Probabilistic methods for high dimensional signal processing

Jean-Baptiste Regli

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of
University College London.

Department of Statistical Science
University College London

February 3, 2019

I, Jean-Baptiste Regli, confirm that the work presented in this thesis, except as noted herein, is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

This thesis investigates the use of probabilistic and Bayesian methods for analysing high dimensional signals. The work proceeds in three main parts sharing similar objectives. Throughout we focus on building data efficient inference mechanisms geared toward high dimensional signal processing. This is achieved by using probabilistic models on top of informative data representation operators. We also improve on the fitting objective to make it better suited to our requirements.

Variational Inference

We introduce a variational approximation framework using direct optimisation of what is known as the *scale invariant Alpha-Beta divergence* (sAB divergence). This new objective encompasses most variational objectives that use the Kullback-Leibler, the Rényi or the gamma divergences. It also gives access to objective functions never exploited before in the context of variational inference. This is achieved via two easy to interpret control parameters, which allow for a smooth interpolation over the divergence space while trading-off properties such as mass-covering of a target distribution and robustness to outliers in the data. Furthermore, the sAB variational objective can be optimised directly by re-purposing existing methods for Monte Carlo computation of complex variational objectives, leading to estimates of the divergence instead of variational lower bounds. We show the advantages of this objective on Bayesian models for regression problems.

Roof-Edge hidden Markov Random Field

We propose a method for semi-local Hurst estimation by incorporating a Markov random field model to constrain a wavelet-based pointwise Hurst estimator. This results in an estimator which is able to exploit the spatial regularities of a piecewise parametric varying Hurst parameter. The pointwise estimates are jointly inferred along with the parametric form of the underlying Hurst function which characterises how the Hurst parameter varies deterministically over the spatial support of the data. Unlike recent Hurst regularisation methods, the proposed approach is flexible in that arbitrary parametric forms can be considered and is extensible in as much as the associated gradient descent algorithm can accommodate a broad class of distributional assumptions without any significant modifications. The potential benefits of the approach are illustrated with simulations of various first-order polynomial forms.

Scattering Hidden Markov Tree

We here combine the rich, over-complete signal representation afforded by the scattering transform together with a probabilistic graphical model which captures hierarchical dependencies between coefficients at different layers. The wavelet scattering network result in a high-dimensional representation which is translation invariant and stable to deformations whilst preserving informative content. Such properties are achieved by cascading wavelet transform convolutions with non-linear modulus and averaging operators. The network structure and its distributions are described using a Hidden Markov Tree. This yields a generative model for high-dimensional inference and offers a means to perform various inference tasks such as prediction. Our proposed scattering convolutional hidden Markov tree displays promising results on classification tasks of complex images in the challenging case where the number of training examples is extremely small. We also use variational methods on the aforementioned model and leverage the objective sAB variational objective defined earlier

to improve the quality of the approximation.

Impact Statement

In the last decade the machine learning and signal processing communities have seen game changing improvements and this has caused the development of many applications in both academia and industry. The work presented in that thesis leverage those methods and improve on top of them.

Recent machine learning methods tend to rely on a having access to a vast amount of correctly annotated examples to perform prediction and "extrapolation". This paradigm is not always true and this work focuses on reducing this dependency. We propose methods allowing to perform accurate complex image classification based on only a very limited number of training examples. This type of methods can prove to be useful in domains where collecting examples is costly (medical studies, physic experiments, rare events...). Those methods also heavily relies on correct annotations. We here develop methods to alleviate that need. This types of methods are valuable in situation where a perfect oracle —i.e. person able to produce annotations— does not exist. This is the case for example for medical image analysis, in analysis of spatial imagery. Those two improvements reduce the cost of using machine learning by reducing the need for big highly curated datasets.

Another pitfall of currently used methods is the lack of measure of uncertainty on the prediction made. In this work we develop methods allowing estimation of the quality of the prediction. This information can be leverage in systems where a wrong decision would have high consequences (medical, military...) to trigger more analysis. This uncertainty can also be used by an higher level control/learning algorithm to explore more the training space in the direction of that uncertainty.

Acknowledgements

I would like to express my deepest gratitude to my supervisors Ricardo Silva and James Nelson, UCL Statistical Science department, DSTL, and my family.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

This thesis explores several methods to perform Bayesian inference and analysis on high dimensional signals. More specifically we are interested in extracting knowledge from a small number of realisations of a potentially corrupted signal whilst —ideally— being able to quantify the uncertainty over this inferred information.

Those constraints are motivated by an application to a problem suggested by the Defence Science and Technology Laboratory (Dstl). They are interested in methods to perform underwater mine detection in sonar images. By nature, the access to training examples is limited. And the reliability of the detection has to be quantifiable due to how catastrophic a false negative could be.

Throughout the thesis we explore how Probabilistic Graphical Models (PGMs) and signal representation methods can be combined. And used to perform inference that is both data efficient and might allow to quantify uncertainty.

1.2 Publications

The thesis structure is based on the following published or soon to be published work.

Variational Inference

- J.-B. Regli and R. Silva. "Alpha-Beta Divergence For Variational Inference". In: *arXiv preprint arXiv:1805.01045* (2018)

Roof-Edge hidden Markov Random Field

- J.-B. Regli and J. Nelson. "Piecewise parameterised Markov random fields for semi-local Hurst estimation". In: *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE. 2015, pp. 1626–1630

Scattering Hidden Markov Tree

- J.-B. Regli and J. Nelson. "Scattering convolutional hidden Markov trees". In: *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE. 2016, pp. 1883–1887

Note that, to this day, some chapters are still unpublished.

1.3 Motivation, contribution, and related work

This section provides a brief introduction to the main themes of this thesis, motivates the research objectives, and points out related work.

1.3.1 Alpha Beta Variational Inference

Modern probabilistic machine learning relies on complex models for which the exact computation of the posterior distribution is intractable. This has motivated the need for scalable and flexible approximation methods. Research on this topic belongs mainly to two families, sampling based methods constructed around Markov Chain Monte Carlo (MCMC) approximations [73], or *variational inference* (VI) [49]. In this work, we focus on the latter, although with the aid of Monte Carlo methods.

Contribution. We propose a variational objective to simultaneously trade-off effects of mass-covering, spread and outlier robustness. This is done by developing a variational inference objective using an extended version of the alpha-beta (AB) divergence [107], a family of divergence governed by two parameters and covering many of the divergences already used for VI as special cases. After reviewing some basic concepts of VI and some useful divergences, we extend it to what we will call the scale invariant AB (sAB) divergence and explain the influence of each parameters. We then develop a framework to perform direct optimisation of the divergence measure which can leverage most of the modern methods to ensure scalability of VI. Finally, we demonstrate the interesting properties of the resulting approximation on regression tasks with outliers.

Related work. The quality of the posterior approximation is a core question in variational inference. When using the KL-divergence [2] averaging with respect to the approximate distribution, standard VI methods such as mean-field underestimate the true variance of the target distribution. In this scenario, such behaviour is sometimes known as *mode-seeking* [75]. On the other end, by (approximately) averaging over the target distribution as in Expectation-Propagation, we might assign much mass to low-probability regions [75]. In an effort to smoothly interpolate between such behaviours, some recent contributions have exploited parameterised families of divergences such as the alpha-divergence [75, 112, 157], and the Rényi-divergence [158]. Another fundamental property of an approximation is its *robustness to outliers*. To that end, divergences such as the beta [42] or the gamma-divergences [84] have been developed and widely used in fields such as matrix factorisation [96, 108]. Recently, they have been used to develop a robust pseudo variational inference method [162].

Roof-Edge hidden Markov Random Fields

The Hurst parameter determines the spectral decay rate of a process with a power-law spectrum. Since such a simple relationship is ubiquitous in many signal and image processing areas and beyond [58, 63] Hurst estimation continues to enjoy many, and disparate, applications including Finance [126], signal/image denoising [83], clutter suppression [99], segmentation [122], the analysis of ECG signals [124, 134], internet traffic flow [58], image texture [53], and turbulence data [85].

The interconnection between wavelets and self-similar processes is a powerful, if not, surprising one. The self-similarity explicitly built into the wavelet basis functions via the two-scale, or refinement, relations provides a natural representation in which to study processes that exhibit power-law behaviour. However, the localised nature of wavelets also facilitates a localised estimation of the Hurst parameter.

Contribution. Since it is reasonable to assume that an image of interest may comprise multiple textures, it is appropriate to consider a piecewise smoothly varying Hurst parameter $H = H(\mathbf{r})$, for $\mathbf{r}$ over some subregion of $\mathbb{R}^2$. Furthermore, we let the way in which this Hurst function varies over space be governed by some parametric form $H = \phi(\mathbf{r};\boldsymbol{\theta})$ with model parameters $\boldsymbol{\theta}$. We would expect these parameters to be fairly constant over certain subregions of the image domain where the image texture is homogeneous. We allow the spatial support to accommodate multiple textures with a suitable partitioning of disjoint subregions. In each subregion, the $\boldsymbol{\theta}$ are assumed constant (or have very small, smooth variations). However, between subregion boundaries, it is allowed to change arbitrarily. As a consequence the Hurst parameter itself will vary smoothly inside a partition and vary arbitrarily across the respective subregions. We here propose a model and inference scheme that exploits this piecewise parametric outlook. The framework utilises a Markov random field prior to constrain, or penalise, the magnitude of parameter variation over the image.

Spatial regularisation of Hurst estimation has been recently considered as a means to exploit prior knowledge about the spatial smoothness of the Hurst parameter [140]. However, the method was based on the generalised lasso and assumed only a piecewise constant varying Hurst parameter. In contrast our model, and corresponding gradient-descent-like algorithm, are more flexible. The framework can accommodate many different kinds of distributional assumptions and arbitrary models that describe how the Hurst parameter varies deterministically in space. On the other hand, the generalised lasso Hurst estimator simply penalises the ℓ_1-norm of the Hurst parameter spatial derivatives (of some specified order). Therefore, along with a fixed Gaussian assumption on the data, the spatial derivatives of the Hurst parameter are assumed to be Laplacian and it is difficult to incorporate other distributional assumptions without making wholesale changes to the inference scheme. Other assumptions would necessitate a change in inference strategy (if one existed). Furthermore, unlike the method proposed here, the lasso inference does not obtain any estimate of the underlying parametric form of the Hurst 'function'.

Related work. Although there are works, such as those based on the multifractal formalism [82, 88], that describe how regularity varies across an image, less attention has been paid to the case where the main interest is to obtain pointwise estimates of a Hurst parameter that is allowed to vary as a smooth, deterministic function. Such a scenario could, for example, present itself in image processing when the texture of an object of interest varies gradually over its spatial support in some assumed manner. In turn this would facilitate tasks such as feature extraction, segmentation, and change detection. Likewise, existing adaptive denoising methods, which are currently based on a piecewise constant Hurst parameter [140], could also be extended to include more general Hurst functions that vary as piecewise parametric functions.

[99] leverage the expressiveness of the dual tree complex wavelet transform [76], to perform efficient global Hurst estimation and apply it to ripple suppression for underwater mine detection.

Scattering Hidden Markov Tree

The standard approach to classify high dimensional signals can be expressed as a two step process. First the data are projected in a feature space where the task at hand is simplified. Then prediction is done using a simple predictor in this new representational space. The mapping can either be hand-built —e.g. Fourier transform, wavelet transform— or learned. In the last decade methods for learning the projection have drastically improved under the impulsion of the so called deep learning. Deep neural networks —often enriched by convolutional architecture— have been able to learn very effective representations for a given dataset and a given task [33, 102, 105]. Such methods have achieved state of the art on many standard problems [114, 115] as well as real world applications [150].

However deep learning methods are only efficient when we have access to a vast quantity of training examples [97]. But in some cases, such as in medical or defence applications for example, datapoints are rare or using an expert for hand-labelling them is time-consuming, costly or subjective. Hence in situations where training examples are expensive to collect, learning has to be performed on smaller datasets. In that case using a fixed, hand crafted set of filters seems to be one of the best solution [56]. Recently Mallat [168] introduced the scattering transform— a fixed bank of wavelet filters used to generate data representation in a convolutional neural networks like architecture. This representational approach was used together with a support vector machine classifier (SVM) and achieved close to state of the art performance on a number of standard datasets [94]. Moreover, it has been shown that this method performs very well on a relatively smaller numbers of training examples [130] —i.e. less that 1000 training samples.

Contribution. We propose to model Mallat's scattering convolutional network [94] using hidden Markov trees. This combines a recently proposed deterministic, analytically tractable transformation inspired by deep convolutional networks with a probabilistic graphical model. It creates a potentially powerful probabilistic tool to handle high-dimensional prediction problems. Unlike previous work on hidden Markov wavelet trees, the use of scattering transforms allow us to exploit their full range of invariances. However, it also compels us to adapt the HMT model to non-homogeneous, non-regular trees. In contrast to simply passing the raw scattering coefficients into a classifier, our proposed framework captures dependencies between different layers in a generative probabilistic model. Moreover, unlike standard classification, once trained our model can tackle not only prediction problems but also other inference tasks such as generation, sensitivity analysis, etc and can also outperform SVMs when only a very small number of training examples are available.

Related work. When only a very small number of training samples are available one-shot learning [79] generative classification methods achieve significantly better results than discriminative models [61], however they require pre-training. Generative probabilistic graphical models have been successfully constructed for various wavelet transforms; in particular, Hidden Markov trees have been used to model the dependencies between the wavelet coefficients [43, 57, 71].

Thesis structure

We begin by introducing some necessary concepts and notations in Part I. This includes a brief introduction to signal representation with a focus on elements used in the remainder of the thesis associated to some element more specific to this work in Chapter 2 and probabilistic graphical models in Chapter 3.

Part II – Flexible Variational Inference

Chapter 4 defines a new objective for variational inference. It defines a new objective based on a flexible divergence measure family allowing more complex approximation properties.

Part III – Roof-Edge hidden Markov Random Field estimation

Chapter 5 provides one way of combining signal representation and probabilistic graphical models to perform high dimensional signal analysis. Those tools are used to perform image segmentation based on the local value of the Hurst coefficient.

Part IV – Scattering Hidden Markov Tree

Chapter 6 introduces a novel method to combine scattering transform and probabilistic graphical model to describe a signal. Chapter 7 extends that framework from exact inference to approximate inference using the objective defined in Chapter 4.

Part V – Conclusions

We finally conclude, and discuss potential future work and recent developments related to this work.

Part I

Background

Chapter 2

Signal representation

This chapter introduces the need for signal representation step prior to performing inference. We then review the desired properties of the operator projecting the signal into that representational space. After reviewing some classical representation operators, we introduce the scattering transform, a representation operator that will be used later in the thesis.

Let us consider the problem of inferring the value of a latent variable y given the observed signal $\mathbf{x}^{new}$. Though some of the frameworks detailed in this documents are more general, we will mainly focus on signals such that $\mathbf{x} = \{x[1] \dots x[d]\}$ with $d \approx 10^6$, $x[.] \in \mathbb{R}$ and $y \in \mathbb{N}$. Let us call f the inference function and $\{\mathbf{x}_i, \hat{y}_i = f(\mathbf{x}_i)\}_{i \leq N}$ the N sampled potentially noisy training values. Example of such signals and latent variables are, speech waveforms of different words, digital photographs of different objects but also electrocardiograms of various heart states, sonar images of multiple features and many others.

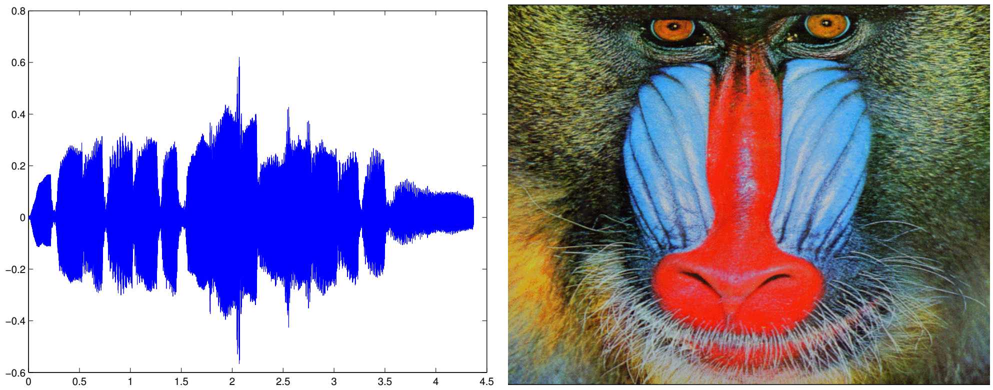

Figure 2.1: Example of high dimensional signals.
Left: Waveform of a flute recording.
Bottom: Image of a Mandrill.

A naive solution to this problem would be to infer the value of the latent variables for a new realisation $\mathbf{x}^{new}$ by looking at its neighbours in the signal space $\mathbb{R}^d$. In essence this approach is similar to the K-Nearest Neighbours (KNN) [26]. Though this type of methods are effective for low-dimensional problems [9], they show limitations in high dimensional cases [48]. Indeed, the number of samples required to find a neighbour within a given distance to a new realisation $\mathbf{x}^{new}$ grows exponentially with the number of dimensions. This issue is known in the statistical learning community as the curse of dimensionality [120] and prevent the use of neighbour based methods directly on high dimensional signals.

One can assume the signal $\mathbf{x}$ to belong to a manifold, say Ω, of $\mathbb{R}^d$. This yields two types of problems. Either the subset Ω is low dimensional or Ω is also high dimensional. In the former, one can mitigate the effect of the curse of dimensionality once the manifold has been isolated. The task at hand is thus a manifold learning problem [86, 119] or a sparse dictionary representation problem [68]. For some signals however, the manifold Ω is also expected to be of high dimensionality. In this case, in order to simplify the inference task, one can only try to reduce the volume of the signal space without losing the crucial information required to characterise it. This can be achieved by reducing the volume of Ω along the invariants in the input signal space [168]. In the remainder of the document we focus on this latter case.

In the remainder of this chapter we focus on designing a mapping, say Φ, which project the signal into a new space where the inference task is simplified. This space should not only capture the main information and discriminatory content in the data but it should also remain stable with respect to appropriate transformations and deformations. Before providing a formal mathematical description of this mapping, it is instructive to consider the following intuitive examples.

2.1 Intuition:

We here provide an intuition of the desired properties of what we will refer as a "good" signal representation. To that end, it is informative to consider the example of image classification. And more specifically to focus on the elements ensuring good generalisation capacities. Using this approach, one can intuit the following properties for the projection Φ:

- The projection has to maintain enough information to permit inference. This means ensuring that Φ preserves separability between the different clusters.

- The mapping also has to be — partially— invariant to translations. Indeed, the information carried by a signal remains the same under limited shifts (see Figure 2.2). This means the transformation Φ has to provide close, if not equal, outputs for shifted versions of the same signal.

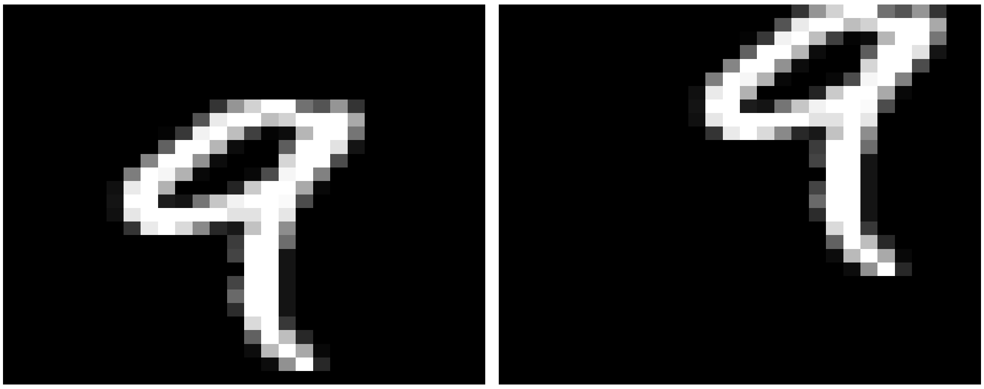

Figure 2.2: The information contained in a signal is invariant to local translations.
Left: Original signal.
Bottom: Translated version of the signal containing the same information.

- To some extent the mapping also has to be stable under deformations. Again, the information contained in a signal remains similar if it has undergone —limited— deformations. Yet if the amplitude of the morphings are too important with regard to the information contained in the signal, then critical features of the signal can be lost (see Figure 2.3). This implies that to a certain degree the projections of morphed realisations of the same signal should be mapped to a same region of the representational space. As opposed to translation, however, one does not want complete invariance to deformation but rather continuous response to it. This is to ensure that the representation created is still informative enough and does not project all inputs to the same region of the space, but instead provide a smooth response to deformations.

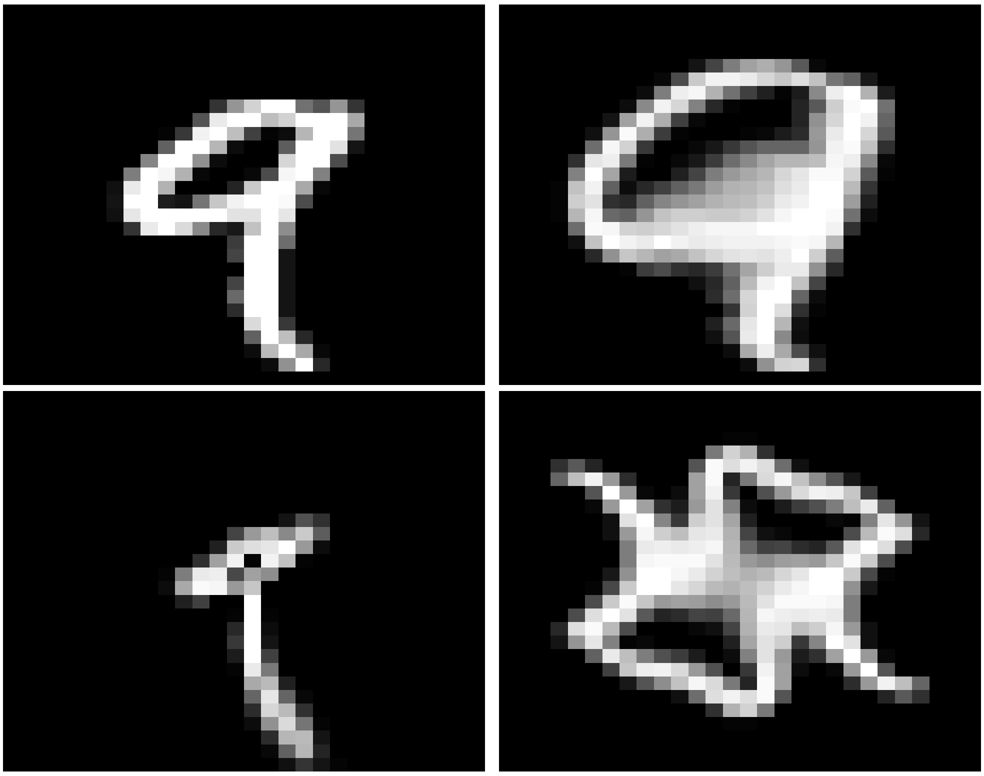

Figure 2.3: The information contained in a signal is stable to deformations limited in amplitude.
Top Left: Original signal.
Top Right: Very lightly deformed version of the signal still containing the same information.
Bottom Left: Lightly deformed version of the signal still containing the same information.
Bottom Right: Highly deformed version of the signal. The information is lost.

- Again to a certain degree, the projection has to be invariant to rotations. Only limited invariance to rotation is wanted because excessive rotation applied to the original signal can be destructive for the information carried (see Figure 2.4). Solutions based on the method described in this document exist [130, 142]. The methods described in Part IV do not incorporate this invariance but could directly be extended to do so.

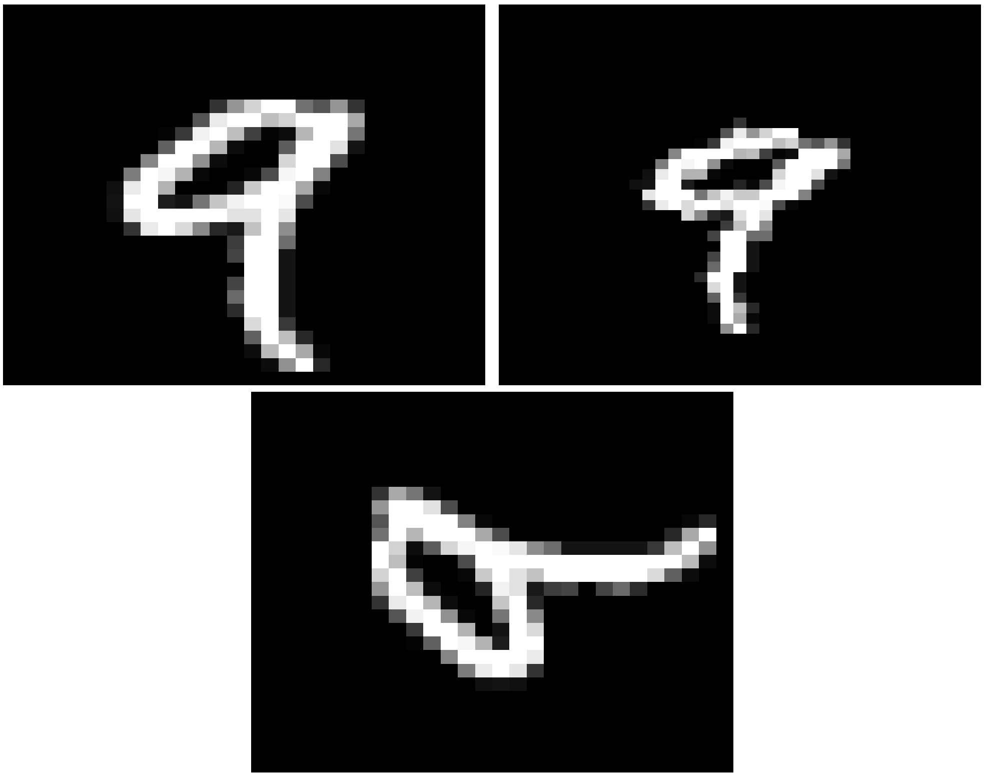

Figure 2.4: The information contained in a signal is to a certain extend invariant to rotation.
Top Left: Original signal.
Top Right: Lightly rotated version of the signal still containing the same information.
Bottom Left: Highly rotated version of the signal. The information is lost.

