | (\omega_j, \omega_i) \in \Omega^2)$. As we will see later, these models play a critical role in the construction of parameterization based on DNNs. Indeed, despite their simple form, they generally provide an interpretable classification.

We now show that under this framework it is possible to derive an unsupervised estimation algorithm which approximates the ML estimate of θ , no matter the choice of the parameterization ψ_θ^y and ψ_θ^x . In particular, we use a direct ML approach rather than an EM one (see Remark 4.2.1) and introduce a pretraining approach for deep parameterizations. This pretraining approach is a novel contribution that will be detailed in the next sections. Once θ has been estimated, we resort to the classical estimation of the posterior distributions $p_\theta(y_t | x_{0:T})$.

4.2.1 Bayesian inference for PMCs

Estimation of θ

Since the hidden variable y_t is discrete, the likelihood $p_\theta(x_{0:T})$ can be computed exactly. This accessibility is a key point in the estimation of θ . Here, the

VI method is not necessary to approximate the likelihood (equivalently, the optimal variational distribution is available). Given the differentiability of the functions ψ_θ^y , ψ_θ^x , ϑ , and ζ , we can propose a gradient ascent method on the likelihood $p_\theta(x_{0:T})$ to approximate the ML estimate of θ . This gradient ascent method is based on the sequential computation of $\alpha_{\theta,t}(y_t) = p_\theta(y_t, x_{0:t})$, for all t , $0 \leq t \leq T$, from which we deduce the likelihood

$$p_\theta(x_{0:T}) = \sum_{y_T} \alpha_{\theta,T}(y_T). \quad (4.12)$$

Based on the Markovian property of (4.1) and on the general parameterization (4.4)-(4.5), the coefficients $\alpha_{\theta,T}(y_T)$ can be computed recursively from (Pieczynski, 2003) as

$$\alpha_{\theta,t}(y_t) = \sum_{y_{t-1}} \alpha_{\theta,t-1}(y_{t-1}) \vartheta(y_t; \psi_\theta^y(y_{t-1}, x_{t-1})) \zeta(x_t; \psi_\theta^x(y_{t-1:t}, x_{t-1})). \quad (4.13)$$

Consequently, the gradient of the likelihood $p_\theta(x_{0:T})$ (or equivalently that of the log-likelihood) w.r.t. θ can be deduced from that of $\alpha_{\theta,t}$, which is itself sequentially computable by using the decomposition (4.13) because $p(y_t, x_{0:t}) = \sum_{y_{t-1}} p(y_{t-1:t}, x_{0:t}) = \sum_{y_{t-1}} p(y_{t-1:t}, x_{0:t-1}) p(y_t, x_t | y_{t-1}, x_{t-1})$. This sequential structure has the advantage that numerical auto-differentiation methods can be used to compute such gradients in practice (Paszke et al., 2019). The estimation of θ can thus be deduced from an iterative gradient ascent method based on a learning rate ϵ and, for example, on the update

$$\theta^{(j+1)} = \theta^{(j)} + \epsilon \nabla_\theta \log p_\theta(x_{0:T}) \Big|_{\theta=\theta^{(j)}}. \quad (4.14)$$

The unsupervised estimation of θ is summarized in Algorithm 3. The gradients can be computed automatically through auto-differentiation tools, e.g. JAX by Bradbury et al. (2018).

Remark 4.2.1. Generally, the parameter estimation procedure for a probabilistic model with hidden r.v. is based on the EM algorithm (Dempster et al., 1977) (see Algorithm 9 in Appendix A). It relies on the computation of

$$Q(\theta, \theta^{(j)}) = \mathbb{E}_{p_{\theta^{(j)}}(y_{0:T}|x_{0:T})} (\log p_\theta(y_{0:T}, x_{0:T}))$$

followed by the maximization of $Q(\theta, \theta^{(j)})$ w.r.t. θ . However, for general parameterizations (4.4)-(4.5), the maximization step cannot be computed analytically. In this case, it is possible to use a gradient-EM approach to replace the maximization step, but it is then strictly equivalent and computationally

more demanding than computing the gradient of the log-likelihood (Xu & Jordan, 1996; Balakrishnan et al., 2017) as we propose in (4.14). Finally, for particular parameterizations for which the maximization step is computable, the comparison between these two approaches is an open question and is out of scope of this thesis.

Algorithm 3 Unsupervised estimation of θ in general PMC models.

Input: A realization $x_{0:T}$, a set of estimated parameters θ^*

Output: θ^* , a set of estimated parameters

- 1: $j = 0$
 - 2: **while** convergence of $\log p_{\theta^{(j)}}(x_{0:T})$ is not attained **do**
 - 3: Compute $\log \alpha_{\theta^{(j)},t}(y_t)$ and $\nabla_\theta \log \alpha_{\theta^{(j)},t}(y_t) \Big|_{\theta=\theta^{(j)}}$, for all $y_t \in \Omega$, for all $0 \leq t \leq T$, with (4.13)
 - 4: Compute $\log p_{\theta^{(j)}}(x_{0:T})$ and $\nabla_\theta \log p_{\theta^{(j)}}(x_{0:T}) \Big|_{\theta=\theta^{(j)}}$, with (4.12)
 - 5: Set $\theta^{(j+1)} = \theta^{(j)} + \epsilon \nabla_\theta \log p_{\theta}(x_{0:T}) \Big|_{\theta=\theta^{(j)}}$
 - 6: $j \leftarrow j + 1$
 - 7: **end while**
 - 8: $\theta^* \leftarrow \theta^{(j)}$
-

Estimation of y_t

Once we have obtained an estimate θ^* of θ , it remains to compute $p_{\theta^*}(y_t | x_{0:T})$, for all t . Since we deal with particular PMCs, it can be done by following the steps of Pieczynski (2003), *i.e.* by using the Markovian property of (4.1) and by introducing the backward coefficients $\beta_{\theta^*,t}(y_t) = p_{\theta^*}(x_{t+1:T} | y_t, x_t)$, for all t , with $\beta_{\theta^*,T}(y_T) = 1$. These coefficients can be computed sequentially from

$$\beta_{\theta^*,t-1}(y_{t-1}) = \sum_{y_t} \beta_{\theta^*,t}(y_t) \vartheta(y_t; \psi_{\theta^*}^y(y_{t-1}, x_{t-1})) \zeta(x_t; \psi_{\theta^*}^x(y_{t-1:t}, x_{t-1})). \quad (4.15)$$

Thus, we deduce

$$p_{\theta^*}(y_{t-1:t} | x_{0:T}) \propto \alpha_{\theta^*,t-1}(y_{t-1}) \times \beta_{\theta^*,t}(y_t) \times \vartheta(y_t; \psi_{\theta^*}^y(y_{t-1}, x_{t-1})) \times \zeta(x_t; \psi_{\theta^*}^x(y_{t-1:t}, x_{t-1})), \quad (4.16)$$

$$p_{\theta^*}(y_t | x_{0:T}) = \sum_{y_{t-1}} p_{\theta^*}(y_{t-1:t} | x_{0:T}). \quad (4.17)$$

The computation of the MAP estimate of y_t is summarized in Algorithm 4.

Algorithm 4 Unsupervised estimation of y_t in general PMC models.

Input: A realization $x_{0:T}$, a set of estimated parameters θ^*

Output: $\hat{y}_{0:T}$, the estimated hidden r.v.

- 1: Compute $\alpha_{\theta^*,t}(y_t)$, for all $y_t \in \Omega$, for all $0 \leq t \leq T$, with (4.13)
 - 2: Compute $\beta_{\theta^*,t}(y_t)$, for all $y_t \in \Omega$, for all $0 \leq t \leq T$, with (4.15)
 - 3: Compute $p_{\theta^*}(y_{t-1:t}|x_{0:T})$, for all $y_{t-1:t} \in \Omega \times \Omega$, for all $0 \leq t \leq T$, with (4.16)
 - 4: Compute $\hat{y}_t = \arg \max p_{\theta^*}(y_t|x_{0:T})$, for all $0 \leq t \leq T$, with (4.17)
-

4.2.2 Deep PMCs for unsupervised classification

We consider the particular parameterization ψ_θ^y and ψ_θ^x of the distributions ϑ and ζ , respectively, where ψ_θ^y and ψ_θ^x are the outputs of two DNNs with (y_{t-1}, x_{t-1}) and $(y_{t-1:t}, x_{t-1})$ as inputs, respectively (as in Section 2.3.1). Note that a unique DNN is used for ψ_θ^y (resp. ψ_θ^x) overtime.

Since ψ_θ^y and ψ_θ^x are differentiable w.r.t. θ and their gradients are computable from the backpropagation algorithm (Rumelhart et al., 1986), Algorithm 3 can be directly applied to estimate θ . However, due to the large number of parameters of these architectures, some problems tend to appear in practice. In particular, a random initialization of θ can lead to convergence issues for the optimization of $\log p_\theta(x_{0:T})$. More importantly, the final r.v. y_t learned by such a model may no longer be interpretable, *i.e.* it is not ensured that y_t coincides with the original class associated to x_t . In other words, a direct application of Algorithm 3 tends to return a final model which gives poorer results than the simple models described in Section 4.2 in terms of classification, as it considers y_t as a latent variable rather than an interpretable label.

We propose a two-step solution based on a constrained output layer and on a pretraining which aims at initializing properly θ . This solution relies on a simple model such as the linear and Gaussian PMC described in Section 4.2 where the linear functions ψ_θ^y and ψ_θ^x in (4.10)-(4.11) can be seen as the output layer of an elementary DNN with no hidden layer. Rather than directly training the DNN associated to ψ_θ^y and ψ_θ^x , we first estimate the linear PMC model (4.10)-(4.11) with Algorithm 3 before adding intermediate layers. These layers are next pretrained from the classification obtained with the elementary model, and are finally finely trained with our ML approach.

Constrained output layer

The main idea of our constrained training step is to make coincide a subset of θ with the parameters of an elementary linear (equivalently a non-deep) PMC model (4.10)-(4.11) which is assumed to provide an interpretable classification. In other words, we first estimate an elementary linear PMC model with Algorithm 3, and we denote the set of associated parameters θ_{fr} , in the sense that these parameters are next *frozen* and will not be further updated. We next consider this linear layer as the output layer of a DNN where the other parameters are denoted θ_{ufr} , which are *unfrozen* in the sense that they have not been estimated yet.

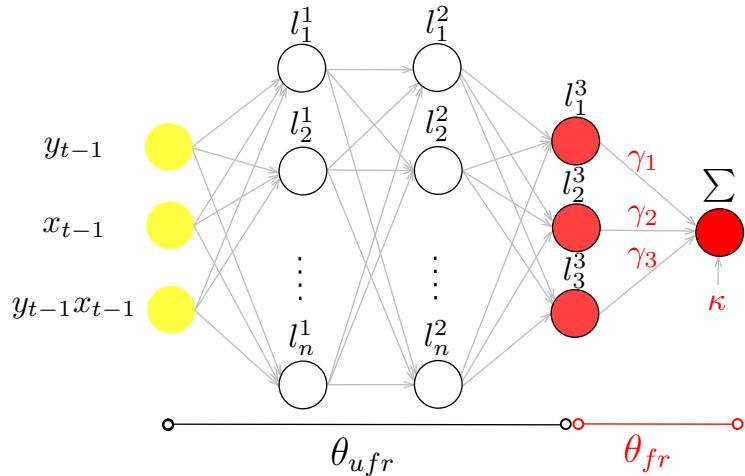


Figure 4.2: DNN architecture with constrained output layer for ψ_θ^y with two hidden layers. $\Sigma = \psi_\theta^y(y_{t-1}, x_{t-1}, y_{t-1}x_{t-1}) = \text{sigm}(\gamma_1 l_1^3 + \gamma_2 l_2^3 + \gamma_3 l_3^3 + \kappa)$, where the last layer parameters $\{\gamma_1, \gamma_2, \gamma_3, \kappa\}$ 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 $\kappa = b_{\omega_1}$.

The parameters θ_{fr} are related to the output layer which computes the function ψ_θ^y of the linear PMC model (4.10). Due to the one-hot encoding of the discrete r.v. y_{t-1} ($y_{t-1} = \omega_1 \leftrightarrow y_{t-1} = 0$ and $y_{t-1} = \omega_2 \leftrightarrow y_{t-1} = 1$), this parameterization is equivalent to that of (4.10) up to the given correspondence between $\theta_{fr} = (\gamma_1, \gamma_2, \gamma_3, \kappa)$ and $(a_{\omega_1}, a_{\omega_2}, b_{\omega_1}, b_{\omega_2})$. When the number of classes C increases, the size of the first and last layer increases due to the one-hot encoding of y_{t-1} . Linear activation functions are used in the last hidden layer in red.

Figure 4.2 describes an example of a constrained DNN architecture for the function ψ_θ^y when $\Omega = \{\omega_1, \omega_2\}$ and $\mathbb{R}^{d_x} = \mathbb{R}$, without loss of generality.

Pretraining by backpropagation

It remains to estimate the parameters θ_{ufr} of the intermediate hidden layers. The idea is to initialize them in a such way that the initial DPMC coincides with the elementary one; in other words, and due to the previous step, the output of the newly added hidden layers aims at coinciding with the identity function after the pretraining. After initializing randomly θ_{ufr} , our pretraining step aims at minimizing cost functions $C_{\psi_\theta^y}$ and $C_{\psi_\theta^x}$ which involve the pre-classification $\hat{y}_{0:T}^{\text{pre}}$. Typically, the cost function $C_{\psi_\theta^y}$ is the averaged overtime cross-entropy between the output of the DNN ψ_θ^y and \hat{y}_t^{pre} and $C_{\psi_\theta^x}$ is the mean square error between the output of ψ_θ^x and the parameters of the elementary linear models associated to $\hat{y}_{t-1:t}^{\text{pre}}$ (see Equation (4.11)). The minimization of these cost functions w.r.t. θ_{ufr} is done with the backpropagation algorithm. Finally, once θ_{ufr} has been properly initialized, it is fine-tuned with Algorithm 3 which approximates the ML estimate of θ . Algorithm 5 summarizes the two estimation steps specific to the DNN parameterization.

Remark 4.2.2. In order to estimate the parameters of our deep PMC, we have used a reverse approach w.r.t. the pretraining approaches proposed at the beginning of 2010s to help supervised learning in DNN (Erhan et al., 2010). Indeed, due to the large number of parameters in these architectures, (Mohamed et al., 2012; Glorot & Bengio, 2010; Hinton et al., 2012) have suggested to first pretrain in an unsupervised way a DNN from a generative probabilistic model which shares common parameters with the original DNN (*e.g.* a Deep Belief Network). The backpropagation algorithm for supervised estimation is next initialized with the (approximated) ML estimate of this probabilistic model. Here, we have started to pretrain our architecture in a supervised way with a pre-classification and next embedded it in our original probabilistic model in which we compute an approximation of the ML estimate.

4.2.3 Simulations

We illustrate the performance of our models with the same binary image segmentation problem as in Chapter 3. In order to highlight our unsupervised approach, we consider the cattle-type images of the Binary Shape Database. The images are transformed into a 1-D signal $x_{0:T}$ with a Hilbert-Peano filling curve (Sagan, 2012). They are blurred with a noise which exhibits non-

Algorithm 5 A general estimation algorithm for deep parameterization of PMC models.

Input: $x_{0:T}$, the observation

Output: $\hat{y}_{0:T}$, the final classification

Linear model: initialization of the output layer of ψ_θ^y and ψ_θ^x (§ 4.2.2)

- 1: Initialize randomly $\theta_{\text{fr}}^{(0)}$
- 2: Estimate θ_{fr}^* using Algorithm 3 with $\theta_{\text{fr}}^{(0)}$
- 3: Estimate $\hat{y}_{0:T}^{\text{pre}}$ using Algorithm 4 with θ_{fr}^*

Pretraining of θ_{ufr} (§ 4.2.2)

- 4: Compute θ_{ufr}^* using Algorithm 3 with $\theta^{(0)} = (\theta_{\text{fr}}^*, \theta_{\text{ufr}}^{(0)})$ (θ_{fr}^* is not updated)
- 5: Compute $\hat{y}_{0:T}$ using Algorithm 4 with $\theta^* = (\theta_{\text{fr}}^*, \theta_{\text{ufr}}^*)$

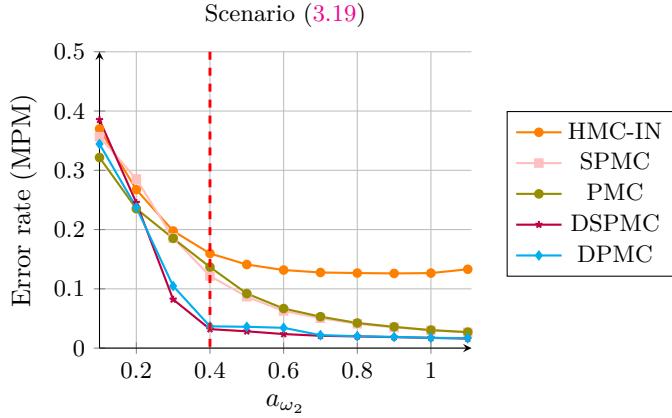
Complete deep model: fine-tuning

- 6: Compute θ_{ufr}^* using Algorithm 3 with $\theta^{(0)} = (\theta_{\text{fr}}^*, \theta_{\text{ufr}}^{(0)})$ (θ_{fr}^* is not updated)
 - 7: Compute $\hat{y}_{0:T}$ using Algorithm 4 with $\theta^* = (\theta_{\text{fr}}^*, \theta_{\text{ufr}}^*)$
-

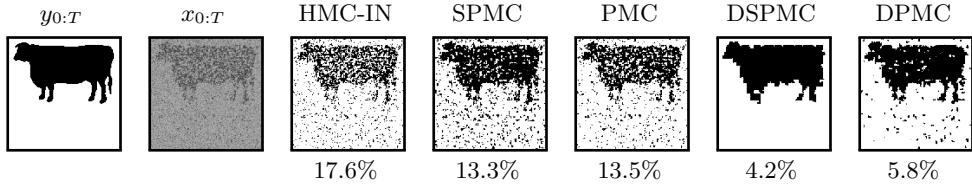
linearities to highlight the ability of the generalized PMC models to learn such a signal corruption generating x_t according to (3.19), with $a_{\omega_1} = 0$, $\sigma^2 = 0.25$ and a_{ω_2} is a varying parameter (see Subsection 3.4.2).

We next focus on two kinds of parameterizations of distributions ϑ and ζ which coincide with (4.8)-(4.9). Each parameterization is applied to the SPMC and PMC models (see Figure 4.1). First, we consider a linear parameterization (SPMC and PMC) based on (4.8)-(4.11). The second parameterization is a deep one (DSPMC and DPMC) and relies on one (unfrozen) hidden layer with 100 neurons and the ReLU activation function. For this architecture, we apply the training constraints discussed in Paragraph 4.2.2.

In Figure 4.3a, we display the averaged error rates for each model over all the selected images as a function of a_{ω_2} . Figure 4.3b displays the results of the classifications for a particular image of the database. As it can be observed, although the same Gaussian distribution ζ is used both models, the general PMC framework that we introduced leads to a great improvement of the elementary HMC model. Next, the deep parameterized models (DPMC and DSPMC) are the most accurate models and are able to capture the complexity by improving the results of their non-deep counterpart. More importantly, note that the gain obtained with our DPMC and DSPMC models does not require any further modeling effort in the sense that they are a particular parameterization in our general framework.



(a) Error rate from the unsupervised segmentations with a noise described by (3.19). 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 Figure 4.3a). Error rates appear below the images.

Figure 4.3: Unsupervised image segmentation with PMC models. Figure 4.3a displays averaged results while Figure 4.3b describes a particular classification.

4.3. TMCs for unsupervised classification

In this section, we extend the integration of a third latent process into our PMC model. This third continuous latent process $z_{0:T}$ can be used to implicitly estimate the nature of the distributions ϑ and ζ of our PMC or to model and learn the continuous non-stationarity of the process $(y_{0:T}, x_{0:T})$ since $p_\theta(y_{0:T}, x_{0:T}) = \int p_\theta(z_{0:T})p_\theta(y_{0:T}, x_{0:T}|z_{0:T})dz_{0:T}$. However, this integration poses computational challenges because a direct computation of the integrals w.r.t. z_t in (4.13) and (4.15) is intractable. Consequently, the likelihood and posterior distributions, $p_\theta(x_{0:T})$ and $p_\theta(y_t|x_{0:T})$ are no longer exactly computable in general. Here, we derive a new estimation algorithm based on VI (see Section 1.2.1), where the ELBO is a particular case of the semi-supervised case presented in Chapter 3. Moreover, a part of the variational distribution q_ϕ can be computed explicitly, which allows adjustments to be made in the model

learning phase. We also propose a modified version of the ELBO, which improves the interpretability of the labels by distinguishing them from the latent variables.

4.3.1 Variational Inference for general TMCs

In Chapter 3, we have introduced a VI framework for the case where the labels are partially observed. In this section, we consider the unsupervised case where all the labels are unobserved. Thus, the ELBO is simpler than in the semi-supervised case (3.7), and a part of the optimal variational distribution can be computed exactly (Proposition 4.3.1).

Let us recall the notation $v_t = (y_t, z_t, x_t)$ for the triplet. The TMC (1.8) can be seen as a PMC (4.4)-(4.5) in augmented dimension, *i.e.* a PMC where $(z_{0:T}, y_{0:T})$ plays the role of the hidden process. If $z_{0:T}$ were a discrete process, it would be possible to apply directly the Bayesian inference framework developed in Section 4.2.1. However, the continuous nature of z_t involves intractable integrals to compute sequentially the equivalent of (4.13), and therefore $p_\theta(x_{0:T})$. To overcome this issue, we introduce a variational distribution $q_\phi(z_{0:T}, y_{0:T}|x_{0:T})$, and deduce the ELBO of the TMC model for the unsupervised case:

$$\begin{aligned} \log p_\theta(x_{0:T}) &\geq \sum_{y_{0:T}} \int q_\phi(z_{0:T}, y_{0:T}|x_{0:T}) \log \left(\frac{p_\theta(z_{0:T}, y_{0:T}, x_{0:T})}{q_\phi(z_{0:T}, y_{0:T}|x_{0:T})} \right) dz_{0:T} \\ &= \mathcal{Q}_{\text{unsup}}(\theta, \phi). \end{aligned}$$

In the context of TMCs with a discrete and continuous latent process, Proposition 4.3.1 exploits the observation that

$$p_\theta(y_{0:T}|z_{0:T}, x_{0:T}) = p_\theta(y_T|z_{0:T}, x_{0:T}) \prod_{t=1}^T p_\theta(y_{t-1}|y_t, z_{0:T}, x_{0:T}) \quad (4.18)$$

is computable (see Appendix E) and shows that it is optimal (in the sense of the value of the ELBO) to restrict the choice of $q_\phi(z_{0:T}, y_{0:T}|x_{0:T})$ to that of $q_\phi(z_{0:T}|x_{0:T})$.

Proposition 4.3.1. Let us denote $\mathcal{Q}_{\text{unsup}}(\theta, \phi)$ and $\mathcal{Q}_{\text{unsup}}^{\text{opt}}(\theta, \phi)$, the ELBOs associated to the variational distributions

$$q_\phi(z_{0:T}, y_{0:T}|x_{0:T}) = q_\phi(z_{0:T}|x_{0:T})q_\phi(y_{0:T}|z_{0:T}, x_{0:T})$$

and

$$q_\phi^{\text{opt}}(z_{0:T}, y_{0:T} | x_{0:T}) = q_\phi(z_{0:T} | x_{0:T}) p_\theta(y_{0:T} | z_{0:T}, x_{0:T}),$$

respectively.

Then, for any (θ, ϕ) , we have

$$\log p_\theta(x_{0:T}) \geq \mathcal{Q}_{\text{unsup}}^{\text{opt}}(\theta, \phi) \geq \mathcal{Q}_{\text{unsup}}(\theta, \phi), \quad (4.19)$$

where

$$\mathcal{Q}_{\text{unsup}}^{\text{opt}}(\theta, \phi) = Q_0^{\text{opt}}(\theta, \phi) + \sum_{t=1}^T Q_{t-1,t}^{\text{opt}}(\theta, \phi) + Q_{0:T}^{\text{opt}}(\theta, \phi), \quad (4.20)$$

and where

$$Q_0^{\text{opt}}(\theta, \phi) = \int \sum_{y_0} q_\phi(z_{0:T} | x_{0:T}) p_\theta(y_0 | z_{0:T}, x_{0:T}) \log p_\theta(v_0) dz_{0:T}, \quad (4.21)$$

$$Q_{t-1,t}^{\text{opt}}(\theta, \phi) = \int \sum_{y_{t-1:t}} q_\phi(z_{0:T} | x_{0:T}) p_\theta(y_{t-1:t} | z_{0:T}, x_{0:T}) \times \log \left(\frac{p_\theta(v_t | v_{t-1})}{p_\theta(y_{t-1} | y_t, z_{0:T}, x_{0:T}) q_\phi(z_{0:T} | x_{0:T})} \right) dz_{0:T}, \quad (4.22)$$

$$Q_{0:T}^{\text{opt}}(\theta, \phi) = - \int \sum_{y_T} q_\phi(z_{0:T} | x_{0:T}) p_\theta(y_T | z_{0:T}, x_{0:T}) \log p_\theta(y_T | z_{0:T}, x_T) dz_{0:T}. \quad (4.23)$$

A proof of Proposition 4.3.1 is given in Appendix E. The practical computation of these integrals will be described later with the modified objective function.

4.3.2 Estimation algorithm for TMCs

Following the approach that we have developed for PMC models, we extend our parameterization framework to the distributions of TMC models. As a direct extension of Section 4.2, functions ψ_θ^y and ψ_θ^x can now depend on $z_{t-1:t}$. We present a particular parameterization of TMCs models derived from the general one introduced in Section 3.2.1. The transition distribution is factorized as follows:

$$p_\theta(v_t | v_{t-1}) = p_\theta(z_t | v_{t-1}) p_\theta(y_t | z_t, v_{t-1}) p_\theta(x_t | y_t, z_t, v_{t-1}). \quad (4.24)$$

Thus, the parameterization of the TMC model given by Equations (3.1)-(3.3) is now given by

$$p_\theta(z_t|v_{t-1}) = \eta(z_t; \psi_\theta^z(v_{t-1})), \quad (4.25)$$

$$p_\theta(y_t|z_t, v_{t-1}) = \vartheta(y_t; \psi_\theta^y(z_t, v_{t-1})), \quad (4.26)$$

$$p_\theta(x_t|y_t, z_t, v_{t-1}) = \zeta(x_t; \psi_\theta^x(y_t, z_t, v_{t-1})). \quad (4.27)$$

Remark 4.3.1. If ψ_θ^z does not depend on v_{t-1} , and if ψ_θ^y and ψ_θ^x are independent of $z_{t-1:T}$, the distribution $p_\theta(y_{0:T}, x_{0:T})$ coincides with that of a PMC built from (4.4)- (4.5).

Joint estimation of θ and ϕ

Classical variational inference algorithms aim at maximizing the ELBO (4.20) when the objective is to estimate the parameters of a generative model, *i.e.* a model in which we do not focus on the interpretability of the hidden r.v. but rather on the modeling power of the distribution $p_\theta(x_{0:T})$. Consequently, in our case, a direct maximization of (4.20) does not guarantee the interpretability of the r.v. $y_{0:T}$. The problem is all the more critical that our hidden process is split into an interpretable one, $y_{0:T}$, and an auxiliary one, $z_{0:T}$. To that end, we propose an adaptation and an interpretation to the sequential case of two techniques introduced in the machine learning community (Higgins et al., 2017; Kingma et al., 2014). The first one relies on a reinterpretation of the ELBO (4.20) as the sum of a reconstruction and a KLD terms; this last one is next penalized. The second technique consists in adding a penalizing term to the resulting ELBO which aims at strengthening the distinct role of $y_{0:T}$ and of $z_{0:T}$ and exploiting the result of previous classifications obtained with an available model.

The β -ELBO

We first start with an alternative decomposition of the ELBO (4.20).

Corollary 1. Let us factorize

$$p_\theta(z_{0:T}, y_{0:T}, x_{0:T}) = \bar{p}_\theta(z_{0:T}, y_{0:T}|x_{0:T})\tilde{p}_\theta(x_{0:T}|z_{0:T}, y_{0:T})$$

with

$$\tilde{p}_\theta(x_{0:T}|z_{0:T}, y_{0:T}) = p_\theta(x_0|y_0, z_0) \prod_{t=1}^T \zeta(x_t; \psi_\theta^x(y_t, z_t, v_{t-1})), \quad (4.28)$$

$$\bar{p}_\theta(z_{0:T}, y_{0:T}|x_{0:T}) = p_\theta(y_0, z_0) \prod_{t=1}^T \eta(z_t; \psi_\theta^z(v_{t-1})) \vartheta(y_t; \psi_\theta^y(z_t, v_{t-1})). \quad (4.29)$$

Then

$$\mathcal{Q}_{\text{unsup}}^{\text{opt}}(\theta, \phi) = \mathcal{L}_1(\theta, \phi) + \mathcal{L}_2(\theta, \phi), \quad (4.30)$$

where

$$\mathcal{L}_1(\theta, \phi) = \mathbb{E}_{q_\phi^{\text{opt}}(z_{0:T}, y_{0:T}|x_{0:T})} (\log \tilde{p}_\theta(x_{0:T}|z_{0:T}, y_{0:T})), \quad (4.31)$$

$$\mathcal{L}_2(\theta, \phi) = -D_{\text{KL}}(q_\phi^{\text{opt}}(z_{0:T}, y_{0:T}|x_{0:T}) || \bar{p}_\theta(z_{0:T}, y_{0:T}|x_{0:T})). \quad (4.32)$$

This decomposition can be seen as a generalization to the sequential case of the decomposition proposed for the β -VAE in (Higgins et al., 2017). Indeed, $\mathcal{Q}_{\text{unsup}}^{\text{opt}}$ involves the sum of (i) a reconstruction term \mathcal{L}_1 between q_ϕ^{opt} and \tilde{p}_θ which measures the ability to reconstruct observations $x_{0:T}$ according to the conditional likelihood \tilde{p}_θ from the latent r.v. $(z_{0:T}, y_{0:T})$ distributed according to q_ϕ^{opt} ; (ii) a KLD term \mathcal{L}_2 between the variational distribution and the conditional prior \bar{p}_θ . However, contrary to the static case, our decomposition involves $\tilde{p}_\theta(x_{0:T}|z_{0:T}, y_{0:T})$ and $\bar{p}_\theta(z_{0:T}, y_{0:T}|x_{0:T})$ rather than $p_\theta(x_{0:T}|z_{0:T}, y_{0:T})$ and $p_\theta(z_{0:T}, y_{0:T})$, respectively. Indeed, except if $T = 0$, the latter two distributions are no longer computable, which makes the classical ELBO decomposition impractical.

The idea underlying our β -ELBO is to penalize the KLD term $\mathcal{L}_2(\theta, \phi)$. To understand why, let us detail the expression of $\mathcal{L}_1(\theta, \phi)$ and of $\mathcal{L}_2(\theta, \phi)$. First, using (4.28) and (4.27), $\mathcal{L}_1(\theta, \phi)$ reads

$$\begin{aligned} \mathcal{L}_1(\theta, \phi) &= \mathbb{E}_{q_\phi^{\text{opt}}(y_0, z_0|x_{0:T})} (\log p_\theta(x_t|y_0, z_0)) + \\ &\quad \sum_{t=1}^T \mathbb{E}_{q_\phi^{\text{opt}}(y_t, z_t|y_{t-1}, z_{0:t-1}, x_{0:T})} (\log p_\theta(x_t|y_t, z_t, v_{t-1})). \end{aligned} \quad (4.33)$$

Following this decomposition, it can be seen that at each time step t , the maximization of (4.33) encourages the model to interpret the latent r.v. (y_t, z_t) as those which explain the best the observation x_t given the past. On the other

hand, using (4.29) and (4.25)-(4.26), the maximization of

$$\begin{aligned}\mathcal{L}_2(\theta, \phi) = & -D_{\text{KL}}\left(q_{\phi}^{\text{opt}}(y_0, z_0 | x_{0:T}) \| p_{\theta}(z_0, y_0)\right) - \\ & \sum_{t=1}^T D_{\text{KL}}\left(q_{\phi}^{\text{opt}}(y_t, z_t | y_{t-1}, z_{0:t-1}, x_{0:T}) \| p_{\theta}(y_t, z_t | v_{t-1})\right)\end{aligned}\quad (4.34)$$

tends to push the posterior variational distribution at each time step to be close to the conditional prior distribution $p_{\theta}(y_t, z_t | v_{t-1})$. As in (Higgins et al., 2017), we penalize $\mathcal{L}_2(\theta, \phi)$ via the introduction of a scalar β_1 . Since a part of the latent r.v. has to be interpretable, and that the interpretability of hidden r.v. is not conditioned by the observations, the interest of this term is to force the posterior distribution q_{ϕ}^{opt} to take into account the prior term at each time step. In other words, this penalization term aims at limiting the impact of the observations on the interpretability of the hidden r.v., particularly in problems where x_t is a very noisy version of y_t .

Cross-entropy penalization

We finally complete our objective function to guide the estimation process into distinguishing the role of $y_{0:T}$ and of $z_{0:T}$ in order to obtain better interpretable estimations of y_t . We assume that we have at our disposal a pre-classification $y_{0:T}^{\text{pre}}$. Next, introduce the KLD between the empirical distribution deduced from this pre-classification, $p^{\text{emp}}(y_{0:T}) = \delta_{y_{0:T}^{\text{pre}}}(y_{0:T})$, and the marginal variational distribution

$$q_{\phi}(y_{0:T} | x_{0:T}) = \int q_{\phi}^{\text{opt}}(z_{0:T}, y_{0:T} | x_{0:T}) dz_{0:T},$$

which aims itself at approximating the true posterior distribution $p_{\theta}(y_{0:T} | x_{0:T})$. Thus, the objective is to push the variational distribution q_{ϕ} to take into account the interpretable labels obtained from an already interpretable pre-classification through the negative cross-entropy

$$\mathcal{L}_3(\theta, \phi) = \mathbb{E}_{p^{\text{emp}}(y_{0:T})} (\log q_{\phi}(y_{0:T} | x_{0:T})) = \log q_{\phi}(y_{0:T}^{\text{pre}} | x_{0:T}), \quad (4.35)$$

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.

Finally, we obtain a new objective function

$$\mathcal{L}(\theta, \phi) = \mathcal{L}_1(\theta, \phi) + \beta_1 \mathcal{L}_2(\theta, \phi) + \beta_2 \mathcal{L}_3(\theta, \phi), \quad (4.36)$$

where $\mathcal{L}_1(\theta, \phi)$, $\mathcal{L}_2(\theta, \phi)$ and $\mathcal{L}_3(\theta, \phi)$ are defined in (4.31), (4.32) and (4.35), respectively. If we set $\beta_1 = 1$ and $\beta_2 = 0$, then $\mathcal{L}(\theta, \phi)$ coincides with the ELBO $\mathcal{Q}_{\text{unsup}}^{\text{opt}}(\theta, \phi)$ in (4.30).

Monte Carlo approximation

It remains to compute and optimize (4.36) in practice. $\mathcal{L}_1(\theta, \phi)$ and $\mathcal{L}_2(\theta, \phi)$ coincide with mathematical expectations according to $q_{\phi}^{\text{opt}}(z_{0:T}, y_{0:T}|x_{0:T}) = q_{\phi}(z_{0:T}|x_{0:T})p_{\theta}(y_{0:T}|z_{0:T}, x_{0:T})$. Using expressions (4.33)-(4.34), expectations according to $p_{\theta}(y_{0:T}|x_{0:T}, z_{0:T})$ are exactly computable. Thus, $\mathcal{L}_1(\theta, \phi)$ and $\mathcal{L}_2(\theta, \phi)$ rely on the approximate computation of expectations according to $q_{\phi}(z_{0:T}|x_{0:T})$. It can be also noted that

$$q_{\phi}(y_{0:T}|x_{0:T}) = \mathbb{E}_{q_{\phi}(z_{0:T}|x_{0:T})}(p_{\theta}(y_{0:T}|z_{0:T}, x_{0:T})),$$

then $\mathcal{L}_3(\theta, \phi)$ also relies on an expectation according to same distribution $q_{\phi}(z_{0:T}|x_{0:T})$ as $\mathcal{L}_1(\theta, \phi)$ and $\mathcal{L}_2(\theta, \phi)$. Consequently, Monte Carlo estimates based on *i.i.d.* samples $z_{0:T}^{(m)} \sim q_{\phi}(z_{0:T}|x_{0:T})$ are estimates of $\mathcal{L}_1(\theta, \phi)$, $\mathcal{L}_2(\theta, \phi)$ and $\mathcal{L}_3(\theta, \phi)$. The choice of the variational distribution is given by the following factorization $q_{\phi}(z_{0:T}|x_{0:T}) = q_{\phi}(z_0|x_{0:T}) \prod_{t=1}^T q_{\phi}(z_t|z_{0:t-1}, x_{0:T})$. Next, $q_{\phi}(z_t|z_{0:t-1}, x_{0:T})$ is chosen such that it is possible to use the reparameterization trick to have a final sample $z_{0:T}^{(m)}$, which as a differentiable function of ϕ . (see Subsection 1.2.1). Finally, we obtain the following estimate of $\mathcal{L}(\theta, \phi)$ in (4.36) given by

$$\widehat{\mathcal{L}}(\theta, \phi) = \widehat{\mathcal{L}}_1(\theta, \phi) + \widehat{\mathcal{L}}_2(\theta, \phi) + \widehat{\mathcal{L}}_3(\theta, \phi), \quad (4.37)$$

where

$$\widehat{\mathcal{L}}_1(\theta, \phi) = \frac{1}{N} \sum_{m=1}^M \mathbb{E}_{p_{\theta}(y_{0:T}|z_{0:T}^{(m)}, x_{0:T})} \left(\log \tilde{p}_{\theta}(x_{0:T}|z_{0:T}, y_{0:T}^{(m)}) \right), \quad (4.38)$$

$$\widehat{\mathcal{L}}_2(\theta, \phi) = \frac{1}{N} \sum_{m=1}^M \mathbb{E}_{p_{\theta}(y_{0:T}|z_{0:T}^{(m)}, x_{0:T})} \left(\log \left(\frac{\bar{p}_{\theta}(z_{0:T}, y_{0:T}^{(m)}|x_{0:T})}{p_{\theta}(y_{0:T}|z_{0:T}^{(m)}, x_{0:T})q_{\phi}(z_{0:T}^{(m)}|x_{0:T})} \right) \right), \quad (4.39)$$

$$\widehat{\mathcal{L}}_3(\theta, \phi) = \log \left(\frac{1}{N} \sum_{m=1}^M p_{\theta}(h_{0:T}^{\text{pre}}|z_{0:T}^{(m)}, x_{0:T}) \prod_{t=1}^T p_{\theta}(y_{t-1}^{\text{pre}}|y_t^{\text{pre}}, z_{0:T}^{(m)}, x_{0:T}) \right), \quad (4.40)$$

where the remaining expectations are computed from (4.18) and from (4.28)-(4.29) and where samples $z_{0:T}^{(m)}$ satisfy the reparameterization concept. The complete estimation algorithm is described in Algorithm 6.

Algorithm 6 Parameter estimation in general TMCs.

Input: $x_{0:T}$, the data; ϵ , the learning rate; M the number of samples**Output:** (θ^*, ϕ^*) , sets of estimated parameters

- 1: Initialize the parameters θ^0 and ϕ^0
 - 2: $j \leftarrow 0$
 - 3: **while** convergence is not attained **do**
 - 4: Sample $z_0^{(m)} \sim q_{\phi^j}(z_0|x_{0:T})$, for all $1 \leq m \leq M$
 - 5: Sample $z_t^{(m)} \sim q_{\phi^j}(z_t|z_{0:t-1}^{(m)}, x_{0:T})$, for all $1 \leq m \leq M$, for all $1 \leq t \leq T$
 - 6: Compute $p_\theta(y_{t-1}|y_t, z_{0:T}^{(m)}, x_{0:T})$, for all $y_{t-1:t} \in \Omega \times \Omega$, for all $1 \leq m \leq M$, for all $1 \leq t \leq T$
 - 7: Evaluate the loss $\hat{\mathcal{L}}(\theta^j, \phi^j)$ from (4.37)-(4.40)
 - 8: Compute the derivative of the loss function $\nabla_{(\theta, \phi)} \hat{\mathcal{L}}(\theta, \phi)$ from (4.37)-(4.40)
 - 9: Update the parameters with gradient ascent
- $$\begin{pmatrix} \theta^{(j+1)} \\ \phi^{(j+1)} \end{pmatrix} = \begin{pmatrix} \theta^j \\ \phi^j \end{pmatrix} + \epsilon \nabla_{(\theta, \phi)} \hat{\mathcal{L}}(\theta, \phi) \Big|_{(\theta^j, \phi^j)} \quad (4.41)$$
- 10: $j \leftarrow j + 1$
 - 11: **end while**
 - 12: $\theta^* \leftarrow \theta^j$
 - 13: $\phi^* \leftarrow \phi^j$
-

Estimation of y_t

Once we have obtained an estimate θ^* of θ , we focus on the computation of $p_{\theta^*}(y_t|x_{0:T})$,

$$p_{\theta^*}(y_t|x_{0:T}) = \int_{z_{0:T}} p_{\theta^*}(y_t|z_{0:T}, x_{0:T}) p_{\theta^*}(z_{0:T}|x_{0:T}) dz_{0:T}, \quad (4.42)$$

where $p_{\theta^*}(y_t|z_{0:T}, x_{0:T})$ is computable from a direct extension of (4.13) and (4.15)-(4.17) (see the proof of Proposition 4.3.1). Since (4.42) is intractable, we propose an MC estimate $\hat{p}_\theta(y_t|x_{0:T})$ deduced from the sequential importance resampling mechanism (Doucet et al., 2001b) and based on the observation that $p_{\theta^*}(z_{0:T}|x_{0:T}) \propto p_{\theta^*}(x_{0:T}, z_{0:T})$ is known up to a constant. Indeed, $p_{\theta^*}(x_{0:T}, z_{0:T})$ can also be computed from a direct extension of (4.12)-(4.13). We thus introduce the estimated variational distribution

$$q_{\phi^*}(z_{0:T}|x_{0:T}) = q_{\phi^*}(z_0|x_{0:T}) \prod_{t=1}^T q_{\phi^*}(z_t|z_{0:t-1}, x_{0:T})$$

as importance distribution due to its proximity with $p_\theta(z_{0:T}|x_{0:T})$. Finally, rewriting (4.42) as

$$p_{\theta^*}(y_t|x_{0:T}) = \frac{\mathbb{E}_{q_{\phi^*}(z_{0:T}|x_{0:T})} \left(\frac{p_{\theta^*}(y_t|z_{0:T}, x_{0:T}) p_{\theta^*}(z_{0:T}, x_{0:T})}{q_{\phi^*}(z_{0:T}|x_{0:T})} \right)}{\mathbb{E}_{q_{\phi^*}(z_{0:T}|x_{0:T})} \left(\frac{p_{\theta^*}(x_{0:T})}{q_{\phi^*}(z_{0:T}|x_{0:T})} \right)}, \quad (4.43)$$

we compute the sequential MC sampler (Doucet & Johansen, 2009) presented in Algorithm 7 consisting of the sequential application of three elementary steps (sampling, weighting and resampling). Note that any improvement of this sequential MC algorithm can be used (Fearnhead et al., 2010).

4.3.3 Deep TMCs for unsupervised classification

Let us now focus on Deep TMCs for unsupervised classification. We adapt the two-step procedure described in Section 4.2.2. The main difference with Section 4.2.2 is that the input of our DNN can now depend on the latent r.v. z_t ; in addition, due to the VI framework that we have proposed in the previous section, we also consider that the conditional variational distribution $q_\phi(z_t|z_{0:t-1}, x_{0:T})$ at the core of our estimation algorithm is parameterized by a DNN.

Algorithm 7 A Sequential Monte Carlo algorithm for Bayesian classification in general TMC.

Input: $x_{0:T}$, the observation; a set of parameters (θ^*, ϕ^*) ; M , the number of samples

Output: $\widehat{y}_{0:T}$ the final classification

- 1: Sample $z_0^{(m)} \sim q_{\phi^*}(z_0|x_{0:T})$,
 - 2: Compute $w_0^{(m)} = \frac{p_{\theta^*}(z_0^{(m)}, x_0)}{q_{\phi^*}(z_0|x_{0:T})} W_0^{(m)} = w_0^{(m)} / \sum_{m=1}^M w_0^{(m)}$, for all $1 \leq m \leq M$
 - 3: **for** $t \leftarrow 1$ to T **do**
 - 4: Sample $z_t^{(m)} \sim q_{\phi^*}(z_{0:t}|z_{0:t-1}, x_{0:T})$, for all $1 \leq m \leq M$
 - 5: Compute
 - $$w_t^{(m)} = w_{t-1}^{(m)} \frac{p_{\theta^*}(z_t^{(m)}, x_{0:t})}{p_{\theta^*}(z_{0:t-1}^{(m)}, x_{t-1}) q_{\phi^*}(z_t^{(m)}|z_{0:t-1}^{(m)}, x_{0:T})}, \text{ for all } 1 \leq m \leq M$$
 - 6: Compute $W_t^{(m)} = w_t^{(m)} / \sum_{m=1}^M w_t^{(m)}$, for all $1 \leq m \leq M$
 - 7: **if** Resampling **then**
 - 8: Sample $l^{(m)} \sim p(l=j) = W_t^{(j)}$, for all $1 \leq m \leq M$
 - 9: Set $z_{0:t}^{(m)} = z_{0:t}^{(l^{(m)})}$ and $W_t^{(m)} = 1/M$ for all $1 \leq m \leq M$
 - 10: **end if**
 - 11: **end for**
 - 12: Compute $p_{\theta^*}(y_{t-1:t}|z_{0:T}^{(m)}, x_{0:T})$, for all $y_{t-1:t} \in \Omega \times \Omega$, for all $1 \leq t \leq T$, using the extension of (4.16)
 - 13: Compute $\hat{p}_{\theta^*}(y_t|x_{0:T}) = \sum_{m=1}^M W_t^{(m)} p_{\theta^*}(y_t|z_{0:T}^{(m)}, x_{0:T})$, for all $y_t \in \Omega$, for all $1 \leq t \leq T$
 - 14: $\hat{y}_t = \arg \max \hat{p}_{\theta^*}(y_t|x_{0:T})$, for all $1 \leq t \leq T$
-

Constrained output layer

The first step is a direct adaptation of Section 4.2.2 and relies on the preliminary estimation of a non-deep TMC model. More precisely, Algorithm 7 is applied to estimate the parameter of a linear TMC model (*i.e.* a TMC which is a direct extension of (4.10)-(4.11) or equivalently a deep TMC model with no hidden layer). Note that since $z_{0:T}$ does not need to be interpretable, $q_\phi(z_t|z_{0:t-1}, x_{0:T})$ are already parameterized by a DNN in the linear TMC models. Next, the DNNs, which parameterize ψ_θ^z , ψ_θ^y and ψ_θ^x , are built according to the same scheme of Figure 4.2, except that the input and the hidden layer before the output also consists of z_{t-1} or of $z_{t-1:t}$. We thus obtain a set of frozen and unfrozen parameters.

Pretraining of the unfrozen parameters

The next step consists in pretraining the unfrozen parameters of the intermediate hidden layers in order to mimic the estimated linear TMC. We use the same approach as the one developed in Section 4.2.2 which relies on a pre-classification $\hat{y}_{0:T}^{\text{pre}}$, but we now take into account the fact that z_t is not observed. Since the objective of the r.v. z_t is to encode the corresponding observation x_t through the DNN related to q_ϕ , we first sample $z_{0:T}$ according to the previously estimated variational distribution $q_\phi(z_{0:T}|x_{0:T})$; we next use the components $z_{t-1:t}$ or z_t as inputs of the DNNs ψ_θ^z , ψ_θ^y and ψ_θ^x . Finally, as in Paragraph 4.2.2, we apply the backpropagation algorithm in order to minimize an adapted cost function w.r.t. θ_{ufr} which depends on $\hat{y}_{0:T}^{\text{pre}}$. Figure 4.4

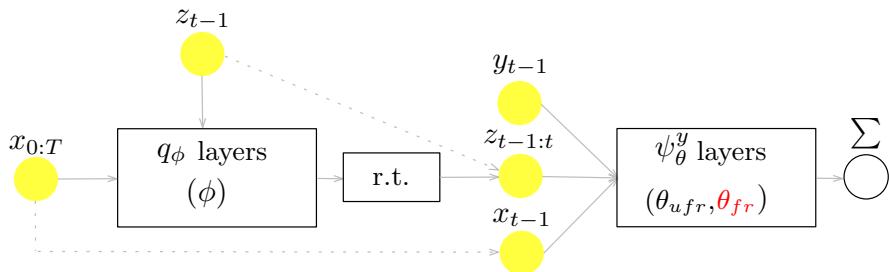


Figure 4.4: Graphical and condensed representation of the parameterization of ψ_θ^y in the DTMC models. *r.t.* stands for reparameterization trick. The dashed arrows represent the fact that some variables are copied. For clarity, we do not represent the block ψ_θ^y which is similar to Figure 4.2, up to the introduction of $z_{t-1:t}$.

summarizes our pretraining procedure for function ψ_θ^y and the final estimation procedure is described in Algorithm 8.