2.2 Formalisation:

Throughout this section, attention is restricted to signals represented by square-integrable d-dimensional functions over the real numbers, namely $\mathbf{x} \in \mathcal{L}^2(\mathbb{R}^d)$. For sake of simplicity, inference is reduced to categorical variable such that f is now a classification function. Let us define the function g and h such that $f = h \circ g$. The function g denotes the soft classification function, i.e. $g(\mathbf{x}) \in \mathbb{R}^K$ where K is the dimension of the mapping space, and represents the distance to the separating surface. The labelling function h is defined such that $y = h \circ g(\mathbf{x})$ is now the label associated to a signal $\mathbf{x}$.

To be informative enough, a representation must preserve separability between elements of different classes. This is encapsulated by the following definition.

Definition 2.2.1. ***(Separability preservation)***
A representation Φ preserves separability if all elements of two different classes are distant of at least a margin C in the representation space,

$$\forall x, x' \in \mathbb{R}^d \quad \exists C \in \mathbb{R}^K \quad s.t. \quad h \circ g(x) \neq h \circ g(x') \quad \Rightarrow \quad \|\Phi(x) - \Phi(x')\| \geq C^{-1}$$

where K is the dimension of the mapping space.

Translations in the input space should not affect the representation. Let $L_{(.)}$ denote the translation operator for the function in $\mathcal{L}^2(\mathbb{R}^d)$, i.e. for $\mathbf{x} \in \mathcal{L}^2(\mathbb{R}^d)$ and $u, c \in \mathbb{R}^d \times \mathbb{R}^d \quad L_c\mathbf{x}(u) = \mathbf{x}(u - c)$. A mapping Φ is translation invariant —respectively canonical translation— if it projects a translated signal to the same point as its original version.

Definition 2.2.2. ***(Translation invariance)***
Let $\mathcal{H}$ be an Hilbert space, an operator $\Phi : \mathcal{L}^2(\mathbb{R}^d) \to \mathcal{H}$ is translation invariant if:

$$\forall c \in \mathbb{R}^d \quad and \quad \forall \mathbf{x} \in \mathcal{L}^2(\mathbb{R}^d) \quad \Phi(L_c\mathbf{x}) = \Phi(\mathbf{x}).$$

Definition 2.2.3. ***(Canonical translation invariant)***
Let $\mathcal{H}$ be an Hilbert space, an operator $\Phi : \mathcal{L}^2(\mathbb{R}^d) \to \mathcal{H}$ is canonical translation invariant if:

$$\forall \mathbf{x} \in \mathcal{L}^2(\mathbb{R}^d) \quad \Phi(L_a\mathbf{x}) = \Phi(\mathbf{x}) \quad where\ a \in \mathbb{R}^d\ is\ function\ of\ \mathbf{x}.$$

For the standard representation operators, instabilities to deformations are known to appear —especially at high frequencies. To prevent this, one would like the representation to be non-expansive.

Definition 2.2.4. ***(Non-expansive representation)***
A representation Φ is non-expansive if,

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{L}^2(\mathbb{R}^d) \quad \|\Phi(\mathbf{x}) - \Phi(\mathbf{x}')\| \leq \|\mathbf{x} - \mathbf{x}'\|.$$

The local stability to deformations of a non-expansive operator can be expressed as its Lipschitz continuity to the action of deformations close to translations [168]. Such a diffeomorphism can be expressed as a canonical translation,

$$\begin{aligned} L_\tau : \mathcal{L}^2(\mathbb{R}^d) &\to \mathcal{L}^2(\mathbb{R}^d) \\ \mathbf{x} \quad &\to \mathbf{x}((.) - \tau(.)) \end{aligned}$$

where $\tau(u) \in \mathbb{R}^d$ is a displacement field — i.e. a transformation associating a displacement vector to each point of the signal.

Definition 2.2.5. ***(Lipschitz continuous)***
A non expansive operator Φ is said to be Lipschitz continuous to the action of $\mathcal{C}^2$ diffeomorphisms if for any compact $\Omega \subset \mathbb{R}^d$ there exists C such that for all $f \in \mathcal{L}^2(\mathbb{R}^d)$ supported in Ω and all $\tau \in \mathcal{C}^2(\mathbb{R}^d)$,

$$\|\Phi(\mathbf{x}) - \Phi(L_\tau \mathbf{x})\|_H \leq C\|\mathbf{x}\| \left(\sup_{u \in \mathbb{R}^d} |\nabla\tau(u)| + \sup_{u \in \mathbb{R}^d} |H\tau(u)| \right) \tag{2.1}$$

where $\nabla\tau(u)$ is a matrix whose norm $|\nabla\tau(u)|$ measures the deformation amplitude at point u, $H\tau(u)$ is the Hessian matrix of the deformation and its sup-norm $|H\tau(u)|$ measures the smoothness of the deformation.

Hence such a Lipschitz continuous operator Φ is almost invariant to deformations by $\tau(.)$, up to the first and second order deformation terms. Equation 2.1 also implies that Φ is invariant to global translations but this is already enforced by the translation invariance requirement from Definion 2.2.2.

This part and more precisely Section 2.3 introduces an analytically tractable, deterministic wavelet based transformation fulfilling all the properties mentioned in this section.

2.2.1 State of the art:

A common representational method is the modulus of the Fourier transform [21]. To a certain extent, this operator is informative enough to allow discrimination between different types of signals [133]. It is also translation invariant [7]. It is well-known, however, that those operators present instabilities to deformation at high frequencies [14] and thus are not Lipschitz continuous to the action of diffeomorphisms.

Another popular representation method is the wavelet transform [27]. Projection of a signal into the wavelet space also provides a representation suitable for inference [52]. We define the wavelet operator by the following set of convolutions,

$$W\mathbf{x} = \begin{pmatrix} \mathbf{x} * \phi \\ \\ \mathbf{x} * \psi_\lambda \end{pmatrix} \begin{matrix} \rightarrow \text{averaging part} \\ \\ \rightarrow \text{high frequency part} \end{matrix}$$

The averaging part expresses is obtained by convolving the signal with a low frequency filter. By grouping high frequencies into dyadic packet, wavelet operators are stable to —small— deformations [121]. However only the averaging part of a wavelet is invariant to translation. Thus wavelets themselves are known to be non-invariant to translations.

Another popular signal representation method are the convolutional neural networks [33]. As opposed to the two methods mentioned previously, those operators are not fixed but learned from the data [70]. Over the past decade they have provided state of the art results on many standard classification tasks, on image datasets such as *MNIST* [151], *CIFAR* [114] and *ImageNet* [115] as well as on speech processing problems such as *TIMIT* [111]. Those good results are used to advocate that those networks are learning "good" representations. There is, however, no mathematical formalisation of this intuition and it seems that in certain cases they learn representation of the data that are, for example, not invariant to deformations [131, 137].

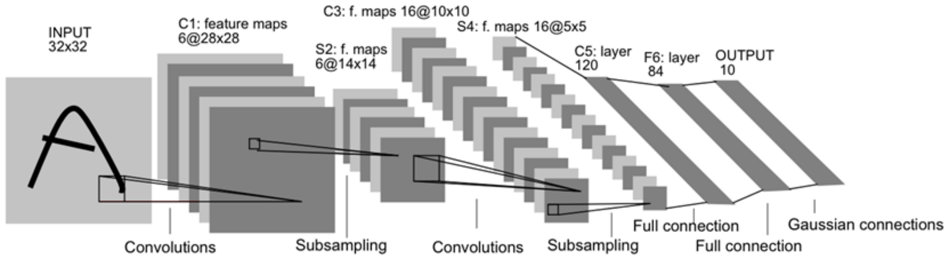

Figure 2.5: Convolutional neural network with 3 convolution/sub-sampling layers and 3 fully connected layers. Image from [45].

2.3 Scattering transform

In this section we describe the construction of a mathematical operator —the scattering transform (ST) [168]— designed to generate what we defined earlier as an interesting representation of signal (see Section 2.2). This operator projects the signal's informative content into scattering decomposition paths, computed by cascading wavelet/modulus operators through an architecture similar to a Convolutional Neural Network (CNN) where the synaptic weights would be given by a wavelet operator instead of learned.

The remainder of this chapter is organised as follows. First, Section 2.3.1 defines the scattering operators. Second, Section 2.3.2 describes how those operators can be stacked to create a Scattering Convolutional Network (SCN), an architecture comparable to a fixed filter CNNs. Then Section 2.3.3 reviews some important properties of the SCNs. And finally, Section 2.3.7 presents how the scattering transform is usually used in classification tasks.

2.3.1 Scattering coefficients:

In this section we focus on the details involved to build an operator fulfilling the properties defined in Section 2.2. As seen in Section 2.2.1, the wavelet wavelet transform already possess some of those properties. Furthermore it can be combined with simple mathematical operators to acquire the missing desired properties.

A two-dimensional directional wavelet is obtained by scaling and rotating a mother wavelet ψ acting as a single band-pass filter. Let G be a discrete, finite rotation group of $\mathbb{R}^2$, multi-scale directional wavelet filters are defined for any scale $j \in \mathbb{Z}$ and rotation $r \in G$ by,

$$\psi_\lambda(u) = \psi_{2^j r}(u) = 2^{2j}\psi(2^j r^{-1} u). \tag{2.2}$$

To simplify the notations, we set $\lambda = \lambda(j,r) \overset{d}{=} 2^j r \in \Lambda \overset{d}{=} G \times \mathbb{Z}$.

A bank of dilated and rotated filters — a wavelet bank of filters— is obtained by simply evaluating Equation 2.2 for different values of $\lambda \in \Lambda$. This bank of filter has no orthogonality properties amongst each other [50]. The wavelet transform projects the signal $\mathbf{x}$ into a representational space using such a bank of filters yielding $\{\mathbf{x} * \psi_\lambda(u)\}_\lambda$. This generates a multiscales and multi-orientations representation of the input signal.

The Morlet wavelet is an example of Wavelet. The mother wavelet ψ_{morlet} is given by,

$$\psi_{morlet}(u) = C_1(e^{iu.\xi} - C_2)e^{\|u\|^2/2\sigma^2},$$

where C_1, ξ and σ are meta-parameters of the wavelet and C_2 is adjusted so that $\int \psi(u)du = 0$. Figure 2.6 shows a Morlet wavelet for $\xi = 3\pi/4$, $\sigma = 0.85$ and $C_1 = 1$. The complete family is obtained by evaluating ψ_{morlet} at different scales and orientations as described in Equation 2.2.

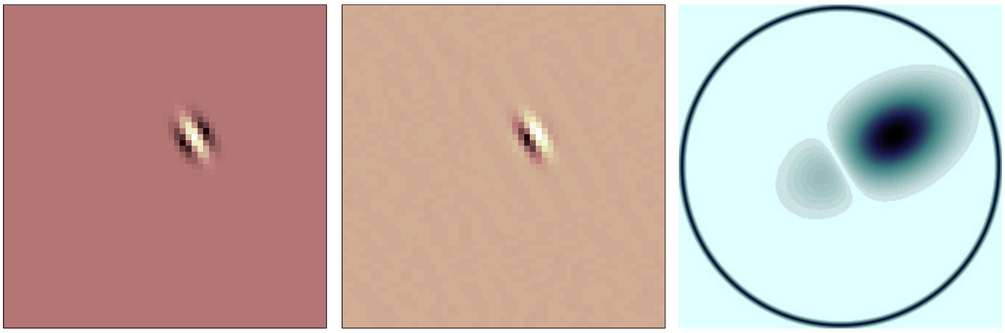

Figure 2.6: Representation of the complex Morlet wavelet for $\xi = 3\pi/4$, $\sigma = 0.85$ and $C_1 = 1$. Redrawn after [113].
Left: Real part of ψ.
Center: Imaginary part of ψ.
Right: Fourier modulus $|\hat{\psi}|$.

As opposed to the Fourier sinusoidal waves, wavelets are operators stable to local $\mathcal{L}^2$ deformations as they can be expressed as localised waveforms [50]. However, as wavelet transform computes a convolution with a wavelet basis, the resulting transform is a translation covariant operator [121].

To make a translation covariant operator translation invariant, one can introduce a non-linearity in the processing pipeline. However one need to make this non linearity such that it does not remove too much of the information contained in the signal. To illustrate that issue, let us consider the wavelet operator as the translation covariant operator and the integration as the non linearity. For any signal $\mathbf{x} \in \mathbb{R}^d$, we get the following trivial invariant,

$$\int \mathbf{x} * \psi_\lambda(u)du = 0.$$

This is because by definition we have, $\int \psi_\lambda(u)du = 0$. This example illustrate the fact that we need to be careful when selecting the non-linearity to introduce in our processing pipeline to avoid removing critical information content from the original signal.

Because the integral of a wavelet operator is known to generate powerful descriptors [72], we want to use an integral based non-linearity. To do so while preserving the informative character of the scattering operator, one has to ensure a non-vanishing integral. A second operator M has to be introduced such that $\int M \circ R(\mathbf{x}) = \int M(\mathbf{x} * \psi_\lambda) \neq 0$. If M was a linear transformation commuting with translation then the integral would still vanish. Hence one has to choose M among the non-linear operator family.

We also want the scattering transform to be stable to deformations. This means we want to define M such that it commutes with deformations,

$$\forall \tau(u)\,,\, ML_\tau = L_\tau M.$$

Adding a weak differentiability condition, Bruna [113] prove that M must necessarily be a point-wise operator — i.e. $M \circ R(\mathbf{x}(u))$ only depends on the value of $\mathbf{x}(u)$.

Finally, by adding the $\mathcal{L}^2(\mathbb{R}^2)$ stability constraint,

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{L}^2(\mathbb{R}^2), \ \|M \circ R(\mathbf{x})\| = \|\mathbf{x}\| \ \text{ and } \ \|M \circ R(\mathbf{x}) - M \circ R(\mathbf{x}')\| \leq \|\mathbf{x} - \mathbf{x}'\|,$$

Bruna [113] shows that necessarily,

$$M(R(\mathbf{x})) = e^{i\alpha}|R(\mathbf{x})|. \tag{2.3}$$

The scattering transform is defined using Equation 2.3 under its simplest form. That is $\alpha = 0$, where it reduces down to the $\mathcal{L}^1(\mathbb{R}^2)$ norms,

$$\|\mathbf{x} * \psi_\lambda\|_1 = \int |\mathbf{x} * \psi_\lambda| du$$

The family of the $\mathcal{L}^1(\mathbb{R}^2)$ normalised wavelets $\{\|\mathbf{x} * \psi_\lambda\|_1\}_\lambda$ generates a crude signal representation which measures the sparsity of the wavelet coefficients.

We have now defined an operator that is both translation invariant and stable to deformations. We need, however to make sure it is expressive and does not discard critical information from the original signal. First, it can be proven that the signal $\mathbf{x}$ can be reconstructed from $\{|\mathbf{x} * \psi_\lambda(u)|\}_\lambda$ up to a multiplicative constant [154]. As a direct consequence, we can say that the information loss in $\{\|\mathbf{x} * \psi_\lambda\|_1\}_\lambda$ occurs during the integration of the absolute value $|\mathbf{x} * \psi_\lambda(u)|$. This integration does indeed removes all non-zero frequencies. However those components can be recovered by calculating the wavelet coefficients $|\mathbf{x} * \psi_{\lambda_1}| * \psi_{\lambda_2}(u)$ of the new signal $|\mathbf{x} * \psi_{\lambda_1}|$. By doing so their $\mathcal{L}^1(\mathbb{R}^2)$ norms define a much larger family of invariants,

$$\forall(\lambda_1, \lambda_2) \in (G \times \mathbb{Z}) \times (G \times \mathbb{Z}) \ \ \||\mathbf{x} * \psi_{\lambda_1}| * \psi_{\lambda_2}\|_1 = \int ||\mathbf{x} * \psi_{\lambda_1}(u)| * \psi_{\lambda_2}| du.$$

One can generate more coefficients with wider translation invariance by further iterating on the "wavelet/modulus" operators. The building block of such a model —the scattering propagator— is thus the absolute value of the convolution between a wavelet and the input signal.

Definition 2.3.1. ***(Scattering propagator)***
The scattering operator U for a given scale and orientation $\lambda \in (G \times \mathbb{Z})$ is defined as the absolute value of the input convoluted with the wavelet operator at this scale and orientation.

$$U[\lambda](\mathbf{x}) \overset{d}{=} |\mathbf{x} * \psi_\lambda|.$$

Definition 2.3.2. ***(Path ordered scattering propagators)***
Let $\forall i \in [\![1,m]\!]$, $\lambda_i \in G \times \mathbb{Z}$. The sequence $p = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ defines a path of length m — i.e. the ordered product of non-linear and non-commuting operators. The p-ordered scattering propagator is defined as,

$$\begin{aligned} \mathcal{U}[p]\mathbf{x} &\overset{d}{=} U[\lambda_m] \ldots U[\lambda_2] U[\lambda_1](\mathbf{x}) \\ &= ||||\mathbf{x} * \psi_{\lambda_1}| * \psi_{\lambda_2}| \ldots | * \psi_{\lambda_m}|. \end{aligned} \tag{2.4}$$

With the convention: $\mathcal{U}[\varnothing]\mathbf{x} = \mathbf{x}$.

We can use the propagators defined in Equation 2.4, to provide a first formal definition of the scattering coefficients.

Definition 2.3.3. ***(Scattering coefficient)***
The scattering coefficient along the path p is defined as the integral of the p-ordered scattering propagator, normalised by the response to a Dirac:

$$\bar{S}[p](\mathbf{x}) \overset{d}{=} \mu_p^{-1} \int \mathcal{U}[p]\mathbf{x}(u) du,$$

with,

$$\mu_p \overset{d}{=} \int \mathcal{U}[p]\delta(u)du.$$

Section 2.3.3 shows that each scattering coefficient $\bar{S}[p](\mathbf{x})$ has the properties listed in Section 2.2. It is invariant to translation of the input signal $\mathbf{x}$, Lipschitz continuous to deformations and yet still informative.

For inference tasks, however, one might want to compute localised descriptors only invariant to translations smaller than a predefined scale 2^J, while keeping the spatial variability at larger scales. This can be achieved by localising the scattering integral with a scaled spatial window $\phi_{2^J}(u) = 2^{-2J}\phi(2^{-2J}u)$. We thus define the windowed scattering transform.

Definition 2.3.4. ***(-Windowed- scattering coefficient of order*** m***)***
If p is a path of length $m \in \mathbb{N}$, the —windowed— scattering coefficient of order m localised at scale 2^J ($J \in \mathbb{N}$) is defined as:

$$\begin{aligned}
S_J[p](\mathbf{x}) &\overset{d}{=} \mathcal{U}[p]\mathbf{x} * \phi_{2^J}(u) \\
&= \int \mathcal{U}[p]\mathbf{x}(v)\phi_{2^J}(u-v)dv \\
&= ||||\mathbf{x} * \psi_{\lambda_1}| * \psi_{\lambda_2}| \dots | * \psi_{\lambda_m}| * \phi_{2^J}(u),
\end{aligned}$$

With the convention: $S_J[\emptyset]\mathbf{x} = \mathbf{x} * \phi_{2^J}$.

So to get invariance up to a given scale $J \in \mathbb{N}^*$, let us define $\mathcal{U}_J[P] \overset{d}{=} \{\mathcal{U}_J[p]\}_{p \in P}$ and $S_J[P] \overset{d}{=} \{S_J[p]\}_{p \in P}$. They respectively define a family of scattering propagators and a family of scattering coefficients indexed by a set of paths P.

Note that in the remainder of this document, we will use, by default, the windowed SC operator, unless stated otherwise. For the sake of reducing the notation clutter we will refer to simply as the scattering operator.

2.3.2 Scattering Convolution Network

This section introduces the scattering convolution network. We choose here to present it as an iterative process over a one-step operator. Similarly to the convolutional neural networks [98], the scattering network is built upon a building bloc comprised of a filtering followed by a non linearity.

Let us recursively build the scattering network. The first layer gathers all the coefficients of order 0. This is $S_J[\varnothing]\mathbf{x} = \mathbf{x} * \phi_{2^J}$. The m-th layer of the scattering network is build by taking all the possible scattering coefficients of order $m-1$. This is $S_J[P_{m-1}]$ where P_{m-1} is the set of all the path of length exactly $m-1$. To construct that layer from the previous ones, it is interesting to notice that for any given path p and an orientation-scale pair λ, we have $U[\lambda]\mathcal{U}[p] = \mathcal{U}[p+\lambda]$. We also remind that $S_J[p](\mathbf{x}) \overset{d}{=} \mathcal{U}[p]\mathbf{x} * \phi_{2^J}(u)$. So we can iteratively compute the nodes of the scattering network by first recursively computing the scattering propagators for all length m up to the pre-determined maximum depth of the network M. Then all the nodes value can be extracted by localising the scattering propagators with a scaled spatial window. Figure 2.7 provides a graphical representation of the scattering network.

In the end, the scattering network can be constructed by iteratively applying a bank of filters and non linearity to an input signal. Thus creating an an architecture similar to a deep convolution network [98]. It has, however, some particularities. Standard CNNs project the signal by applying a succession of convolution/pooling steps and extract the features used for inference at the final layer. The scattering network however outputs features at each layers (see Figure 2.7). Also, while convolutional neural networks use kernel filters learned from the data with back-propagation algorithm, SCNs use a fixed wavelet filter bank.

It is interesting to take a closer look to the second layer of the SCN. The coefficients of the second layer are defined as $\{|\mathbf{x} * \psi_\lambda| * \phi_{2^J}(u)\}$. Using the DAISY approximation [104], one can recognise the SIFT coefficients [72], $S_J[2^j r] = |\mathbf{x} * \psi_{2^j r}| * \phi_{2^J}(u)$, where $\psi_{2^j r}$ is the partial derivative of a Gaussian computed at the finest image scale 2^j and for 8 different rotations r. The averaging filter ϕ_{2^J} is a scaled Gaussian. So the second layer of the SCN is equivalent to the SIFT filters. The difference with them is in the fact that the information pipeline is iterated over to create more complex features.

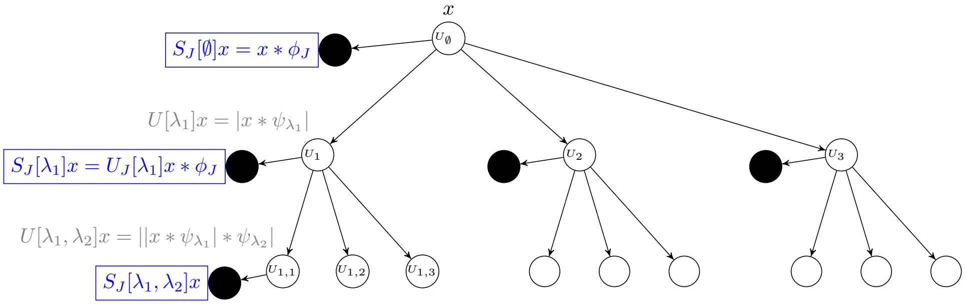
x
$U_\emptyset$
$S_J[\emptyset]x = x * \phi_J$
$U[\lambda_1]x = |x * \psi_{\lambda_1}|$
U_1
U_2
U_3
$S_J[\lambda_1]x = U_J[\lambda_1]x * \phi_J$
$U[\lambda_1, \lambda_2]x = ||x * \psi_{\lambda_1}| * \psi_{\lambda_2}|$
$U_{1,1}$
$U_{1,2}$
$U_{1,3}$
$S_J[\lambda_1, \lambda_2]x$

Figure 2.7: A scattering propagator U_J applied to a signal $\mathbf{x}$ computes each $U[\lambda_i]\mathbf{x} = |\mathbf{x} * \psi_{\lambda_i}|$ and outputs $S[\emptyset]\mathbf{x} = \mathbf{x} * \phi_{2^J}$. Applying U_J to each $U[\lambda_i]\mathbf{x}$ computes all $U[\lambda_i, \lambda_j]\mathbf{x}$ and outputs $S_j[\lambda_i] = U[\lambda_i] * \phi_{2^J}$. Applying iteratively U_J to each $U[p]\mathbf{x}$ outputs $S_J[p]\mathbf{x} = U[p]\mathbf{x} * \phi_{2^J}$ and computes the next path layer.

2.3.3 Properties of the scattering transform:

The scattering coefficient having been defined, one can be interested in the characteristics of such a data representation. This section provides an overview of some of the properties of the scattering transform. It also introduces an approximation to the scattering convolution network defined in the previous section, leading to computationally tractable networks.

Note. Formal proofs for most of those properties can be found in [168].

2.3.4 Non-expansivity:

The path ordered scattering propagator $\mathcal{U}_J[p]\mathbf{x}$ results of the composition of an unitary wavelet transform W_J with a non-expansive modulus operator —as $\forall(a,b) \in \mathbb{C}^2 ||a| - |b|| \leq |a - b|$— and is thus also non-expansive. Since the scattering transform $S_J[\mathcal{P}_J]$ iterates on $\mathcal{U}_J$, one can prove that $S_J[\mathcal{P}_J]$ is also non-expansive (proof adapted from [46]).

Proposition 1. ***(Non-expansive)***
The scattering transform is non expansive.