Algorithm 8 A general estimation algorithm for deep parameterizations of TMC models

Input: $x_{0:T}$, the observation; q_ϕ a class of variational distribution

Output: $\hat{y}_{0:T}$ the final classification

Initialization of the output layer of ψ_θ^z , ψ_θ^y and ψ_θ^x

1: Estimate $(\theta_{\text{fr}}^*, \tilde{\phi})$ and $\hat{y}_{0:T}^{\text{pre}}$ with Algorithm 6-7, using the related non-deep TMC model

Pretraining of θ_{ufr}

2: $\theta_{\text{ufr}}^{(0)} \leftarrow \text{Backprop}(\hat{y}_{0:T}^{\text{pre}}, x_{0:T}, \theta_{\text{fr}}^*, \tilde{\phi}, \mathcal{C}_{\psi_\theta^z}, \mathcal{C}_{f_\theta}, \mathcal{C}_{g_\theta})$

Fine-tuning of the complete model

3: Compute $(\theta_{\text{ufr}}^*, \phi^*)$ with Lines 2-13 of Algorithm 6

4: Compute $\hat{y}_{0:T}$ with Algorithm 7

4.3.4 Simulations

We continue to illustrate the performance of our models with the same binary image segmentation problem as Section 4.2.3. We focus our experiments on the relevance of the latent process $z_{0:T}$. To that end, we focus on a particular TMC model in which the role of the latent process $z_{0:T}$ is to complexify the conditional distribution ζ of the noise but not ϑ . We first present the particular model and next the results. β_1 and β_2 are tuned manually by taking into account the characteristics of the studied models.

The minimal TMCs

In order to highlight the role of $z_{0:T}$ w.r.t. the other characteristics of our models, we introduce the Minimal TMC (MTMC) model which exhibits a reduced number of direct dependencies. In this model, $z_{0:T}$ is an independent process and given $z_{0:T}$, $(y_{0:T}, x_{0:T})$ is a HMC where only the observations depend on z_t ; in other words, ψ_θ^z in (4.25) does not depend on v_{t-1} , ψ_θ^y in (4.26) only depends on (y_{t-1}) and ψ_θ^x in (4.27) only depends on (z_t, y_t) . The joint distribution of $v_{0:T}$ can be rewritten as

$$p_\theta(v_{0:T}) = \underbrace{\prod_{t=0}^T \eta(z_t; \psi_\theta^z)}_{p_\theta(z_{0:T})} \underbrace{\prod_{t=1}^T \vartheta(y_t; \psi_\theta^y(y_{t-1}))}_{p_\theta(y_{0:T}|z_{0:T}) = p_\theta(y_{0:T})} \underbrace{\prod_{t=0}^T \zeta(x_t; \psi_\theta^x(z_t, y_t))}_{p_\theta(x_{0:T}|z_{0:T}, y_{0:T})}, \quad (4.44)$$

With this model, the latent process $z_{0:T}$ affects the conditional distribution of the observations.

We next consider three instances of MTMCs. The first one is the continuous linear MTMC in which $z_t \in \mathbb{R}$ are distributed according to standard normal distribution (so η is the Gaussian distribution and $\psi_\theta^z = [0; 1]$), ψ_θ^y , ψ_θ^x , ϑ and ζ coincide with our first illustrative example in Section 4.2, see (4.6)-(4.7), up to the dependency in z_t . We also consider a deep version of the MTMC (DMTMC) in which ψ_θ^x is parameterized by a DNN (with one hidden layer of 100 neurons and ReLU activation function). For both continuous versions of the MTMC, we use the variational distribution

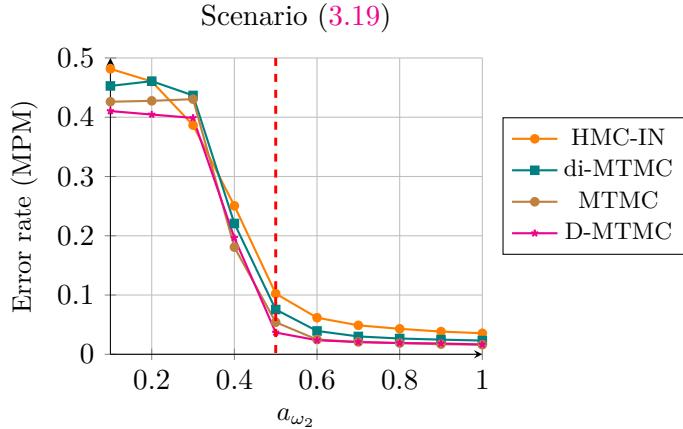
$$q_\phi(z_{0:T}|x_{0:T}) = \prod_{t=1}^T q_\phi(z_t|z_{t-1}, x_t) = \prod_{t=1}^T \mathcal{N}(z_t; \nu_\phi(z_{t-1}, x_t)). \quad (4.45)$$

where $\nu_\phi(z_{t-1}, x_t)$ is parameterized by a DNN with one hidden layer of 100 neurons and a ReLU activation function.

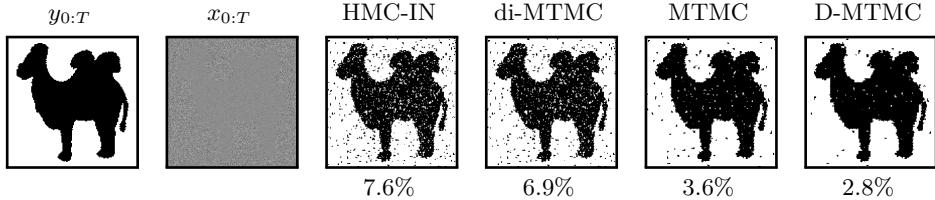
The motivation underlying this choice of variational distribution is that $z_{0:T}$ is an independent process and that x_t only depends on (y_t, z_t) given the past; consequently, it is reasonable to assume that the posterior distribution of z_t only depends on z_{t-1} and x_t . In addition, more complex variational distributions tend to be more difficult to estimate. And indeed, it has been observed that the choice of the variational distribution does not impact the results in the case of Scenario (4.44), see Appendix E. Finally, we also consider a discrete version of the MTMC (di-MTMC) in which $z_t \in \{\nu_1, \nu_2\}$ is discrete (Gorynin et al., 2018; Li et al., 2019; Chen & Jiang, 2020). For this model, Algorithm 3 and 4 can be directly applied in the augmented space $\{\omega_1, \omega_2\} \times \{\nu_1, \nu_2\}$.

Experiments and results

We now consider two scenarios in which binary images are corrupted with non elementary noises. In the first scenario, the hidden images $y_{0:T}$ are the *camel*-type images of the Binary Shape Database and are corrupted with the stationary multiplicative noise given in (3.20) in Section 3.4.2, where $z_t \sim \mathcal{N}(0, 1)$, $a_{\omega_1} = 0$, a_{ω_2} is a varying parameter and $b_{\omega_1} = b_{\omega_2} = 0.2$. Figure 4.5a displays the results for the setting $\beta_1 = 5$, $\beta_2 = 1$ in our variational approach. Scalar β_1 can be interpreted as enforcing the standardized Gaussian prior on the learnt latent variables, which is seemingly favorable on this example because of the way $z_{0:T}$ is generated. β_2 is also needed and seems to guide the optimization so that the estimated $\hat{y}_{0:T}$ corresponds to the desired segmentation.



(a) Error rate from the unsupervised segmentations of Scenario (3.19). 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. 4.5a). Error rates appear below the images.

Figure 4.5: Unsupervised image segmentation with General Triplet Markov Chains (Scenario (3.19)).

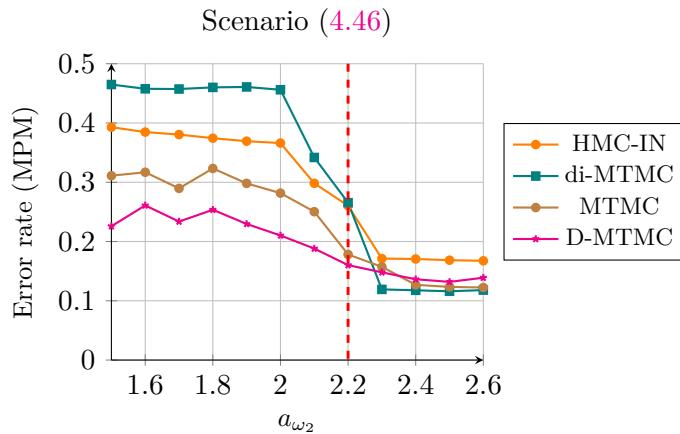
A particular classification is also displayed in Figure 4.5b. As we see, our MTMC models improve the performance (up to a 7%-point improvement) of the HMC-IN. This comparison illustrates the interest of the third latent process $z_{0:T}$. A slight advantage goes to the models with continuous $z_{0:T}$ (MTMC and DMTMC) over the di-MTMC which still performs better than the HMC-IN model. Note that in the case where we optimize directly the ELBO (*i.e.* $\beta_1 = 1$ and $\beta_2 = 0$), it has been observed that the classification obtained is not interpretable. This observation validates experimentally our strategy to adapt the objective function.

In the second scenario, the hidden images $y_{0:T}$ are the *dog*-type images of the Binary Shape Database. They are corrupted by a non-stationary general

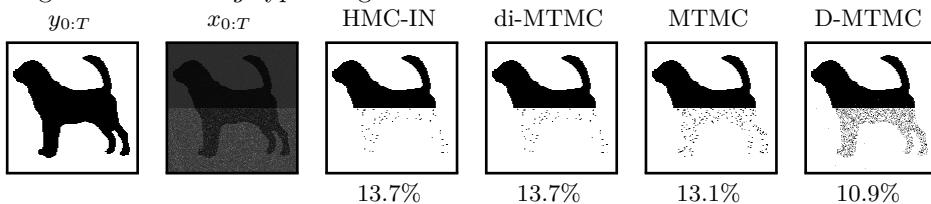
noise,

$$\begin{cases} x_t|y_t \sim \mathcal{N}(a_{y_t}; \sigma^2), & \text{if } k \in \left\{1, \dots, \left\lfloor \frac{T}{2} \right\rfloor\right\}, \\ x_t|y_t \sim a_{y_t} + \mathcal{E}(\vartheta), & \text{if } k \in \left\{\left\lfloor \frac{T}{2} \right\rfloor + 1, \dots, K\right\}, \end{cases} \quad (4.46)$$

where $\mathcal{E}(\vartheta)$ is the exponential probability distribution of parameter ϑ , $a_{\omega_1} = 0$, a_{ω_2} is a varying parameter, $\sigma = 0.2$ and $\vartheta = 1.4$. The main difficulty of this scenario is that the images are corrupted by two different noises with a relatively low level for both areas and have to be fitted in a unique model. For this scenario, we set $\beta_1 = 0.1$ and $\beta_2 = 0$. A small value of β_1 can be interpreted as a way to better fit the observations. Indeed, more flexibility seems to be needed to learn such a complex non-stationary noise.



(a) Error rate from the unsupervised segmentations of Scenario (4.46). Results are averaged on all the *dog*-type images from the database.



(b) Selected illustrations for $a_{\omega_2} = 2.2$ (signaled by the red vertical line on Fig. 4.6a). Error rates appear below the images.

Figure 4.6: Unsupervised image segmentation with General Triplet Markov Chains (Scenario (4.46)).

The reason why β_2 is set to 0 is that the pre-classification obtained with the HMC-IN is poor and should not be used to learn the parameters in the MTMC. It has been observed that other values deteriorate the final classification obtained with MTMC models. The results are displayed in Figure 4.6a and Figure 4.6b displays a particular classification. It is clear that the TMC models with a continuous auxiliary latent r.v. (MTMC and DMTMC) offer a greater flexibility and are able to learn this complex multi-stationary noise. On the other hand the average classification provided by the di-MTMC or the HMC-IN models are irrelevant as soon as $a_{\omega_2} < 2$. This experiment illustrates the interest of a continuous auxiliary latent r.v. over discrete auxiliary latent r.v.; the latter being the only option that has been considered in the literature so far (Gorynin et al., 2018; Li et al., 2019; Chen & Jiang, 2020). These experiments show the interesting capabilities of the generalized models to provide results in presence of very general noises. Coupled to the deep parameterization, a continuous third latent process enables our models to bypass the need of an explicit expression of the conditional distribution of the noise.

Remark 4.3.2. We also propose an alternative use of the latent process, where our objective is to characterize explicitly the relationship between the pair (y_t, x_t) and the past observations x_{t-1} when $z_{0:T}$ is deterministic given the observations. Thus, a closed-form expression of $p_\theta(y_t, x_t | y_{t-1}, x_{t-1})$ is available contrary to the general TMC introduced before. A direct advantage of the resulting TMC model is that it can be interpreted as the combination of a PMC model (4.1) with an RNN (Rumelhart et al., 1986; Mikolov et al., 2015), and that the distributions of interest can be computed exactly, without any approximation. This model is called a Partially Pairwise Markov Chain (PPMC), which is detailed in the Appendix E.

4.4. Experiments on real datasets

We finally experiment our models on two real datasets. The first one is devoted to a medical images. The main challenge of this kind of data is that the noise associated to such images is unknown and non-usual; that is why we introduce our TMCs to measure the impact of the third latent process. The next dataset is related to human activity recognition. For this problem, the dependencies between the r.v. (the class and the observed r.v.) are critical; that is why we focus on the impact of our PMCs.

4.4.1 Unsupervised segmentation of biomedical images

We first illustrate the potential of the generalized TMC models on real biomedical data. The task consists in the segmentation of micro-computed tomography X-ray scans of human arteries containing a metallic stent biomaterial¹. These images are reminiscent of the synthetic experiment of Scenario (4.46): some regions exhibit a particular type of correlated noise (because of the beam hardening artifacts caused by the interactions between X-rays and the metallic stent) and some regions do not. However, the noise is unknown and has not been simulated contrary to Scenario (4.46).

Table 4.1 and Figure 4.7 summarize the experiment. It can be seen that the classical models (HMC-IN and di-MTMC) are unable to handle the non-stationarity of the noise. The di-MTMC model even fail to provide any improvement over the HMC-IN model. On the other hand, major improvements can be seen when using the TMC models with a continuous auxiliary process, suggesting that the latter model offers more flexibility and that our parameter estimation algorithm enables to take advantage of it. These results on real-world data corroborates the results found in the synthetic experiment given in Section 4.3.4. Note that, in this case, we set $\beta_1 = 5$, $\beta_2 = 1$ and used the HMC-IN classification as a pre-segmentation. The network configurations are the same as in Section 4.3.4.

4.4.2 Unsupervised clustering for human activity recognition

We now illustrate the performances of classical PMC models, deep PMC models and deep PPMC models on a real clustering task linked with human activity recognition. We use the Human Activity and Postural Transition (HAPT) dataset described in (Reyes-Ortiz et al., 2016)². It consists of three-dimensional time series that we wish to cluster into two classes: *movement* and *no movement*. To solve this task, the models we used are the same as those introduced before, namely ψ_θ^y , ψ_θ^x , ϑ and ζ coincide with our first illustrative example in Section 4.2 ((4.6)-(4.7)). In the case of the deep parameterizations, ψ_θ^y and ψ_θ^x have one (unfrozen) hidden layer with 100 neurons and the ReLU activation function. Moreover, in the case of the deep PPMC models, ψ_θ^z is composed of two independent standard RNNs with ReLU activation function, *i.e.* $z_t = [z_t^1, z_t^2] = [\psi_\theta^{z1}(z_{t-1}^1, x_{t-1}), \psi_\theta^{z2}(z_{t-1}^2, x_{t-1})]$, with 10 hidden neurons.

The results are given in Table 4.2 for models sharing the same configura-

¹Data provided by Dr. Salomé Kuntz (GEPROMED, Strasbourg, France)

²<http://archive.ics.uci.edu/ml/datasets/smartphone-based+recognition+of+human+activities+and+postural+transitions>

Slice	HMC-IN	di-MTMC	MTMC	DMTMC
Average	8.6	8.6	7.6	6.5

Table 4.1: Averaged error rates (%) in unsupervised image segmentation with all the generalized TMCs assessed on ten micro-computed tomography slices. The detailed scores are given in Appendix E.

Data	HMC-IN	SPMC	DSPMC	DPSPMC	PMC	DPMC	DPPMC
Average	25.2	21.3	16.8	16.7	17.1	16.8	16.8

Table 4.2: Averaged error rates (%) in the binary clustering of the first twenty raw entries of the HAPT dataset (Reyes-Ortiz et al., 2016). The detailed scores are given in Appendix E.

tions with the models in Section 4.2.3 and E. First of all, the modelization using the pairwise models seems very relevant in this application since we notice up to a 9%-point improvement over the HMC-IN model. In the case of the SPMCs, we clearly see the advantage of using deep parameterizations over the shallow models. The advantage of the deep parameterization is less significant in the PMC case. The contributions of the DPSPMC and DPPMC models are also less significant. The absence of gains in error rate when using the most complex models might be related to the limited length of the training sequences in this application (sequences of length between 15000 and 20000).

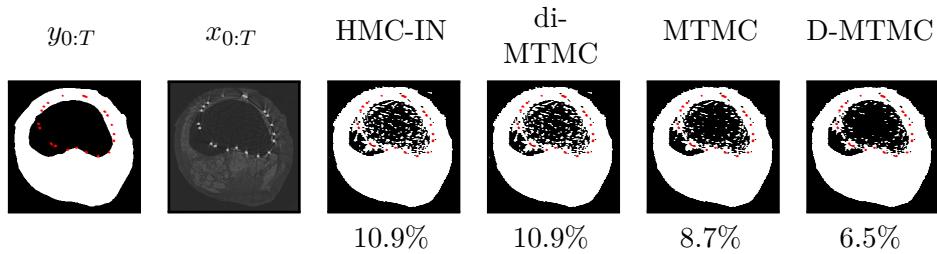


Figure 4.7: Illustration of the unsupervised segmentation of slice B, as reported in Table 4.1. The D-MTMC appears to better fit the non-stationary noise, offering a 4%-point improvement in the error rate. The stent components appearing in red are segmented beforehand with a thresholding technique and are considered as image borders during the segmentation using the probabilistic models.

4.5. Conclusions

In this chapter, we have proposed a general framework for PMC and TMC models which fully exploits the modeling power offered by such models for unsupervised signal processing. Contrary to previous work on TMCs with a discrete hidden data, we have introduced a continuous latent process. For these models, we have derived Bayesian inference algorithms for estimating their parameters and the associated hidden r.v. and we have emphasized the case where the parameterization relies on DNNs. Our algorithms rely on an objective function deduced from the variational Bayesian inference but which has been modified to include the interpretability of the discrete hidden r.v.

This contribution enables us to propose an efficient answer to three recurrent questions linked with the practical applications of complex probabilistic graphical models for sequential data: which probability distributions to choose, how to parameterize them, and how to estimate their parameters in an unsupervised way. For several applications, it has indeed been shown that our global procedure leads to new models that consistently perform better than the classical ones. Importantly, the ability of these models to tackle more complex noises comes without no additional effort from the signal processing point of view. Our experiments also suggest that it is possible to model complex noises by using the universal approximating properties of DNNs and by training them in an unsupervised way with the new algorithms that we propose.

On the other hand, while being invisible to a potential practitioner, these new capabilities permitted by the embedded DNNs and by the third auxiliary latent process come at the price of a more complex training procedure. The latter is indeed cast in the context of variational inference with inherent difficulties regarding the approximation of the lower bound, the choice of the variational distribution or the choice of the penalizing coefficients. However, since variational inference is a very popular research topic, it could inspire many improvements for future works with the Generalized Hidden Markov Models framework.

CHAPTER 5

Medical Perspectives

Contents

5.1	Context and motivation	96
5.2	Data and preprocessing	97
5.2.1	Data availability	97
5.2.2	Challenges	100
5.2.3	Pre-processing of the CT and micro CT images	101
5.3	Medical image segmentation	104
5.3.1	Results	105
5.4	Remaining challenges	107
5.5	Conclusions	109

5.1. Context and motivation

Cardiovascular diseases (CVDs) represent a leading global cause of mortality, as highlighted by data from the World Health Organization¹. CVDs include a wide range of conditions that affect the heart and blood vessels. Among these, atherosclerosis is the most common cause of CVDs, which is characterized by the build-up of plaque inside the arteries. This atheromatous plaque is made up of fat, cholesterol, calcium, and other substances found in the bloodstream. Over time, this plaque hardens, leading to the obstruction of the arteries and can cause serious health problems (Insull Jr, 2009). For example, it can limit the flow of oxygen-rich blood to the organs and other parts of the body (Rafieian-Kopaei et al., 2014). Figure 5.1 shows an example of a normal artery (A) and an artery with atherosclerosis (B)².

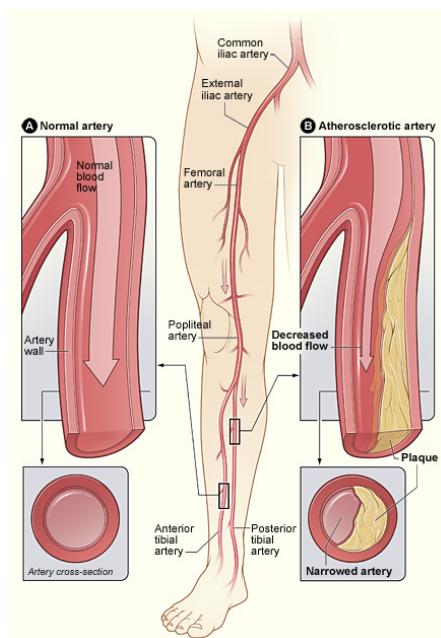


Figure 5.1: Peripheral arterial disease results from narrowing or blockage of the arteries of the legs.

In this context, the GEPROMED (European Research Group on Prostheses Applied to Vascular Surgery) has been established to develop new bioma-

¹<https://www.who.int/>

²<http://vascularsurgeon.ie/peripheral-arterial-disease-pad/>

terials and surgical techniques for vascular surgery. The group has access to a database of medical images, which they have made available to us for the purpose of developing new methods for medical images processing in vascular surgery. The images in this database are Computed Tomography (CT) or micro-Computed Tomography (micro CT) images of the femoro-popliteal arterial segment (SAFP) of different patients. The SAFP is one of the longest arteries in the human body, subject to diverse mechanical forces (*e.g.* torsion, flexion, and extension) due to the movement of the lower limbs.

Atherosclerosis disease comprises 3 categories of plaque: calcified (calcium) ($\approx 70\%$), fibrous ($\approx 20\%$), and lipid ($\approx 10\%$) (Kuntz et al., 2021). To treat these, some endovascular techniques have been developed, such as angioplasty and stenting. There is no non-invasive method (imaging) that can accurately differentiate lesions along the SAFP. The analysis is usually based on the preoperative CT scan (low resolution images), but there are high-resolution scanners that allow a quasi-histological analysis of the tissue. In other words, we have a micro CT scanner *ex vivo*³, and then correlate the images with the histology⁴. Gangloff (2020) has already proposed a method to segment micro CT images of the SAFP. However, a major limitation of this method is that it is not possible to directly segment the CT images of the SAFP, which are of low resolution. This problem is the motivation for the work presented in this chapter.

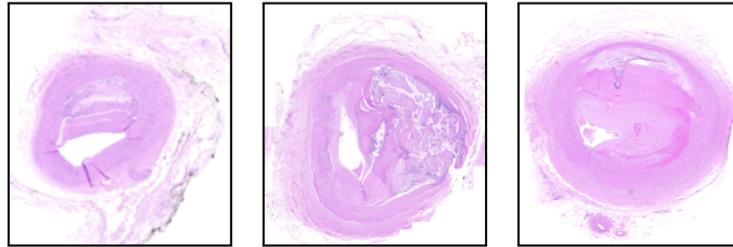
5.2. Data and preprocessing

5.2.1 Data availability

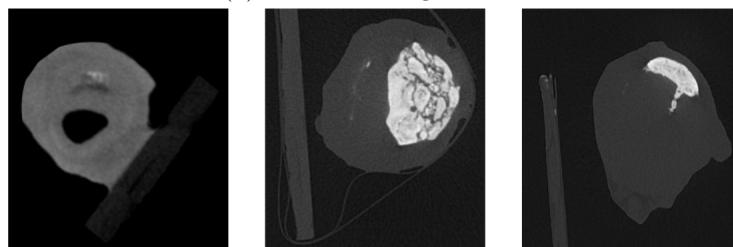
A protocol was developed to obtain the data for this study, which is described in (Kuntz et al., 2021) and is detailed in Appendix F. Figure 5.2 shows the available data. We have the histologic slices 5.2a, the (2D) micro CT images 5.2b, and their corresponding ground truth 5.2c. The ground truth is composed of 6 classes, which are described in Figure 5.3. These annotations are only available for some slices of the 3D scan as shown in Figure 5.4. Here, the red rectangles represent all the 2D slices of the 3D micro CT image. However, the combined information (depicted as light gray and purple rectangles) is not uniformly distributed across these slices; it is only present in certain slices without following a specific pattern. Figure 5.5 shows the correlation

³The term *ex vivo* refers to experimentation performed on tissue samples outside the living organism.

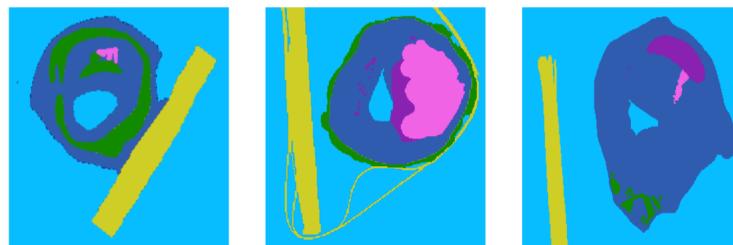
⁴Histology is considered as a gold or criterion standard for the diagnosis of many diseases.



(a) Three histologic slices.



(b) Three microCT images correlated with their histologic truth.



(c) Three expert annotated microCT images obtained.

Figure 5.2: Illustration of part of the available data for the study. Figure taken from ([Gangloff, 2020](#)).

	<i>soft tissue (ST)</i>		<i>fatty tissue (FT)</i>
	<i>sheet calcification (SC)</i>		<i>nodular calcification (NC)</i>
	<i>specimen holder (SH)</i>		<i>background (Ba)</i>

Figure 5.3: Notation of the classes of the ground truth 5.2c. Figure taken from ([Gangloff, 2020](#)).

between the CT scanner and the micro CT scanner, that is predominantly available in segments of the artery where specific lesions, particularly calcifications, are present.

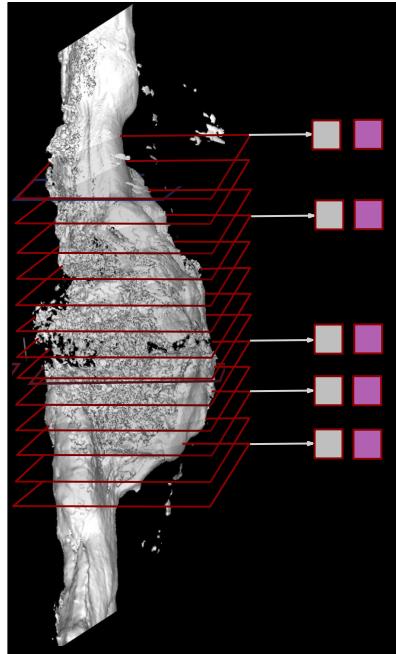


Figure 5.4: Illustration of the available pair of information: 2D micro CT image (light gray rectangle) and its corresponding ground truth (purple rectangle). The pairs of information are only available for some slices of the 3D micro CT image. The red rectangles represent all the 2D slices of the 3D micro CT image. Figure based on ([Kuntz et al., 2021](#))

The aim of this study is to assess the technical feasibility of histological segmentation using the SAFFP algorithm based on the preoperative CT scan. The results of this study will provide initial data to assess the value of a subsequent, larger-scale study to validate the diagnostic capabilities of automated segmentation. As far as we know, there is no non-invasive method (imaging) that can accurately differentiate lesions along the SAFFP. Characterization of AOMI plaques will enable a patient-centred treatment strategy to be devised, based on the type of plaque in the lesion. Automated segmentation will be a tool that will make it possible to dispense with histopathological analysis and detect the type of plaque on the preoperative CT scan. The expected long-term benefits for other patients are very significant. They can be offered individualized treatment depending on the nature of their lesions by adapting the medical device. Previous work has shown that it is possible to segment the micro CT images of the SAFFP. However, a segmentation of the CT images is also necessary, since the micro CT images are not available for all patients.

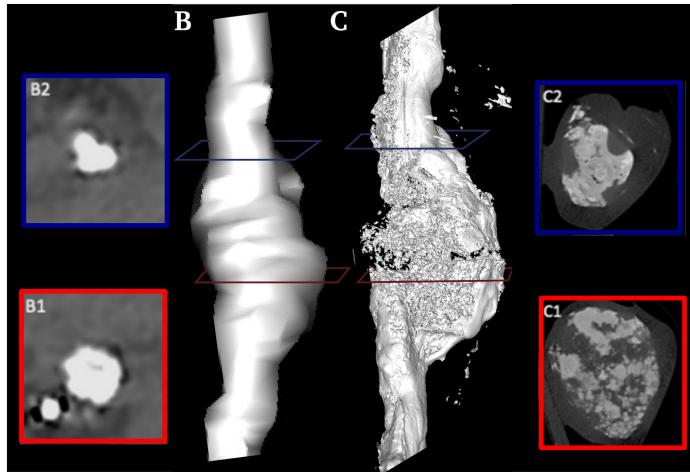


Figure 5.5: Illustration of the available correlation between CT scanner [B, B1, B2] and micro CT scanner [C, C1, C2] using standard references after Step 6. They represent different types of calcifications in SAFP plaques. Figure based on ([Kuntz et al., 2021](#)).

In this chapter, the objective is to perform image segmentation on CT images of the SAFP. Our particular focus lies on the most common type of plaques, the calcifications (sheet and nodular), which is a first step to segment other types of plaques in the future.

5.2.2 Challenges

While we have achieved success in segmenting micro CT images, a notable limitation remains: the segmentation of CT images. Extending our segmentation to CT images is a challenging task for several reasons. First, the low resolution of CT images introduces additional complexity into the segmentation process, in contrast to the higher-resolution micro CT images that [Gangloff \(2020\)](#) has been working with. As depicted in Figure 5.5, the discrepancy in image quality between the CT scanner (B1 and B2) and the micro CT scanner (C1 and C2) is evident. Moreover, the limited availability of data poses a significant challenge. While we possess both 3D micro CT images and their corresponding 3D CT images, establishing a clear and precise correspondence between the two sequences of images is far from straightforward. These images are not perfectly aligned, and their correspondence is not as simple as a ‘mirror’ image. Furthermore, the availability of annotations for only some

slices of the micro CT image is a notable limitation when it comes to training a segmentation algorithm for the CT images.

5.2.3 Pre-processing of the CT and micro CT images

We have developed a workflow to segment the calcifications in the CT images depicted in Figure 5.6. First, we select the region of interest within the images, which corresponds to the artery segment containing the calcifications. The selection process in the micro CT images is guided by annotations and employs a box detection algorithm. In contrast, the selection process in the CT images utilizes a centerline algorithm, that is provided by an expert. Normalization of the images is carried out to facilitate the subsequent super resolution, and segmentation processes.

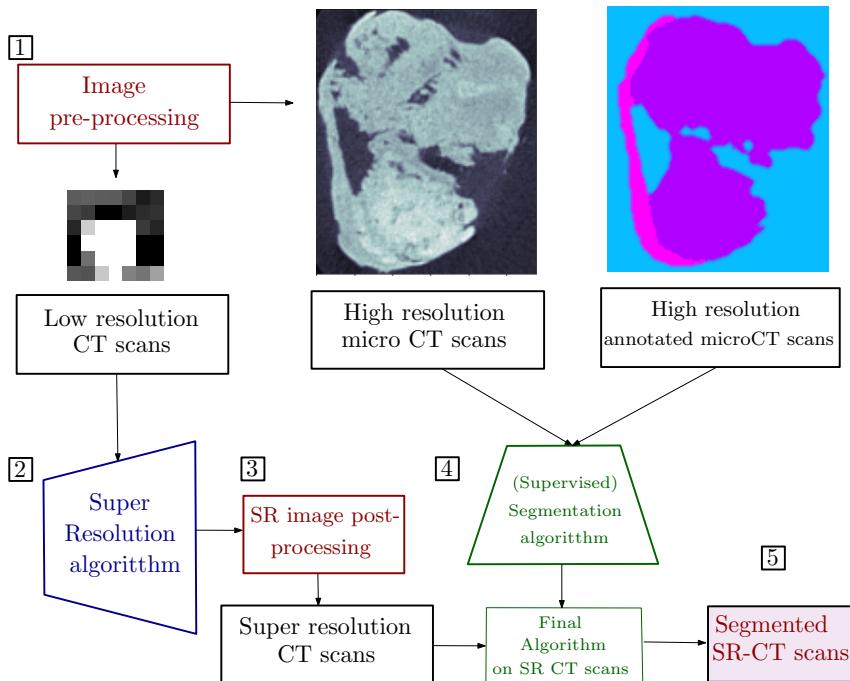


Figure 5.6: Our workflow for segmenting sheet and nodular calcifications in CT images of the SAFP is structured into five steps. First, we perform pre-processing of the CT and micro CT images, followed by a super resolution algorithm on the CT images, post-processing of the SR-CT images, supervised segmentation, and segmentation on the SR-CT images.

Moreover, the original size of the CT images containing calcifications is often small (5×5 to 12×12 pixels). Thus, we apply a Super Resolution (SR) algorithm to increase the image resolution to facilitate the segmentation process. In our case, a primary concern is the preservation of details in the CT images. Focusing on methods that enhance resolution without losing crucial information is key. Different SR techniques have been suggested, encompassing optimization methods and deep learning approaches. The latter have emerged as the most promising, with exponential growth (Li et al., 2021b). We have considered the LapSRN algorithm proposed by Lai et al. (2017), which utilizes a Laplacian pyramid framework. This algorithm has been selected for its ability to accurately reconstruct high-frequency details and reduce visual artifacts, which are crucial for medical images, especially CT scans. We also studied SR algorithms via VAEs, from a point of view of the applicability to medical images, we won't be able to use those algorithms (more details in Appendix F). This choice may not be definitive, and we will continue to explore other SR algorithms in the future.

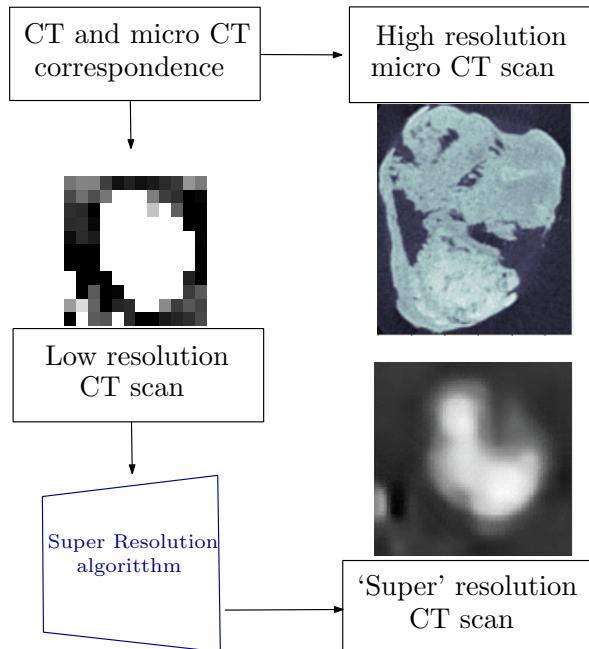


Figure 5.7: Example of a CT image of the SAFP and its corresponding micro CT image. In addition, the corresponding Super Resolution CT image after applying the LapSRN algorithm with a factor of up-scaling of 8.

Figure 5.7 shows an example of a SR-CT image obtained with the LapSRN algorithm. The original CT image is of size 12×12 pixels and the SR-CT image is of size 96×96 pixels, which is a factor of up-scaling of 8. After applying the SR algorithm, a post-processing phase is undertaken to eliminate any noise introduced by the SR algorithm. This involves an analysis of the SR-CT images in comparison to the micro CT images, enabling a medical interpretation of the results. The analysis entails a histogram comparison and a visual inspection of the images to determine the quality of the SR-CT images.

Figure 5.8 shows an example of a sequence of CT images where the calcifications are present. These new sequences of SR-CT images will be used for the segmentation of the calcifications in the next steps.

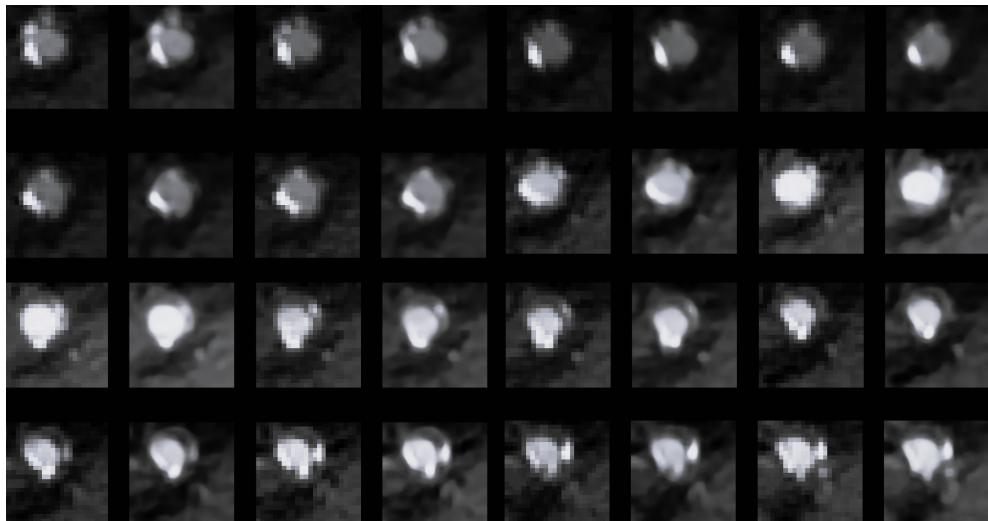


Figure 5.8: Example of a sequence of CT and SR-CT images. From left to right, the pairs of images (CT, SR-CT). The CT images correspond to a sequence of 2D slices of a 3D CT image, where the calcifications are present. The corresponding Super Resolution CT images are obtained with the LapSRN algorithm with a factor of up-scaling of 8.

5.3. Medical image segmentation

Semantic segmentation is a well-studied problem in the field of computer vision. The objective is to assign a label to each pixel of an image. In the context of medical imaging, this task is particularly challenging due to the complexity of the images and the limited availability of annotated data. In this section, we present the segmentation of the calcifications in the CT images of the SAFFP. This segmentation is performed in two steps. First, we perform a supervised segmentation using the pre-processed micro CT (HR images), and their corresponding ground truth data. Once a final segmentation model is obtained, we perform a segmentation on the SR-CT images. Our results are based on the U-Net model (Ronneberger et al., 2015), and the Probabilistic U-Net (Kohl et al., 2018).

The U-Net architecture is a type of convolutional neural network (CNN) that was specifically designed for biomedical image segmentation tasks proposed by Ronneberger et al. (2015). The U-Net architecture is a fully convolutional network that consists of a contracting path (encoder) and an expansive path (decoder), which gives it the U-shape. The contracting path follows the typical architecture of a convolutional network (Figure 5.9). This architecture has been remarkably successful due to its efficiency in learning from a limited number of samples while accurately segmenting images. The U-Net and its (non-stochastic) variants have been used in a variety of medical image segmentation tasks such as the bone segmentation (Caron et al., 2023; Ganesharaj et al., 2022), and the pancreas segmentation (Sriram et al., 2020).

The Probabilistic U-Net was introduced by Kohl et al. (2018), and designed to address the inherent ambiguities in real-world vision problems, especially in medical imaging. With ambiguous problems, there is no single correct answer. For example, the same image can be segmented in different ways by different experts, leading to different possible segmentations. The overlap between structures in the image can also lead to ambiguities. The Probabilistic U-Net incorporates a Conditional Variational AutoEncoder (CVAE) into the U-Net architecture (more details in Appendix F). The latent space is a low-dimensional space where the segmentation variants are represented as probability distributions. A sample from the latent space is drawn and then injected into the U-Net to produce the corresponding segmentation map. This model is trained using a variational inference approach (see Subsection 1.2.1), which allows the model to learn the distribution of segmentations in the latent space.

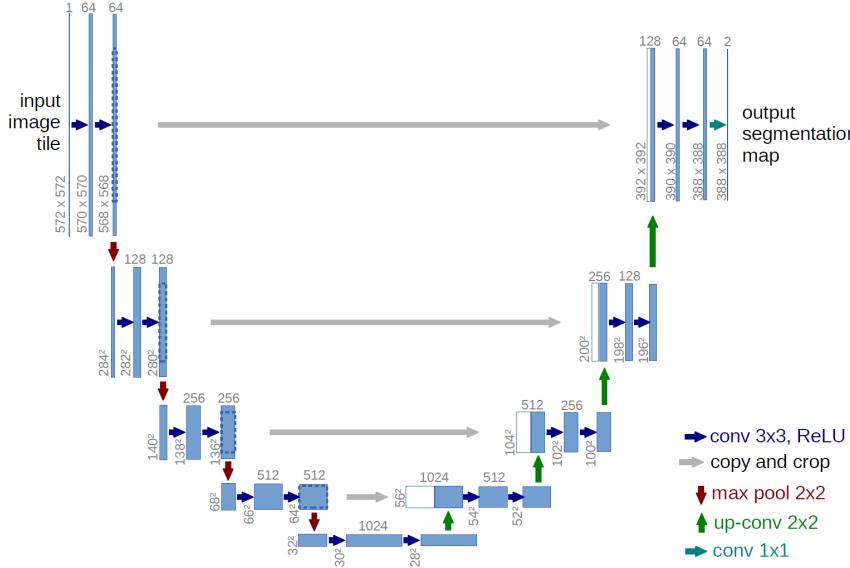


Figure 5.9: U-net architecture (example for 32×32 pixels in the lowest resolution). Each blue box corresponds to a multichannel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations. Figure taken from (Ronneberger et al., 2015)

5.3.1 Results

We aim to specifically segment calcifications in SR-CT images, which show areas of calcification in the artery. The segmentation algorithms are trained on a dataset which contains micro CT images, and their corresponding ground truth data representing the calcification zones. We present the results obtained with both, the U-Net, and Probabilistic U-Net models. We evaluate their effectiveness by calculating the Dice score (F.1) on the test set. This score provides an assessment of each model's performance in accurately segmenting, and classifying each class within CT images, crucial for informed doctor analysis

Three class segmentation: Table 5.1 summarizes the Dice score on the test set for the segmentation of the micro CT images. Three classes are considered: background (Ba), nodular calcifications (NC), and sheet calcifications (SC). In terms of the Dice score, the Probabilistic U-Net model outperforms the U-Net model for the calcifications classes. Once the segmentation model

is obtained, we perform a segmentation on the SR-CT images.

Model	Dice score on the test set		
	Ba	NC	SC
U-Net	0,7688	0,6032	0,5967
Probabilistic U-Net	0,7178	0,6214	0,6141

Table 5.1: Dice score on the test set for the U-Net and Probabilistic U-Net models. Three classes are considered: background (Ba), nodular calcifications (NC), and sheet calcifications (SC).

Figure 5.10 shows an example of segmentation of the micro CT image and its corresponding SR-CT image with the U-Net and Probabilistic U-Net models. We can see that both models are able to segment the calcifications in the SR-CT images, however, this analysis is not possible with all the sequence of SR-CT images. The results need to be analyzed carefully by a medical expert.

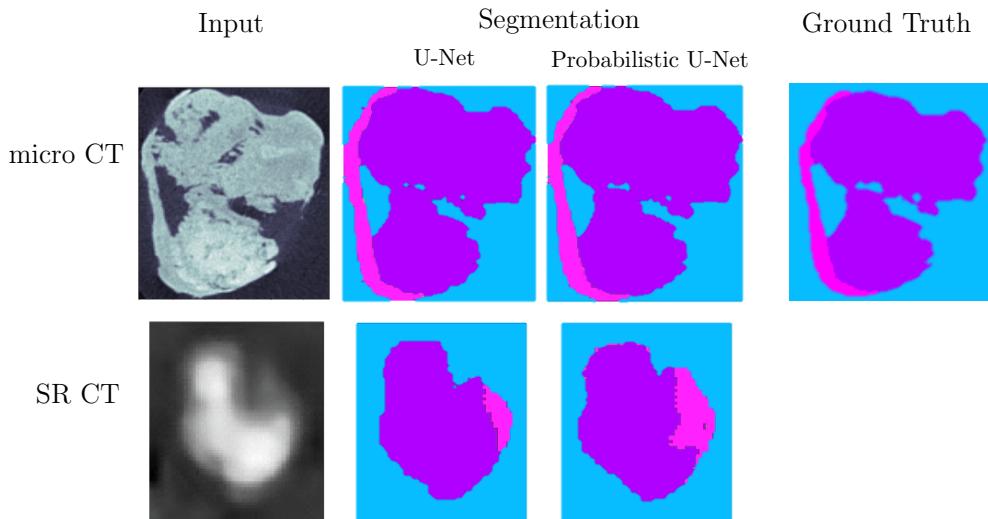


Figure 5.10: Example of a segmentation of the micro CT (first row) and its corresponding SR-CT images (second row) with the U-Net and Probabilistic U-Net models.

Four class segmentation: Initially, our segmentation model for CT images was designed to differentiate among three classes (background, nodular and sheet calcifications). However, we observed that calcifications were not consistently present across the entire sequence of CT images. This led us to introduce an additional class into our model: the soft tissue (ST) class, that encompasses both the arterial wall and the surrounding tissue, making its segmentation vital for an accurate doctor’s interpretation of the results. Table 5.2 shows the Dice scores obtained on the test set, now configured to identify four classes.

Model	Dice score on the test set			
	Ba	ST	NC	SC
U-Net	0,6027	0,6363	0,5807	0,6016
Probabilistic U-Net	0,6256	0,6439	0,5817	0,6751

Table 5.2: Dice score on the test set for the U-Net and Probabilistic U-Net models, with four classes: background (Ba), soft tissue (ST), nodular calcifications (NC), and sheet calcifications (SC).

Figure 5.11 presents some slices of the SR-CT images and their corresponding segmentation with Probabilistic U-Net models, with three and four classes. When we examine these results with the doctors, it becomes apparent that four-class segmentation offers greater interpretability compared to three-class segmentation.

5.4. Remaining challenges

We have made significant progress in the segmentation of sheet and nodular calcifications in CT images of the SAFP. However, several challenges remain to be addressed. First, the super-resolution algorithm itself does not take into account the sequential nature of the images, as it is applied independently to each slice. This can lead to inconsistencies across the image sequence. To address this problem, we have explored a post-processing technique for SR-CT images that aims to improve the consistency of results across the image sequence prior to segmentation.