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{L}^2(\mathbb{R}^d) \quad \|S_J[\mathcal{P}_J]\mathbf{x} - S_J[\mathcal{P}_J]\mathbf{x}'\| \leq \|\mathbf{x} - \mathbf{x}'\|$$

2.3.5 Energy preservation:

Each scattering propagator $U[\lambda]\mathbf{x} = |\mathbf{x} * \psi_\lambda|$ captures the frequency energy contained in the signal $\mathbf{x}$ over a frequency band covered by the Fourier transform $\hat{\psi}_\lambda$ and propagates this energy towards lower frequencies. It can thus be proved that under some assumptions on the wavelet —admissible wavelets—, the whole scattering energy ultimately reaches the minimum frequency 2^{-J} and is trapped by the low-pass filter ϕ_{2^J}. Thus the energy propagated by a —windowed— scattering transform goes to 0 as the path length increases, implying that $\|S_J[\mathcal{P}_J]\| = \|x\|$

But prior to showing this, one must states the necessary assumptions to be made on the wavelet used.

Note. The notation $\hat{(.)}$ is used to design the Fourier transform.

Definition 2.3.5. ***(Admissible scattering wavelet)***
A scattering wavelet ψ is admissible if there exist $\eta \in \mathbb{R}^d$ and $\rho \in \mathcal{L}^2(\mathbb{R}^d)$ positive,

with $|\hat{\rho}(\omega)| \leq |\hat{\phi}(2\omega)|$ *and* $\hat{\rho}(\omega) = 0$*, such that the function,*

$$\hat{\Upsilon}(\omega) = |\hat{\rho}(\omega - \eta)|^2 - \sum_{k=1}^{+\infty} k(1 - |\hat{\rho}(2^{-k}(\omega - \eta)|^2),$$

satisfies,

$$\alpha = \inf_{1 \leq |\omega| \leq 2} \sum_{j=-\infty}^{+\infty} \sum_{r \in G} \hat{\Upsilon}(2^{-j} r^{-1} \omega) |\hat{\psi}(2^{-j} r^{-1} \omega)|^2 > 0.$$

For an admissible wavelet one can prove that the scattering transform conserves the energy of the signal.

Theorem 2.3.6. ***(Energy conservation)***
If the scattering wavelet ψ *is admissible, then for all signal* $\mathbf{x} \in \mathcal{L}^2(\mathbb{R}^d)$*,*

$$\lim_{m \to +\infty} \|\mathcal{U}[\Lambda_J^m]\mathbf{x}\|^2 = \lim_{m \to +\infty} \sum_{n=m}^{+\infty} \|S_J[\Lambda_J^n]\mathbf{x}\|^2 = 0,$$

and

$$\|S_J[P_J]\mathbf{x}\|^2 = \|\mathbf{x}\|^2.$$

The proof of the Theorem 2.3.6 also shows that the scattering energy propagates progressively towards lower frequencies and that the energy of $\mathcal{U}[p]\mathbf{x}$ is mainly concentrated along frequency decreasing paths $p = (\lambda_k)_{k \leq m}$, i.e. for which $|\lambda_{k+1}| \leq |\lambda_k|$. The energy contained in the other paths is negligible and thus for the applications in this document only frequency decreasing paths are considered.

Moreover, the decay of $\sum_{n=m}^{+\infty} \|S_J[\Lambda_J^n]x\|^2$ implies that there exist a path length $M > 0$ after which all longer paths can be neglected. For signal processing applications, this decay appears to be exponential. And for classification applications, paths of length $M = 3$ provides the most interesting results [94, 106].

The restrictions stated above yield an easier parameterisation of a scattering network. Indeed when only the frequency decreasing paths up to a given order are considered, a scattering network is completely defined by:

- ψ: The admissible wavelet used. In the remainder of the document, unless stated otherwise, the Morlet wavelet is used.
- M: The maximum path length considered.
- J: The finest scale level considered.
- L: The number of orientation considered, which can be defined as the cardinality of the previously define ensemble G.

Hence for a given set of parameter (ψ, M, J, L), one can generate one and only one frequency decreasing paths scattering network. Let $ST_{(\psi,M,J,L)}(\mathbf{x})$ now denotes the frequency decreasing windowed scattering convolutional network of parameter (ψ, M, J, L) evaluated for signal $\mathbf{x}$. Each node i of this network generates a -possibly empty- set of of nodes of size $(j_i - 1) \times L$ where j_i is the scale of node i and L is the number of orientations considered. Finally the number of nodes O of this network is,

$$O = \sum_{m=0}^{M-1} \binom{J}{m} L^m \tag{2.5}$$

and it has the architecture displayed by Figure 2.8.

Translation invariance:

The translation invariance of the scattering transform $S_J[\mathcal{P}_J]$ can be proved for a limit metric when J goes to infinity. To do so one can first prove

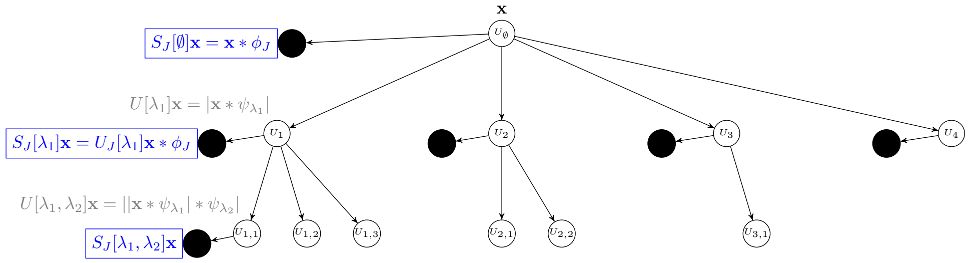
$\mathbf{x}$
$v_\emptyset$
$S_J[\emptyset]\mathbf{x} = \mathbf{x} * \phi_J$
$U[\lambda_1]\mathbf{x} = |\mathbf{x} * \psi_{\lambda_1}|$
$S_J[\lambda_1]\mathbf{x} = U_J[\lambda_1]\mathbf{x} * \phi_J$
v_1
v_2
v_3
v_4
$U[\lambda_1, \lambda_2]\mathbf{x} = ||\mathbf{x} * \psi_{\lambda_1}| * \psi_{\lambda_2}|$
$S_J[\lambda_1, \lambda_2]\mathbf{x}$
$v_{1,1}$
$v_{1,2}$
$v_{1,3}$
$v_{2,1}$
$v_{2,2}$
$v_{3,1}$

Figure 2.8: Frequency decreasing scattering convolution network with $J = 4$ scales, $L = 1$ orientation and $M = 2$ layers. A node i at scale j_i generates $(j_i - 1) \times L$ nodes.

that the scattering distance $\|S_J[\mathcal{P}_J]\mathbf{x} - S_J[\mathcal{P}_J]\mathbf{x}'\|$ converges when J goes to infinity — as it is non-increasing when J increases (see Section 2.3.4). From there one can bound the distance between the scattering transform of the signal and the one of its translated version $\|S_J[\mathcal{P}_J]\mathcal{L}_c\mathbf{x} - S_J[\mathcal{P}_J]\mathbf{x}\|$ and prove that this bound tends to 0 when J goes to infinity. This proves the translation invariance.

Theorem 2.3.7. ***(Translation invariance)***
For admissible scattering wavelets,

$$\forall \mathbf{x} \in \mathcal{L}^2(\mathbb{R}^d), \forall c \in \mathbb{R}^d \quad \lim_{J\to\infty} \|S_J[\mathcal{P}_J]\mathcal{L}_c\mathbf{x} - S_J[\mathcal{P}_J]\mathbf{x}\| = 0$$

Lipschitz continuity to the action of diffeomorphisms:

The Lipschitz continuity to the action of diffeomorphisms of $\mathbb{R}^d$ can be proved for deformations sufficiently close to translations. Such diffeomorphisms map u to $u - \tau(u)$ where $\tau(u)$ is a displacement field such that $\|\nabla\tau\|_\infty < 1$ —i.e. invertible transformations [121]. Let $L_\tau\mathbf{x}(u) = \mathbf{x}(u - \tau(u))$ denotes the action of such diffeomorphisms on the signal $\mathbf{x}$. Once again, one can find an upper bound to the distance between the scattering transform of the signal and the one of its deformed version $\|S_J[\mathcal{P}_J]\mathcal{L}_\tau\mathbf{x} - S_J[\mathcal{P}_J]\mathbf{x}\|$. With a bit of work on this bound, one can then proved that the consequences of the action of L_τ is bounded by a translation term proportional to $2^{-J}\|\tau\|_\infty$ and a deformation error proportional to $\|\nabla\tau\|_\infty$. Finally some more work on the bounding term provides the Lipschitz continuity.

Theorem 2.3.8. ***(Lipschitz continuity to the action of diffeomorphisms)*** *There exists C such that all $\mathbf{x} \in \mathcal{L}(\mathbb{R}^d)$ with $\|U[\mathcal{P}_J]\mathbf{x}\|_1 < \infty$ and all $\tau \in \mathcal{C}^2(\mathbb{R}^d)$ with $\|\nabla\tau\|_\infty < \frac{1}{2}$ satisfy,*

$$\|S_J[\mathcal{P}_J]\mathcal{L}_\tau\mathbf{x} - S_J[\mathcal{P}_J]\mathbf{x} + \tau.\nabla S_J[\mathcal{P}_J]\mathbf{x}\| \leq C\|U[\mathcal{P}_J]\mathbf{x}\|_1 K(\tau), \tag{2.6}$$

with

$$K(\tau) = 2^{-2J}\|\tau\|_\infty^2 + \|\nabla\tau\|_\infty\left(\max\left(\log\frac{\|\Delta\tau\|_\infty}{\|\nabla\tau\|_\infty}, 1\right)\right) + \|H\tau\|_\infty.$$

Remark. If the case where $2^J \gg \|\tau\|_\infty$ and $\|\nabla\tau\|_\infty + \|H\tau\|_\infty \ll 1$, then $K(\tau)$ becomes negligible and the displacement field $\tau(u)$ can be estimated at each $u \in \mathbb{R}^d$. This can be done by solving the linear equation resulting from Equation 2.6 under the assumptions mentioned above,

$$\forall p \in \mathcal{P}_J \|S_J[p]\mathcal{L}_\tau\mathbf{x} - S_J[p]\mathbf{x} + \tau.\nabla S_J[p]\mathbf{x}\| \approx 0.$$

This estimate of the displacement field can be used for many applications such as object tracking in video sequences or image sequence restoration [35].

2.3.6 Extensions

In the previous sections we have introduced a signal projection operator with local translation invariance, stability to deformations and yet still expressive. Building upon that basic architecture of wavelet filter followed by a non linearity, extensions of the scattering transform with extra properties have been developed. The results in Chapter Chapter 6 and

Chapter Chapter 7 can be, with little extra work, extended to those more complex transforms.

Sifre and Mallat [130] develop an extension to the scattering network offering partial rotation invariance to the projection operator. This extra invariance offers more robustness for natural image classification. Oyallon and Mallat [142] further improve and develop that concept.

Singh and Kingsbury [164] develop the scatternet. They follow the same general architecture of wavelet transform/non linearity, but use a parametric log transformation with Dual-Tree complex wavelets (DTCW) [76]. Leveraging the invertibility of the DTCW, they design a invertible projection.

2.3.7 Application to classification:

The scattering transform maps a given realisation of a high-dimensional signal into an even higher-dimensional space where the classification task is simplified due to the inherent properties described in the previous section yielding easily separable data clusters in the "scattering" space.

The scattering transform has been successfully applied in classification of a wild variety of signals such as audio signals [106], images [142] or electrocardiograms [135] and in the vast majority —if not all— the classification task has been done using the features generated by the transform of the dataset as inputs for a discriminative classifier, e.g. Support Vector Machine classifier. The new input vector is obtained by concatenating the scattering coefficients of all orders, scale and orientations into a unique 1-D vector -for 2-D signal the scattering coefficients are also flattened. Leveraging the richness of the representation generated the scattering transform combined to an SVM classifier provides performance comparable to those of a -small- deep convolutional neural network [129]. This section proposes to test this framework on the handwritten digit dataset *MNIST* [151].

MNIST is composed of 28×28 binary and centered images of handwritten digits. The dataset is split into a training set of 50000 images and a testing set of 10000 images and the task at hand is a 10 classes classification problem.

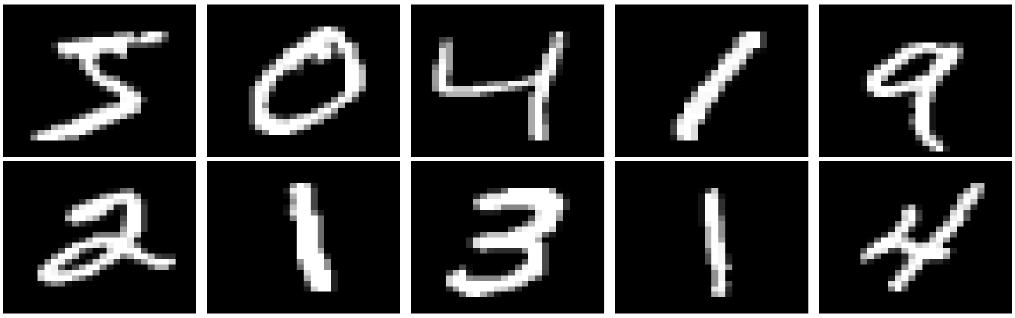

Figure 2.9: Samples from the MNIST handwritten digits recognition dataset.

For this task the frequency decreasing scattering convolutional network has $M = 3$ layers, breaking down the images into $J = 3$ scales and $L = 6$ orientations. For each input image this networks generates 127 scattering coefficients (see Equation 2.5) and thus yields a 99568 dimensional feature vector (Number of scattering coefficients × image dimensions — i.e. $127 \times 28 \times 28$). The discriminative classifier used is a set of binary SVM classifiers with a Gaussian radial basis function kernel [41]. This classifier have two meta-parameters. γ defines how influential a single training example is and C the trade off between misclassification of training examples and simplicity of the decision surface. Those meta-parameters are fine-tuned by cross-validation to $C = 3$ and $\gamma = 0.0018$.

Using this set-up, the trained model scores 99.47% accuracy on the test set, i.e. 9947 true positive out of 10000 realisations. This accuracy is of the same order of what can be obtained using a convolutional neural network [90, 151]. For reference, when apply directly to the raw pixel a linear classifiers reachs near 88% accuracy [45]. The improvement provided by the scattering network projection is a compeling argument for the need of signal representation prior to inference.

This approach of classification have been used successfully for many more applications but unfortunately it does not directly leverage the structure created by the scattering transform and the possible information contained into it. Nor that it provides a generative models of the data, with all the advantages encompassed (see Chapter 3). In Part IV, we focus on building a generative model describing a scattering convolutional network.

Chapter 3

Probabilistic Graphical Models

Probabilistic Graphical Models (PGMs) offer an efficient framework to express joint distributions and conditional independencies. They rely on the usage of a graph based representation of conditional dependence between a set of random variables. Such graphs can then be used to encode a complete distribution over a multi-dimensional space in a compact —or factorised— manner. Probabilistic graphical models exist under many forms but they can be split into two main families, the Bayesian Networks (BNs) and the Markov models (MMs). Both families encompass the properties of factorisation and independence defined by the graph, but differ when it comes to the specificities of the set of independence they can encode as well as the factorisation of the distribution that they can induce [77].

In Part IV, we will use a probabilistic graphical model to describe the scattering network defined in Section 2.3.2. This chapter aims at providing the necessary prior knowledge for this work. Note, however, that we do not aim here at providing a complete overview of the probabilistic graphical models field but rather at introducing some concepts that are used in the remainder of this document. A reader further interested PGMs could refer to Heckerman [44], [91] or Bishop [77] for a more complete introduction to those models.

This chapter introduces the two main classes of probabilistic graphical models. Section 3.1 focuses on Bayesian networks, while Section 3.2 provides more details about Markov models. Finally Section 3.3 introduces the basics of the approximate inference method known as variational inference.

3.1 Bayesian Networks:

A BN is subclass of probabilistic graphical model where the set of random variables and their conditional dependencies are expressed via a Directed Acyclic Graph (DAG). Those models can be used to describe either continuous or discrete random variables as well as system governed by a mix of those. The architecture of Bayesian Networks is further explained in Section 3.1.1. Section 3.1.2 describes the inference mechanism for those networks and Section 3.1.3 presents a brief overview of the learning mechanisms for BNs.

3.1.1 Architecture:

A BN is a graphical model encoding a joint probability distribution via a DAG.

Definition 3.1.1. ***Bayesian Network***
For a set of random variables $\mathbf{R} = \{R_i\}_{i \in [\![1,N]\!]}$, a Bayesian network consists of a direct acyclic graph $\mathcal{G}$ encoding a set of conditional independence assertions about the random variables in $\mathbf{R}$ and a set $\mathbf{P}$ of local probability distribution associated with each variable.

Each node of $\mathcal{G}$ encodes one of the random variable R_i and each edge $E_{i \to j}$ represents the possible conditional dependence between nodes R_i and R_j.

Such networks encodes the conditional independence properties of the distribution [47].

Proposition 2. ***(Conditional independence for Bayesian networks)***
In a BN, each node of the graph is conditionally independent of all its nondescen-

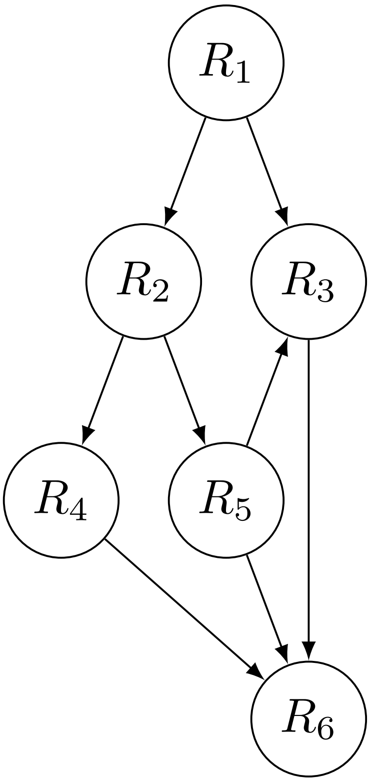

Figure 3.1: A simple Bayesian network.

dants in the graph given the value of all its parents.

$$P(\{R_i\}_{i\in[\![1,N]\!]}) = \prod_{i=1}^{N} P(R_i|R_{\rho(i)})$$

where $R_{\rho(i)}$ are the parents of the node R_i.

As a direct consequence to Property 2, one can say that a node with no parents is not conditioned on any other random variable considered. It defines a prior probability.

Property 2 allows to simplify the computation of the joint probability distribution represented by a Bayesian network. For example, for the network defined in Figure 3.1, the joint distribution can be obtained using the chain rule and theory on conditional independece,

$$\begin{aligned}
&P(R_1, R_2, R_3, R_4, R_5, R_6)\\
&\quad = P(R_6|R_3, R_4, R_5)P(R_1, R_2, R_3, R_4, R_5)\\
&\quad = P(R_6|R_3, R_4, R_5)P(R_3|R_1, R_5)P(R_4|R_2)P(R_5|R_2)P(R_1, R_2)\\
&\quad = P(R_6|R_3, R_4, R_5)P(R_3|R_1, R_5)P(R_4|R_2)P(R_5|R_2)P(R_2|R_1)P(R_1).
\end{aligned}$$

This example shows how BNs offer a convenient way to encode independence and an intuitive way to decompose the joint distributions.

3.1.2 Inference:

A Bayesian network encodes the full joint distribution of the studied random variables. This knowledge can be used to perform several interesting inference tasks among which are:

- Belief updating: Given some evidences —i.e. values for some nodes of the network $\{R_j\}_{j \in J}$ where J is a subset of the graph— we compute the probability associated with an unobserved variable,

  $$R_i^* = P(R_i|\{R_j\}_{j \in J}). \tag{3.1}$$

  R_i such that the probability from Equation 3.1 is maximised defines a prediction for this node. This is one of the advantages of belief updating over other prediction methods, it can provide a probabilistic prediction even when given incomplete observations —i.e. a set $\{R_j\}_{j \in J}$ such that $J \cup \{i\} \neq \mathbf{R}$. Belief updating can be extended to the prediction of a set of unobserved variables.

- Optimal decision: A probabilistic graphical model can be used to express actions taken by an agent to modify the state of an uncertain world. In this case given some evidence $\{R_j\}_{j\in J}$ where J is a subset of the graph $\mathcal{G}$ one is interested in finding the set of action $\{A_i\}_{i\in\mathcal{A}}$ where $\mathcal{A}$ is the set of all possible actions. To do so one also needs a reward function $O_i(A_i)$ expressing the outcome of the action A_i, maximising the probability of the outcome ,

$$\{A_i^*\}_{i\in\mathcal{A}} = \operatorname*{argmax}_{\mathcal{A}} P(\{O_i(A_i)\}_{i\in\mathcal{A}}|\{A_i\}_{i\in\mathcal{A}}, \{R_j\}_{j\in J}).$$

This type of inference is useful in Reinforcement learning framework where one is interested in learning the optimal set of actions to complete a task.

- Sensitivity analysis: Given some evidences —i.e. : values for some nodes of the network $\{R_j\}_{j\in J}$ where J is a subset of the graph— used for belief updating, one can be interested in assessing which among those random variables has the most influence on the prediction quality. This means find,

  $$\Delta_k^* = \operatorname*{argmax}_{k\in J} \Delta_k$$

  where Δ_k defines the difference between the probabilities given the full set of evidences and given the set minus the k-th evidence,

  $$\Delta_k = P(R_i|\{R_j\}_{j\in J}) - P(R_i|\{R_j\}_{j\in J\setminus\{k\}}).$$

  This type of inference can be useful in the case where the evidence is expensive to collect, or when prediction has to be provided within a certain time. Then the sensitivity analysis allows to focus the effort into collecting/incorporating the most important piece of information.

3.1.3 Learning:

In most applications, the full characterisation of the BN is not provided but has to be learned from a set of observations $\mathbf{X} = \{\mathbf{x}_n\}_{n \in [\![1,N]\!]}$. One can split the learning problem into two main categories:

- Learning the local probability distributions: In this case the structure of the graph $\mathcal{G}$ is known and fixed before hand. It can be provided by an expert (e.g. IBM trouble shooting system [65], disease diagnostic [40]) or be imposed by some construction rules (e.g. Boltzmann Machine [19], Restricted Boltzmann Machine [23] ...). The task at hand is then to learn the parameters θ governing the local probability distributions of the network.

- Learning the architecture and local probability distributions: In this case the architecture of the network $\mathcal{G}$ has to be learned along side with the local probability distributions' parameters θ. This problem is not developed in the rest of this document, but one could refer to [69] for an introduction to the existing methods.

Leaving aside the case where the network architecture has to be learned, the problem of learning the parameter of a Bayesian network can again be split into two main categories.

Complete data:

In this case each training example of the set $\mathbf{X}$ contains the value of the full set of random variable $\mathbf{R}$ of the graph. In such a case one can use methods such as the Maximum Likelihood estimates where the parameters of the network are selected to maximise the log-likelihood of the data given the model,

$$\theta_{ML} = \underset{\theta}{\operatorname{argmax}} \log P(\mathbf{X}|\theta).$$

Another common estimator used is the Maximum A Posteriori (MAP) estimate where one maximise the posterior distribution of the network's parameters given the data,

$$\theta_{MAP} = \underset{\Theta}{\operatorname{argmax}} \log P(\theta|\mathbf{X}) = \underset{\theta}{\operatorname{argmax}} \log(P(\mathbf{X}|\theta)P(\theta)).$$

Incomplete data:

In this case each training example of the set $\mathbf{X}$ only contains the value of some random variables $\{R_i\}$ of the graph. In such a case one can use a two steps iterative algorithm named the Expectation-Maximisation (EM) algorithm [16]. The first step (Expectation) aims at estimating the values of the unobserved random variables given the current estimate of the parameters θ. The maximisation step aims at providing a ML estimate of θ once all the variable are known or estimated. This procedure is described in more details in Section 6.4.

3.2 Markov Models:

Markov models are a subclass of graphical models useful when it comes to describing a system whose observations are randomly changing over an event. The key assumption in those models is that the upcoming a node only depends on a finite number of previous ones.

$$\forall t \in \mathbb{N} \quad P(\{O_k\}_{k \in \mathbb{N}}) \quad = \quad P(O_{t+1} \mid \{O_k\}_{k \in [\![t-l,t]\!]})$$

where $l \in \mathbb{N}$ characterised the number of past steps used to condition the next observation. Most of the time, for sake of computational tractability as well as because this constraint seems to be sufficient, the future observation is assumed to be only dependent on the present one. This is called the order-1 Markov dependence and can be expressed as,

$$\forall t \in \mathbb{N} \quad P(O_{t+1} \mid \{O_k\}_{k \in \mathbb{N}}) \quad = \quad P(O_{t+1} \mid O_t)$$

Maybe the most famous class of Markov model is the Markov chain. It has applications in finance (Brownian motion [54]), in Internet page ranking (Google page rank [67]) or as a sampling procedure (Markov Chain Monte-Carlo (MCMC) [74]). Those models are not covered in this document but Kemeny and Snell [5] provides a good entry point to the field for reader with further interests for Markov chains.

Another flavor of Markov models are the Hidden Markov Models (HMMs) [8]. The remainder of this section is dedicated to providing a general introduction to those models. While Section 3.2.1 formalises the HMM modelling, Section 3.2.2 and Section 3.2.3 respectively describes the inference mechanism and the learning method for HMMs.

3.2.1 Architecture:

An HMM is a stochastic finite automaton, where each hidden state generates —i.e. emits— an observation. Let O_t be the observation at step t and H_t denotes the hidden state at this step. Let also K_t be the number of possible states at step t such that $H_t \in [\![1, K_t]\!]$. The observations in an HMM can be discrete, continuous or mixed.