On the other hand, segmentation with the U-Net, and Probabilistic U-Net models present the limitation related to their static nature. That is, when applied to SR-CT images, the segmentation performed by these models treats each slice independently, without taking into account the sequential context

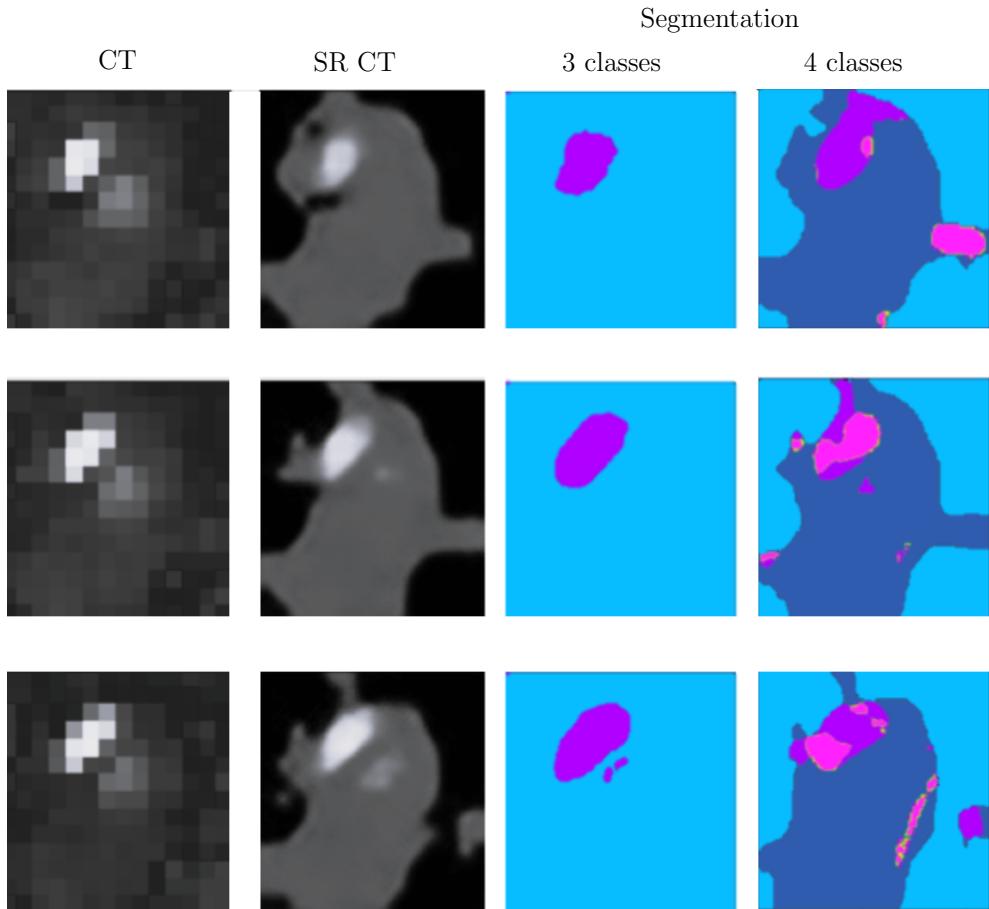


Figure 5.11: CT slides, their corresponding SR-CT images, and their corresponding 3 class and 4 class segmentations of SR-CT images.

that the images are part of a 3D image. This led to inconsistent segmentation results, with different results between two consecutive slices. To address this problem, we adapted the input of each model to include not only the target slice, but also the anterior and posterior slices, creating a “sliding window” effect. This modification is intended to incorporate some degree of sequential context into the segmentation process. In the future, we plan to explore other approaches to overcome these challenges, with the goal of developing more sophisticated models that can more accurately reflect the sequential and dynamic nature of the medical data, *e.g.* a sequential probabilistic U-Net model, a sequential SR VAE.

5.5. Conclusions

In this chapter, we have described a structured workflow for the segmentation of sheet and nodular calcifications in the CT images of the SAFP. This workflow encompasses five steps: pre-processing of the CT and micro CT images, application of the SR algorithm on the CT images, followed by post-processing of the SR-CT images, and finally the segmentation on the SR-CT images. First significant results from our research include those obtained using the LapSRN algorithm for the SR task, and the U-Net and Probabilistic U-Net models for segmentation. In particular, the Probabilistic U-Net model demonstrated superior performance to the U-Net model in segmenting the calcification classes. In addition, we observed that segmentation into four classes produces more detailed results, allowing a clearer distinction between calcifications and soft tissue, which is vital for a proper doctor's interpretation.

Conclusions and Perspectives

Throughout this thesis, our research has integrated traditional probabilistic models with modern deep learning techniques to address various challenges in machine learning. We have focused on generative sequential modeling, supervised, semi-supervised, unsupervised Bayesian classification, and a collaboration with the GEPROMED project to address the segmentation of medical images.

First, we introduced a novel generative model based on Pairwise Markov Chains, which effectively combines the strengths of Hidden Markov Models, Recurrent Neural Networks, and the Stochastic RNNs. This model considers observed and latent variables as well as the interactions between them, providing a more comprehensive representation of sequential data. We developed a new parameter estimation method leveraging the variational inference framework, which is both computationally efficient and straightforward to implement. The integration of deep parameterizations within this PMC model demonstrated superior performance on different datasets compared to traditional RNN and Stochastic RNN models. We also highlighted the linear and stationary Gaussian PMC's ability to model complex Gaussian distributions more effectively than previous models, by using the covariance function.

Moreover, if we consider the latent variables as discrete, *i.e.* the labels associated with each the observations, we demonstrated the potential of the PMC for supervised, and unsupervised classification tasks. In a supervised setting, our first variational framework can be easily adapted. In an unsupervised setting, we can use the PMC with traditional Bayesian parameter estimation methods, since the likelihood is tractable, *i.e.* VI is not necessary. However, the use of this Bayesian framework with neural networks in an unsupervised context is difficult due to the interpretation of the latent variables as discrete

labels. Thus, we proposed an alternative approach to address this issue by using a constrained output layer and a pretraining step to initialize the neural network.

Next, we extended our generative model to Triplet Markov Chains that incorporate an additional (continuous or discrete) process to model the interactions between the observed features and their corresponding labels. We illustrated the feasibility of creating diverse generative models based on variational inference, which is particularly advantageous for datasets with partially labeled observations or missing labels. We proposed a new adapted parameter estimation methods for the TMC model, that combines the variational inference framework, which is both computationally efficient, and interpretable in the context of sequential data classification. Each context, semi-supervised and unsupervised classification, has its own challenges, and we proposed different techniques to address them. For the semi-supervised context, we proposed a relaxation of the discrete variables using the Gumbel-Softmax trick, and for the unsupervised classification, we proposed a constrained output layer and pre-classification.

In addition, our collaboration with the GEPROMED project allowed us to know the challenges of medical image analysis. We proposed an adapted workflow to address the segmentation of medical images, which is a helpful tool for clinicians. We also applied classic super-resolution and segmentation techniques to medical images, which are essential for improving the interpretability of medical images. However, the results are not always satisfactory or adapted to the available data. We applied a probabilistic segmentation model that incorporates the variational framework with conditional VAEs.

In conclusion, the integration of traditional probabilistic models with modern deep learning techniques has shown promising results in various applications. The proposed models have demonstrated superior performance compared to traditional models, and the variational inference framework has proven to be a powerful tool for parameter estimation. Future work should continue to explore the integration of deep neural networks with other probabilistic models to develop more robust and efficient generative models. Research into various neural architectures (*e.g.* U-net), and training paradigms could further improve model performance and broaden their applicability. While this thesis primarily focused on medical imaging, the proposed methods and models have potential applications in other fields such as natural language processing, bioinformatics, and finance. Future research could explore these domains to validate the versatility and robustness of our models.

Moreover, the application of these models to medical image analysis should be further explored to improve the interpretability of medical images and enhance the workflow of clinicians. For example, the integration of the TMCs with the U-NET architecture could provide a more comprehensive representation of the sequential data, where the labels become the segmented images, and the observed features are the medical images.

In practice, semi-supervised classification tasks are challenging due to the discrete nature of the latent variables. The relaxation of the discrete variables using the Gumbel-Softmax trick provides a workaround, but it introduces a trade-off between the optimization of discrete variables and the quality of the approximation. Researchers continue to explore ways to improve the optimization of models with discrete variables, making them more tractable and effective for a wider range of applications. In the unsupervised case, we also noted that the DNN pretraining and the interpretability constraint require an available pre-classification. A future line of research involving self-supervised learning might prove itself as an efficient way to relax this requirement.

Finally, the stochastic realization theory can be also used to describe the covariance series which can be produced by linear and stationary PMCs, similar to the one used in the context of the GUM. The main difficulty is that they do not admit a state-space model representation due to the new dependencies introduced by the pairwise interactions, which makes the analysis more complex. The trick is to interpret the PMC as a particular HMC in augmented dimension. However, the theoretical analysis of the covariance series produced by general linear and stationary PMCs remains an open question.

Appendices

APPENDIX A

Additional material

Algorithm 9 Expectation Maximization ([Dempster et al., 1977](#))

Input: x , the observations.

Output: $\hat{\theta}$ the set of estimated parameters.

- 1: Initialize the parameters θ^0
- 2: $j \leftarrow 0$
- 3: **while** convergence is not attained **do**
- E-step:**
- 4: Define $\mathcal{Q}(\theta|\theta^j)$ by

$$\mathcal{Q}(\theta|\theta^j) = \mathbb{E}_{p(y|x,\theta^j)} [\log p(x, y|\theta)]. \quad (\text{A.1})$$

M-step:

- 5: Estimate the new set of parameters

$$\theta^{j+1} \leftarrow \arg \max_{\theta} \mathcal{Q}(\theta|\theta^j) \quad (\text{A.2})$$

- 6: $j \leftarrow j + 1$
 - 7: **end while**
 - 8: $\hat{\theta} \leftarrow \theta^j$
-

Lemma A.0.1. (Rao, 1973)

Let $x \in \mathbb{R}^p$, $y \in \mathbb{R}^q$, $F \in \mathbb{R}^{p \times q}$, $d \in \mathbb{R}^p$, $m \in \mathbb{R}^q$, Σ_1 and Σ_2 be $p \times p$ and $q \times q$ positive definite matrices, respectively. Then the following equality holds

$$\int_{y \in \mathbb{R}^q} \mathcal{N}(x; Fy + d, \Sigma_1) \mathcal{N}(y; m, \Sigma_2) dy = \mathcal{N}(x; Fm + d, \Sigma_1 + F\Sigma_2 F^T).$$

Conditional Variational Autoencoder

Let x , y , and z be the input image, the corresponding ground truth, and the latent representation, respectively. The CVAE is an extension of VAE (see Example 1.2.1) to conditional tasks such as image segmentation. Each component of the model is conditioned on some observed image x .

The ELBO objective function for the CVAE is defined as follows:

$$\mathcal{Q}_{\text{CVAE}}(x, y) = \mathbb{E}_{q_\phi(z|x, y)} [\log p_\theta(y|x, z)] - \text{KL}(q_\phi(z|x, y) || p(z|x)).$$

APPENDIX B

Generative Pairwise Markov Models

B.1. Proof of Theorem 2.5.2

Let's recall that in the stationary case, the function from \mathbb{N} to \mathbb{R} that associates r_k to any k is a covariance function (or a covariance sequence) if and only if, for any $T \geq 0$, the Toeplitz matrix with the first row $[r_0, r_1, \dots, r_T]$ is a covariance matrix, *i.e.* it is positive semi-definite. This set of constraints thus restricts the set of possible sequences, and we aim to characterize this set. $\{r_0, r_1, r_2, \dots\}$ is a covariance function if and only if $r_0 \geq 0$, and if

$$C(z) = r_0 + 2 \sum_{k=1}^{+\infty} r_k z^k$$

is a function of the Carathéodory class, *i.e.* $C(z)$ has a positive real part for z in the open unit disk (Carathéodory-Toeplitz theorem ([Akhiezer, 1965](#)))

Thus, we look for values of \tilde{A} and \tilde{B} such that the covariance matrix Σ_T^x with first row $[1, \tilde{B}, \tilde{A}^2, \tilde{A}^2 \tilde{B}, \tilde{A}^4, \tilde{A}^4 \tilde{B}, \dots]$ satisfies:

$$\forall T \in \mathbb{N}^*, \Sigma_T^x \geq 0 \iff \forall z \in \{u \in \mathbb{C}; |u| < 1\}, \Re \left(1 + 2(\tilde{B}z + \tilde{A}^2 z^2) \sum_{\tau=0}^{\infty} (\tilde{A}^2 z^2)^{\tau} \right) \geq 0,$$

which is derived from:

$$\begin{aligned}
C(z) &= 1 + 2(\tilde{B}z + \tilde{A}^2z^2 + \tilde{A}^2\tilde{B}z^3 + \tilde{A}^4z^4 + \tilde{A}^4\tilde{B}z^5 + \dots) \\
&= 1 + 2[\tilde{B}z((\tilde{A}^2z^2)^0 + (\tilde{A}^2z^2)^1 + (\tilde{A}^2z^2)^2 + (\tilde{A}^2z^2)^3 + \dots) \\
&\quad + \tilde{A}^2z^2((\tilde{A}^2z^2)^0 + (\tilde{A}^2z^2)^1 + (\tilde{A}^2z^2)^2 + (\tilde{A}^2z^2)^3 + \dots)] \\
&= 1 + 2(\tilde{B}z + \tilde{A}^2z^2) \sum_{\tau=0}^{\infty} (\tilde{A}^2z^2)^{\tau}
\end{aligned}$$

The positive real part condition is equivalent to:

$$\begin{aligned}
&\Re\left(1 + 2(\tilde{B}z + \tilde{A}^2z^2) \sum_{\tau=0}^{\infty} (\tilde{A}^2z^2)^{\tau}\right) \geq 0 \\
&\iff \Re\left(1 + 2 \frac{\tilde{B}z + \tilde{A}^2z^2}{1 - \tilde{A}^2z^2}\right) \geq 0 \\
&\iff \Re\left(\frac{1 + 2\tilde{B}z + \tilde{A}^2z^2}{1 - \tilde{A}^2z^2}\right) \geq 0 \\
&\stackrel{(ii)}{\iff} \Re\left(\frac{1 + 2\tilde{B}re^{i\theta} + \tilde{A}^2r^2e^{2i\theta}}{1 - \tilde{A}^2r^2e^{2i\theta}}\right) \geq 0 \\
&\iff \Re\left(\frac{(1 + 2\tilde{B}re^{i\theta} + \tilde{A}^2r^2e^{2i\theta})(1 - \tilde{A}^2r^2e^{-2i\theta})}{|1 - \tilde{A}^2r^2e^{2i\theta}|^2}\right) \geq 0 \\
&\iff \Re\left((1 + 2\tilde{B}re^{i\theta} + \tilde{A}^2r^2e^{2i\theta})(1 - \tilde{A}^2r^2e^{-2i\theta})\right) \geq 0 \\
&\iff 1 + 2\tilde{B}r \cos(\theta) - 2\tilde{A}^2\tilde{B}r^3 \cos(-\theta) - \tilde{A}^4r^4 \geq 0 \\
&\stackrel{(iii)}{\iff} 1 + 2\tilde{B}r \cos(\theta) - 2\tilde{A}^2\tilde{B}r^3 \cos(\theta) - \tilde{A}^4r^4 \geq 0 \\
&\iff 1 + 2\tilde{B}r \cos(\theta)(1 - \tilde{A}^2r^2) - \tilde{A}^4r^4 \geq 0,
\end{aligned}$$

where we used the following arguments:

- (i) $|\tilde{A}^2z^2| < 1$ since $\tilde{A} \in [-1, 1]$ and $|z| < 1$.
- (ii) Writing $z = re^{i\theta}$, for all $r \in [0, 1)$ and $\theta \in [-\pi, \pi]$.
- (iii) Cosine is an even function.

Thus, we need to analyze the expression:

$$1 + 2\tilde{B}r \cos(\theta)(1 - \tilde{A}^2r^2) - \tilde{A}^4r^4 \geq 0, \quad (\text{B.1})$$

and we can distinguish four cases:

1. Case $\tilde{A} = 0$: Let us first consider the case where $\tilde{A} = 0$. In this case, (B.1) simplifies to:

$$1 + 2\tilde{B}r \cos(\theta) \geq 1 - 2|\tilde{B}| \geq 0,$$

which implies $|\tilde{B}| \leq \frac{1}{2}$.

2. Case $\tilde{B} = 0$: We then have the condition $|\tilde{A}| \leq 1$, which is true.
3. Case $\tilde{B} > 0$:

$$\begin{aligned} & 1 + 2\tilde{B}r \cos(\theta)(1 - \tilde{A}^2 r^2) - \tilde{A}^4 r^4 \\ & \geq 1 - 2\tilde{B}(1 - \tilde{A}^2) - \tilde{A}^4. \end{aligned}$$

Note that $1 - \tilde{A}^2 r^2 \geq 0$ and $\tilde{A}^4 r^4 \geq 0$. Therefore,

$$\begin{aligned} & 1 + 2\tilde{B}r \cos(\theta)(1 - \tilde{A}^2 r^2) - \tilde{A}^4 r^4 \geq 0 \\ \iff & \tilde{B} \leq \frac{\tilde{A}^2 + 1}{2}. \end{aligned}$$

4. Case $\tilde{B} < 0$:

$$\begin{aligned} & 1 + 2\tilde{B}r \cos(\theta)(1 - \tilde{A}^2 r^2) - \tilde{A}^4 r^4 \\ & \geq 1 + 2\tilde{B}(1 - \tilde{A}^2) - \tilde{A}^4. \end{aligned}$$

Note that $1 - \tilde{A}^2 r^2 \geq 0$ and $\tilde{A}^4 r^4 \geq 0$. Therefore,

$$\begin{aligned} & 1 + 2\tilde{B}r \cos(\theta)(1 - \tilde{A}^2 r^2) - \tilde{A}^4 r^4 \geq 0 \\ \iff & \tilde{B} \geq -\frac{\tilde{A}^2 + 1}{2}. \end{aligned}$$

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

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

Now, the objective is to determine if any such probability distribution function can be modeled by some PMC model. For this, we study the inverse mapping of:

$$\phi : \theta \mapsto (\tilde{A} = \tilde{A}(\theta), \tilde{B} = \tilde{B}(\theta)), \quad (\text{B.2})$$

where θ represents the set of parameters of the model.

We set $\gamma = b$, and f either as 0 or $-a - bc$ (two particular cases of the PMC), that coincide with (2.23). The following expressions for \tilde{A} and \tilde{B} are obtained:

$$\begin{cases} \tilde{A} = \sqrt{ce} & \text{and } \tilde{B} = b(c(1 - b^2\eta) + e\eta) & \text{if } f = 0, \\ \tilde{A} = \sqrt{e^2\eta + a^2(1 - b^2\eta)} & \text{and } \tilde{B} = be\eta - a(1 - b^2\eta) & \text{if } f = -a - bc. \end{cases}$$

First, the case $f = 0$, $\gamma = b$ implies that $a = -bc$, so the set of parameters is b, c, e, η since a, f , and γ are functions of these parameters. Thus, ϕ can be written as:

$$\phi : (b, c, e, \eta) \mapsto (\tilde{A} = \sqrt{ce}, \tilde{B} = b(c(1 - b^2\eta) + e\eta)). \quad (\text{B.3})$$

The domain (\tilde{A}, \tilde{B}) has been characterized to obtain a covariance matrix, *i.e.* $\tilde{A} \in [-1, 1]$ and $-\frac{\tilde{A}^2+1}{2} \leq \tilde{B} \leq \frac{\tilde{A}^2+1}{2}$, which defines a surface \mathcal{S} . We obtain an inverse mapping ϕ^{-1} of Equation (B.3), showing that for some $(\tilde{A}, \tilde{B}) \in \mathcal{S}$, there exists at least one PMC which yields an observation probability distribution. For simplicity, we do not show the detailed inverse mappings and their calculations here, as they are lengthy and complex. The important result is that such a mapping exists and is consistent with the conditions stated above.

Next, the case $f = -a - bc$, $\gamma = b$ implies $c = e\eta - ab\eta$. The set of parameters is then a, b, e, η , and ϕ can be written as:

$$\phi : (a, b, e, \eta) \mapsto (\tilde{A} = \sqrt{e^2\eta + a^2(1 - b^2\eta)}, \tilde{B} = be\eta - a(1 - b^2\eta)). \quad (\text{B.4})$$

Similarly, we can obtain the inverse mapping ϕ^{-1} of Equation (B.4), showing that for some $(\tilde{A}, \tilde{B}) \in \mathcal{S}$, there exists at least one PMC which yields an observation probability distribution.

APPENDIX C

Supervised Bayesian classification

PMCs can be adapted for the supervised classification task by considering an observed variable in an augmented dimension $x_t \leftarrow (x_t, y_t)$. We add a discrete variable y_t label associated to x_t , for all $t \in \mathbb{N}$. The parameter estimation is realized by maximizing the ELBO with respect to θ and ϕ where the general ELBO in (2.5) is still valid and reads as

$$\begin{aligned} \mathcal{Q}_{\text{sup}}(\theta, \phi) = & - \int \log \left(\frac{q_\phi(z_0|x_{0:T}, y_{0:T})}{p(x_0, y_0, z_0)} \right) q_\phi(z_0|x_{0:T}, y_{0:T}) dz_{0:T} \\ & - \sum_{t=1}^T \int \log \left(\frac{q_\phi(z_t|z_{0:t-1}, x_{0:T}, y_{0:T})}{p_\theta(z_t, x_t, y_t|z_{t-1}, x_{t-1}, y_{t-1})} \right) q_\phi(z_{0:T}|x_{0:T}, y_{0:T}) dz_{0:T}. \end{aligned}$$

The transition distribution $p_\theta((x_t, y_t), z_t|(x_{t-1}, y_{t-1}), z_{t-1})$ can be factorized in two terms as shown in (2.1). Without loss of generality, we can consider the following factorization,

$$\begin{aligned} p_\theta(z_t, x_t, y_t|z_{t-1}, x_{t-1}, y_{t-1}) = & p_\theta(x_t, y_t|z_{t-1:t}, x_{t-1}, y_{t-1}) p_\theta(z_t|z_{t-1}, x_{t-1}, y_{t-1}) \\ = & p_\theta(x_t|z_{t-1:t}, x_{t-1}, y_{t-1}) p_\theta(y_t|z_{t-1:t}, x_{t-1:t}, y_{t-1}) \times \\ & p_\theta(z_t|z_{t-1}, x_{t-1}, y_{t-1}), \end{aligned} \tag{C.1}$$

which is nothing more than a TMC with transition (C.1). The ELBO now reads

$$\mathcal{Q}_{\text{sup}}(\theta, \phi) = \mathcal{L}_1(\theta, \phi) + \mathcal{L}_2(\theta, \phi) \tag{C.2}$$

with

$$\begin{aligned}\mathcal{L}_1(\theta, \phi) &= \mathbb{E}_{q_\phi(z_0|x_{0:T}, y_{0:T})} \log p_\theta(x_0|z_0) + \mathbb{E}_{q_\phi(z_0|x_{0:T}, y_{0:T})} \log p_\theta(y_0|z_0, x_0) \\ &\quad + \sum_{t=1}^T \mathbb{E}_{q_\phi(z_t|z_{0:t-1}, x_{0:T}, y_{0:T})} \log p_\theta(x_t|z_{t-1:t}, x_{t-1}, y_{t-1}) \\ &\quad + \sum_{t=1}^T \mathbb{E}_{q_\phi(z_t|z_{0:t-1}, x_{0:T}, y_{0:T})} \log p_\theta(y_t|z_{t-1:t}, x_{t-1:t}, y_{t-1}), \\ \mathcal{L}_2(\theta, \phi) &= -\text{D}_{\text{KL}}(q_\phi(z_0|x_{0:T}, y_{0:T})||p_\theta(z_0)) \\ &\quad - \sum_{t=1}^T \text{D}_{\text{KL}}(q_\phi(z_t|z_{0:t-1}, x_{0:T}, y_{0:T})||p_\theta(z_t|z_{t-1}, x_{t-1}, y_{t-1})).\end{aligned}$$

The training procedure of the PMC model presented in Algorithm 1, can be adapted for the supervised classification task. The only distinction with the previous algorithm is the set of parameters θ , which now includes the parameters of the conditional distribution of the labels $p_\theta(y_t|z_{t-1:t}, x_{t-1:t}, y_{t-1})$.

APPENDIX D

Semi-supervised Bayesian classification

The VSL (Chen et al., 2018) is based on conditional VAEs (Pagnoni et al., 2018), where at each time step t , the observation x_t is generated according its associated context u_t , which consists of the observations other than x_t . The lower bound of the log-likelihood at each time step t is given by

$$\begin{aligned}\log p_\theta(x_t|u_t) &\leq \mathbb{E}_{q_\phi(z_t|x_{0:T})} [\log p_\theta(x_t|z_t, u_t) p_\theta(z_t|u_t) p_\theta(y_t|z_t, u_t)], \text{ for all } t \in \mathbf{O}. \\ \log p_\theta(x_t|u_t) &\leq \mathbb{E}_{q_\phi(z_t, y_t|x_{0:T})} [\log p_\theta(x_t|z_t, u_t) p_\theta(z_t|u_t) p_\theta(y_t|z_t, u_t)], \text{ for all } t \in \mathbf{H}.\end{aligned}$$

The VSL model simplifies some dependencies by assuming that $p_\theta(y_t|z_t, u_t) = p_\theta(y_t|z_t)$ and $p_\theta(x_t|z_t, u_t) = p_\theta(x_t|z_t)$. While the associated variational distribution is given by

$$q_\phi(z_{0:T}, y_T^{\mathbf{H}}|x_{0:T}, y_T^{\mathbf{O}}) = \prod_{t=0}^T q_\phi(z_t|x_{0:T}) \prod_{t \in \mathbf{H}} q_\phi(y_t|z_t),$$

which satisfies the factorization (3.8) with

$$\begin{aligned}q_\phi(z_t|z_{t-1}, y_{t-1}, x_{0:T}, y_{t+1:T}^{\mathbf{O}}) &= q_\phi(z_t|x_{0:T}), \\ q_\phi(y_t|y_{t-1}, z_t, x_{0:T}, y_{t+1:T}^{\mathbf{O}}) &= p_\theta(y_t|z_t), \text{ for all } t \in \mathbf{H}.\end{aligned}$$

Our proposed variation of this model considers $u_t = (x_{t-1}, z_t)$, i.e. we assume the context u_t depends on the previous observation x_{t-1} and the current

latent variable z_t . The associated ELBO (3.10) for the VSL model is given by

$$\begin{aligned} \mathcal{Q}_{\text{semi}}(\theta, \phi) &\stackrel{\text{mVSL}}{=} \sum_{t \in \mathbf{O}} \mathbb{E}_{q_\phi(z_t|x_{0:T})} (\log p_\theta(y_t|z_t)) + \\ &\quad \sum_{t=0}^T \left[\mathbb{E}_{q_\phi(z_t|x_{0:T})} \log p_\theta(x_t|z_t) - \text{D}_{\text{KL}}(q_\phi(z_t|x_{0:T}) || p_\theta(z_t|x_{t-1}, z_{t-1})) \right]. \end{aligned}$$

It consists of two terms and that the previous assumptions enable us to interpret it as an expectation according to $q_\phi(z_{0:T}|x_{0:T})$. Thus, it is not necessary to sample discrete variables according to the G-S trick. Moreover, a regularization term β can be introduced in the second part of the ELBO in order to encourage good performance on labeled data while leveraging the context of the noisy observations during reconstruction. While this model simplifies the inference, it should be noted that in the generative process, the observation x_t is conditionally independent of its associated label and may not be adapted to some applications.

APPENDIX E

Unsupervised Bayesian classification

E.1. Partially Pairwise Markov Chains

In this section, we propose a particular class of TMC which aims at extending the PMC model proposed in Section 4.2. The main motivation underlying this particular model is to introduce an explicit dependency on the past observations x_{t-1} of the pair (y_t, x_t) , for all t . This dependency is introduced through the continuous latent process $z_{0:T}$ and enables us to build an explicit joint distribution $p_\theta(y_{0:T}, x_{0:T})$ which does not satisfy the Markovian property of the PMC (4.1). The main difference with Section 4.3 is that $z_{0:T}$ is now a conditional deterministic latent process. The resulting model is called a Partially Pairwise Markov Chain (PPMC). As we will see, this particular construction enables us to use directly the Bayesian inference framework developed in Section 4.2. Finally, since PMCs appears as particular TMCs, the pretraining of deep parameterized PPMCs is a direct adaptation of Section 4.3.3.

E.1.1 Deterministic TMCs

Let us focus on a particular case of the TMC (4.24)-(4.27). From now on, we consider that the conditional distribution η coincides with the Dirac distribution δ , and that function ψ_θ^z only depends on (z_{t-1}, x_{t-1}) . Thus, z_t becomes deterministic given (z_{t-1}, x_{t-1}) ,

$$z_t = \psi_\theta^z(z_{t-1}, x_{t-1}). \quad (\text{E.1})$$

Each variable z_t can be interpreted as a summary of all the past observations x_{t-1} . Consequently, it is easy to see that (4.26) and (4.27) now coincide with

$p_\theta(y_t|y_{t-1}, x_{t-1})$ and $p_\theta(x_t|y_{t-1:t}, x_{t-1})$, respectively, and marginalizing (1.8) w.r.t. $z_{0:T}$ gives the explicit distribution of $(y_{0:T}, x_{0:T})$,

$$p_\theta(y_{0:T}, x_{0:T}) = p_\theta(y_0, x_0) \prod_{t=1}^T \underbrace{\vartheta(y_t; \psi_\theta^y(zt-1:t, y_{t-1}, x_{t-1}))}_{p_\theta(y_t|y_{t-1}, x_{t-1})} \times \underbrace{\zeta(x_t; \psi_\theta^x(zt-1:t, y_{t-1:t}, x_{t-1}))}_{p(x_t|y_{t-1:t}, x_{t-1})}, \quad (\text{E.2})$$

where z_t satisfies (E.1). It can be noted that $(y_{0:T}, x_{0:T})$ is no longer Markovian. Remark that this property is also satisfied by the general TMC (1.8). However, $p_\theta(y_{0:T}, x_{0:T})$ is now available in a closed-form expression and the relationship between the pair (y_t, x_t) and the past observations is fully characterized by the function ψ_θ^z .

This kind of parameterization has an advantage in terms of Bayesian inference. Since z_t is a deterministic function of (z_{t-1}, x_{t-1}) (and so of x_{t-1} , by induction), the conditional posterior distribution $p_\theta(z_t|z_{t-1}, x_{0:T})$ reduces to $\delta_{\psi_\theta^z(z_{t-1}, x_{t-1})}$. Consequently, Algorithm 3 and Algorithm 4 can be directly applied to estimate θ and y_t , for all t , by introducing the dependency in $z_{t-1:t}$ in functions ψ_θ^y and ψ_θ^x of Section 4.2.1. An alternative point of view is that when z_t is deterministic, Algorithm 6 can be seen as a particular instance of Algorithm 3 in which we have set $q_\phi(z_{0:T}|x_{0:T}) = p_\theta(z_{0:T}|x_{0:T})$, $\beta_1 = 1$ and $\beta_2 = 0$. Indeed, for this particular setting the objective function (4.37) coincides with the ELBO but also with the log-likelihood $p_\theta(x_{0:T})$.

E.1.2 Deep PPMCs

As previous models, we consider the case where PPMCs (E.2) are parameterized with DNN. Such models will be referred to as Deep Partially Pairwise Markov Chain (DPPMC). In the particular case of PPMCs, ψ_θ^z can be seen as a RNN, *i.e.* a neural network which admits the output of the network at previous time $t-1$ as input at time t (Hochreiter & Schmidhuber, 1997). It is thus possible to directly combine our models with powerful RNN architectures such as Long Short Term Memory (LSTM) RNNs or Gated Recurrent Unit (GRU) RNNs which have been developed to introduce emphasize sequential dependencies. Note that the gradient of ψ_θ^z w.r.t. θ can also be computed with a version of the backpropagation algorithm adapted to RNNs (Hochreiter & Schmidhuber, 1997; Chung et al., 2014).

The pretraining of this deep architecture is direct. The constrained output layer step is an application of Paragraph 4.3.3 with $q_\phi(z_{0:T}|x_{0:T}) = p_\theta(z_{0:T}|x_{0:T})$,

$\beta_1 = 1$ and $\beta_2 = 0$; so it can be seen as the step described for PMCs in Paragraph 4.2.2 up to the additional input $z_{t-1:t}$.

The second step of our pretraining procedure of Paragraph 4.3.3 can also be simplified. Since in this particular case we have implicitly computed the optimal conditional variational distribution $q_\phi^{\text{opt}}(z_t|z_{0:t-1}, x_{0:T}) = \delta_{\psi_\theta^z(z_{t-1}, x_{t-1})}(z_t)$, the reparameterized sample $z_{t-1:t}$ of Figure 4.4 is now deterministic and coincides directly with the output of ψ_θ^z , as shown in Figure E.1. Note that the parameters of ψ_θ^z are unfrozen. The training process is summarized in Algorithm 10.

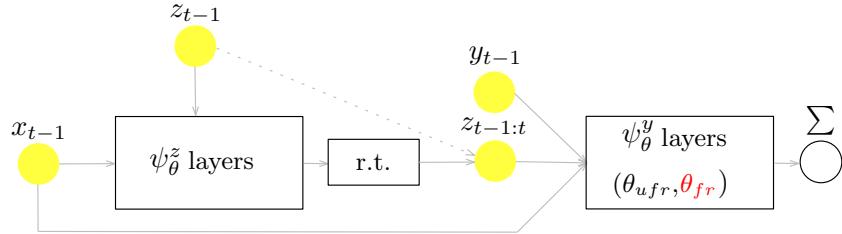


Figure E.1: Graphical and condensed representation of the parameterization of ψ_θ^y in the DPPMC model. The dashed arrows represent the fact that some variables are copied.

Algorithm 10 A general estimation algorithm for deep parameterizations of PPMC models.

Input: $x_{0:T}$, the observation

Output: $\hat{y}_{0:T}$, the final classification

Initialization of the output layer of ψ_θ^y and ψ_θ^x

1: Estimate θ_{fr}^* and $\hat{y}_{0:T}^{\text{pre}}$ with Lines (1)-(3) of Algorithm 5

Pretraining of θ_{ufr}

2: $\theta_{\text{ufr}}^{(0)} \leftarrow \text{Backprop}(\hat{y}_{0:T}^{\text{pre}}, x_{0:T}, \theta_{\text{fr}}^*, \mathcal{C}_{f_\theta}, \mathcal{C}_{g_\theta})$

Fine-tuning of the complete model

3: Update all the models parameters (except θ_{fr}) with Algorithm 3

4: Compute $\hat{y}_{0:T}$ with Algorithm 4

E.1.3 Simulations

We start again with the same experiments as those in Section 4.2.3, but we use an alternative noise which aims at introducing longer dependencies on

the observations. We now set

$$x_t|y_t, x_{t-2:t-1} \sim \mathcal{N}\left(\sin(a_{y_t} + 0.2(x_{t-1} + x_{t-2})); \sigma^2\right). \quad (\text{E.3})$$

where $a_{\omega_1} = 0$, $\sigma^2 = 0.25$ and a_{ω_2} is a varying parameter. We compare the deep models of Section 4.2 (DSMPC and DPMC) with their natural extensions developed in this section (DPSPMC and DPPMC).

Figure E.2a illustrates the results involving the models we have just introduced. For ψ_θ^z we use two independent standard RNNs with ReLU activation function, i.e. $z_t = [z_t^1, z_t^2] = [\psi_\theta^{z1}(z_{t-1}^1, x_{t-1}), \psi_\theta^{z2}(z_{t-1}^2, x_{t-1})]$; ψ_θ^y (resp. ψ_θ^x) depends on $z_{t-1:t}^1$ (resp. $z_{t-1:t}^2$). In this setting, we found that the models worked the best when the dimensions of z_t^1 and of z_t^2 is 5. We can see that the more general parameterizations embedded in DPSPMC and DPPMC lead to an improvement of the DPMC models; each DPPMC model leading to a better accuracy than its DPMC counterpart. The ability to model long term dependencies proves to be important to better solve the correlated noise. This experiment illustrates a way to take advantage of a deterministic auxiliary process: by strengthening the sequential dependencies between the hidden random variables.

E.2. Additional material

E.2.1 Proof of Proposition 4.3.1

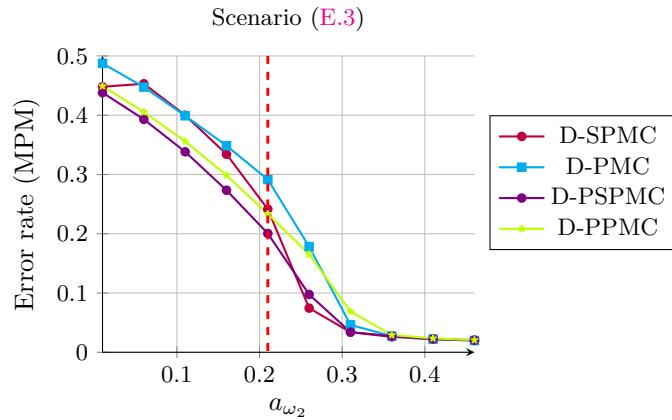
The ELBO

$$Q(\theta, \phi) = \sum_{y_{0:T}} \int q_\phi(y_{0:T}, z_{0:T}|x_{0:T}) \log \left(\frac{p_\theta(y_{0:T}, z_{0:T}, x_{0:T})}{q_\phi(y_{0:T}, z_{0:T}|x_{0:T})} \right) dz_{0:T} \quad (\text{E.4})$$

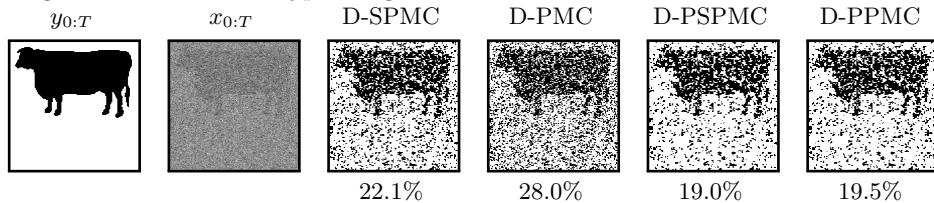
can be decomposed as

$$\begin{aligned} Q(\theta, \phi) &= \int \overbrace{\left[\sum_{y_{0:T}} q_\phi(y_{0:T}|z_{0:T}, x_{0:T}) \right]}^1 q_\phi(z_{0:T}|x_{0:T}) \log \left(\frac{p_\theta(z_{0:T}, x_{0:T})}{q_\phi(z_{0:T}|x_{0:T})} \right) dz_{0:T} \\ &\quad - \int q_\phi(z_{0:T}|x_{0:T}) \text{D}_{\text{KL}}(q_\phi(y_{0:T}|z_{0:T}, x_{0:T}) || p_\theta(y_{0:T}|z_{0:T}, x_{0:T})) dz_{0:T}, \end{aligned} \quad (\text{E.5})$$

$$\leq \int q_\phi(z_{0:T}|x_{0:T}) \log \left(\frac{p_\theta(z_{0:T}, x_{0:T})}{q_\phi(z_{0:T}|x_{0:T})} \right) dz_{0:T} = Q^{\text{opt}}(\theta, \phi). \quad (\text{E.6})$$



(a) Error rate from the unsupervised segmentations of Scenario (E.3). 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 E.2a). Error rates appear below the images.

Figure E.2: Unsupervised image segmentation with Partially Pairwise Markov Chains.

We have $Q(\theta, \phi) = Q^{\text{opt}}(\theta, \phi)$ when the KLD term in (E.5) is null, *i.e.* when $q_\phi(y_{0:T}|z_{0:T}, x_{0:T}) = p_\theta(y_{0:T}|z_{0:T}, x_{0:T})$. It remains to compute $Q^{\text{opt}}(\theta, \phi)$. Starting again from (E.4) where we set

$$q_\phi(y_{0:T}, z_{0:T}|x_{0:T}) = q_\phi(y_{0:T}|x_{0:T})p_\theta(y_{0:T}|z_{0:T}, x_{0:T}),$$

the Markovian structure of $p_\theta(y_{0:T}, z_{0:T}, x_{0:T})$ and the additive property of the logarithm function give the decomposition (4.21)-(4.23).

Note that the computation of $Q^{\text{opt}}(\theta, \phi)$ via (4.21)-(4.23) relies on the distribution $p_\theta(y_{t-1:t}|z_{0:T}, x_{0:T})$. It can be computed from a direct extension of the intermediate quantities $\alpha_{\theta,k}$ and $\beta_{\theta,k}$ which are now defined as $\alpha_{\theta,k}(y_t) = p_\theta(y_t, z_{0:t}, x_{0:t})$ and $\beta_{\theta,k}(y_t) = p_\theta(z_{t+1:K}, x_{t+1:K}|y_t, z_t, x_t)$. Their computation is similar to (4.13) and (4.15), except that they now involve the transition $p(y_t, z_t, x_t|y_{t-1}, z_{t-1}, x_{t-1})$ rather than $p(y_t, x_t|y_{t-1}, x_{t-1})$.

E.2.2 Detailed error rates for experiments 4.4.1 and 4.4.2

This section provides the full results of the real world experiments described in Section 4.4. Table E.1 provides a comprehensive comparison of the error rates achieved by different generalized Triplet Markov Chains in the context of unsupervised image segmentation. The table presents detailed error rates for ten micro-computed tomography slices, evaluated across four models: HMC-IN, di-MTMC, MTMC, and DMTMC.

Slice	HMC-IN	di-MTMC	MTMC	DMTMC
A	8.5	8.5	6.5	5.4
B	10.9	10.9	8.7	6.5
C	6.9	7.0	6.0	5.2
D	10.0	10.1	8.3	6.1
E	6.5	6.3	6.2	5.4
F	11.5	11.5	10.8	9.3
G	4.6	4.6	3.9	3.7
H	8.6	8.6	8.5	7.7
I	11.5	11.5	10.1	9.2
J	7.2	7.2	6.9	6.5
Average	8.6	8.6	7.6	6.5

Table E.1: Detailed error rates (%) in unsupervised image segmentation with all the generalized TMCs assessed on ten micro-computed tomography slices. See Section 4.4.1.

E.2.3 Additional experiments

In this section, we provide additional experiments. The first one consists in introducing experiments in the case where the number of classes is $C > 2$. In the second one, we study experimentally the impact of the variational distribution for the TMC model of Scenario (4.46).

Multi-class extension

In this section, we illustrate an extension of our models when $C > 2$. For $C > 2$, Eq. (4.6) becomes a vector of softmax function,

$$f_\theta(y_{t-1}, x_{t-1}) = \left[\frac{e^{b_{\omega_1, y_{t-1}}}}{\sum_{j=1}^C e^{b_{\omega_j, y_{t-1}}}}, \dots, \frac{e^{b_{\omega_C, y_{t-1}}}}{\sum_{j=1}^C e^{b_{\omega_j, y_{t-1}}}} \right], \quad (\text{E.7})$$

while the distribution $\lambda(y_t, f_\theta(y_{t-1}, x_{t-1}))$ coincides with the Categorical distribution whose parameters are described by $f_\theta(y_{t-1}, x_{t-1})$, *i.e.*

$$p_\theta(y_t = \omega_i | y_{t-1}, x_{t-1}) \stackrel{\text{HMC}}{=} p_\theta(y_t = \omega_i | y_{t-1}) = \frac{e^{b_{\omega_i, y_{t-1}}}}{\sum_{j=1}^C e^{b_{\omega_j, y_{t-1}}}}. \quad (\text{E.8})$$

Data	HMC-IN	SPMC	DSPMC	DPSPMC	PMC	DPMC	DPPMC
acc_exp01_user01	15.0	29.0	20.9	17.8	20.9	19.9	20.1
acc_exp02_user01	16.0	20.3	13.3	12.4	13.1	18.2	14.6
acc_exp03_user02	25.7	16.1	11.7	9.8	11.7	5.6	12.7
acc_exp04_user02	24.3	15.2	10.9	11.5	10.9	5.6	11.7
acc_exp05_user03	21.1	28.8	23.2	15.3	22.4	22.7	23.4
acc_exp06_user03	26.3	15.6	12.9	11.0	12.3	19.9	14.2
acc_exp07_user04	23.3	19.2	14.4	13.4	23.3	21.9	14.6
acc_exp08_user04	26.3	17.1	13.1	12.3	12.9	10.4	12.9
acc_exp09_user05	24.3	19.0	14.9	12.3	14.7	12.3	15.5
acc_exp10_user05	25.8	48.3	24.5	25.4	24.3	27.6	24.3
acc_exp11_user06	27.7	15.1	12.7	10.9	12.7	12.6	11.9
acc_exp12_user06	36.9	43.5	42.8	43.2	42.8	42.1	41.5
acc_exp13_user07	26.1	18.2	14.6	16.5	14.4	13.9	13.9
acc_exp14_user07	26.0	18.5	14.5	21.9	14.4	18.9	13.6
acc_exp15_user08	22.2	16.7	12.9	9.0	12.8	10.0	13.0
acc_exp16_user08	26.2	19.4	16.5	14.7	16.5	15.8	14.3
acc_exp17_user09	25.6	17.0	13.1	17.9	12.9	14.0	11.0
acc_exp18_user09	24.8	13.8	10.9	11.3	10.8	8.1	12.3
acc_exp19_user10	26.1	13.3	10.4	21.4	10.3	8.0	15.2
acc_exp20_user10	34.9	22.1	27.2	26.8	27.1	29.1	25.9
Average	25.2	21.3	16.8	16.7	17.1	16.8	16.8

Table E.2: Detailed Error rates (%) in the binary clustering of the first twenty raw entries of the HAPT dataset ([Reyes-Ortiz et al., 2016](#)).

Figure E.3 displays an extension of Scenario (3.19) to the multi-class case. One can note that the relative performances of the models remain similar to those established in the article.

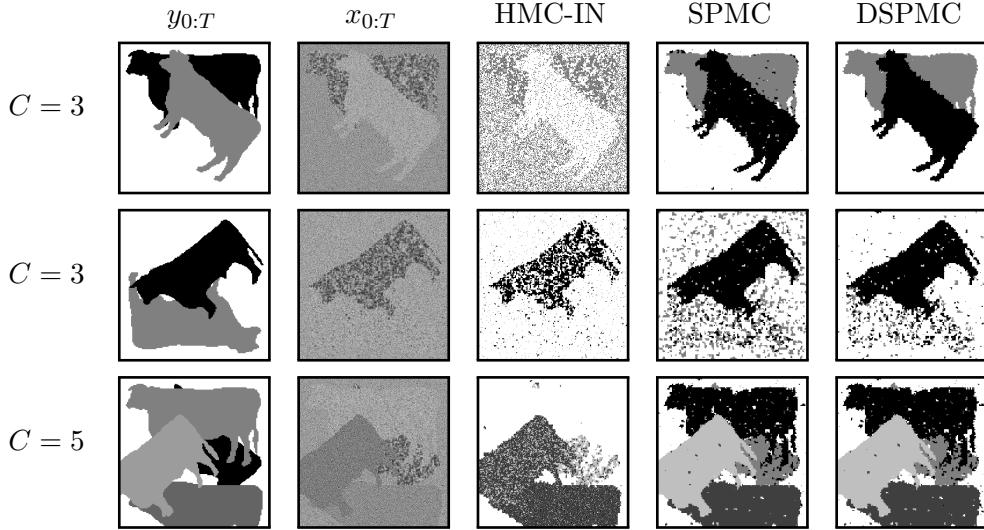


Figure E.3: Multi-class segmentations with the HMC-IN, SPMC and DSPMC models. The noisy image is simulated according to Eq. (3.19) from the paper with $a_{\omega_1} = 0, a_{\omega_2} = 1$ and $a_{\omega_3} = 2$ for the top row, $a_{\omega_1} = 0, a_{\omega_2} = 0.5$ and $a_{\omega_3} = 1$ for the middle row and $a_{\omega_1} = 0, a_{\omega_2} = 0.75, a_{\omega_3} = 1.5, a_{\omega_4} = 2.25$ and $a_{\omega_5} = 3$ for the bottom row. Note that the segmentation can be affected by label switching, which is another different problem out of scope of the article.