The model's parameters are:

- The initial state distribution $\pi(i) = P(H_0 = i)$ where π is a multinomial distribution.
- The transition model at step t, $A_t^{(ij)} = P(H_{t+1} = j | H_t = i)$ where A_t is a stochastic matrix.
- The emission model $P(O_t | H_t)$. Usually the emission model is defined by a parametric distribution governed by $\rho_{k,t}$. In the case of discrete observations, it is defined by a multinomial distributions such that,

$$\forall t \geq 0 \;\; \forall (l,k) \in |O_t| \times [\![1, K_t]\!] \;\; P(O_t = l | H_t = k) = P_{\rho_{k,t}}(l).$$

where $|O_t|$ is the set of value for O_t. In the continuous case, the observation model is a continuous parametric distribution such that,

$$\forall t \geq 0 \ \ \forall (l,k) \in \mathbb{R}^{d_t} \times [\![1, K_t]\!] \ \ P(O_t = l | H_t = k) = P_{\rho_{k,t}}(l).$$

where d_t is the dimension of O_t.

A common simplification to those model is to assume stationarity. This means stating that the transition matrices and observation models are shared across steps.

3.2.2 Inference:

HMMs are probabilistic graphical models. Thus, by definition, they encode the joint distribution of the system. They can be used to perform similar inference tasks as those described in Section 3.1.2 for Bayesian networks, with the difference that for HMMs, the values of the hidden states have to be inferred prior to solving any specific request.

Inferring the value of the hidden states can be done using a MAP estimate adapted to HMMs. The initial MAP algorithm is due to Viterbi [10] and was originally designed to analyse Markov processes observed in memory-less noise. Forney Jr [15] expressed this algorithm as being equivalent to finding the shortest path in a graph with weighted edges.

Note. This procedure is described at length in the special case of hidden Markov trees in Section 6.5 but this section aims at providing an informal explanations on the methodology.

The observation behind Viterbi's MAP algorithm for HMMs is that for any state at step t, we can easily find a most likely path to this state. Therefore, one can simplify the computation by replacing several paths converging to a given state at step t by simply the most likely one. Applying this method at each step of the model reduces the computation complexity from $\mathcal{O}(K^t)$ to $\mathcal{O}(tK^2)$.

3.2.3 Learning:

Learning the parameters of an HMM from data is somehow similar to learning the parameters of a Bayesian network in the case of incomplete data, as only parts of the nodes of an HMM are observed. Hence the parameters of an HMM model can be learned using the offline maximum likelihood (ML) estimation method known as the EM — or Baum-Welch— algorithm [16].

Let $\{O^n_{[1:T]}\}_{n \in [\![1,N]\!]}$ be a set of observed nodes of an HMM used as training set. The learning procedure would be straight forward if one had access to the sequences of hidden state $H^n_{[0:T]}$ for all $n \in [\![1,N]\!]$. The ML estimate of the transition matrix, for example, could be computed by normalising the matrix of co-occurrences,

$$A^{(ij)}_{t,ML} = \frac{C_t(i,j)}{\sum_{k=1}^{K} C_t(i,k)}$$

where

$$C_t(i,j) = \sum_{n=1}^{N} \mathbb{1}(H_{t+1} = j, H_t = i)$$

and $\mathbb{1}(\text{event})$ is the binary indicator of occurrence of a event. The initial distribution and the observation model could be estimated in a similar fashion.

However since $H^n_{[0:T]}$ is hidden, one has to estimate the hidden states prior to performing the ML update. The general idea of the EM-algorithm is to estimate the hidden states given the observations using a variant of the Maximum A Posteriori approach described in the previous section with the current set of parameters, which compute the corresponding expected values of the hidden states given the observation. This is the Expectation (E) step. Those estimated sequences of hidden states are then used to update the parameters' estimates. This is the Maximisation (M) step.

One can prove [12, 16] the convergence of this procedure toward a —local— maximum of the likelihood.

3.3 Variational inference

As seen in the Section 3.1 and Section 3.2, the key to performing probabilistic inference in graphical models reduces down to the task of computing a conditional probability distribution over the values of the hidden nodes, given the values of the observed ones. Using the notation defined in the previous sections, this can be written,

$$P(H|O) = \frac{P(H,O)}{P(O)}$$

Though exact inference algorithms exist [29, 32, 36] and can used in practice (see Chapter 6 for an example of application). The problem at hand, however, quickly becomes intractable when increasing the complexity of the model. This calls for approximate inference method.

In that field two main paradigms exist. Markov Chain Monte-Carlo (MCMC) [13, 24]. The basic idea behind MCMC is to construct an ergodic Markov chain on the hidden variables H such that its stationary distribution is the posterior $P(H|O)$. We sample from that chain to collect estimates of the posterior. Finally, we use —part of — those samples to construct an empirical estimate of the posterior. MCMC sampling is a well established and yet still very active field of research [3, 13, 18, 138] and has successfully been applied to Bayesian statistics [24].

Some problems, however, are not easily suitable for this approach. MCMC based methods show limitation in term of scalability to large datasets and very complex models. Theoretically the MCMC method will converge but it might be prohibitively slow. In these settings, variational inference provides a good alternative approach to approximate Bayesian inference.

Rather than setting the task as a sampling problem, variational inference posit the posterior estimation task as an optimisation problem. We assume the densities over the latent variable lay in a family $\mathcal{Q}$. And we use as approximation the member of that family that minimise the variational objective to the exact posterior,

$$q^*(H) = \underset{q(H)\in\mathcal{Q}}{\operatorname{argmin}} D_{.}(q(H), p(H|O)). \tag{3.2}$$

Where the objective $D_{.}$ is a divergence measure [1, 4] and can be modified to change the properties of the approximation (see Chapter 4). However in its simplest form the variational objective is defined using the Kullback-Leibler (KL) divergence [2]. In that case, the objective from Equation 3.2 can bewritten,

$$D_{KL}(q(H), p(H|O)) = -\int q(h) \log \frac{p(H|O)}{q(H)} dH \tag{3.3}$$

This can be interpreted as an expectation taken with regard to $q(H)$. Equation 3.3 reveals a dependency of this objective on the intractable probability of the observation $p(O)$ by expressing $p(H|O)$ as $p(H,O)/p(O)$. One cannot directly minimise the objective defined in Equation 3.2 for the

KL divergence. However, in Equation 3.3, one can notice that,

$$\begin{aligned}
&D_{KL}(q(H),p(H|O)) \\
&= -\int q(H)\left[\log\frac{p(H,O)}{p(O)} - \log q(H)\right]dH \\
&= \int q(H)\log q(H)dH - \int q(H)\log p(H,O)dH + \int q(H)\log P(O)dH \\
&= \mathcal{L}_{KL}(q(H),p(H,O)) + \log p(O).
\end{aligned} \tag{3.4}$$

From Equation 3.4, we get,

$$\log p(O) = \mathcal{D}_{KL}(q(H),p(H|O)) + \mathcal{L}_{KL}(q(H),p(H,O)),$$

where $\mathcal{L}_{KL}(q(H),p(H,O))$ defines the Evidence Lower BOund (ELBO) for the Kullback-Leibler divergence and can be expressed as,

$$\mathcal{L}_{KL}(q(H),p(H,O)) = \int q(H)\log\frac{p(H,O)}{q(H)}dH. \tag{3.5}$$

We know $\log P(O)$ is constant and $D_{KL}(q(H),p(H|O)) \geq 0$, so minimising $D_{KL}(q(H),p(H|O))$ is equivalent to maximising $\mathcal{L}_{KL}(q(H),p(H,O))$. While optimising directly the KL-divergence is intractable, this equivalent optimisation problem is tractible.

This optimisation of an ELBO rather than the true variational objective is common in the VI literature and has been used to extend the objective defined in Equation 3.5 to broader families of divergences (see Part II).

Despite being, by design, better suited to large datasets than MCMC sampling in the recent years work as been done to allow VI to scale to the dimension of modern datasets. To that end, general method such as stochastic variational inference [123] have been developped. It allows the fitting of variational posteriors without iterating over the complete dataset at every step of the procedure but only on a subset. The quantity needed for the updates are estimated from the fraction of data observed.

In the remainder of this document we both improve the core theory of variational inference and propose an application on a complex graphical model. Chapter 4 defines a new variational objective to overcome some weaknesses of the KL-objective. In Chapter 7, we use a variational approximation to simplify learning the parameter of the SHMT model (see Chapter 6).

Part II

Flexible Variational Inference

Chapter 4

Alpha-Beta variational Inference

This chapter introduces a variational approximation framework using direct optimisation of what is known as the *scale invariant Alpha-Beta divergence* (sAB divergence). This new objective encompasses most variational objectives that use the Kullback-Leibler, the Rényi or the gamma divergences. It also gives access to objective functions never exploited before in the context of variational inference. This is achieved via two easy to interpret control parameters, which allow for a smooth interpolation over the divergence space while trading-off properties such as mass-covering of a target distribution and robustness to outliers in the data. Furthermore, the sAB variational objective can be optimised directly by re-purposing existing methods for Monte Carlo computation of complex variational objectives, leading to estimates of the divergence instead of variational lower bounds. We show the advantages of this objective on Bayesian models for regression problems.

Chapter outline

We propose here a variational objective to simultaneously trade-off effects of mass-covering, spread and outlier robustness. This is done by developing a variational inference objective using an extended version of the alpha-beta (AB) divergence [107], a family of divergence governed by two parameters and covering many of the divergences already used for VI as special cases. Section 4.1 provides further details on the motivation behind that new objective. After reviewing some basic concepts of VI and some useful divergences in Section 4.2, we extend it to what we will call the scale invariant AB (sAB) divergence and explain the influence of each parameters (see Section 4.3). In Section 4.4, we then develop a framework to perform direct optimisation of the divergence measure which can leverage most of the modern methods to ensure scalability of VI. Finally, in Section 4.5, we demonstrate the interesting properties of the resulting approximation on regression tasks with outliers.

4.1 Introduction

Modern probabilistic machine learning relies on complex models for which the exact computation of the posterior distribution is intractable. This has motivated the need for scalable and flexible approximation methods. Research on this topic belongs mainly to two families, sampling based methods constructed around Markov Chain Monte Carlo (MCMC) approximations [73], or optimisation based approximations collectively known under the name of *variational inference* (VI) [49]. In this chapter, we focus on the latter, although with the aid of Monte Carlo methods.

The quality of the posterior approximation is a core question in variational inference. When using the KL-divergence [2] averaging with respect to the approximate distribution, standard VI methods such as mean-field underestimate the true variance of the target distribution. In this scenario, such behaviour is sometimes known as *mode-seeking* [75]. On the other end, by (approximately) averaging over the target distribution as in Expectation-Propagation, we might assign much mass to low-probability regions [75]. In an effort to smoothly interpolate between such behaviours, some recent contributions have exploited parameterised families of divergences such as the alpha-divergence [75, 112, 157], and the Rényi-divergence [158]. Another fundamental property of an approximation is its *robustness to outliers*. To that end, divergences such as the beta [42] or the gamma-divergences [84] have been developed and widely used in fields such as matrix factorisation [96, 108]. Recently, they have been used to develop a robust pseudo variational inference method [162]. A cartoon depicting stylised examples of these different types of behaviour is shown in Figure 4.1,

4.2 Background

This section briefly reviews the basis of variational inference (for a longer introduction to the concept please refer to Section 3.3). It also introduces some divergence measures which have been used before in the context of VI, and which will be used as baselines in this chapter.

4.2.1 Variational Inference

We first review the variational inference method for posterior approximation, as typically required in Bayesian inference tasks. Unless stated otherwise, the notation defined in this section will be used throughout this chapter.

Let us consider a set of N i.i.d samples $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^{N}$ observed from a probabilistic model $p(\mathbf{x}|\theta)$ parameterised by a random variable θ that is drawn from a prior $p_0(\theta)$. Bayesian inference involves computing the posterior distribution of the unknowns given the observations:

$$p(\theta|\mathbf{X}) = \frac{p_0(\theta)\prod_{n=1}^{N} p(\mathbf{x}_n|\theta)}{p(\mathbf{X})}$$

This posterior is in general intractable due to the normalising constant.

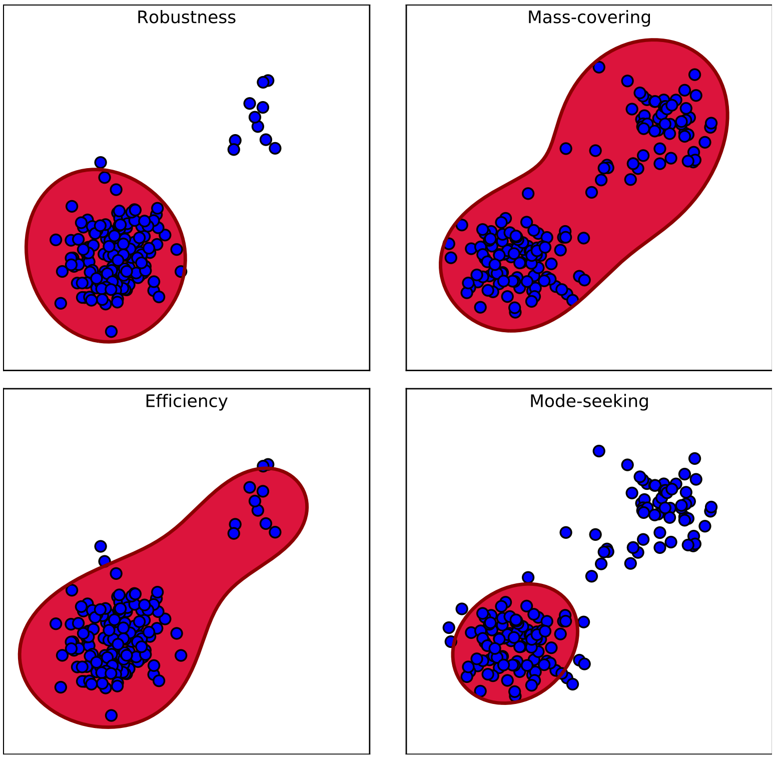

Figure 4.1: Illustration of the robustness/efficiency properties (left) and mass-covering/mode-seeking (right). The red region is a stylised representation of a high probability region of a model approximated to fit training data (blue points). Mass-covering and mode-seeking are well-established concepts described by [75]. Efficiency refers to the ability of capturing the correct distribution from data, including tail behaviour. Robustness is defined here as the ability of ignoring points contaminated with noise that are judged not to be representative of test-time behaviour if their probability is too small, according to a problem-dependent notion of outliers.

The idea behind variational inference is to reduce the inference task to an optimisation problem rather than an integration problem. To do so, it introduces a probability distribution $q(\theta)$ from a tractable family $\mathcal{Q}$, optimised to approximate the true posterior to an acceptable standard. The approximation is found by minimising a divergence $D[q(\theta)||p(\theta|\mathbf{X})]$ between the approximation and the true posterior. For the vast majority of divergences, this objective remains intractable as it usually involves computing $p(\mathbf{X})$. VI circumvents the issue by considering the equivalent maximisation a lower-bound (“ELBO,” short for *evidence lower-bound*) of that objective,

$$\mathcal{L}_D(q,\mathbf{X},\boldsymbol{\varphi}) \equiv \log p(\mathbf{X}|\boldsymbol{\varphi}) - D(q(\theta)||p(\theta|\mathbf{X},\boldsymbol{\varphi})) \tag{4.1}$$

where $D(.||.)$ is a divergence measure and $\mathcal{L}_D(.)$ denotes the objective function associated with D.

4.2.2 Notable Divergences and their Families

A key component for successful variational inference lies in the choice of the divergence metric used in Equation (4.1). A different divergence means a different optimisation objective and results in the approximation having different properties. Over the years, several have been proposed. The review below here does not intend to be exhaustive, but focuses only on the divergences of interest in the context of this document.

Arguably, the most famous divergence within the VI community is the Kullback-Leibler divergence [49],

$$D_{KL}(q||p) = \int q(\theta) \log\left(\frac{q(\theta)}{p(\theta)}\right) d\theta. \tag{4.2}$$

It offers a relatively simple to optimise objective. However, because the KL-divergence considers the log-likelihood ratio q/p, it tends to penalise more the region where $q > p$ —i.e, for any given region over-estimating the true posterior is penalised more than underestimating it. The approximation derived tends to poorly cover regions of small probability in the target model [110] while focusing on a number of modes according to what is allowed by the constraints of $\mathcal{Q}$.

To mitigate this issue, efforts have been made to use broader families of divergences, where one meta-parameter can be tuned to modify the mass-covering behaviour of the approximation. In the context of variational inference, the alpha-divergence [112] has been used to develop power EP [75] and the black-box alpha divergence [157]. In this chapter, however, we focus on the Rényi divergence [6, 146] [1],

$$D_R^{\alpha}(p||q) = \frac{1}{\alpha - 1} \log \int p(\theta)^{\alpha} q(\theta)^{1-\alpha} d\theta, \tag{4.3}$$

used in Rényi VI [158]. For this family, the meta-parameter α can be used to control the influence granted to likelihood ratio p/q on the objective in regions of over/under estimation. This flexibility has allowed for improvements on traditional VI on complex models, by fine-tuning the meta-parameter to the problem [155].

KL-divergence also suffers from the presence of outliers in the training data [163]. To perform robust distribution approximation, families of divergences such as the beta-divergence [42] have been developed and used to define a pseudo variational objective [156]. Instead of solving the optimisation problem defined in Equation (4.1), they use a surrogate objective function motivated by the beta-divergence. In this part, however, we focus on the gamma-divergence [84],

$$\begin{aligned} D_{\gamma}^{\beta}(p||q) &= \frac{1}{\beta(\beta+1)} \log \int p(\theta)^{\beta+1} d\theta \\ &+ \frac{1}{\beta+1} \log \int q(\theta)^{\beta+1} d\theta - \frac{1}{\beta} \log \int p(\theta) q(\theta)^{\beta} d\theta. \end{aligned} \tag{4.4}$$

This family has a Pythagorean relation property [84], meaning that,

$$D_{\gamma}^{\beta}(p_{\epsilon}||q) - D_{\gamma}^{\beta}(p||q) \approx D_{\gamma}^{\beta}(p_{\epsilon}||p), \tag{4.5}$$

where p_{ϵ} is a perturbated version of p such that their density are still approximately similar. As direct consequence of Equation 4.5, one can say that the parameter β controls how much importance is granted to small perturbation in the target distribution. The upshot is that in the case the data is contaminated with *outliers* — here interpreted as data points contaminated with noise, which are assumed to be spurious and must not be

[1] In this chapter we focus on divergences of the general form $\log \int(.)$. In Section 4.3.1, we will see that this type of divergence allows to simplify the computation of the variational objective to something computationally tractable.

covered by the model, although not easy to clean manually in multivariate distributions — then the tail behaviour of the model will be compromised[2]. If the divergence measure is not flexible enough, accommodating outliers may have unintended effects elsewhere in the model. [162] propose a framework to use the gamma-divergences for pseudo VI. Here again their method only proposes a pseudo-Bayesian variational updates where the objective does not satisfy Equation (4.1). Despite that they obtain a posterior robust to outliers.

As flexible as the divergences defined in Equations (4.3) and (4.4) are, they control only either the mass-covering property or the robustness property, respectively. The AB-divergence [107] allows for both properties to be tuned independently, but to the best of our knowledge it has not yet been used in the context of variational inference.

4.3 Scale invariant AB Divergence

In this section, we extend the definition of the scale invariant AB-divergence [107] (sAB), as well as defining it for continuous distributions. We also describe how it compares to other commonly used divergence measures.

4.3.1 A two degrees of freedom family of divergences

Under its simplest form, the AB-divergence cannot be used for variational inference as it does not provide any computationally tractable form for the loss function $\mathcal{L}_{AB}(.)$ as defined in Equation (4.1) as one cannot isolate the terms involving computing the marginal likelihood $p(\mathbf{X})$. Detailed computations are available in Appendix .1. One could use the AB-divergence to perform pseudo variational updates as described in [162]. However, in that

[2] The point being is that we should not focus on changing the model to accommodate noise, which might not exist out-of-sample, but to change the estimator. The difference between estimator and model is common in frequentist statistics, with the Bayesian counterpart being less clear at the level of generating a posterior distribution. One could consider a measurement error model that accounts for noise at training time, to be removed at test time, for instance, at the cost of complicating inference. The estimator is considered, in our context, as the choices made in the approximation to the posterior.

case we would lose the guarantees of divergence minimisation. Consider instead, as our primary divergence of interest, the scale invariant version of the AB-divergence. This concept was briefly introduced by [107],

$$\begin{aligned} D^{\alpha,\beta}_{sAB}(p||q) \equiv & \frac{1}{\beta(\alpha+\beta)} \log \int p(\theta)^{\alpha+\beta} d\theta \\ & + \frac{1}{\alpha(\alpha+\beta)} \log \int q(\theta)^{\alpha+\beta} d\theta \\ & - \frac{1}{\alpha\beta} \log \int p(\theta)^{\alpha} q(\theta)^{\beta} d\theta, \end{aligned} \tag{4.6}$$

for $(\alpha,\beta) \in \mathbb{R}^2$ such that $\alpha \neq 0$, $\beta \neq 0$ and $\alpha+\beta \neq 0$. The divergence from Equation 4.6 is called scale invariant as for all $c_1, c_2 \in \mathbb{R}^*_+$, $D^{\alpha,\beta}_{sAB}(c_1 p||c_2 q) = D^{\alpha,\beta}_{sAB}(p||q)$. This invariance property is valuable when working with non normalised distributions.

4.3.2 Extension by continuity

In Equation (4.6), the sAB divergence is not defined on the complete $\mathbb{R}^2$ space. We extend this definition to cover all values $(\alpha,\beta) \in \mathbb{R}^2$ for the purpose of comparison with other known divergences, as shown in Equation(4.7). Detailed computations are available in Appendix .2.

$$D_{sAB}^{\alpha,\beta}(p||q) \equiv
\begin{cases}
\frac{1}{\alpha\beta}\log\frac{\left(\int p(\theta)^{\alpha+\beta}d\theta\right)^{\frac{\alpha}{\alpha+\beta}}.\left(\int q(\theta)^{\alpha+\beta}d\theta\right)^{\frac{\beta}{\alpha+\beta}}}{\int p(\theta)^{\alpha}q(\theta)^{\beta}d\theta}, \\
\qquad\qquad\qquad \text{for } \alpha \neq 0, \beta \neq 0, \alpha+\beta \neq 0 \\
\frac{1}{\alpha^2}\left(\log\int\left(\frac{p(\theta)}{q(\theta)}\right)^{\alpha}d\theta - \int\log\left(\frac{p(\theta)}{q(\theta)}\right)^{\alpha}d\theta\right), \\
\qquad\qquad\qquad \text{for } \alpha = -\beta \neq 0 \\
\frac{1}{\alpha^2}\left(\log\frac{\int q(\theta)^{\alpha}d\theta}{\int p(\theta)^{\alpha}d\theta} - \alpha\log\int q(\theta)^{\alpha}\log\frac{q(\theta)}{p(\theta)}d\theta\right), \\
\qquad\qquad\qquad \text{for } \alpha \neq 0, \beta = 0 \\
\frac{1}{\beta^2}\left(\log\frac{\int p(\theta)^{\beta}d\theta}{\int q(\theta)^{\beta}d\theta} - \beta\log\int p(\theta)^{\beta}\log\frac{p(\theta)}{q(\theta)}d\theta\right), \\
\qquad\qquad\qquad \text{for } \alpha = 0, \beta \neq 0 \\
\frac{1}{2}\int(\log p(\theta) - \log q(\theta))^2 d\theta, \qquad \text{for } \alpha = 0, \beta = 0
\end{cases}
\tag{4.7}$$

For $\alpha = 0$ or $\beta = 0$, the sAB-divergence reduces to a KL-divergence scaled by a power term. For $\alpha = 0$ and $\beta = 0$, we get a log-transformed Euclidean distance [149]. As we will see in Section 4.4, the sAB-divergence can be used in the variational inference context.

One can notice that the scale invariance property does not hold to the limits. Even if desirable, this behavior is not critical for the rest of our analysis. It is also interesting to note that the Rényi-divergence [6, 146] can be obtained by applying the same transformation to the alpha-divergence, similarly the gamma-divergence [84] is the transformed version of the beta-divergence.

4.3.3 Special cases

In this section, we describe how some specific choice of parameters (α, β) simplifies the sAB-divergence into some known divergences or families of divergences.

When $\alpha = 0$ and $\beta = 1$ the sAB-divergence reduces down to the Kullback-Leibler divergence as defined in Equation (4.2). By symmetry, the reverse KL is obtained for $\alpha = 1$ and $\beta = 0$.

More generally, when $\alpha + \beta = 1$, Equation (4.7) becomes,

$$D_{sAB}^{\alpha+\beta=1}(p||q) = \frac{1}{\alpha(\alpha-1)} \log \int p(\theta)^{\alpha} q(\theta)^{1-\alpha} d\theta,$$

and the sAB-divergence is proportional to the Rényi-divergence defined in Equation (4.3).

When $\alpha = 1$ and $\beta \in \mathbb{R}$, Equation (4.7) becomes

$$D_{sAB}^{\alpha=1,\beta}(p||q) = \frac{1}{\beta(\beta+1)} \log \int p(\theta)^{\beta+1} d\theta$$
$$+ \frac{1}{\beta+1} \log \int q(\theta)^{\beta+1} d\theta - \frac{1}{\beta} \log \int p(\theta) q(\theta)^{\beta} d\theta.$$

and the sAB-divergence is equivalent to gamma-divergence.

A mapping of the (α, β) space is shown in Figure 4.2. To summarise, the sAB-divergence allows smooth interpolation between many known divergences.