Influence of the variational distribution

In this section, we performed new simulations in the case of the non-stationary noise experiment (Scenario (4.46)) with 3 different variational distributions for the DMTMC model, namely:

$$q_{\phi}^1(z_{0:T}|x_{0:T}) = \prod_{t=1}^T \mathcal{N}(z_t; \nu_{\phi}(x_t)),$$

$$q_{\phi}^2(z_{0:T}|x_{0:T}) = \prod_{t=1}^T \mathcal{N}(z_t; \nu_{\phi}(z_{t-1}, x_t)),$$

and

$$q_{\phi}^3(z_{0:T}|x_{0:T}) = \prod_{t=1}^T \mathcal{N}(z_t; \nu_{\phi}(z_{k-2}, z_{t-1}, x_t)).$$

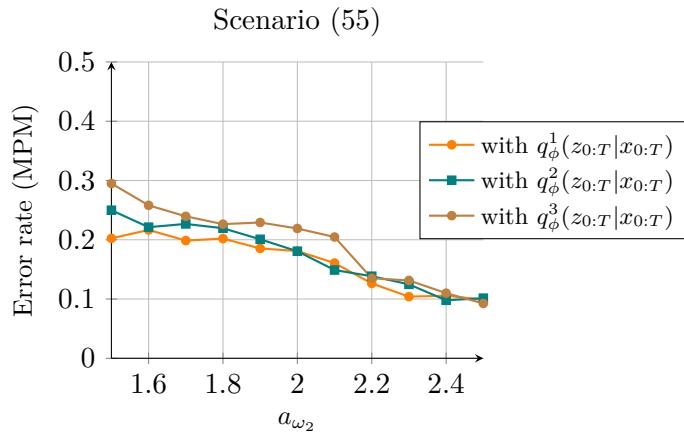


Figure E.4: Error rate from the unsupervised segmentations of Scenario (4.46). Results are averaged on all the *dog*-type images from the database.

Figure E.4 summarizes this additional experiment.

It can be observed that the choice of the variational distribution does not lead to significant changes in the results as compared to, for example, the Mean-Field variational distribution with fully independent random variables. However, adding more dependencies led to worse results probably because of the complexity of the noise to estimate.

APPENDIX F

Medical image segmentation

F.1. Protocol and Database Construction

We describe the protocol and the database construction as follows.

- **Patients and explant recovery:**

1. Patients scheduled for transfemoral amputation in the vascular surgery department are informed of the procedure and asked not to object during the pre-operative consultation scheduled for the day before the operation.
2. Management of the patient in accordance with current practice, with an injected preoperative CT scan.
3. During routine transfemoral amputation surgery: recovery of the sample (portion of the amputation including the damaged artery) to be analysed after rinsing the artery lumen.
4. Recovery of the explant with macroscopic analysis at GEPROMED and storage on their premises. The subjects' participation in the research ends after the surgery.

- **GEPROMED:** Ex-vivo microscanner imaging at GEPROMED.

1. The microCT 3D images of the arteries are acquired at the CVPATH Institute, Inc. (Gaithersburg, MD, USA).
2. Histology are performed on the specimens as described in [Torii et al. \(2019\)](#).

3. Co-registration are subsequently performed manually between the microCT images and the histologic slices obtained during the steps described above. The result of this step consists in pairs of data: the micro CT 2D image with its histologic ground truth.
4. An expert annotate the micro CT images using the histologic ground truths in the GIMP software⁷. They are 6 classes:
 - soft tissue (ST): soft tissue, formaldehyde, thrombus, fibrous plaque.
 - fatty tissue (FT): fatty tissue, lipid pool.
 - sheet calcification (SC).
 - nodular calcification (NC).
 - specimen holder (SH).
 - background (Ba)
5. Collection and analysis of *dicom* data (CT images): Centerline information is available for the CT images. The centerline is given by an expert and is used to select the interest region of the images, since the lesion represents a small area of the artery, *i.e.* of the CT image.
6. Correlation between CT scanner and micro CT scanner using standard references (collaterals, branches and specific lesions).

F.2. Previous Work

[Gangloff \(2020\)](#) has proposed different methods to segment the micro CT images of the SAFFP. They used pairs of micro CT images histologically annotated micro-CT images, which constituted the training set. In other words, they mainly used the pairs of information obtained until Step 4 of the protocol described above. The additional information obtained after that step, which is related to the correlation between the CT scanner and the micro CT scanner, were not exploited from a segmentation point of view. The authors have used a CNN based on the U-Net architecture ([Ronneberger et al., 2015](#)) to segment the micro CT images into 6 classes. We describe this technique with more details in Subsection 5.3. The number of classes was selected based on the histopathologists' advice. Notice that their work is a 2D supervised segmentation, since the pairs (micro CT image, ground truth) are only available for some slices of the 3D micro CT image.

The measure of performance is the Dice score, which is a measure of the similarity between two sets of data. In the context of image segmentation,

for example, the Dice score can be used to evaluate the similarity between a predicted segmentation mask and the ground truth segmentation mask. The Dice score is defined as follows:

$$\text{Dice score} = \frac{2|A \cap B|}{|A| + |B|}, \quad (\text{F.1})$$

where A and B are the two sets of data. This score is a number between 0 and 1, where 0 indicates no similarity and 1 indicates perfect similarity.

F.3. Super resolution

F.3.1 Super resolution via VAEs

Super-resolution (SR) techniques, while sharing a common objective of enhancing image resolution, employ a variety of methods to achieve this goal. Super-resolution VAEs architectures have been proposed by ([Gatopoulos et al., 2020](#)), which requires a dataset of high-resolution images and their corresponding low-resolution images. On the one hand, an unsupervised real image de-

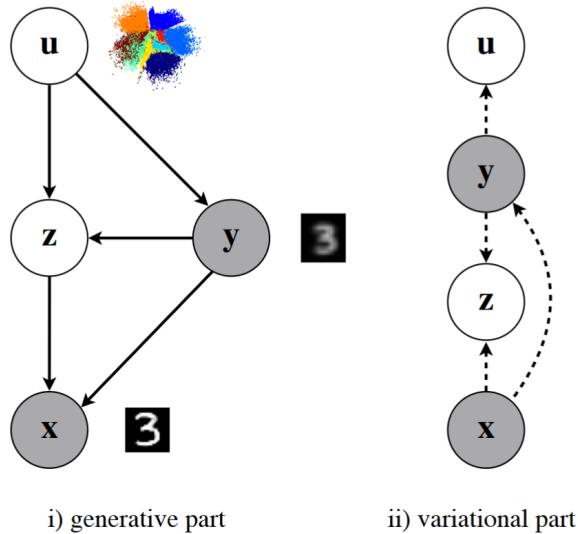


Figure F.1: Stochastic dependencies of the proposed model. Our approach takes advantage of a compressed representation y of the data in the variational part, that is then utilized in the super-resolution in the generative part. Figure taken from ([Gatopoulos et al., 2020](#))

noising and Super-Resolution approach via Variational AutoEncoder (dSR-

VAE) was proposed by (Liu et al., 2020). The architecture of the proposed model is shown in Figure F.2.

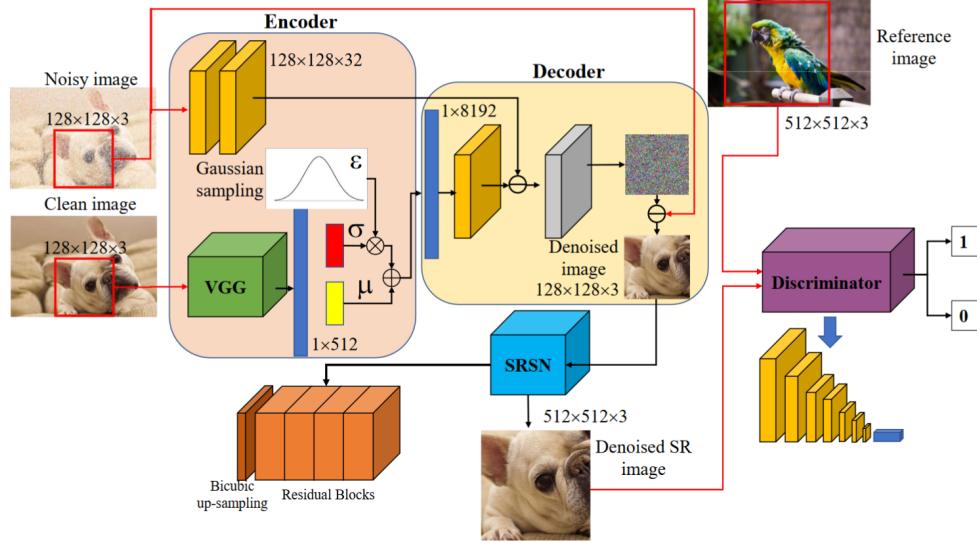


Figure F.2: Complete structure of the proposed dSRVAE model. It includes Denoising AutoEncoder (DAE) and Super-Resolution SubNetwork (SRSN). The discriminator is attached for photo-realistic SR generation. Figure taken from (Liu et al., 2020)

SR models based on VAEs are a branch of image processing focused on generating high-resolution images from low-resolution ones (Appati et al., 2023; Liu et al., 2020; Gatopoulos et al., 2020; Hyun & Heo, 2020). These models have gained popularity due to their effectiveness in modeling high-resolution images, traditionally dominated by autoregressive models (Li & Orchard, 2001; Joshi et al., 2005), and GANs (Chira et al., 2022). Images generated by VAEs present, in general, blurry details, which is a limitation of these models.

Applicability to medical images: Since the LR-CT images we have are small (5×5 to 12×12) pixels and the details are important for the segmentation, the factor of up-scaling is important. In addition, the images are noisy, which is a common problem in medical images. The SR algorithms based on VAEs presented above are promising, however, they are not suitable for our problem due to the up-scaling factor, and the noise in the images. The SR algorithm based on VAEs presented in (Gatopoulos et al., 2020) proposes a factor of up-scaling of 2, which is not enough for our problem. Moreover, the

input for training is an HR (micro CT) center line of the artery, which is not yet available. From a point of view of the applicability to medical images, we won't be able to use this algorithm. Although, it is a promising algorithm for future researches in this field which can be related to the work we have presented in previous chapters.

F.3.2 Laplacian pyramid SR network

The Laplacian Pyramid Super-Resolution Network (LapSRN), presented in (Lai et al., 2017), is a method for single-image super-resolution using CNNs. It progressively reconstructs the sub-band residuals of high-resolution (HR) images without requiring bicubic interpolation, which reduces computational complexity. Figure F.3 shows the LapSRN architecture where we can see the different layers of the network. LapSRN directly extracts feature maps from low-resolution images and progressively predicts sub-band residuals in a coarse-to-fine manner using transposed convolutional layers for upsampling. It is trained end-to-end with deep supervision using a robust Charbonnier loss function, which improves accuracy and reduces visual artifacts. LapSRN stands out for its fast processing speed, accuracy, and ability to generate multi-scale predictions in one feed-forward pass, making it suitable for resource-aware applications.

Remark F.3.1. The Charbonnier loss function is a variant of the L1 loss function, commonly used in image processing and computer vision tasks, particularly for regression problems like image super-resolution. This loss function is defined as follows:

$$\mathcal{L}_{\text{Charbonnier}}(x) = \sqrt{(y_{\text{pred}} - y_{\text{true}})^2 + \epsilon^2}$$

where y_{pred} is the predicted value, y_{true} is the ground truth value, and ϵ is a small constant which ensures numerical stability and prevent division by zero. The inclusion of the ϵ term allows the Charbonnier loss to be less sensitive to outliers than the L_2 loss, while being smoother and less abrupt than the L_1 loss, which can be beneficial in training neural networks for tasks like image super-resolution.

Lai et al. (2017) have compared LapSRN with other classic SR algorithms such as Super Resolution Convolutional Neural Network (SRCNN) (Dong et al., 2015), Fast Super Resolution Convolutional Neural Network (FSR-CNN) (Dong et al., 2016), Very Deep Convolutional Neural Network (VDSR) (Kim et al., 2016), and some other state-of-the-art SR algorithms. They have shown that LapSRN outperforms these algorithms in terms of accuracy and visual quality.

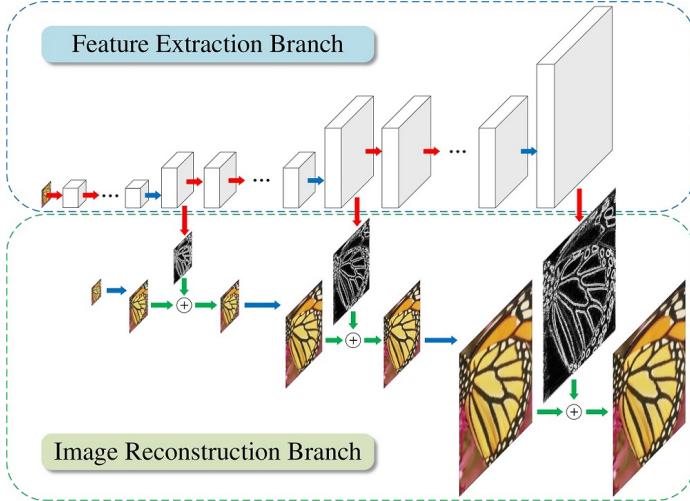


Figure F.3: Red arrows indicate convolutional layers. Blue arrows indicate transposed convolutions (upsampling). Green arrows denote element-wise addition operators, and the orange arrow indicates recurrent layers. Figure taken from ([Lai et al., 2017](#))

F.4. Probabilistic U-Net architecture

The central component of the Probabilistic U-Net is the latent space, which is the key to modeling the ambiguity of the segmentation problem. The latent space is a low-dimensional space where the segmentation variants are represented as probability distributions. A sample from the latent space is drawn and then injected into the U-Net to produce the corresponding segmentation map S , defined as follows:

$$S(x, z) = f_{\text{comb}}(f_{U-\text{Net}}(x), z).$$

Here, $f_{U-\text{Net}}$ is the U-Net architecture and f_{comb} is the function that combines the information obtained from the latent space and the output of the U-Net.

Figure F.4, (a) represents the sampling process, where a sample is drawn from the prior distribution $p(z|x)$. Next, the segmentation map S is obtained. Figure F.4(b) represents the training process, where the model is trained with the standard training procedure for conditional VAEs. The ELBO objective function for the Probabilistic U-Net reads

$$\mathcal{Q}_{\text{P-U-Net}}(x, y) = \mathbb{E}_{q_\phi(z|x,y)} [\log p_\theta(y|S(x, z))] - \beta \text{ KLD} (q_\phi(z|x,y) || p(z|x)) ,$$

where β is a hyperparameter that controls the trade-off between the reconstruction loss and the KLD term (Higgins et al., 2017). The reconstruction loss is the cross-entropy between the segmentation map S and the ground truth y . The KLD term is the Kullback-Leibler divergence between the approximate posterior $q_\phi(z|x, y)$ and the prior $p(z|x)$.

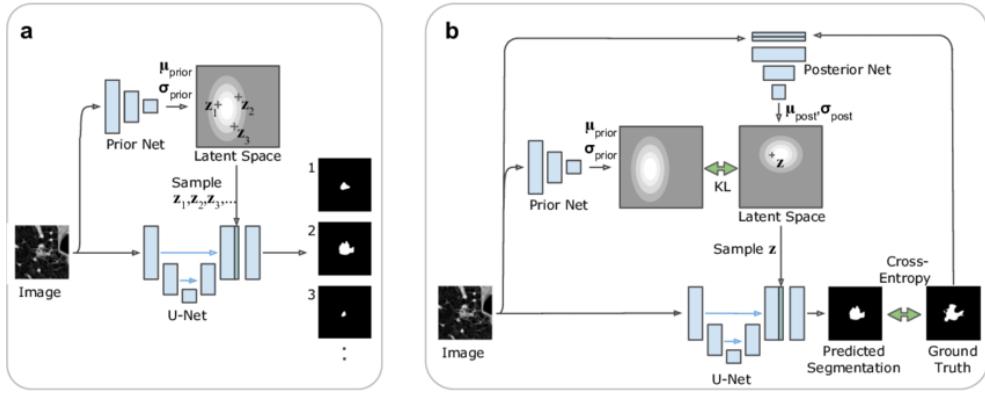


Figure F.4: The Probabilistic U-Net. (a) Sampling process. The heatmap represents the probability distribution in the low-dimensional latent space. (b) Training process illustrated for one training example. Figure taken from (Kohl et al., 2018).

List of Figures

- 1.1 Graphical representation of a Recurrent Neural Network. The recurrent connections between the nodes highlight the network’s ability to process sequences of data by maintaining a ‘memory’ of previous inputs through the hidden states. 15
- 1.2 Illustration of the reparameterization trick. In the original form, we cannot compute the gradient of f w.r.t ϕ . While in the reparameterized form, gradient of f w.r.t ϕ is easily computed. Diamonds indicate no stochasticity, while blue circles highlight its presence. Figure based on (Kingma & Welling, 2014). 19
- 1.3 Illustration of a Gaussian-Variational AutoEncoder model. 20
- 1.4 Illustration of the Gumbel-Max and Gumbel-Softmax tricks with $C = 3$. The blue circle represents the Gumbel samples drawn from $\text{Gumbel}(0, 1)$. The result of the Gumbel-Max trick is the index c of the maximum value and the result of the Gumbel-Softmax trick is a C -dimensional vector z^{G-S} with values in $[0, 1]^C$, which is a continuous, differentiable approximation of the arg max. 21
- 2.1 Conditional dependencies of the HMC, RNN, GUM, and PMC. In the RNN, the hidden states z_t are shown as diamonds to stress that they are no source of stochasticity. The HMC, RNN, and GUM are particular cases of the PMC. 32
- 2.2 Illustration of a deep parameterization of the distribution $p_\theta(z_t|z_{t-1}, x_{t-1})$, where the parameters μ_θ^z and σ_θ^z of the Gaussian distribution are the output of a DNN. 34

2.3 Examples of generated images from estimated $p_\theta(x_{0:t})$ for the MNIST data set with the PMC-II model.	41
3.1 Semi-supervised image segmentation with d-TMC models with 40% of unlabeled pixels.	63
3.2 Semi-supervised image segmentation with d-TMC models with 60% of unlabeled pixels.	64
4.1 Graphical representations of the HMC, SPMC, and PMC models.	67
4.2 DNN architecture with constrained output layer for ψ_θ^y with two hidden layers. $\Sigma = \psi_\theta^y(y_{t-1}, x_{t-1}, y_{t-1}x_{t-1}) = \text{sigm}(\gamma_1 l_1^3 + \gamma_2 l_2^3 + \gamma_3 l_3^3 + \kappa)$, 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 $\kappa = b_{\omega_1}$. The parameters θ_{fr} are related to the output layer which computes the function ψ_θ^y of the linear PMC model (4.10). Due to the one-hot encoding of the discrete r.v. y_{t-1} ($y_{t-1} = \omega_1 \leftrightarrow y_{t-1} = 0$ and $y_{t-1} = \omega_2 \leftrightarrow y_{t-1} = 1$), this parameterization is equivalent to that of (4.10) up to the given correspondence between $\theta_{\text{fr}} = (\gamma_1, \gamma_2, \gamma_3, \kappa)$ and $(a_{\omega_1}, a_{\omega_2}, b_{\omega_1}, b_{\omega_2})$. When the number of classes C increases, the size of the first and last layer increases due to the one-hot encoding of y_{t-1} . Linear activation functions are used in the last hidden layer in red.	72
4.3 Unsupervised image segmentation with PMC models. Figure 4.3a displays averaged results while Figure 4.3b describes a particular classification.	75
4.4 Graphical and condensed representation of the parameterization of ψ_θ^y in the DTMC models. <i>r.t.</i> stands for reparameterization trick. The dashed arrows represent the fact that some variables are copied. For clarity, we do not represent the block ψ_θ^y which is similar to Figure 4.2, up to the introduction of $z_{t-1:t}$	85
4.5 Unsupervised image segmentation with General Triplet Markov Chains (Scenario (3.19)).	88
4.6 Unsupervised image segmentation with General Triplet Markov Chains (Scenario (4.46)).	89

4.7 Illustration of the unsupervised segmentation of slice B, as reported in Table 4.1. The D-MTMC 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.	92
5.1 Peripheral arterial disease results from narrowing or blockage of the arteries of the legs.	96
5.2 Illustration of part of the available data for the study. Figure taken from (Gangloff, 2020).	98
5.3 Notation of the classes of the ground truth 5.2c. Figure taken from (Gangloff, 2020).	98
5.4 Illustration of the available pair of information: 2D micro CT image (light gray rectangle) and its corresponding ground truth (purple rectangle). The pairs of information are only available for some slices of the 3D micro CT image. The red rectangles represent all the 2D slices of the 3D micro CT image. Figure based on (Kuntz et al., 2021)	99
5.5 Illustration of the available correlation between CT scanner [B, B1, B2] and micro CT scanner [C, C1, C2] using standard references after Step 6. They represent different types of calcifications in SAFP plaques. Figure based on (Kuntz et al., 2021).	100
5.6 Our workflow for segmenting sheet and nodular calcifications in CT images of the SAFP is structured into five steps. First, we perform pre-processing of the CT and micro CT images, followed by a super resolution algorithm on the CT images, post-processing of the SR-CT images, supervised segmentation, and segmentation on the SR-CT images.	101
5.7 Example of a CT image of the SAFP and its corresponding micro CT image. In addition, the corresponding Super Resolution CT image after applying the LapSRN algorithm with a factor of up-scaling of 8.	102
5.8 Example of a sequence of CT and SR-CT images. From left to right, the pairs of images (CT, SR-CT). The CT images correspond to a sequence of 2D slices of a 3D CT image, where the calcifications are present. The corresponding Super Resolution CT images are obtained with the LapSRN algorithm with a factor of up-scaling of 8.	103

5.9 U-net architecture (example for 32×32 pixels in the lowest resolution). Each blue box corresponds to a multichannel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations. Figure taken from (Ronneberger et al., 2015)	105
5.10 Example of a segmentation of the micro CT (first row) and its corresponding SR-CT images (second row) with the U-Net and Probabilistic U-Net models.	106
5.11 CT slides, their corresponding SR-CT images, and their corresponding 3 class and 4 class segmentations of SR-CT images.	108
E.1 Graphical and condensed representation of the parameterization of ψ_θ^y in the DPPMC model. The dashed arrows represent the fact that some variables are copied.	129
E.2 Unsupervised image segmentation with Partially Pairwise Markov Chains.	131
E.3 Multi-class segmentations with the HMC-IN, SPMC and DSPMC models. The noisy image is simulated according to Eq. (3.19) from the paper with $a_{\omega_1} = 0, a_{\omega_2} = 1$ and $a_{\omega_3} = 2$ for the top row, $a_{\omega_1} = 0, a_{\omega_2} = 0.5$ and $a_{\omega_3} = 1$ for the middle row and $a_{\omega_1} = 0, a_{\omega_2} = 0.75, a_{\omega_3} = 1.5, a_{\omega_4} = 2.25$ and $a_{\omega_5} = 3$ for the bottom row. Note that the segmentation can be affected by label switching, which is another different problem out of scope of the article.	135
E.4 Error rate from the unsupervised segmentations of Scenario (4.46). Results are averaged on all the <i>dog</i> -type images from the database.	136
F.1 Stochastic dependencies of the proposed model. Our approach takes advantage of a compressed representation y of the data in the variational part, that is then utilized in the super-resolution in the generative part. Figure taken from (Gatopoulos et al., 2020)	139
F.2 Complete structure of the proposed dSRVAE model. It includes Denoising AutoEncoder (DAE) and Super-Resolution SubNetwork (SRSN). The discriminator is attached for photo-realistic SR generation. Figure taken from (Liu et al., 2020)	140

- F.3 Red arrows indicate convolutional layers. Blue arrows indicate transposed convolutions (upsampling). Green arrows denote element-wise addition operators, and the orange arrow indicates recurrent layers. Figure taken from (Lai et al., 2017) 142
- F.4 The Probabilistic U-Net. (a) Sampling process. The heatmap represents the probability distribution in the low-dimensional latent space. (b) Training process illustrated for one training example. Figure taken from (Kohl et al., 2018). 143

List of Tables

2.1	Configuration of the dependencies for different deep generative PMCs. In each model, the sequence of latent variables $\{h_t\}_{t \in \mathbb{N}}$ is treated as a deterministic variable given the observations. As a result, η coincides with the Dirac measure. The distribution λ is typically chosen to be Gaussian, while ζ depends on the nature of the observations. Remember that in a classical RNN, $\{z_t\}_{t \in \mathbb{N}}$ is not considered.	39
2.2	Dimensions of latent variables for each Deep PMC. ψ_θ^h , ψ_θ^z , ψ_θ^x and ψ_ϕ^z are implemented as neural networks with two hidden layers. The number of neurons on each layer coincide with d_h	40
2.3	Averaged ELBO and approximated log-likelihood (approx. LL) of the observations on the test set with two different configurations. For the RNN, the ELBO coincides with the (exact) log-likelihood.	41
2.4	Approximated likelihoods on the polyphonic music data sets. For the RNN, the exact log-likelihood is computed.	42
3.1	Average error rates of the reconstruction of the unobserved pixels on different sets of images with different percentages of unobserved pixels.	63
4.1	Averaged error rates (%) in unsupervised image segmentation with all the generalized TMCs assessed on ten micro-computed tomography slices. The detailed scores are given in Appendix E.	92
4.2	Averaged error rates (%) in the binary clustering of the first twenty raw entries of the HAPT dataset (Reyes-Ortiz et al., 2016). The detailed scores are given in Appendix E.	92

5.1	Dice score on the test set for the U-Net and Probabilistic U-Net models. Three classes are considered: background (Ba), nodular calcifications (NC), and sheet calcifications (SC).	106
5.2	Dice score on the test set for the U-Net and Probabilistic U-Net models, with four classes: background (Ba), soft tissue (ST), nodular calcifications (NC), and sheet calcifications (SC).	107
E.1	Detailed error rates (%) in unsupervised image segmentation with all the generalized TMCs assessed on ten micro-computed tomography slices. See Section 4.4.1.	133
E.2	Detailed Error rates (%) in the binary clustering of the first twenty raw entries of the HAPT dataset (Reyes-Ortiz et al., 2016).	134

Bibliography

- Abdel-Hamid, O., Deng, L., & Yu, D. (2013). Exploring convolutional neural network structures and optimization techniques for speech recognition. In *Interspeech 2013*. ISCA. 6, 14
- Akhiezer, N. (1965). The classical moment problem, hafner publ. Co., New York, (pp. 299–2). 47, 119
- An, J., & Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1), 1–18. 7
- Andriyash, E., Vahdat, A., & Macready, B. (2018). Improved gradient-based optimization over discrete distributions. *preprint arXiv:1810.00116*. 56
- Appati, J. K., Gyamenah, P., Owusu, E., & Yaokumah, W. (2023). Deep residual variational autoencoder for image super-resolution. In *International Conference on Information, Communication and Computing Technology*, (pp. 91–103). Springer. 140
- 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. 70
- Bayer, J., & Osendorfer, C. (2014). Learning stochastic recurrent networks. In *NIPS 2014 Workshop on Advances in Variational Inference*. 37
- Bengio, Y., Boulanger-Lewandowski, N., & Pascanu, R. (2013). Advances in optimizing recurrent networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, (pp. 8624–8628). IEEE. 42
- Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern Recognition and Machine Learning*, vol. 4. Springer. 18

- Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518), 859–877. [7](#), [8](#), [17](#), [22](#), [30](#), [33](#)
- Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., & Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs. [69](#)
- Butepage, J., Kjellstrom, H., & Krägic, D. (2019). Predicting the what and how: A probabilistic semi-supervised approach to multi-task human activity modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, (pp. 0–0). [51](#), [58](#)
- Caron, R., Londono, I., Seoud, L., & Villemure, I. (2023). Segmentation of trabecular bone microdamage in x-ray microct images using a two-step deep learning method. *Journal of the Mechanical Behavior of Biomedical Materials*, 137, 105540. [104](#)
- Chan, W., Jaitly, N., Le, Q., & Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (pp. 4960–4964). IEEE. [6](#), [14](#)
- Chen, M., Tang, Q., Livescu, K., & Gimpel, K. (2018). Variational sequential labelers for semi-supervised learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (pp. 215–226). [51](#), [58](#), [62](#), [125](#)
- Chen, S., & Jiang, X. (2020). Modeling repayment behavior of consumer loan in portfolio across business cycle: A triplet Markov model approach. *Complexity*, 2020. [87](#), [90](#)
- Chira, D., Haralampiev, I., Winther, O., Dittadi, A., & Liévin, V. (2022). Image super-resolution with deep variational autoencoders. In *European Conference on Computer Vision*, (pp. 395–411). Springer. [7](#), [140](#)
- Chopin, N., Papaspiliopoulos, O., et al. (2020). *An introduction to sequential Monte Carlo*, vol. 4. Springer. [25](#)
- Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*. [16](#), [128](#)

- Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., & Bengio, Y. (2015). A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems*, (pp. 2980–2988). [32](#), [37](#), [38](#)
- Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference*, vol. 307 of *ACM International Conference Proceeding Series*, (pp. 160–167). ACM. [6](#), [14](#)
- Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4), 303–314. [6](#), [14](#)
- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1), 1–22. [8](#), [17](#), [18](#), [25](#), [69](#), [117](#)
- Deng, L., Hinton, G., & Kingsbury, B. (2013). New types of deep neural network learning for speech recognition and related applications: An overview. In *2013 IEEE international conference on acoustics, speech and signal processing*, (pp. 8599–8603). IEEE. [6](#), [14](#)
- Derrode, S., & Pieczynski, W. (2004). Signal and image segmentation using pairwise Markov chains. *IEEE Transactions on Signal Processing*, 52(9), 2477–2489. [8](#), [25](#)
- Dong, C., Loy, C. C., He, K., & Tang, X. (2015). Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2), 295–307. [141](#)
- Dong, C., Loy, C. C., & Tang, X. (2016). Accelerating the super-resolution convolutional neural network. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*, (pp. 391–407). Springer. [141](#)
- Douc, R., & Moulines, E. (2012). Asymptotic properties of the maximum likelihood estimation in misspecified hidden Markov models. *Annals of Statistics*, 40(5), 2697–2732. [24](#)
- Douc, R., Moulines, E., & Rydén, T. (2004). Asymptotic properties of the maximum likelihood estimator in autoregressive models with Markov regime. [24](#)

- Doucet, A., De Freitas, N., & Gordon, N. (2001a). An introduction to sequential monte Carlo methods. *Sequential Monte Carlo Methods in Practice*, (pp. 3–14). [22](#)
- Doucet, A., de Freitas, N., & Gordon, N. (2001b). An introduction to Sequential Monte Carlo methods. In *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science, (pp. 3–14). Springer. [25](#), [83](#)
- Doucet, A., & Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656–704), 3. [66](#), [83](#)
- Erhan, D., Courville, A., Bengio, Y., & Vincent, P. (2010). Why does unsupervised pre-training help deep learning? In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS*, vol. 9 of *JMLR Proceedings*, (pp. 201–208). JMLR.org. [73](#)
- Fausett, L. V. (1994). *Fundamentals of neural networks: architectures, algorithms and applications*. Pearson Education India. [14](#)
- Fearnhead, P., Wyncoll, D., & Tawn, J. (2010). A sequential smoothing algorithm with linear computational cost. *Biometrika*, 97(2), 447–464. [83](#)
- Fraccaro, M., Sønderby, S. K., Paquet, U., & Winther, O. (2016). Sequential neural models with stochastic layers. *Advances in Neural Information Processing Systems*, 29. [32](#)
- Fu, J., Zheng, H., & Mei, T. (2017). Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 4438–4446). [6](#), [14](#)
- Gales, M., Young, S., et al. (2008). The application of hidden Markov models in speech recognition. *Foundations and Trends in Signal Processing*, 1(3), 195–304. [8](#)
- Ganeshaaraj, G., Kaushalya, S., Kondarage, A., Karunaratne, A., Jones, J., & Nanayakkara, N. (2022). Semantic segmentation of micro-ct images to analyze bone ingrowth into biodegradable scaffolds. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, (pp. 3830–3833). IEEE. [104](#)

- Gangloff, H. (2020). *Probabilistic Models for Image Processing: Applications in Vascular Surgery*. Ph.D. thesis, Strasbourg. 9, 97, 98, 100, 138, 147
- Gangloff, H., Morales, K., & Petetin, Y. (2021). A general parametrization framework for pairwise Markov models: An application to unsupervised image segmentation. In *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, (pp. 1–6). IEEE. 10
- Gangloff, H., Morales, K., & Petetin, Y. (2022). Chaînes de Markov cachées à bruit généralisé. *Colloque GRETSI 2022*, (pp. 17–20). 10
- Gangloff, H., Morales, K., & Petetin, Y. (2023). Deep parameterizations of pairwise and triplet Markov models for unsupervised classification of sequential data. *Computational Statistics & Data Analysis*, 180, 107663. 10
- Gatopoulos, I., Stol, M., & Tomczak, J. M. (2020). Super-resolution variational auto-encoders. *preprint arXiv:2006.05218*. 139, 140, 148
- Glorot, X., & 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*, vol. 9 of *JMLR Proceedings*, (pp. 249–256). 73
- Goldberg, Y. (2017). Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1), 1–309. 6, 14
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139–144. 7
- Gorynin, I., Gangloff, H., Monfrini, E., & Pieczynski, W. (2018). Assessing the segmentation performance of pairwise and triplet Markov Models. *Signal Processing*, 145, 183–192. 25, 26, 87, 90
- Gumbel, E. J. (1948). *Statistical Theory of Extreme Values and Some Practical Applications: A Series of Lectures*, vol. 33. US Government Printing Office. 20
- Harshvardhan, G., Gourisaria, M. K., Pandey, M., & Rautaray, S. S. (2020). A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review*, 38, 100285. 7
- Hecht-Nielsen, R. (1992). Theory of the backpropagation neural network. In *Neural networks for perception*, (pp. 65–93). Elsevier. 14, 34

- Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., & Lerchner, A. (2017). Beta-VAE: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR*. OpenReview.net. 78, 79, 80, 143
- Hinton, G. E., Deng, L., Yu, D., Dahl, G. E., A-R., M., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., & 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. 73
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. 16, 128
- Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2), 251–257. 6, 14
- Huang, L. (2023). Comparative study of deep learning neural networks for image classification. *Highlights in Science, Engineering and Technology*, 62, 78–83. 6
- Huber, P. J. (1967). The behavior of maximum likelihood estimates under nonstandard condition. In N. LeCam, & J. Neyman (Eds.) *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, (pp. 221–233). Berkeley, CA, USA: University of California Press. 17
- Hyun, S., & Heo, J.-P. (2020). VarSR: Variational super-resolution network for very low resolution images. In *European Conference on Computer Vision*, (pp. 431–447). Springer. 140
- Insull Jr, W. (2009). The pathology of atherosclerosis: Plaque development and plaque responses to medical treatment. *The American Journal of Medicine*, 122(1), S3–S14. 96
- Jaakkola, T. S., & Jordan, M. I. (2000). Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10, 25–37. 7, 8, 17, 30
- Jang, E., Gu, S., & Poole, B. (2017). Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR 2017)*. 18, 21, 56, 60
- Joshi, M. V., Chaudhuri, S., & Panuganti, R. (2005). A learning-based method

- for image super-resolution from zoomed observations. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(3), 527–537. [140](#)
- Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., & Chopin, N. (2015). On particle methods for parameter estimation in state-space models. *Statistical Science*, 30(3), 328–351. [25](#)
- Kim, J., Lee, J. K., & Lee, K. M. (2016). Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 1646–1654). [141](#)
- Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*. [39](#), [62](#)
- Kingma, D. P., Mohamed, S., Rezende, D. J., & 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*, (pp. 3581–3589). [56](#), [78](#), [80](#)
- Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014*. [7](#), [18](#), [19](#), [36](#), [145](#)
- Klys, J., Snell, J., & Zemel, R. (2018). Learning latent subspaces in variational autoencoders. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems*, (pp. 6445–6455). [80](#)
- Kohl, S., Romera-Paredes, B., Meyer, C., De Fauw, J., Ledsam, J. R., Maier-Hein, K., Eslami, S., Jimenez Rezende, D., & Ronneberger, O. (2018). A probabilistic U-net for segmentation of ambiguous images. *Advances in neural information processing systems*, 31. [104](#), [143](#), [149](#)
- Kumar, S., Pradeep, J., & Zaidi, H. (2021). Learning robust latent representations for controllable speech synthesis. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP*, vol. ACL/IJCNLP 2021 of *Findings of ACL*, (pp. 3562–3575). Association for Computational Linguistics. [80](#)
- Kuntz, S. H., Jinnouchi, H., Kutyna, M., Torii, S., Cornelissen, A., Sakamoto,

- A., Sato, Y., Fuller, D. T., Schwein, A., Ohana, M., et al. (2021). Co-registration of peripheral atherosclerotic plaques assessed by conventional ct angiography, microct and histology in patients with chronic limb threatening ischaemia. *European Journal of Vascular and Endovascular Surgery*, 61(1), 146–154. 97, 99, 100, 147
- Kupiec, J. (1992). Robust part-of-speech tagging using a hidden Markov model. *Computer speech & language*, 6(3), 225–242. 8
- Lai, W.-S., Huang, J.-B., Ahuja, N., & Yang, M.-H. (2017). Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 624–632). 102, 141, 142, 149
- Lanchantin, P., Lapuyade-Lahorgue, J., & Pieczynski, W. (2008). Unsupervised segmentation of triplet Markov chains hidden with long-memory noise. *Signal Processing*, 88(5), 1134–1151. 26
- Lanchantin, P., & Pieczynski, W. (2004). Unsupervised non stationary image segmentation using triplet Markov chains. *Advanced Concepts for Intelligent Vision Systems (ACVIS 04)*. 26
- Le Cam, S., Salzenstein, F., & Collet, C. (2008). Fuzzy pairwise Markov chain to segment correlated noisy data. *Signal processing*, 88(10), 2526–2541. 25
- LeCun, Y. (1998). The MNIST database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>. 40
- Li, H. (2018). Deep learning for natural language processing: advantages and challenges. *National Science Review*, 5(1), 24–26. 6
- Li, H., Derrode, S., & Pieczynski, W. (2019). An adaptive and on-line IMU-based locomotion activity classification method using a triplet Markov model. *Neurocomputing*, 362, 94–105. 87, 90
- Li, J., Lee, J.-Y., & Liao, L. (2021a). A new algorithm to train hidden Markov models for biological sequences with partial labels. *BMC bioinformatics*, 22, 1–21. 8
- Li, X., & Orchard, M. T. (2001). New edge-directed interpolation. *IEEE transactions on image processing*, 10(10), 1521–1527. 140
- Li, Y., Sixou, B., & Peyrin, F. (2021b). A review of the deep learning methods for medical images super resolution problems. *Irbm*, 42(2), 120–133. 102

- Liang, S., & Srikant, R. (2016). Why deep neural networks for function approximation? In *International Conference on Learning Representations*. **6**, 14
- Liu, Z.-S., Siu, W.-C., Wang, L.-W., Li, C.-T., & Cani, M.-P. (2020). Unsupervised real image super-resolution via generative variational autoencoder. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, (pp. 442–443). **140**, 148
- Lu, Z., Pu, H., Wang, F., Hu, Z., & Wang, L. (2017). The expressive power of neural networks: a view from the width. *Advances in neural information processing systems*, **30**. **6**, 14
- Maddison, C. J., Mnih, A., & Teh, Y. W. (2017). The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR 2017)*. **18**, 21, 22
- Maddison, C. J., Tarlow, D., & Minka, T. (2014). A* sampling. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, (pp. 3086–3094). **20**
- Medsker, L. R., & Jain, L. (2001). Recurrent neural networks. *Design and Applications*, **5**(64-67), 2. **14**, 30
- Mikolov, T., Joulin, A., Chopra, S., Mathieu, M., & Ranzato, M.-A. (2015). Learning longer memory in recurrent neural networks. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*. **15**, 90
- Mohamed, A., Dahl, G. E., & Hinton, G. (2012). Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech and Language Processing*, **20**(1), 14–22. **73**
- Morales, K., & Petetin, Y. (2021). Variational Bayesian inference for pairwise Markov models. In *2021 IEEE Statistical Signal Processing Workshop (SSP)*, (pp. 251–255). **10**
- Morales, K., & Petetin, Y. (2022). Pairwise Markov chains as generative models. *Colloque GRETSI 2022*, (pp. 649–652). **10**
- Morales, K., & Petetin, Y. (2023). A probabilistic semi-supervised approach with triplet Markov chains. In *2023 IEEE 33rd International Workshop on Machine Learning for Signal Processing (MLSP)*, (pp. 1–6). **10**

- Nassif, A. B., Shahin, I., Attili, I., Azzeh, M., & Shaalan, K. (2019). Speech recognition using deep neural networks: A systematic review. *IEEE access*, 7, 19143–19165. [6](#)
- Pagnoni, A., Liu, K., & Li, S. (2018). Conditional variational autoencoder for neural machine translation. *preprint arXiv:1812.04405*. [59, 125](#)
- 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., & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems*. [69](#)
- Paul, A., Purkayastha, B. S., & Sarkar, S. (2015). Hidden Markov model based part of speech tagging for nepali language. In *2015 international symposium on advanced computing and communication (isacc)*, (pp. 149–156). IEEE. [8](#)
- Pieczynski, W. (2002). Chaines de Markov triplet. *Comptes Rendus de l'Academie des Sciences - Mathematiques*, 335, 275–278. In French. [8, 26](#)
- Pieczynski, W. (2003). Pairwise Markov chains. *IEEE Transactions on pattern analysis and machine intelligence*, 25(5), 634–639. [8, 25, 30, 69, 70](#)
- Pieczynski, W. (2007). Multisensor triplet Markov chains and theory of evidence. *International Journal of Approximate Reasoning*, 45(1), 1–16. [26](#)
- Pieczynski, W., & Desbouvries, F. (2005). On triplet Markov chains. In *International Symposium on Applied Stochastic Models and Data Analysis, (ASMDA)*. [8, 26, 52](#)
- Pinkus, A. (1999). Approximation theory of the MLP model in neural networks. *Acta numerica*, 8, 143–195. [6, 14](#)
- Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A., & Carin, L. (2016). Variational autoencoder for deep learning of images, labels and captions. *Advances in Neural Information Processing Systems*, 29. [7](#)
- Rabiner, L. R. (1989). A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257–286. [8, 30](#)

- Rafieian-Kopaei, M., Setorki, M., Doudi, M., Baradaran, A., & Nasri, H. (2014). Atherosclerosis: Process, indicators, risk factors and new hopes. *International Journal of Preventive Medicine*, 5(8), 927. [96](#)
- Rao, C. R. (1973). *Linear Statistical Inference and Its Applications*, vol. 2. Wiley New York. [118](#)
- Reyes-Ortiz, J.-L., Oneto, L., Samà, A., Parra, X., & Anguita, D. (2016). Transition-aware human activity recognition using smartphones. *Neurocomputing*, 171, 754–767. [91](#), [92](#), [134](#), [151](#), [152](#)
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III* 18, (pp. 234–241). Springer. [104](#), [105](#), [138](#), [148](#)
- Ruder, S. (2016). An overview of gradient descent optimization algorithms. *preprint arXiv:1609.04747*. [14](#)
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536. [71](#), [90](#)
- Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. (1985). Learning internal representations by error propagation. Technical report, California University San Diego La Jolla Institute for Cognitive Science. [14](#), [34](#)
- Sagan, H. (2012). *Space-filling curves*. Springer Science & Business Media. [60](#), [73](#)
- Salaün, A., Petetin, Y., & Desbouvries, F. (2019). Comparing the modeling powers of rnn and hmm. In *2019 18th IEEE International Conference on Machine Learning And Applications (ICMLA)*, (pp. 1496–1499). IEEE. [30](#), [31](#), [32](#), [44](#), [48](#)
- Sherstinsky, A. (2020). Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404, 132306. [16](#)
- Sriram, S. A., Paul, A., Zhu, Y., Sandfort, V., Pickhardt, P. J., & Summers, R. M. (2020). Multilevel unet for pancreas segmentation from non-contrast ct scans through domain adaptation. In *Medical Imaging 2020: Computer-Aided Diagnosis*, vol. 11314, (pp. 116–121). SPIE. [104](#)

- Torii, S., Mustapha, J. A., Narula, J., Mori, H., Saab, F., Jinnouchi, H., Yahagi, K., Sakamoto, A., Romero, M. E., Narula, N., et al. (2019). Histopathologic characterization of peripheral arteries in subjects with abundant risk factors: Correlating imaging with pathology. *JACC: Cardiovascular Imaging*, 12(8 Part 1), 1501–1513. [137](#)
- Traore, B. B., Kamsu-Foguem, B., & Tangara, F. (2018). Deep convolution neural network for image recognition. *Ecological Informatics*, 48, 257–268. [6, 14](#)
- Tzikas, D. G., Likas, A. C., & Galatsanos, N. P. (2008). The variational approximation for Bayesian inference. *IEEE Signal Processing Magazine*, 25(6), 131–146. [18](#)
- White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica*, 50(1), 1–25. [17](#)
- Xu, L., & Jordan, M. I. (1996). On convergence properties of the EM algorithm for Gaussian mixtures. *Neural computation*, 8(1), 129–151. [70](#)
- Xu, W., Sun, H., Deng, C., & Tan, Y. (2017). Variational autoencoder for semi-supervised text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31. [7](#)
- Yoon, B.-J. (2009). Hidden Markov models and their applications in biological sequence analysis. *Current genomics*, 10(6), 402–415. [8](#)
- Zheng, H., Fu, J., Mei, T., & Luo, J. (2017). Learning multi-attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE international conference on computer vision*, (pp. 5209–5217). [6, 14](#)



Titre : Modèles de Markov génératifs pour la classification séquentielle bayésienne

Mots clés : apprentissage approfondi, modèles probabilistes, données séquentielles, chaînes de Markov

Résumé : Cette thèse vise à modéliser des données séquentielles à travers l'utilisation de modèles probabilistes à variables latentes et paramétrés par des architectures de type réseaux de neurones profonds. Notre objectif est de développer des modèles dynamiques capables de capturer des dynamiques temporelles complexes inhérentes aux données séquentielles tout en étant applicables dans des domaines variés tels que la classification, la prédiction et la génération de données pour n'importe quel type de données séquentielles.

Notre approche se concentre sur plusieurs problématiques liés à la modélisation de ce type de données, chacune étant détaillé dans un chapitre de ce manuscrit. Dans un premier temps, nous balayons les principes fondamentaux de l'apprentissage profond et de l'estimation bayésienne. Par la suite, nous nous focalissons sur la modélisation de données séquentielles par des modèles de Markov cachés qui constitueront le socle commun des modèles génératifs développés par la suite. Plus

précisément, notre travail s'intéresse au problème de la classification (bayésienne) séquentielle de séries temporelles dans différents contextes : supervisé (les données observées sont étiquetées) ; semi-supervisé (les données sont partiellement étiquetées) ; et enfin non supervisés (aucune étiquette n'est disponible). Pour cela, la combinaison de réseaux de neurones profonds avec des modèles probabilistes markoviens vise à améliorer le pouvoir génératif des modélisations plus classiques mais pose de nombreux défis du point de vue de l'inférence bayésienne : estimation d'un grand nombre de paramètres, estimation de lois à postériori et interprétabilité de certaines variables cachées (les labels). En plus de proposer une solution pour chacun de ces problèmes, nous nous intéressons également à des approches novatrices pour relever des défis spécifiques en imagerie médicale posés par le Groupe Européen de Recherche sur les Prothèses Appliquées à la Chirurgie Vasculaire (GEPROMED).

Title : Generative Markov models for sequential Bayesian classification

Keywords : deep learning, probabilistic models, sequential data, Markov chains

Abstract : This thesis explores and models sequential data through the application of various probabilistic models with latent variables, complemented by deep neural networks. The motivation for this research is the development of dynamic models that adeptly capture the complex temporal dynamics inherent in sequential data. Designed to be versatile and adaptable, these models aim to be applicable across domains including classification, prediction, and data generation, and adaptable to diverse data types. The research focuses on several key areas, each detailed in its respective chapter. Initially, the fundamental principles of deep learning, and Bayesian estimation are introduced. Sequential data modeling is then explored, emphasizing the Markov chain models, which set the stage for the generative models discussed in subsequent chapters. In particular, the research delves into the sequential Bayesian classi-

fication of data in supervised, semi-supervised, and unsupervised contexts. The integration of deep neural networks with well-established probabilistic models is a key strategic aspect of this research, leveraging the strengths of both approaches to address complex sequential data problems more effectively. This integration leverages the capabilities of deep neural networks to capture complex nonlinear relationships, significantly improving the applicability and performance of the models.

In addition to our contributions, this thesis also proposes novel approaches to address specific challenges posed by the Groupe Européen de Recherche sur les Prothèses Appliquées à la Chirurgie Vasculaire (GEPROMED). These proposed solutions reflect the practical and possible impactful application of this research, demonstrating its potential contribution to the field of vascular surgery.