4.3.4 Robustness of the divergence

To develop a better understanding on why using the sAB-divergence might be good as a variational objective, we describe how the governing parameters affect the optimisation problem for various divergences. Let us assume here that the approximation q is a function of a vector of parameters $\boldsymbol{\varphi}$. Detailed computations are available in Appendix .4.

Let us first consider as a baseline the usual KL-divergence $D_{KL}(q||p)$. In that case, the optimal estimated parameter $\hat{\boldsymbol{\varphi}}$ is solution of,

$$\frac{d}{d\boldsymbol{\varphi}} D_{KL}(q||p) = -\int \frac{dq(\theta)}{d\boldsymbol{\varphi}} \left(\log \frac{p(\theta)}{q(\theta)} - 1 \right) d\theta = 0. \tag{4.8}$$

The log-term in Equation (4.8) increases with the cost over-estimating p and hence causes the underestimation of the posterior variance [110].

In order to gain more flexibility in the approximation behaviour, some

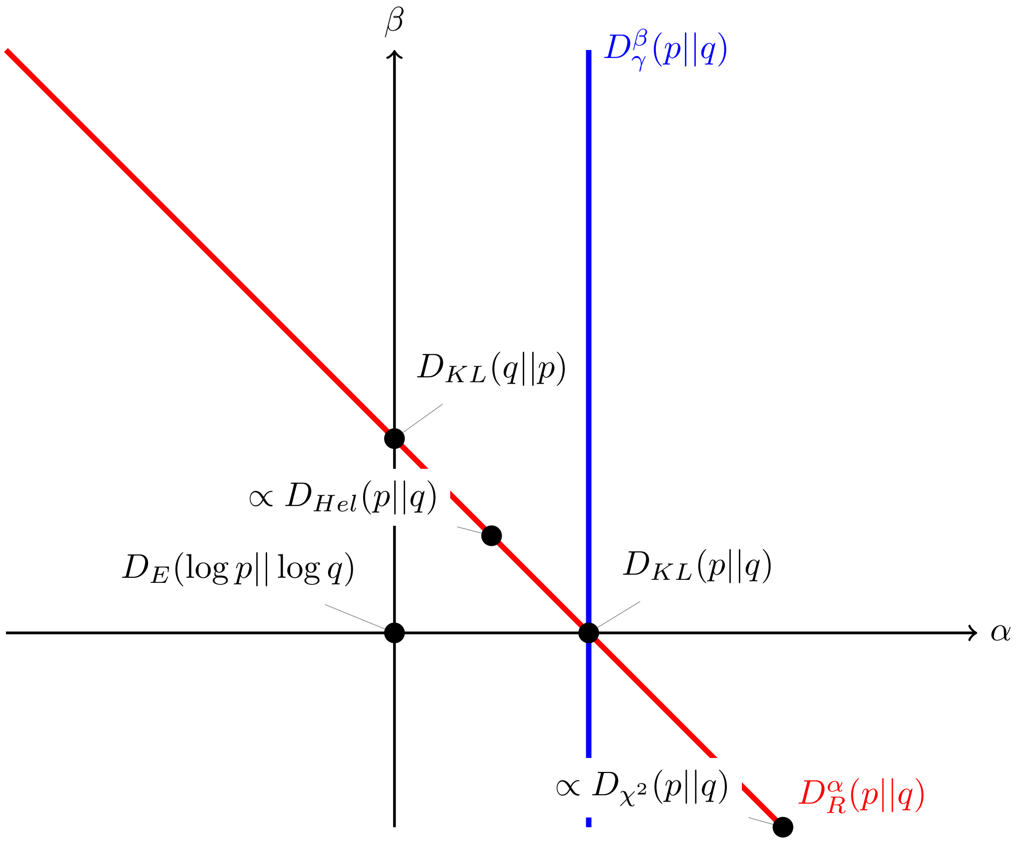

Figure 4.2: Mapping of the (α, β) space. The sAB-divergence reduces down to many known divergences but also interpolates smoothly in between them and cover a much broader spectrum than the Rényi or the gamma-divergence. For (α, β) equals $(0.5, 0.5)$ and $(2, -1)$ the sAB divergence is proportional to respectively the Hellinger and the Chi-square divergences. Detailed expressions for the divergences mentioned in that Figure are available in Appendix .3.

have suggested using broader families of divergences to formulate the variational objective. The Rényi divergence [158] is an example of such divergence. The estimator $\hat{\boldsymbol{\varphi}}$ obtained with the Rényi divergence is a solution of,

$$\frac{d}{d\boldsymbol{\varphi}} D_R^\alpha(q||p) = -\frac{\alpha}{1-\alpha} \frac{\int \frac{dq(\theta)}{d\boldsymbol{\varphi}} \left(\frac{p(\theta)}{q(\theta)}\right)^{1-\alpha} d\theta}{\int q(\theta)^\alpha p(\theta)^{1-\alpha} d\theta} = 0. \tag{4.9}$$

This simplifies to,

$$\int \frac{dq(\theta)}{d\boldsymbol{\varphi}} \left(\frac{p(\theta)}{q(\theta)}\right)^{1-\alpha} d\theta = 0. \tag{4.10}$$

When using the Rényi-divergence as an objective, the influence of the ratio of p/q is deformed by a factor α. This allows the practitioner to select whether to emphasise the relative importance of the large ratios (i.e. set $\alpha < 0$) or on the small ones (i.e. set $\alpha > 0$), thus going from respectively mass-covering to mode-seeking behaviour. However this does not provide any mechanism to handle outliers or rare events.

In the case of the gamma-divergence discussed by [162], the estimator $\hat{\boldsymbol{\varphi}}$ is solution of,

$$\begin{aligned}\frac{d}{d\boldsymbol{\varphi}}D_{\gamma}^{\beta}(q||p) = -\frac{1}{\beta}\Bigg(&\frac{\int \frac{dq(\theta)}{d\boldsymbol{\varphi}}q(\theta)^{\beta}\frac{p(\theta)}{q(\theta)}d\theta}{\int q(\theta)^{\beta}p(\theta)d\theta} \\ &-\beta\frac{\int \frac{dq(\theta)}{d\boldsymbol{\varphi}}q(\theta)^{\beta}d\theta}{\int q(\theta)^{\beta+1}d\theta}\Bigg) = 0.\end{aligned} \tag{4.11}$$

When using the gamma-divergence, the influence of the ratio p/q in the gradient is weighted by the factor $q(\theta)^{\beta}$. For $\beta < 1$, its influence is reduced for small values of q causing robustness to outliers. For $\beta > 1$, the influence of ratios where q is large is reduced instead causing a focus on outliers - i.e. an artificially increased influence of the low probability events. By setting β to values slightly below 1, one can achieve robustness to outliers whilst maintaining the efficiency of the objective [84].

Finally the sAB-divergence with regard to $\boldsymbol{\varphi}$ yields an estimator $\hat{\boldsymbol{\varphi}}$ solution of,

$$\begin{aligned}&\frac{d}{d\boldsymbol{\varphi}}D_{sAB}^{\alpha,\beta}(q||p) = \\ &-\frac{1}{\beta}\left(\frac{\int \frac{dq(\theta)}{d\boldsymbol{\varphi}}q(\theta)^{\alpha+\beta-1}\left(\frac{p(\theta)}{q(\theta)}\right)^{\beta}d\theta}{\int q(\theta)^{\alpha}p(\theta)^{\beta}d\theta}\right. \\ &\qquad\left.-\alpha\beta\frac{\int \frac{dq(\theta)}{d\boldsymbol{\varphi}}q(\theta)^{\alpha+\beta-1}d\theta}{\int q(\theta)^{\alpha+\beta}d\theta}\right) = 0.\end{aligned} \tag{4.12}$$

The two meta-parameters of the sAB-divergence allow us to combine the effects of both the gamma and the Rényi divergences. All the terms similar to Equation (4.11) are controlled by the parameter $\alpha + \beta - 1$. For the sake of clarity, in the reminder of the chapter we will use the expression $\lambda = \alpha + \beta$ and parameterised the AB divergence in terms of λ and β. One can control the robustness of the objective by varying λ. By setting it to small values below 2, one can achieve robustness to outliers while maintaining the efficiency of the objective. The terms responsible for the "mode-seeking" behaviour as seen in Equation (4.10) are here governed by the term $1 - \beta$. Thus for $\beta > 1$, one gets the objective to promote a mass-covering behaviour. For $\beta < 1$, it promotes mode-seeking behaviour.

Figure 4.3 provides a visual explanation of the influence of each parameters.

In the remainder of the document, we will report the values used to instantiate the sAB-divergence using $\lambda = \alpha + \beta$ instead of α to get a direct understanding in terms of robustness and mass covering properties.

To further illustrate the flexibility offered by the two control parameters of the sAB-divergence, Figure 4.4 shows the approximation q minimising $D_{sAB}(\alpha,\beta)(q||p)$. Here p is set to be a mixture of two skewed unimodal densities — a tall and narrow one combined with a short and wide density. Density q is required to be a single (non skewed) Gaussian with arbitrary mean and variance. The optimal solution is found by performing a greedy search on the paramater space of θ — i.e. the mean and the variance of the variational posterior.

The sAB divergence allows to smoothly tune the properties of the objective between "mass covering" and "robustness to outliers." In this sense, it is a richer objective than either the Rényi or the gamma divergences, which can only affect respectively the "mass covering" or the "robustness" properties.

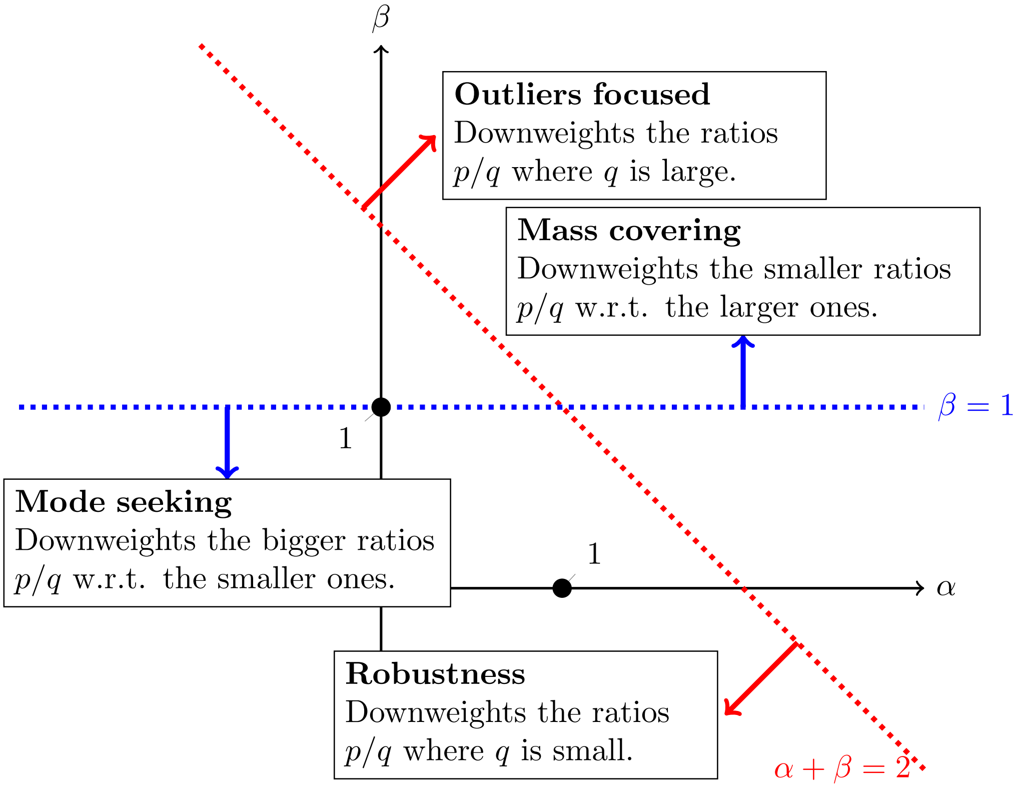
β
Outliers focused
Downweights the ratios
p/q where q is large.
Mass covering
Downweights the smaller ratios
p/q w.r.t. the larger ones.
$\beta = 1$
1
Mode seeking
Downweights the bigger ratios
p/q w.r.t. the smaller ones.
1
α
Robustness
Downweights the ratios
p/q where q is small.
$\alpha + \beta = 2$

Figure 4.3: Graphical illustration of the influence of the set control parameters (α, β).The red line $< \alpha + \beta = 2 >$ shows the region where the robustness factor $q(\theta)^{\alpha+\beta-1}$ in Equation (4.12) is uniform. The blue line $< \beta = 1 >$ shows the region where the ratio p/q in the mass-seeking term $(p(\theta)/q(\theta))^{\beta}$ is constant and equal to that of the standard Kullback-Leibler divergence.

4.4 sAB-divergence Variational Inference

In this section we present how the sAB-divergence can be used for approximate inference.

Let us consider a posterior distribution of interest $p(\theta|\mathbf{X})$ as well as a probability distribution $q(\theta)$ set to approximate the true posterior and let us derive the associated sAB variational objective.

4.4.1 sAB Variational Objective

As seen in Section 4.2.1, the variational approximation is fitted by minimising the divergence between the true distribution and the approximated posterior. Using the sAB-divergence defined in Equation (4.7) we get the

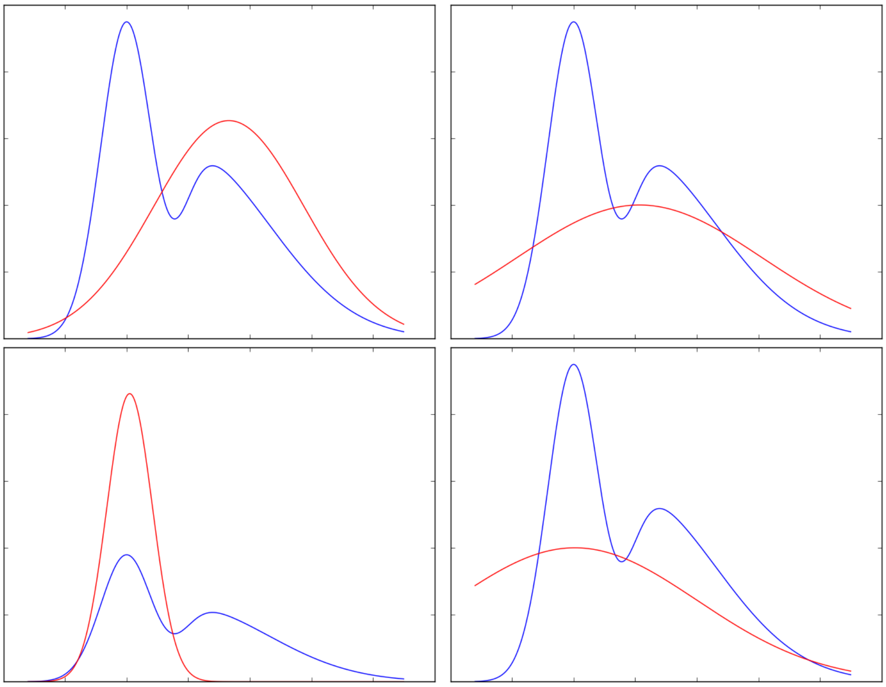

Figure 4.4: Approximation of a mixture of 2 skewed densities p by a Gaussian q for various parameters λ and β. $\lambda < 2$ causes the objective to be robust to outliers, while $\lambda > 2$ increases their weight. $\beta > 1$ causes the objective to have a mass-covering property, whilst $\beta < 1$ enforce mode-seeking.
Top Left: $\lambda = 2.4$, $\beta = -1.0$.
Top Right: $\lambda = 2.4$, $\beta = 2.0$.
Bottom Left: $\lambda = 1.8$, $\beta = -1.0$.
Bottom Right: $\lambda = 1.8$, $\beta = 2.0$.

following objective,

$$\begin{aligned} & D_{sAB}^{\alpha,\beta}(q(\theta)||p(\theta|\mathbf{X})) \\ & = \frac{1}{\alpha(\alpha+\beta)} \log \mathbb{E}_q \left[\frac{p(\theta,\mathbf{X})^{\alpha+\beta}}{q(\theta)} \right] \\ & \quad + \frac{1}{\beta(\alpha+\beta)} \log \mathbb{E}_q \left[q(\theta)^{\alpha+\beta-1} \right] \\ & \quad - \frac{1}{\alpha\beta} \log \mathbb{E}_q \left[q(\theta)^{\alpha+\beta-1} \left(\frac{p(\theta,\mathbf{X})}{q(\theta)} \right)^{\beta} \right] \end{aligned} \tag{4.13}$$

Details of the computation as well as the extension to the complete domain of definition are detailed in Appendix .5. Note that we here compute $D_{sAB}^{\alpha,\beta}(q(\theta)||p(\theta|\mathbf{X}))$ (as opposed to $D_{sAB}^{\alpha,\beta}(p(\theta|\mathbf{X})||q(\theta))$ in Equation 4.6). We do so in order to match the direct KL optimisation objective $D_{KL}(q(\theta)||p(\theta|\mathbf{X}))$ in standard VI. The scale invariant AB-divergence between the true posterior and the variational approximation can be expressed as a sum of expectations with regard to the variational approximation. Usually in variational inference, the term corresponding the marginal likelihood $p(\mathbf{X})$ is dropped, so that the objective function is not the divergence itself but an expression that can be interpreted as a lower bound on the marginal likelihood, the ELBO. Here, we optimise directly on the divergence itself as the terms involving the probability of the data $p(\mathbf{X})$ cancel each other. At least in principle, this provide a way of directly comparing different choices of q regarding the quality of their approximation. This however does not mean that the computation of Equation (4.13) can be done exactly, as we will resort to Monte Carlo approximations in the next section. One have to keep in mind, however, that we will not leverage MCMC-like methods. Simply perform expectations estimation using MC sampling.

Equation (4.13) has three main components,

- The first term ensure the objective satisfies the properties of a divergence. D_{sAB} is always positive and it is equal to 0 if and only if $p = q$.

- The second element and the weighting of the ratio $p(\theta,\mathbf{X})/q(\theta)$ in the third element by $q(\theta)^{\alpha+\beta-1}$ control the sensibility to outliers as seen in Section 4.3.4, by setting $\lambda = \alpha + \beta$ to small values below 2, one can achieve robustness to outliers whilst maintaining the efficiency of the objective.

- The scaling on the ratio $p(\theta,\mathbf{X})/q(\theta)$ by a power β in the last element is similar to the bound objective of [158] and favours the mass-covering property.

4.4.2 Optimisation framework

Unfortunately, in general the objective defined in Equation (4.13) still remains intractable and further approximations need to be made. As observed in Section 4.3.3, the sAB-divergence has a form very similar to the Rényi divergence, so we here use the same approximations as in [158]. However, theoretically this objective could be used with any optimisation method as long we are able to compute $p(\theta, \mathbf{X})$ and $q(\theta)$ independently (i.e. not computing the ratio of the two).

To simplify the computation of the objective, a simple Monte Carlo (MC) method is deployed, which uses finite samples $\theta_k \sim q(\theta)$, $k = 1, \dots, K$ to approximate $D_{sAB}^{\alpha,\beta} \approx \hat{D}_{sAB}^{\alpha,\beta,K}$,

$$\begin{aligned}
&\hat{D}_{sAB}^{\alpha,\beta,K}(q(.)||p(.|\mathbf{X})) \\
&= \frac{1}{\alpha(\alpha+\beta)} \log \frac{1}{K} \sum_{k=1}^{K} \frac{p(\theta_k, \mathbf{X})^{\alpha+\beta}}{q(\theta_k|\mathbf{X})} \\
&+ \frac{1}{\beta(\alpha+\beta)} \log \frac{1}{K} \sum_{k=1}^{K} q(\theta_k|\mathbf{X})^{\alpha+\beta-1} \\
&- \frac{1}{\alpha\beta} \log \frac{1}{K} \sum_{k=1}^{K} \left[q(\theta_k|\mathbf{X})^{\alpha+\beta-1} \left(\frac{p(\theta_k, \mathbf{x})}{q(\theta_k|\mathbf{X})} \right)^{\beta} \right].
\end{aligned} \tag{4.14}$$

We also use the reparametrization trick [125], along with more robust gradient descent based optimisation methods as explained in the next section.

Equation 4.14 describe a complex optimisation objective, with potentially high variance. Empirically, however, we observe that the variance of the optimisation target is low enough to yield satisfying experimental results for sensible values of the pair (α, β). Furthermore this variational objective is, in essence, very similar to the Réni variational objective introduced by Li and Turner [158] whose variance was also small enough even with a reasonable number of samples.

4.5 Experiments

To demonstrate the advantages of the sAB-divergence over a simpler objective, we use it to train variational models on regression tasks on both synthetic and real dataset corrupted with outliers. The following experiments have been implemented using *tensorflow* [152] and *Edward* [161] and the code is publicly available at `https://github.com/jbregli/edward/tree/ab_divergence`.

4.5.1 Regression on synthetic dataset

First, similarly to [162], we fit a Bayesian linear regression model [116] to a two dimensional toy dataset where 5% of the data points are corrupted and observe how the generalisation performances are affected for various training objectives on a non corrupted test set. We use a fully factorised Gaussian approximation to the true posterior $q(\theta)$. A detailed experimental setup is provided in Appendix .6. In such a regression setup one could see outliers as rare events not fitting the main trend of the data.

The mean of the predictive distributions for various values of (α,β) are displayed in Figure 4.5 and Table 4.1. As expected, the network trained with standard VI is highly sensitive to outliers and thus has poor predictive abilities at test time, where contamination did not happen. On the other end, when trained with $(\lambda,\beta) = (1.8, 0.8)$ —for this values the sAB-divergence is equivalent to a gamma distribution set up to be robust to outliers—, the predictive distribution ignores the corrupted values. More complex behaviour can be obtained by tuning the values of the pair (α,β) but only yield little improvement on such a simple problem.

4.5.2 Image classification with outliers

In this section we consider training a Bayesian Neural Network [117] for multi-class classification on data where some training input have been mislabelled. To do so we use MNIST [45] and randomly flip the label associated to an input with probability $p = 0.1$. The objective here is not

(λ,β)	MAE	MSE
(1,0,0.0) (KL)	0.58 ± 0.001	0.53 ± 0.003
(1.0,0.3) (Renyi)	0.58 ± 0.003	0.51 ± 0.007
(1.8,0.8) (Gamma)	0.34 ± 0.025	0.21 ± 0.030
$(\mathbf{1.9},-\mathbf{0.3})$ **(sAB)**	$\mathbf{0.34 \pm 0.025}$	$\mathbf{0.21 \pm 0.030}$

Table 4.1: Average Mean Square Error and Mean Absolute Error over 40 regression experiments on the same toy dataset where the training data contain a 5% proportion of corrupted values.

to reach a new state of the art accuracy on the non corrupted test set but rather to show how careful selection of the meta-parameters (α,β) allows to limit the drop in accuracy compared to the same model trained on non corrupted data.

The network used here has one fully connected stochastic layer between the input and the output and reaches a baseline of 92.3% accuracy when trained on without outliers using standard VI. The detailed results can be observed in Table 4.2. As expected from the results in Section 4.3.4, the KL objective does not cope well with outliers and we observe a huge drop in performance when the same network is trained using partly mislabelled data. Exact accuracy figures are reported in Table 4.2. Again a gamma-divergence —i.e. sAB objective with $\alpha = 1$ and $\beta \in \mathbb{R}$— provides a good robustness to outliers. However even better testing accuracy can be achieved, using an objective non described by one of the special cases (i.e. $(\alpha,\beta) = (1.1,-0.3)$) For some set of parameters the sAB-objective offers classification accuracy much closer to the baseline.

4.5.3 UCI datasets regression

In this section, we show that cross validation can be used to fine-tune the parameters (α,β) to outperform standard variational inference with a KL-objective.

We use here a Bayesian neural network regression model [117] with Gaussian likelihood on datasets collected from the UCI dataset repository

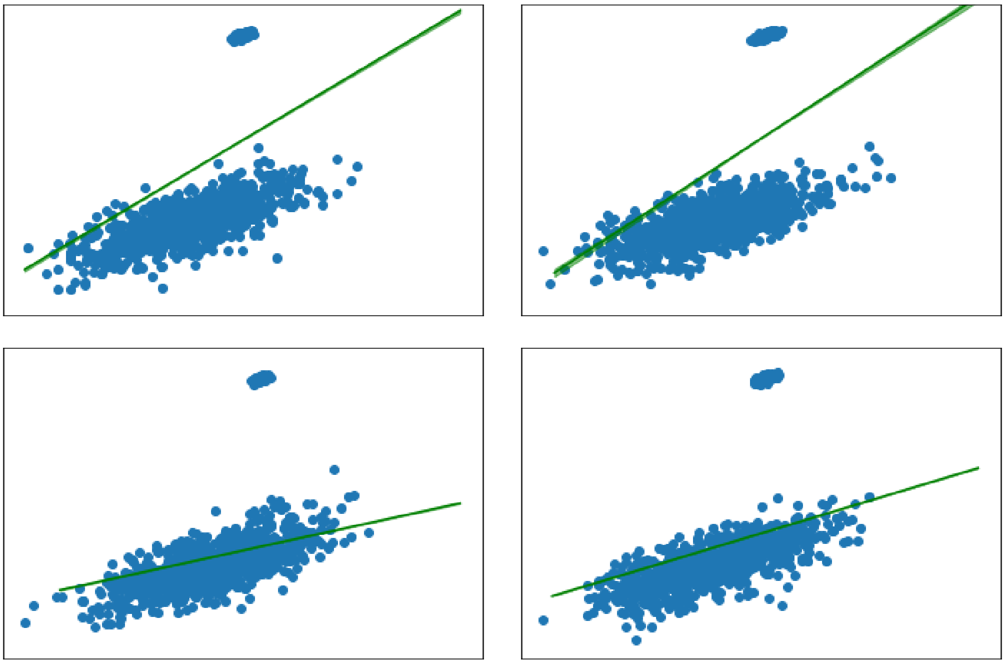

Figure 4.5: Bayesian linear regression fitted to a dataset containing outliers using several sAB objectives. The parameters α and β can be used to ensure robustness to outliers.
Top Left: $\lambda = 1.0$ $\beta = 0.0$.
Top Right: $\lambda = 1.0$ $\beta = 0.3$.
Bottom Left: $\lambda = 1.8$ $\beta = 0.8$.
Bottom Right: $\lambda = 1.9$ $\beta = -0.3$.

[127]. We also artificially corrupt part of the outputs in the training data to test the influence of outliers.

For all the experiments, we use a two-layers neural network with 50 hidden units with ReLUs activation functions. We use a fully factorised Gaussian approximation to the true posterior $q(\theta)$. Independent standard Gaussian priors are given to each of the network weights. The model is optimised using ADAM [139] with learning rate of 0.01 and the standard settings for the other parameters. We perform nested cross-validations [95] where the inner validation is used to select the optimal parameters α and β within the $[-0.5, 2.5] \times [-1.5, 1.5]$ (with step 0.25). Table 4.3 reports the Root Mean Squared Error (RMSE) for the two best pairs (α, β) and for the KL (i.e. $(\alpha, \beta) = (1, 0)$).

In the case of uncorrupted data, KL-divergence is often the best choice

(α, β)	p_{outliers}	accuracy
BASELINE (KL)	0%	92.30%
(1,0,0.0) (KL)	5%	90.11%
$(1.3, -0.4)$ (sAB)	5%	90.32%
$(1.3, -0.4)$ (sAB)	5%	90.29%
(1,0,0.0) (KL)	10%	83.08%
(0.6,0.4) (Renyi)	10%	83.18%
$(1.3, -0.4)$ (Gamma)	10%	88.58%
$(1.0, -0.3)$ (sAB)	10%	88.96%
$(\mathbf{1.1}, \mathbf{-0.3})$ **(sAB)**	**10%**	**89.35%**

Table 4.2: .
Classification accuracy of a one layer BNN trained on data corrupted by p_{outliers}% of mislabeled data points.

of objective though other set of values for (α, β) geared toward mode seeking can yield comparable predictive performances. As expected when contaminated with outliers, a carefully selected set of parameters such that $\alpha + \beta < 2$ allows to achieve better generalisation performances on a non corrupted test set compared to VI with KL. In most of the cases —with and without outliers— the best test score is achieved with $\beta < 1$, corresponding to a mode-seeking type of objective.

4.6 Conclusion

We introduced the extended sAB divergence and its associated variational objective. This objective minimise directly the divergence and does not require to define an equivalent objective via a lower bound. Furthermore this family of divergence covers most of the already known methods and extend them into a more general framework which taps into the growing literature of Monte Carlo methods for complex variational objectives. As the resulting objective functions are not bounds, they provide a way of directly comparing different approximating posterior families. This flexibility is, however coming at the price of a more complex and harder to optimise objective. Successful approximation relies on the Monte Carlo error to be not too difficult to control. Empirically we show that this is feasible for

$(\alpha+\beta,\beta)$	RMSE
Boston housing - $p_{\text{outliers}} = 0\%$	
$(\mathbf{1,0,0.0})$ **(KL)**	$\mathbf{0.99 \pm 0.031}$
$(1.0,0.25)$ (sAB)	1.01 ± 0.015
$(0.0,-0.75)$ (sAB)	1.03 ± 0.021
Boston housing - $p_{\text{outliers}} = 10\%$	
$(1,0,0.0)$ (KL)	1.13 ± 0.043
$(\mathbf{1.25,-0.5})$ **(sAB)**	$\mathbf{1.07 \pm 0.016}$
$(1.75,-0.25)$ (sAB)	1.12 ± 0.029
Concrete - $p_{\text{outliers}} = 0\%$	
$(1,0,0.0)$ (KL)	1.01 ± 0.002
$(\mathbf{1.0,-1.0})$ **(sAB)**	$\mathbf{0.99 \pm 0.001}$
$(1.5,-0.5)$ (sAB)	1.02 ± 0.003
Concrete - $p_{\text{outliers}} = 10\%$	
$(1,0,0.0)$ (KL)	1.16 ± 0.002
$(\mathbf{1.5,-0.25})$ **(sAB)**	$\mathbf{1.07 \pm 0.008}$
$(1.25,-0.5)$ (sAB)	1.08 ± 0.003
Yacht - $p_{\text{outliers}} = 0\%$	
$(\mathbf{1,0,0.0})$ **(KL)**	$\mathbf{0.98 \pm 0.021}$
$(1.0,0.5)$ (sAB)	1.00 ± 0.011
$(1.0,-1.0)$ (sAB)	1.01 ± 0.003
Yacht - $p_{\text{outliers}} = 10\%$	
$(1,0,0.0)$ (KL)	1.09 ± 0.025
$(\mathbf{1.25,-0.25})$ **(sAB)**	$\mathbf{1.05 \pm 0.011}$
$(1.75,-0.5)$ (sAB)	1.06 ± 0.017

Table 4.3: Regression accuracy of a two layer Bayesian neural network trained on datasets from the UCI bank of datasets with corrupted by $p_{\text{outliers}}\%$ training points. The flexibility offered by the sAB-objective allows us to outperform KL-VI in most of the cases where there is noise contamination.

sensible values of the pair (α,β).

We show that the two governing meta-parameters of the objective allow to control independently the mass-covering character and the robustness of the approximation. Experimental results point out the interest of this flexible objective over the already existing ones for data corrupted with outliers. We also show that the variance of the Monte-Carlo estimate of the objective is controlled and quickly becomes negligible with the number of samples.

In Chapter 7, we will leverage the flexibility of the AB variational objective to fit complex graphical models to limited amount of datapoints.

Part III

Roof-Edge hidden Markov Random Field

Chapter 5

Roof-Edge hidden Markov Random Field

Semi-local Hurst estimation is considered by incorporating a Markov random field model to constrain a wavelet-based pointwise Hurst estimator. This results in an estimator which is able to exploit the spatial regularities of a piecewise parametric varying Hurst parameter. The pointwise estimates are jointly inferred along with the parametric form of the underlying Hurst function which characterises how the Hurst parameter varies deterministically over the spatial support of the data. Unlike recent Hurst regularisation methods, the proposed approach is flexible in that arbitrary parametric forms can be considered and is extensible in as much as the associated gradient descent algorithm can accommodate a broad class of distributional assumptions without any significant modifications. The potential benefits of the approach are illustrated with simulations of various first-order polynomial forms. This shows that such a regularisation method can be used to infer te value of a specific feature of the signal more robustly.

5.1 Introduction

The Hurst parameter determines the spectral decay rate of a process with a power-law spectrum. Since such a simple relationship is ubiquitous in many signal and image processing areas and beyond [58, 63] Hurst estimation continues to enjoy many, and disparate, applications including Finance [126], signal/image denoising [83], clutter suppression [99], segmentation [122], the analysis of ECG signals [124, 134], internet traffic flow [58], image texture [53], and turbulence data [85]. Furthermore, following the work of [99], one can use the Hurst coefficients to perform ripple suppression (see Section 6.6.2) and improve the efficiency of mine detection systems.

The interconnection between wavelets and self-similar processes is a powerful, if not, surprising one. The self-similarity explicitly built into the wavelet basis functions via the two-scale, or refinement, relations provides a natural representation in which to study processes that exhibit power-law behaviour. However, the localised nature of wavelets also facilitates a localised estimation of the Hurst parameter.

Although there are works, such as those based on the multifractal formalism [82, 88], that describe how regularity varies across an image, less attention has been paid to the case where the main interest is to obtain pointwise estimates of a Hurst parameter that is allowed to vary as a smooth, deterministic function. Such a scenario could, for example, present itself in image processing when the texture of an object of interest varies gradually over its spatial support in some assumed manner. In turn this would facilitate tasks such as feature extraction, segmentation, and change detection. Likewise, existing adaptive denoising methods, which are currently based on a piecewise constant Hurst parameter [140], could also be extended to include more general Hurst functions that vary as piecewise parametric functions.

Since it is reasonable to assume that an image of interest may comprise multiple textures, it is appropriate to consider a piecewise smoothly varying Hurst parameter $H = H(\mathbf{r})$, for $\mathbf{r}$ over some subregion of $\mathbb{R}^2$. Furthermore, we let the way in which this Hurst function varies over space be governed by some parametric form $H = \phi(\mathbf{r}; \boldsymbol{\theta})$ with model parameters $\boldsymbol{\theta}$. We would expect these parameters to be fairly constant over certain subregions of the image domain where the image texture is homogeneous. We allow the spatial support to accommodate multiple textures with a suitable partitioning of disjoint subregions. In each subregion, the $\boldsymbol{\theta}$ are assumed constant (or have very small, smooth variations). However, between subregion boundaries, it is allowed to change arbitrarily. As a consequence the Hurst parameter itself will vary smoothly inside a partition and vary arbitrarily across the respective subregions. We here propose a model and inference scheme that exploits this piecewise parametric outlook. The framework utilises a Markov random field prior to constrain, or penalise, the magnitude of parameter variation over the image.

Spatial regularisation of Hurst estimation has been recently considered as a means to exploit prior knowledge about the spatial smoothness of the Hurst parameter [140]. However, the method was based on the generalised lasso and assumed only a piecewise constant varying Hurst parameter. In contrast our model, and corresponding gradient-descent-like algorithm, are more flexible. The framework can accommodate many different kinds of distributional assumptions and arbitrary models that describe how the Hurst parameter varies deterministically in space. On the other hand, the generalised lasso Hurst estimator simply penalises the ℓ_1-norm of the Hurst parameter spatial derivatives (of some specified order). Therefore, along with a fixed Gaussian assumption on the data, the spatial derivatives of the Hurst parameter are assumed to be Laplacian and it is difficult to incorporate other distributional assumptions without making wholesale changes to the inference scheme. Other assumptions would necessitate a change in inference strategy (if one existed). Furthermore, unlike the method proposed here, the lasso inference does not obtain any estimate of the underlying parametric form of the Hurst 'function' [64].

In Section 5.2 we present the requisite background of wavelet-based Hurst estimation and Li's piecewise (roof-edge) parameterised Markov random field model [62]. We fuse these two concepts in Section 5.3, propose our parameterised MRF Hurst estimation framework, and describe the inferential machinery. In Section 5.4 we perform estimation on a selection of simulated imagery where the Hurst parameter is varied according to several first-order polynomial forms. Each one manifests unique roof-like edges in the Hurst parameter and presents different challenges to the estimators. We draw conclusions in Section 5.5.

5.2 Background

The Hurst parameter controls the spectral slope of a self-similar stochastic process which obeys a power-law relationship. Myriad estimation approaches exist [63]. We here follow the popular wavelet-based framework [59].

5.2.1 Wavelet-based Hurst estimation

Consider a stochastic field z defined on a subregion of $\mathbb{R}^2$ with weak statistical self-similarity namely $\mathbb{E}z(\lambda\cdot) = \lambda^H\mathbb{E}z$ and $\mathbb{E}z(\lambda\mathbf{r})z(\lambda\cdot) = \lambda^{2H}\mathbb{E}z(\mathbf{r})z(\cdot)$. Then, it is well known (see e.g. [100]), that

$$\mathbb{E}\big|(\mathcal{W}z)(\cdot\,;k,\alpha)\big|^2 \propto 2^{2k(H+1)} \tag{5.1}$$

where $\mathcal{W}$ is the wavelet transform operator defined by $(\mathcal{W}z)(\mathbf{r};k,\alpha) := 2^{-k}\langle z, \psi_\alpha(2^{-k}\cdot - \mathbf{r})\rangle$, with wavelet ψ defined over space $\mathbf{r}$, orientation α, and kth finest scale level.

In practice the expectation in Equation (5.1) is approximated by the sample second moment of the wavelet coefficients magnitudes. When the Hurst parameter varies over space it is still possible to estimate the slope by simply using the squared magnitude of the wavelet coefficients. This pointwise estimate, $E_{k,\alpha}(\cdot) := \left|(\mathcal{W}z)(\cdot;k,\alpha)\right|^2$, approximately satisfies the power-law, namely $E_{k,\alpha} \sim 2^{2k(H_\alpha(\mathbf{r})+1)}$. Estimation of H is then performed by taking the log of both sides and regressing the log wavelet magnitude on scale. The Hurst parameter is then easily obtained from the slope of the regression line. Generally, H can also vary with orientation too. In this case, one can perform separate regressions in each direction as appropriate (cf. [99, 128]). Alternatively, if we assume that the Hurst parameter is isotropic there are two main options. Firstly, one could perform separate regressions over the different orientations and then compute the average. Secondly, one could perform one regression over the orientation -averaged wavelet magnitude. As such, without loss of generality, we can drop any orientation notation and write the log wavelet magnitudes about the spatial location $\mathbf{r}_i$ as $\gamma_k[i]$ where $i \in \mathbb{I}$ simply indexes the spatial locations or 'sites' in Markov random field modelling parlance. This furnishes the set of equations $\boldsymbol{\gamma}[i] = \mathbf{A}\boldsymbol{\beta}[i]$, with

$$\boldsymbol{\gamma}[i] = \begin{bmatrix} \gamma_{k_-}[i] \\ \vdots \\ \gamma_{k_+}[i] \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 1 & k_- \\ \vdots & \vdots \\ 1 & k_+ \end{bmatrix}, \quad \boldsymbol{\beta}[i] = \begin{bmatrix} \beta_1[i] \\ \beta_2[i] \end{bmatrix},$$

where only the k_-th to the k_+th finest wavelet scale levels are used— the coarsest levels will give poor spatial location and the finest levels will typically have low signal-to-noise ratio. Solving in the least-squares sense gives the ordinary least squares (OLS) estimate

$$\hat{\boldsymbol{\beta}}[i] := \operatorname{argmin} \left\|\boldsymbol{\gamma}[i] - \mathbf{A}\boldsymbol{\beta}[i]\right\|_2 = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\boldsymbol{\gamma}[i],$$

and then the estimate of the Hurst parameter can be recovered from the second element of the $\boldsymbol{\beta}$ vector, namely $\hat{H}(\mathbf{r}_i) = (\hat{\beta}_2[i]/2 - 1)$.

5.2.2 Roof edge model

The roof edge model was introduced by Li [62] as a means to recover piecewise planar surfaces from noisy observations. Assuming that the parameters of the underlying true surface are the same, or similar, over contiguous regions of the spatial domain, a Markov random field prior can be introduced to aid inference. This introduces the notion of a Markovian label field $f = \{f_1, \ldots f_m\}$ with the property that, conditioned on its neighbours, the field at a site is conditionally independent of all other sites. This allows us to write the full conditional of f as the local conditional: $\mathbb{P}(f_i|f_{-i}) = \mathbb{P}(f_i|f_{\mathbb{I}_i})$.

As a consequence of the Hammersley-Clifford Theorem, the joint prior takes the form $\mathbb{P}(f) \propto \exp(-U(f))$. The prior energy term $U(f)$ therefore determines the manner in which spatially incoherent label configurations are penalised. Given observations d, this is counter-balanced to some extent by the likelihood energy $U(d|f)$. By Bayes rule the posterior $\mathbb{P}(f|d)$ has (posterior) energy $U(f|d) = U(d|f) + U(f)$. Observations are assumed to follow some parametric surface, corrupted by noise $d_i := \phi(\mathbf{r}_i; \boldsymbol{\theta}_i) + \epsilon_i$ but where the underlying labels of the parameters $\boldsymbol{\theta}_i$ satisfy the Markov model. For our problem we exploit this to impose piecewise smooth constraints on the Hurst function model parameters $\boldsymbol{\theta}_i$ and, as a consequence, on the Hurst parameter itself. In Li's basic roof edge model, $\phi(\mathbf{r}_i; \boldsymbol{\theta}_i) := \boldsymbol{\theta}_i^\top \boldsymbol{\rho}_i$, with $\boldsymbol{\theta}_i^\top := (\theta_0[i], \theta_1[i], \theta_2[i])$, and $\boldsymbol{\rho}_i^\top := (1, x_i, y_i)$ but higher-order polynomials can easily be accommodated.

Given data d, the distributional assumptions of ϵ (i.e. the likelihood), and our prior model of the underlying configuration label field (the prior), the goal then is to estimate the maximum a posteriori, namely $f^* = \operatorname{argmin}_f U(f|d)$.

5.3 Parameterised MRF Hurst estimation

We assume that the Hurst parameter varies as a piecewise parametric function. The parameters which describe how H varies are therefore assumed to change little within a given subregion. However, the parameters may change at the boundaries between subregions. We therefore introduce a Markov random (label) field to assign sites and model parameters to labels.

5.3.1 Markov random field model

The ordinary least squares estimate $\hat{\boldsymbol{\beta}}[i] = (\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top \gamma[i]$ gives rise to a 'noisy' version of the true value of $\boldsymbol{\beta}$, namely $\hat{\boldsymbol{\beta}}[i] = \boldsymbol{\beta}[i] + \boldsymbol{\epsilon}[i]$. For notational convenience, and without generality, denote the observed spectral log-slope (i.e. $\hat{\beta}_2[i]$) as $\hat{\beta}[i]$. Assume that the true spectral slope follows some parametric model: $\beta[i] = \phi(\mathbf{r}_i;\boldsymbol{\theta}_i)$, where $\mathbf{r}_i = (x_i,y_i)$ determines pixel location and where $\boldsymbol{\theta}_i$ denotes the parameters of β. Then, assuming that the noise is iid Gaussian [1] $\epsilon_i \overset{iid}{\sim} \mathcal{N}(0,\sigma^2)$ we have the likelihood energy

$$U(\hat{\beta}|f) = \lambda \sum_{i\in\mathbb{I}} \left(\hat{\beta}[i] - \phi(\mathbf{r}_i;\boldsymbol{\theta}_i)\right)^2 \tag{5.2}$$

Exploiting the Markov structure of the label field, we use a prior energy function of the same form as Li [34, 62]:

$$U(f) = \sum_{i\in\mathbb{I}} \sum_{i'\in\mathbb{I}_i} g\left(\left\|\mathbf{W}(\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i'})\right\|_2\right),$$

where $\mathbb{I}_i$ is the neighbourhood of site i and the diagonal weight matrix $\mathbf{W}$ provides the option to penalise the lack of smoothness of each parameter to different degrees. Li [34] describe the conditions that g has to satisfy. The key to edge preserving property is to define a g such that $lim_{z\to\infty}g(z) = C$ where C is a constant greater or equal to 0. We choose$g(z) = \ln(1+z^2)$. The

[1] Strictly speaking there exists a small bias term due to non-linearities introduced by the log function [58] but we neglect them here and leave such considerations as further work

choice of ϕ determines the complexity with which the underlying Hurst parameter is assumed to vary. In contrast to the work of Nafornita et al [140], who considered a piecewise constant Hurst, we here consider a Hurst parameter which varies as a piecewise order-1 polynomial. However, it should be noted that higher-order terms can easily be accommodated by recalling that $\phi(\mathbf{r}_i;\boldsymbol{\theta}_i) = \boldsymbol{\theta}_i^\top \boldsymbol{\rho}_i$ and noting that the vectors $\boldsymbol{\rho}_i$ and $\boldsymbol{\theta}_i$ can be extended accordingly. For example higher order products $(x^{p_0}y^{p_1})_{p_0,p_1}$ can be concatenated on to the vector $\boldsymbol{\rho}_i$ for suitable ranges of p_0 and p_1.

In that model one can think of β as a feature extracted from the preprocessing of the original data. We then craft a HMRF roof-edge model for this specific feature.

5.3.2 Inference

Given the least-squares estimate of the Hurst parameter and the Markov random field roof-edge piecewise parametric model, we find the Maximum A Posteriori (MAP) solution to the problem, namely

$$U(f|\hat{\beta}) := U(\hat{\beta}|f) + U(f).$$

This is an unconstrained optimization problem and can be solved using a gradient-descent-like algorithm. The derivatives with respect to the model parameters can be expressed analytically as

$$\begin{aligned}\frac{1}{2}\frac{\partial U(f|\hat{\beta})}{\partial \boldsymbol{\theta}_i} = -\lambda\left(\hat{\beta}[i] - \phi(\mathbf{r}_i;\boldsymbol{\theta}_i)\right)\boldsymbol{\rho}_i \\ + \sum_{i'\in\mathbb{I}_i} g'\left(\left\|\mathbf{W}(\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i'})\right\|_2\right)\mathbf{W}(\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i'}),\end{aligned}$$

where g' is the derivative of g with regard to its parameters. In our implementation we use the unconstrained version of the BFGS algorithm proposed by Yuan [25] instead of a simple gradient descent. It is a variation of second order newton's method where the Hessian matrix is estimated rather than computed at every steps. The optimization procedure is detailed in Algorithm 1. Therein, for a given step ℓ, we define $\boldsymbol{\theta}^{(\ell)} := \big(\boldsymbol{\theta}_i^{(\ell)}\big)_{i=1}^m \in \mathbb{R}^{3\times m}$, $\boldsymbol{\rho}^{(\ell)} := \big(\boldsymbol{\rho}_i^{(\ell)}\big)_{i=1}^m \in \mathbb{R}^{3\times m}$, $\mathbf{B}_{(\ell)} := \big(\mathbf{B}_{(\ell)}[i]\big)_{i=1}^m \in \mathbb{R}^{3\times 3\times m}$ and where the products between the elements are defined pixel-wise, namely: $\boldsymbol{\theta}_{(\ell)}^\top \boldsymbol{\rho}^{(\ell)} = \big(\boldsymbol{\theta}_i^{(\ell)\top} \boldsymbol{\rho}_i^{(\ell)}\big)_{i=1}^m$ and m is the number of pixels, or sites.

The meta-parameter λ in Equation (5.2) is used to control the importance of the likelihood over the prior. The weights in the diagonal matrix allows variable emphasis to be placed on each of the model parameters $\boldsymbol{\theta}_i$.

```
Meta-parameters:
$\lambda$, $\mathbf{W}$
Initialization:
$\ell = 0$ and $\mathbf{B}_{(0)}[i] = \mathbf{I}_3 \; \forall i \in [\![1, m]\!]$
while convergence do
    - Descent direction:
    $\mathbf{p}^{(\ell)} = -\mathbf{B}_{(\ell)}^{-1} \nabla U(f|\hat{\beta}^{(\ell)})$
    - Optimal step in the direction $\mathbf{p}^{(\ell)}$:
    $\mu^{(\ell)} = \operatorname{argmin}_{\mu\in\mathbb{R}} \left[U(f|(\boldsymbol{\theta}_{(\ell)} + \mu\mathbf{p}^{(\ell)})^\top \boldsymbol{\rho})\right]$
    $\boldsymbol{\theta}^{(\ell+1)} = \boldsymbol{\theta}^{(\ell)} + \mu^{(\ell)}\mathbf{p}^{(\ell)}$
    $\hat{\beta}^{(\ell+1)} = \boldsymbol{\theta}^{(\ell)\top}\boldsymbol{\rho}$
    - Hessian matrix estimate:
    $\boldsymbol{\eta}^{(\ell)} = \nabla U(f|\hat{\beta}^{(\ell+1)}) - \nabla U(f|\hat{\beta}^{(\ell)})$
    $\mathbf{B}_{(\ell+1)} = \mathbf{B}_{(\ell)} + \frac{\boldsymbol{\eta}^{(\ell)}\boldsymbol{\eta}^{(\ell)\top}}{\mu^{(\ell)}\boldsymbol{\eta}^{(\ell)\top}\mathbf{p}^{(\ell)}} - \frac{\mathbf{B}_{(\ell)}\mathbf{p}^{(\ell)}\mathbf{p}^{(\ell)\top}\mathbf{B}_{(\ell)}}{\mathbf{p}^{(\ell)\top}\mathbf{B}_{(\ell)}\mathbf{p}^{(\ell)}}$
    $l = l + 1$
end
```

Algorithm 1: Minimization of the posterior energy

5.4 Experiments

Experiments were carried out to test the utility of the proposed method for scenarios where the Hurst parameter varied as a first-order polynomial surface. In particular, the behaviour of the estimator was investigated when H varied as a selection of different roof-edge-like functions. These might model the way in which a texture becomes gradually smoother or rougher

Table 5.1: Mean absolute error (and standard deviation) of the OLS and MRF Hurst estimators

Type	OLS	MRF
1	0.2145 (0.1502)	**0.1335** (0.0955)
2	0.1906 (0.1355)	**0.1330** (0.0983)
3	0.1874 (0.1376)	**0.1286** (0.1038)
4	0.1495 (0.1174)	**0.1117** (0.0928)

in space. The second column of Figures (5.1, 5.2, 5.3, 5.4) illustrates the different roof-edge shapes. For simplicity, we let H vary as a function of its ℓ_∞-norm distance from the centre of the image, namely $H(\mathbf{r}) = h(\|\mathbf{r}\|)$. The function h is a projection of the Hurst values onto the ℓ_∞-ball; we shall refer to it as the *Hurst signature*. The signatures of the four different roof-types are plotted in the first column of Figures (5.1, 5.2, 5.3, 5.4).

5.4.1 Simulation

The data was synthesised by adapting the incremental Fourier synthesis approach of Kaplan and Kuo [37], as implemented in the Fraclab toolbox [144]. We partition the spatial domain into disjoint ℓ_∞ tori: $\mathbb{I}[j] := \{i \in \mathbb{I} \colon \|\mathbf{r}_i\|_\infty \in [j\Delta r, (j+1)\Delta r)\}$. Then, fractional Brownian surfaces are simulated which have a Hurst parameter of $h(j\Delta r)$ on the region $\mathbb{I}[j]$ and which take zero values elsewhere (and which all have the same global white noise driving process). Finally, the surfaces are simply summed over all j. The result is a fractional Brownian surface with a piecewise, order-one polynomial, varying Hurst parameter.

5.4.2 Hurst estimation

Hurst estimation was performed on the four image types 'Hip', 'Pavillion', 'Gambrel', and 'Bonnet' illustrated in Figures (5.1, 5.2, 5.3, 5.4). Ordinary least-squares estimates were used as a baseline for our proposed MRF-based approach although we note that a direct comparison is not necessarily fair as we were free to select an optimal value of λ in our approach

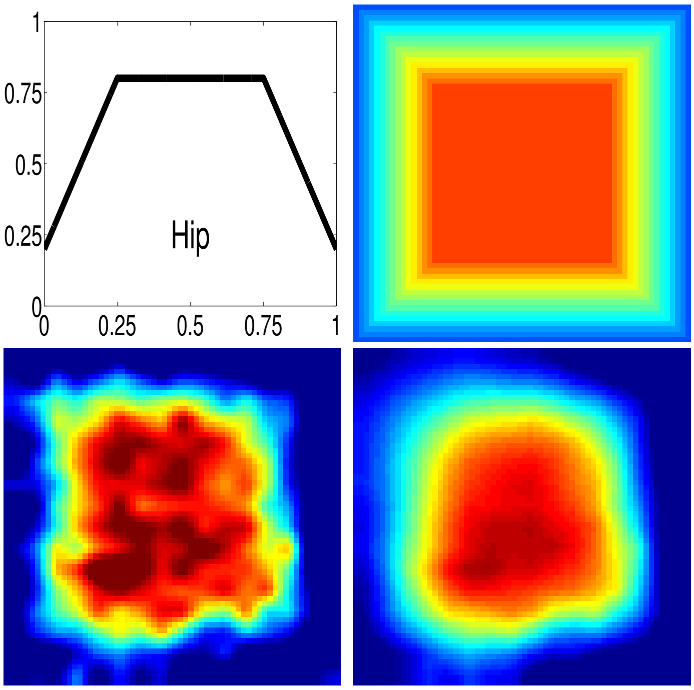

Figure 5.1: Indicative Hurst estimates of the 'Hip' fractional Brownian surfaces.
Top Left: True Hurst projected onto ℓ_∞ ball
Top Right: Spatial map of true Hurst.
Bottom Left: Ordinary Least Square
Bottom Right: Markov Random Field regularisation.

to balance the effects of the prior and likelihood functions. Nevertheless, the comparison does offer some intuition as to some of the advantages that one might buy from the addition of an extra parameter. For example, the bottom left and right plots of Figures (5.1, 5.2, 5.3, 5.4) depict the Hurst parameter estimates from the OLS and MRF methods, respectively. The spatial regularisation, or smoothing, effect of the MRF method can be clearly seen for all edge types.

Experiments were performed over 100 instances of each of the edge types. The value of λ was chosen by testing over a smaller subset of data as 0.001 in all cases. For simplicity, we used equal weights: $\mathbf{W} = \mathbf{I}$. The mean absolute errors are listed in Table (5.1) and the error histograms are plotted in Figure (5.6). The advantage of exploiting the spatial smoothness of the Hurst parameter is evident. However, this advantage is not as marked in the 'Bonnet' image. The reason for this can be seen by inspecting the error

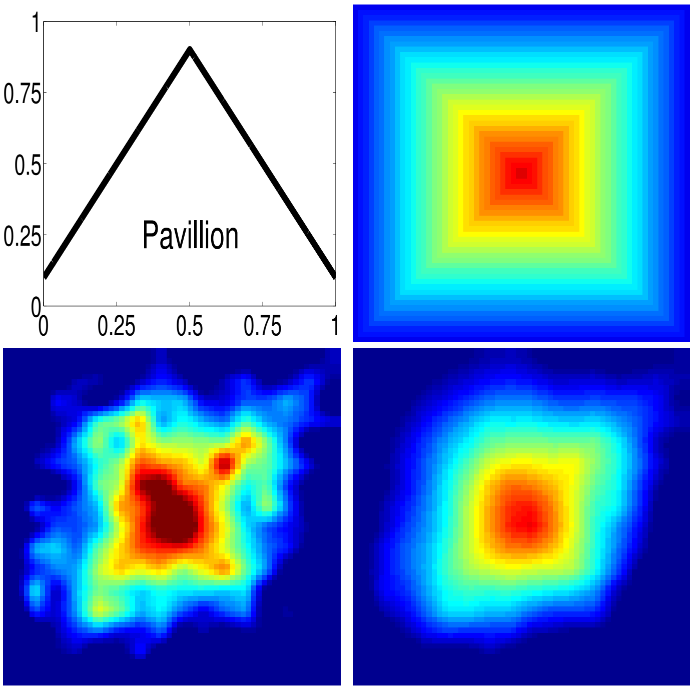

Figure 5.2: Indicative Hurst estimates of the "Pavillon" fractional Brownian surfaces.
Top Left: True Hurst projected onto ℓ_∞ ball
Top Right: Spatial map of true Hurst.
Bottom Left: Ordinary Least Square
Bottom Right: Markov Random Field regularisation.

as a function of the Hurst signature— i.e. the distance from the centre as measured by the ℓ_∞-norm. We see, in Figure (5.5), that the MRF method's tendency to smooth the edge features somewhat is more pronounced when the edge is sharp or concave. Nevertheless, MRF still holds an advantage here because the OLS method overshoots the edge point. For convex ridge shapes, the advantage becomes significant.

5.5 Conclusion

A piecewise parameterised Markov random field was introduced to jointly estimate a spatially regularised pointwise Hurst parameter and the model parameters which govern how it varies over the spatial support. The model is flexible in that the model can easily accommodate other likelihood or prior assumptions without any significant changes in the gradient-descent-

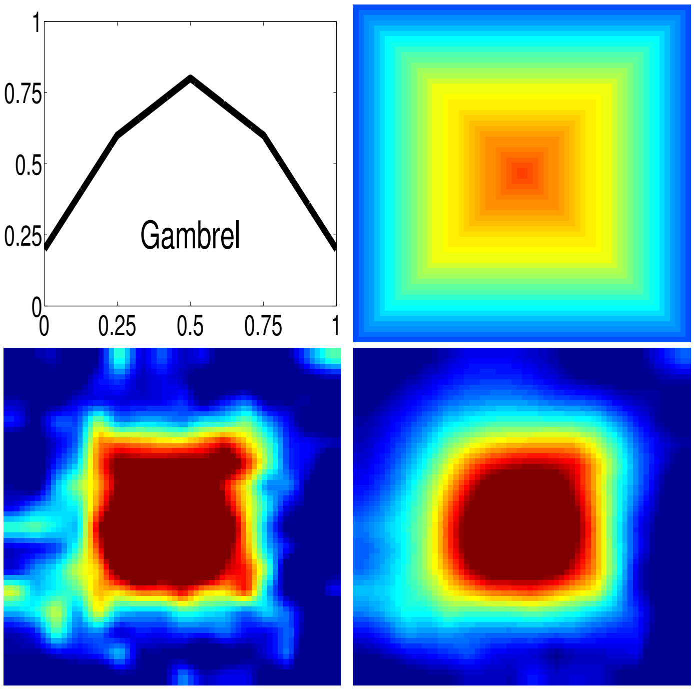

Figure 5.3: Indicative Hurst estimates of the "Grambrel" fractional Brownian surfaces.
Top Left: True Hurst projected onto ℓ_∞ ball
Top Right: Spatial map of true Hurst.
Bottom Left: Ordinary Least Square
Bottom Right: Markov Random Field regularisation.

like inferential machinery. Experiments confirm that the introduction of the Markov random field prior successfully furnishes spatially regularised Hurst estimates with more accuracy than ordinary least squares although this advantage is tempered somewhat when the Hurst function displays concave ridge shapes.

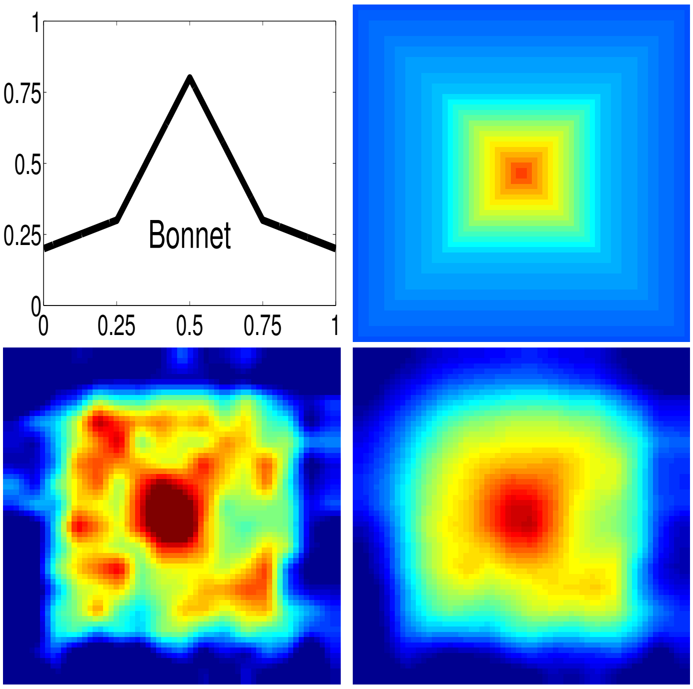

Figure 5.4: Indicative Hurst estimates of the "Bonnet" fractional Brownian surfaces.
Top Left: True Hurst projected onto ℓ_∞ ball
Top Right: Spatial map of true Hurst.
Bottom Left: Ordinary Least Square.
Bottom Right: Markov Random Field regularisation.

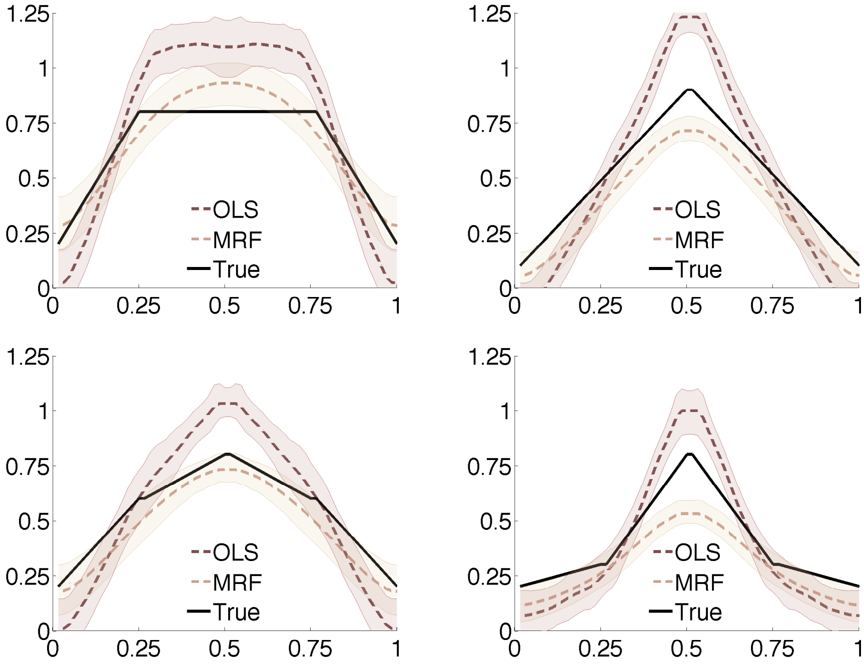

Figure 5.5: Mean Hurst estimates of the four fractional Brownian surfaces projected onto the ℓ_∞ ball. the shaded error bars indicate the upper- and lower-quantiles over all experiments and pixel estimates.
top left: 'hip' profile.
top right: 'pavillon'profile.
bottom left: 'gambrel' profile.
bottom right: 'bonnet' profile.

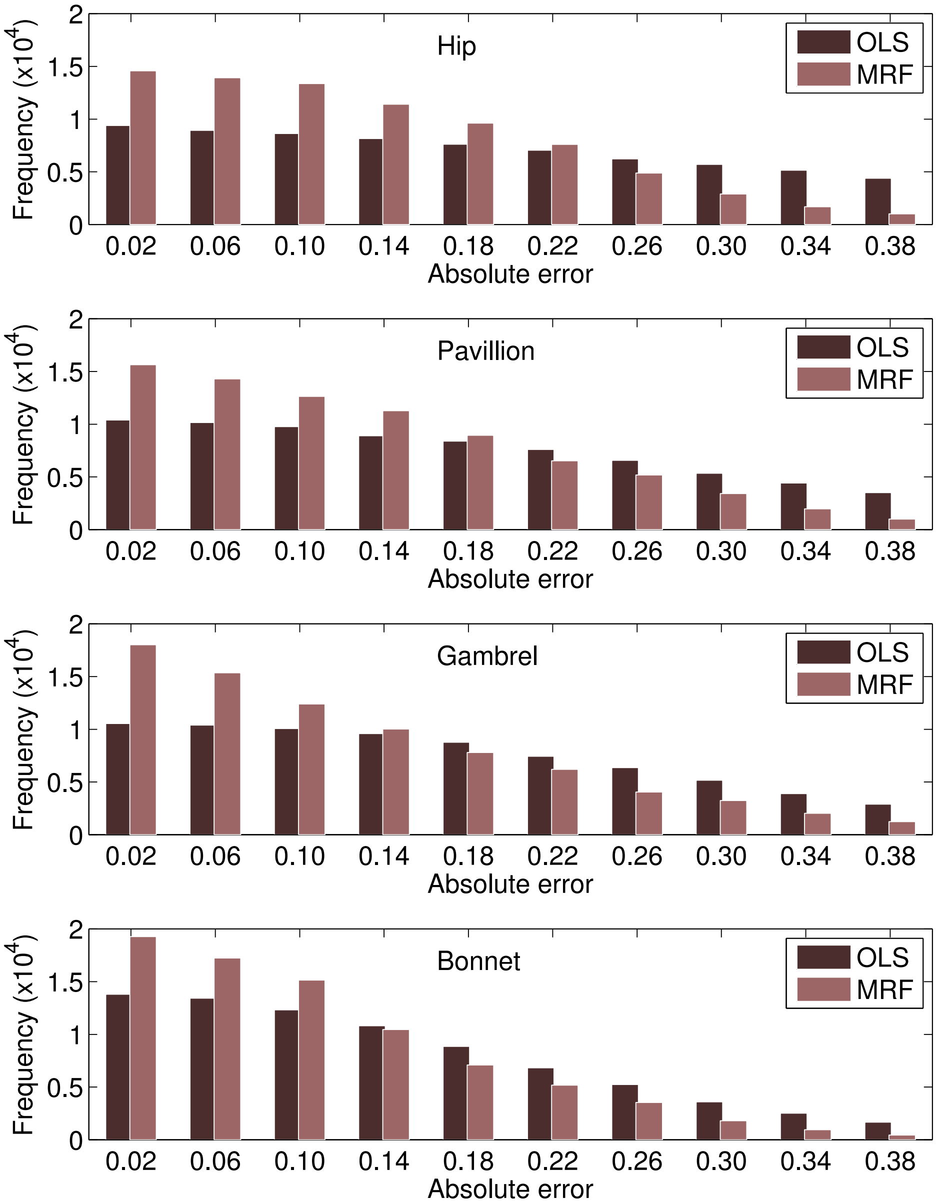

Figure 5.6: Absolute error histograms for the OLS and MRF Hurst estimates over 100 instances of each roof-edge type.
top left: True Hurst projected onto ℓ_∞ ball.
top right: Spatial MAP of the true Hurst.
bottom left: Ordinary Least Square.
bottom right: Markov Random Field regularisation.

Part IV

Scattering Hidden Markov Tree

Chapter 6

Scattering Hidden Markov Tree

In this chapter we combine the rich, overcomplete signal representation afforded by the scattering transform together with a probabilistic graphical model which captures hierarchical dependencies between coefficients at different layers. The wavelet scattering network result in a high-dimensional representation which is translation invariant and stable to deformations whilst preserving informative content. Such properties are achieved by cascading wavelet transform convolutions with non-linear modulus and averaging operators. The network structure and its distributions are described using a Hidden Markov Tree. This yields a generative model for high-dimensional inference and offers a means to perform various inference tasks such as prediction. Our proposed scattering hidden Markov tree displays promising results on classification tasks of complex images in the challenging case where the number of training examples is extremely small.

Chapter outline

In this chapter we introduce the Scattering Hidden Markov Tree, a Bayesian modeling of the Scattering Convolutional network introduced in Section 2.3. In Section 6.1, we provide more detailled motivation for such a model. Section 6.2 introduces some related works. In Section 6.3, we provide a first description of our proposed SHMT model and details the hypothesis needed to develop this model as well as provides some justifications on their validity. We detail the learning and inference algorithms in Section 6.4 and Section 6.5 respectively. Finally Section 6.6 provides some experimental results using our proposed model.

6.1 Introduction

In Section 2.3.7 we have described how the scattering network could be used combined with a support vector machine classifier to achieve competitive classification performance. This approach, however, only provides a Boolean label for each class. Methods to express the output of an SVM as a probability exists [51]. This method is, however, not widely accepted as a true probabilistic approach and shows some theoretical limitations [109]. If one is interested in a true probabilistic model to describe the scattering coefficients, it is quite natural to try expressing them as a probabilistic graphical model. Indeed if one ignores the propagation step from the scattering transform (see Section 2.3.2) the scattering network defines the tree structure displayed in Figure 6.1.

To simplify notations in the remainder of this section, let $\mathcal{T}$ denote the tree structure defined by a scattering convolutional network $ST_{(\psi,J,M,L)}(\cdot)$ restricted to frequency decreasing path of length shorter than M, considering J scales and L orientations. Such a tree $\mathcal{T}$ is depicted in Figure 6.1. Let also I denote the total number of nodes —i.e. scattering coefficients— and let S_i for $i \in [\![0, I-1]\!]$ denote one of the nodes of $\mathcal{T}$ for a given path

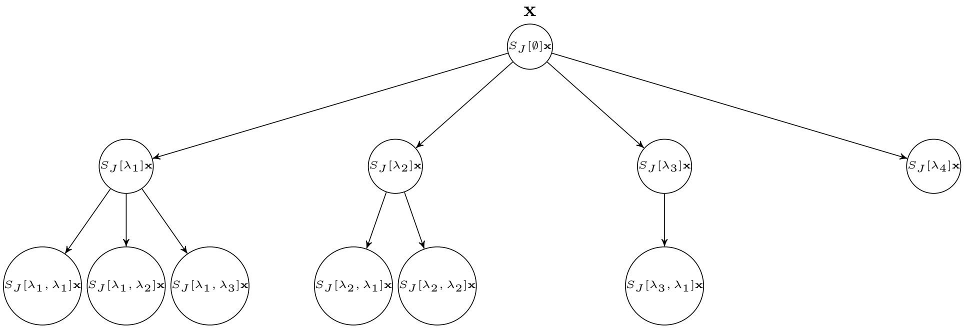
x
$S_J[\emptyset]x$
$S_J[\lambda_1]x$
$S_J[\lambda_2]x$
$S_J[\lambda_3]x$
$S_J[\lambda_4]x$
$S_J[\lambda_1,\lambda_1]x$
$S_J[\lambda_1,\lambda_2]x$
$S_J[\lambda_1,\lambda_3]x$
$S_J[\lambda_2,\lambda_1]x$
$S_J[\lambda_2,\lambda_2]x$
$S_J[\lambda_3,\lambda_1]x$

Figure 6.1: Scattering transform tree with $J = 4$ scales, $L = 1$ orientation and $M = 2$ layers.

$p_i = [\lambda_0 \ldots \lambda_u]$ ($u \in \mathbb{N}$). Note that S_i represents a node and does not depend upon the signal $\mathbf{x}$. For a given signal $\mathbf{x}$, the realisation of the node i for signal $\mathbf{x}$ is denoted by $s_i = S[p_i]\mathbf{x}$. Note also that in the remainder of the document, the shorter notation $i \in \mathcal{T}$ will be used to denote the path p_i to the node i. Let us also use the convention $S_0 = S[\emptyset]$. Finally let ρ_i and $\mathcal{C}_i$ denote respectively the parent of a node i and the set of children of the node i. A node S_i can have no children, in such a case this node is a leaf of the tree.

6.2 Background

We first review the usage of graphical model to describe signal representation methods. Unless stated otherwise, the notation defined in this section will be used throughout this chapter.

Let us consider the case of a hierarchical signal representation, the idea is to assume that it can be efficiently expressed as a graphical model (see Chapter 3). The nodes of this PGM are the representation values —e.g. the wavelet coefficient, the scattering coefficient, the neurons of a neural network...— and the edges can either be aligned with the information flow — i.e. following the signal propagation through the representational structure — or be independant of it.

Models trying to describe directly the correlation across coefficients at different scales have been studied for traditional wavelet transforms [38] but are in conflict with the compression property of the wavelet —i.e. the fact that most wavelet representations are sparse [43]. Thus it seems that a simple one-step Markovian assumption across scales is not satisfying to describe the complex relationship between wavelet or scattering coefficients.

A common approach when a direct Markovian model does not hold is to introduce hidden states and assume the Markovian property across those unobserved values. The observed nodes now are then only dependent on their respective state. This architecture has been adopted to create the SHMT. This model is represented by Figure 6.2. As the scattering transform is closely related to wavelet transform it is not surprising to find similar ideas exploited for wavelet trees. Crouse et al. [43] have developed a model where a hidden Markov tree model is used to model the wavelet coefficients of a standard wavelet trees. Later Kingsbury [57] has adapted Crouse's model to the Dual Wavelet Complex Trees (DWCTs). The resulting hidden Markov tree models provides better classification performance than the Wavelet Hidden Markov Tree (WHMT) as the wavelet used generates a "better" representation of the signal in the sense defined in Section 2.1. Indeed this version can leverage the quasi-translation invariance property of the DWCTs. This improvement in performance due to the quasi-invariance property provides a good motivation for using a hidden Markov tree model on the scattering transform as they have even "better" representational properties (see Section 2.3.3). The parameters of the original WHMT are trained using a version of the Expectation-Maximization adapted to binary hidden Markov trees. However since this learning method suffers from underflowing issues [20], Durand et al. [71]

proposed a smoothed version of the training algorithm preventing this from happening.

We propose an adaptation of those models to create a scattering convolutional hidden Markov tree composed of a set of visible nodes $\{\mathbf{S}_i\}_{i\in\mathcal{T}}$ and a set of hidden node $\{\mathbf{H}_i\}_{i\in\mathcal{T}}$. Both sets are organised in a similar tree structure with the following characteristics, but different nodes.

6.3 SHMT model

In this section we introduce the Scattering Hidden Markov Tree (SHMT) used to model the scattering convolutional network. We also list and argument the modeling assumptions.

6.3.1 Model

The idea behind the SHMT model is to assume that the more detailed representations of the signal are somehow correlated to the less detailed ones from which they are generated. More formally this means that for a signal $\mathbf{x}$, s_i is somehow correlated to $s_{\rho(i)}$.

In its simplest form this assumption implies we can model the scattering network by a Markov tree and assumes

$$P(S_i|\mathcal{T}) = P(S_i|S_{\rho(i)}).$$

Those independence properties are encoded in the graph displayed in Figure 6.1. As seen in Section 6.2, however, modeling directly the correlation is not sensible. It violates the sparsity property of the wavelet coefficients.

One can, however, use hidden coefficients which are themselves Markov independent. Such a modeling of the SCN is represented by Figure 6.2 and it induces the following independence properties,

$$P(H_i|\mathcal{T}) = P(H_i|H_{\rho(i)}),$$

$$P(S_i|\mathcal{T}) = P(S_i|H_i).$$

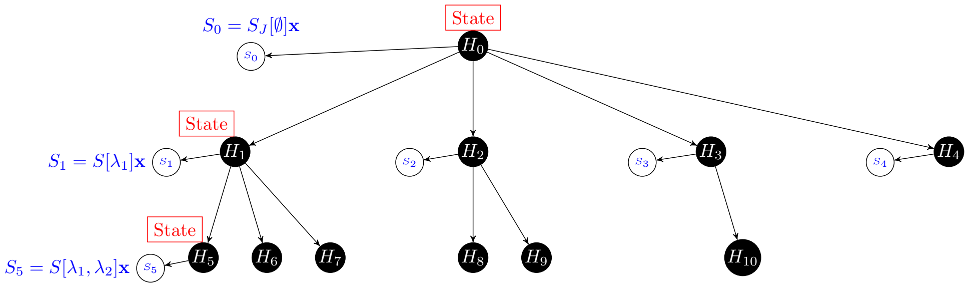
$S_0 = S_J[\emptyset]\mathbf{x}$
State
H_0
s_0
State
$S_1 = S[\lambda_1]\mathbf{x}$
s_1
H_1
s_2
H_2
s_3
H_3
s_4
H_4
State
$S_5 = S[\lambda_1, \lambda_2]\mathbf{x}$
s_5
H_5
H_6
H_7
H_8
H_9
H_{10}

Figure 6.2: Scattering hidden Markov tree.

This is the architecture we will use. The propose SHMT model is thus composed of a set of visible nodes $\{\mathbf{S}_i\}_{i\in\mathcal{T}}$ and a set of hidden node $\{\mathbf{H}_i\}_{i\in\mathcal{T}}$. Both sets are organised in a similar tree structure with the following characteristics,

- For any index i of the tree, $S_i \in \mathbb{R}$ and $H_i \in [\![1, K]\!]$ where K is the number of possible hidden states.

- The initial hidden state is drawn from a discrete possibly non uniform initial distribution π_0 such that:

$$\forall k \in [\![1, K]\!] \quad \pi_0(k) = P(H_0 = k).$$

- For any index i of the tree, the emission distribution describes the probability of the visible node S_i conditional to the hidden state H_i,

$$\forall i \in \mathcal{T} \;\; \forall k \in [\![1,K]\!] \text{ and } \forall s \in \mathbb{R} \quad P(S_i = s | H_i = k) = P(s|\phi_{k,i}), \quad (6.1)$$

  where $P(.|\phi_{k,i})$ belongs to a parametric distribution family and where $\phi_{k,i}$ is the vector of emission parameters for the state k and the node i. In the remainder of the document the emission distribution is Gaussian. So Equation 6.1 becomes,

$$\forall i \in \mathcal{T} \;\; \forall k \in [\![1,K]\!] \text{ and } \forall s \in \mathbb{R} \quad P(S_i = s | H_i = k) = \mathcal{N}(s|\mu_{k,i},\sigma_{k,i}),$$

  where $\phi_{k,i} = (\mu_{k,i},\sigma_{k,i})$ with $\mu_{k,i}$ and $\sigma_{k,i}$ being respectively the mean and the variance of the Gaussian for the k-th value of the mixture and the node i.

- For any index i of the tree, the probability for the hidden node H_i to be in a state k given the father's state g is characterised by a transition probability,

$$\forall i \in \mathcal{T} \backslash \{0\} \;\; \forall g,k \in [\![1,K]\!]^2 \quad A_i^{(gk)} = P(H_i = k | H_{\rho(i)} = g),$$

  where A_i defines a transition probability matrix such that,

$$\forall i \in \mathcal{T} \backslash \{0\} \;\; \forall k \in [\![1,K]\!] \quad P(H_i = k) = \sum_{g=1}^{K} A_i^{(gk)} P(H_{\rho(i)} = g). \quad (6.2)$$

Thus for a given scattering architecture —i.e. fixed M, J and L— the SHMT model is fully parametrized by,

$$\theta = \big(\pi_0, \{A_i, \{\phi_{k,i}\}_{k\in[\![1,k]\!]}\}_{i\in\sqcup}\big). \tag{6.3}$$

And we the joint distribution factorizes in,

$$P(\mathbf{H},\mathbf{S}) = \pi_0(H_0)P(S_0|H_0)\prod_{i}^{|\mathcal{T}|} P(H_i|H_{\rho(i)}, A_i)p(S_i|H_i,\phi_i).$$

Our SHMT model differs from the previous works by the properties of its tree structure. First, previous work on HMT models are based on regular binary trees where all the leaves are found at the same depth. The scattering tree, however, is both irregular and non-binary. Indeed, as seen in Section 2.3, each node has a variable number of children. This yields an architecture where the number of descendants is not constant and where leaves can be found at any depth of the tree. Second, the SHMT is expected to have non-homogeneous transition matrix. Indeed by the nature of the scattering transform one can expect a non homogeneous transmission of the information across the orders and especially across the orientations. Hence non-homogeneous transition matrices across nodes from a same father and across images themselves are allowed. Section 6.4 describes an adaptation of Durand et al. [71] learning algorithm to irregular, non-homogeneous and non-binary trees.

Even though the theoretical framework of SHMT holds for any $K \in \mathbb{N}^*$, in all the applications of the SHMT we set $K = 2$. This means that the scattering coefficients are described by a mixture of two Gaussians. Those two states match the sparsity of the wavelet described in [57]. A wavelet coefficient is either "low" —i.e. no information— or "high" —i.e. contain information. This model yields a sparser representation of the scattering coefficient as the number of hidden states is highly constrained.

6.3.2 Hypothesis

Expressing the dependencies between the scattering coefficients as a hidden Markov tree requires two modelling assumptions. The statement in Section 6.3.2 expresses the fact that there is a —simple— parameterisation for the distribution of the scattering coefficients. In Section 6.3.2, we assert that the coefficients are correlated across layers.

K populations

This assumption reflects the fact that the scattering coefficients can effectively be expressed by K hidden states.

Assumption 1. K **populations:**
Each scattering coefficient of a signal can be described by K clusters. The smooth regions are represented by small scattering coefficients, while edges,ridges, and other singularities are represented by large coefficients.

This assumption is common for $K = 2$ and standard or complex wavelets [57]. Since the scattering coefficients of order m can simply be seen as the modulus of the wavelet transform of a "new" signal —i.e. the scattering coefficient of order $m - 1$, the two-populations assumption for scattering network is sensible.

This intuition can be confirmed by Figure 6.3 and 6.4 displaying the scattering coefficients at a given node obtained for several signals. Figure 6.3 shows the scattering coefficients of a noisy binary square. Note that for sake of clarity a "small" network has been used. This does not affect the observations that can be made and one can notice that the largest values of the scattering coefficient are obtained on highly informative pixels (edges in this case) while the less informative pixels are represented by scattering coefficients near 0. Similar observations can be made for more complex signals —such as the one displayed in Figure 6.4.

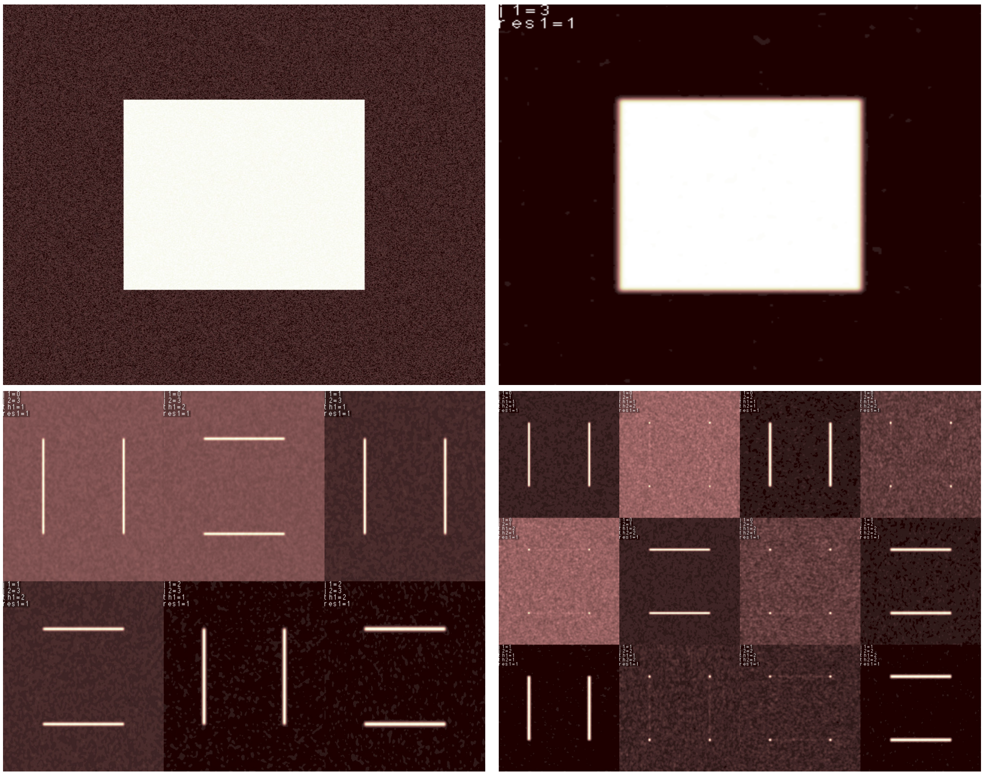

Figure 6.3: K populations - Experiment 1: The signal is a binary square (0: background, 1: square) with noise. The scattering network has $M = 2$ layers, $J = 3$ scales and $L = 2$ orientations.
Top Left: Original signal.
Top Right: Layer 0.
Bottom Left: Layer 1.
Bottom Right: Layer2.

A statistical interpretation of the K populations assumption implies that scattering coefficients have non-Gaussian marginal statistics, that is, their marginal probability density functions have a large peak at zero due to the many small coefficients and heavy tails due to a few large coefficients are observed. Finally since many real-world signals (photograph-like images, for example) consist mostly of smooth regions separated by a few singularities, the K populations assumption tells us that the scattering coefficients are a sparse representation for these signals (this notion of sparsity can be made mathematically precise; see for example [30] or [28]). Most of the scattering coefficient magnitudes are small, while a few of them

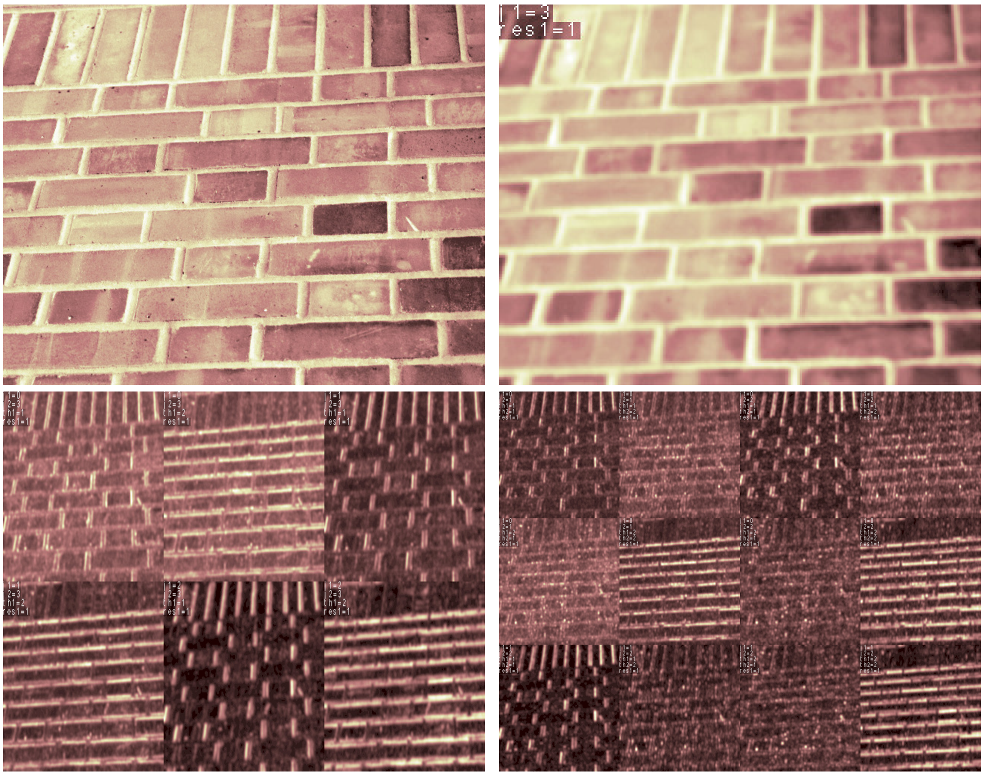

Figure 6.4: K populations - Experiment 2: The signal is a realisation of the class "brick wall" of the *CUReT* texture dataset. The scattering network has $M = 2$ layers, $J = 3$ scales and $L = 2$ orientations.
Top Left: Original signal.
Top Right: Layer 0.
Bottom Left: Layer 1.
Bottom Right: Layer2.

encoding the singularities and the informative content are large.

Persistence

This assumption expresses the smoothness of the states across the scattering transform tree.

Assumption 2. **Persistence:**

Along a scattering path, high and low scattering coefficient values cascade across the scattering orders.

This assumption codifies how the hidden states are structured. Smooth regions/singularities are assumed to be represented by low/high values at every layer. Persistence leads to scattering coefficient values that are statistically dependent along the branches of the scattering tree. This means that one can expect the transitions matrices to have higher diagonal coefficients —i.e. same state transitions. A statistical interpretation of the K populations assumption implies that scattering coefficients are —to some extent— correlated across layers.

Figure 6.5 displays the magnitude of the scattering coefficients for a given node i of the tree against those of its father $\rho(i)$. One expects to see a strong positive correlation but also expects the difference of orientations between the father and the child to have an influence on how strong this correlation is. One could intuit that the closer the orientations the higher the correlation. This intuition can be supported by the difference between the left and right figures of Figure 6.5. The left figure displays the correlation between third order scattering coefficients and their second order fathers in the case where the whole lineage has the same orientation. In this case a high correlation coefficient is observed. The right figure also displays the correlation between third order scattering coefficients and their second order fathers but in the case where the members of this lineage have different orientations. Not surprisingly, a lower correlation coefficient is observed. Table 6.1 reports the average correlation across all the pairs (father, child) of a SHMT. Those two experiments tend to confirm the existence of a correlation as well as a potential dependency over the delta in orientation between the father and the child.

In addition to those experimental intuitions, one can —under some restrictions— prove that the correlation between scattering coefficients across layers is decreasing exponentially with the difference between scales. For the remainder of this section, the scattering operator is restricted to its 1-D version and is hence only function of the scale j —as opposed to scale

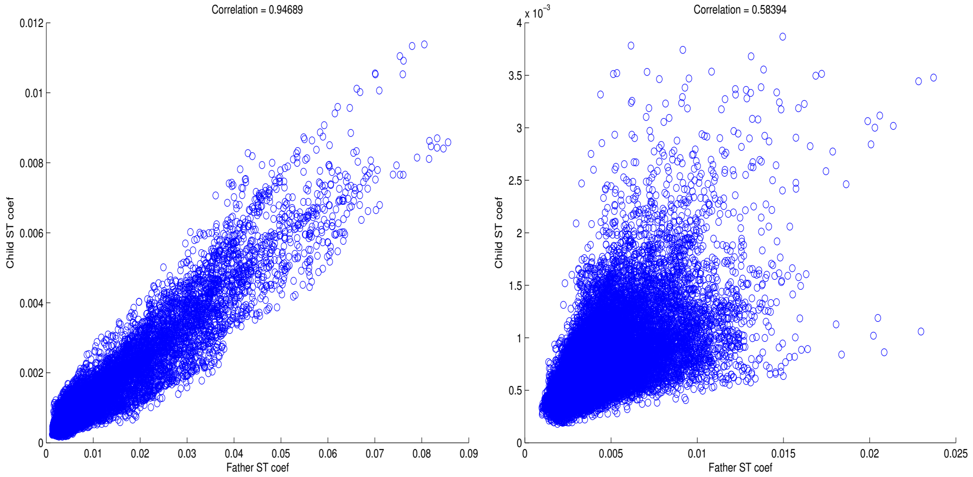

Figure 6.5: Persistence - Experiment 1: Magnitude of the scattering coefficients obtain for a realization of the class "brick wall" of the *CUReT* texture dataset at a given index i of the tree against those of its father $\rho(i)$. The scattering network has $M=3$ layers, $J=4$ scales and $L=4$ orientations.
Left: Same orientation for the two layers.
Right: Different orientations for the two layers.

Classification results		
Signal:	Correlation mean	Correlation variance
diagonal:	0.909	0.260
Square:	0.811	0.300
Circle:	0.876	0.164
uiuc brick:	0.647	0.241
Mandrill:	0.503	0.255
Lena:	0.727	0.236

Table 6.1: Persistence - Experiment 2: Average correlation across nodes of the scattering transform applied to different signals. The scattering network has $M = 3$ layers, $J = 4$ scales and $L = 4$ orientations.

j and orientation θ for a 2-D operator. This scattering transform is applied to a self-similar process $\mathbf{x}$ having stationary increments H. Examples of such processes are the fractional Brownian motions or the α-stable Lévy processes.

We know from [148] that the scattering coefficients can be expressed as stated in the following proposition.

Proposition 3. *(Scattering transform of self-similar processes)*

If $\mathbf{x}$ is a self-similar process with stationary increments then for all scale $j_1 \in \mathbb{Z}$,

$$\widetilde{S}[j_1]\mathbf{x} = 2^{j_1 H},$$

and for all $(j_1, j_2) \in \mathbb{Z}^2$,

$$\widetilde{S}[j_1, j_2]\mathbf{x} = \bar{S}[j_2 - j_1]\widetilde{\mathbf{x}},$$

where,

$$\widetilde{S}[j_1]\mathbf{x} = \frac{\bar{S}[j_1]\mathbf{x}}{\bar{S}[0]\mathbf{x}},$$

$$\widetilde{S}[j_1, j_2]\mathbf{x} = \frac{\bar{S}[j_1, j_2]\mathbf{x}}{\bar{S}[j_1]\mathbf{x}},$$

and

$$\widetilde{\mathbf{x}} = \frac{|\mathbf{x} * \lambda(j_1)|}{\mathbb{E}[|\mathbf{x} * \lambda|]}.$$

They also prove the following theorem for a signal $\mathbf{x}$ belonging to the fractional Brownian family.

Theorem 6.3.1. ***(Scattering transform of fractional Brownian)***
Let $\mathbf{x}$ be a fractional Brownian motion with Hurst exponent $0 < H < 1$. There exists a constant $C > 0$ such that, for all $j_1 \in \mathbb{Z}$,

$$\lim_{j' \to \infty} 2^{j'/2} \widetilde{S}[j_1, j_1 + j']\mathbf{x} = C$$

Hence by combining Proposition 3 and Theorem 6.3.1, one can state the following corollary.

Corollary 6.3.2. ***(Scattering coefficients correlation)***
Let $\mathbf{x}$ be a fractional Brownian motion. Then for all $j_1 \in \mathbb{Z}$,

$$\mathbb{E}\left[\widetilde{S}[j_1]\mathbf{x}\widetilde{S}[j_1, j_1 + j']\mathbf{x}\right] \simeq 2^{-j'/2} \tag{6.4}$$

Corollary 6.3.2 shows that the covariance drops exponentially between the two layers indexed by j_1 and $(j_1 + j')$ as j' —i.e. the difference between the two different scale levels— goes large. In other words, this shows that there is some dependency between similar scales. Note that this is not absolutely perfectly, since it proves the non correlation for very different scales. However since we limit the scattering networks to frequency decreasing paths of scale smaller than a given J, we can assume that we are never in the case where $j_2 - j_1$ is large enough for Equation 6.4 to be validated and the correlation to be small.

Note. We are currently working on defining a more pertinent bounding for $\mathbb{E}\left[\widetilde{S}[j_1]\mathbf{x}\widetilde{S}[j_1, j_1 + j']\mathbf{x}\right]$.

6.4 Learning the tree parameters

As seen in Section 3.2.3, hidden Markov models can be trained using Expectation-Maximization methods. Hidden Markov chains use a version of the EM algorithm called Forward-Backward (FB) algorithm allowing the propagation of the hidden states along the chain. Crouse et al. [43] propose an adaptation to the hidden Markov trees of the FB algorithm called the Upward-Downward (UD) procedure. Those procedures are suffering from underflowing issues [60], preventing from fitting them to large models. Devijver [20] propose an smoothing trick for the FB procedure. Durand et al. [71] adapt this smoothing procedure to tree models. This section proposes our rewritten version of the smoothed EM algorithm adapted to irregular, non-homogeneous and non-binary HMTs.

To do so one needs to introduce the following notation:

- $\forall i \in \mathcal{T}$, let n_i be the number of children of the node i.

- $\forall i \in \mathcal{T}$, let $\bar{\mathcal{S}}_i = \bar{s}_i$ be the observed sub-tree rooted at node i. By convention $\bar{\mathcal{S}}_0$ denotes the entire observed tree.

- $\forall i \in \mathcal{T}$, let $\bar{\mathcal{S}}_{\mathcal{C}_i} = \bar{s}_{\mathcal{C}_i}$ be the entire -possibly empty collection of observed sub-trees rooted at the children of node i (i.e. the sub-tree $\bar{s}_i$ except its root s_i).

- If $\bar{\mathcal{S}}_i$, is a sub-tree of $\bar{\mathcal{S}}_j$, then $\bar{\mathcal{S}}_{j\backslash i} = \bar{s}_{j\backslash i}$ is the sub-tree rooted at node j except the sub-tree rooted at node i.

- $\forall i \in \mathcal{T}$ let $\bar{\mathcal{S}}_{0\backslash \mathcal{C}_i} = \bar{s}_{0\backslash \mathcal{C}_i}$ be the entire tree except for the sub-trees rooted at children of node i.

Note. Those notations transpose to the hidden state and for instance $\bar{\mathcal{H}}_i = \bar{h}_i$ is the state sub-tree rooted at node i.

It is interesting to express the logic developed for the EM algorithm to a dynamic programming approach [11, 103, 120]. The complex parameter learning problem is being successively broken down into sub-problems until solving them becomes tractable.

6.4.1 E-Step

The smoothed version of the E-step requires the computation of the conditional probability distributions $\xi_i(k) = P(H_i = k|\bar{\mathcal{S}}_i = \bar{s}_i)$ (smoothed probability) and $P(H_i = k, H_{\rho(i)} = g|\bar{\mathcal{S}}_i = \bar{s}_i)$ for each node $i \in \mathcal{T}$ and states k and g. The smoothed probability adapted to the HMT structure can be decomposed as,

$$\xi_i(k) = \frac{P(\bar{\mathcal{S}}_{0\backslash i} = \bar{s}_{0\backslash i}|H_i = k)}{P(\bar{\mathcal{S}}_{0\backslash i} = \bar{s}_{i\backslash i}|\bar{\mathcal{S}}_1 = \bar{s}_i)} P(H_i = k|\bar{\mathcal{S}}_i = \bar{s}_i)$$

The smoothed upward-downward algorithm requires the introduction the following quantities,

$$\begin{aligned}
\beta_i(k) &= P(H_i = k|\bar{\mathcal{S}}_i = \bar{s}_i) \\
\beta_{\rho(i)i}(k) &= \frac{P(\bar{\mathcal{S}}_i = \bar{s}_i|H_{\rho(i)} = k)}{P(\bar{\mathcal{S}}_i = \bar{s}_i)} \\
\alpha_i(k) &= \frac{P(\bar{\mathcal{S}}_{0\backslash i} = \bar{s}_{0\backslash i}|H_{\rho(i)} = k)}{P(\bar{\mathcal{S}}_{0\backslash i} = \bar{s}_{0\backslash i}|\bar{\mathcal{S}}_i = \bar{s}_i)}
\end{aligned} \tag{6.5}$$

The smoothed upward-downward algorithm also requires the preliminary knowledge of the marginal state distributions $P(H_i = k)$ for each node i. However this can simply be achieved by a downward recursion initialised at the root node with $P(H_0 = k) = \pi_0(k)$ and then cascading the information down the tree using the recursive Formula 6.2.

Upward recursion

The upward algorithm is initialised at all the leaves of the tree, by computing $\beta_i(k)$ using,

$$\begin{aligned}
\beta_i(k) &= P(H_i = k|\bar{\mathcal{S}}_i = \bar{s}_i) \\
&= P(H_i = k|\mathcal{S}_i = s_i) \\
&= \frac{P(S_i = s_i|H_i = k)P(H_i = k)}{P(S_i = s_i)} \\
&= \frac{P(s_i|\phi_{i,k})P(H_i = k)}{N_i},
\end{aligned}$$

where the normalization factor for the leaves N_i is given by,

$$N_i = P(S_i = s_i) = \sum_{k=1}^{K} P(s_i|\phi_{i,k})P(H_i = k).$$

Then one can recursively —upward recursion— compute $\beta_i(k)$ for the remaining nodes of the tree using,

$$\begin{aligned}
\beta_i(k) &= P(H_i = k|\bar{\mathcal{S}}_i = \bar{s}_i) \\
&= \left[\prod_{j\in\mathcal{C}_i} P(\bar{\mathcal{S}}_j = \bar{s}_j|H_i = k)\right] P(S_i = s_i|H_i = k)\frac{P(H_i = k)}{P(\bar{\mathcal{S}}_i = \bar{s}_i)} \\
&= \left[\prod_{j\in\mathcal{C}_i} \frac{P(\bar{\mathcal{S}}_j = \bar{s}_j|H_i = k)}{P(\bar{\mathcal{S}}_j = \bar{s}_j)}\right] P(S_i = s_i|H_i = k)P(H_i = k)\frac{\prod_{j\in\mathcal{C}_i} P(\bar{\mathcal{S}}_j = \bar{s}_j)}{P(\bar{\mathcal{S}}_i = \bar{s}_i)} \\
&= \frac{\left[\prod_{j\in\mathcal{C}_i} \beta_{ij}(k)\right] P(s_i|\phi_{i,k})P(H_i = k)}{N_i},
\end{aligned}$$

where the normalization factor for the non-leaf nodes N_i is given by,

$$\begin{aligned}
N_i &= \frac{P(\bar{\mathcal{S}}_i = \bar{s}_i)}{\prod_{j\in\mathcal{C}_i} P(\bar{\mathcal{S}}_j = \bar{s}_j)} \\
&= \sum_{k=1}^{K}\left[\prod_{j\in\mathcal{C}_i} \beta_{ij}(k)\right] P(s_i|\phi_{i,k})P(H_i = k).
\end{aligned}$$

For all nodes i, the quantities $\beta_{\rho(i)i}(k)$ can be extracted from β_i using,

$$\begin{aligned}
\beta_{\rho(i)i}(k) &= \frac{P(\bar{\mathcal{S}}_i = \bar{s}_i|H_{\rho(i)} = k)}{P(\bar{\mathcal{S}}_i = \bar{s}_i)} \\
&= \frac{\sum_{g=1}^{K} P(\bar{\mathcal{S}}_i = \bar{s}_i|H_i = g)P(H_i = g|H_{\rho(i)} = k)}{P(\bar{\mathcal{S}}_i = \bar{s}_i)} \\
&= \sum_{g=1}^{K} \frac{P(H_i = g|\bar{\mathcal{S}}_i = \bar{s}_i)P(H_i = g|H_{\rho(i)} = k)}{P(H_i = g)} \\
&= \sum_{g=1}^{K} \frac{\beta_i(g)A_i^{(kg)}}{P(H_i = g)}.
\end{aligned}$$

Using those relationships, one can derive the upward Algorithm 2.

```
Meta-parameters:
$K$
Initialization:
// $P(s_i|\phi_{i,k})$:
for All the nodes $i$ of the tree $\mathcal{T}$ do
    $P(s_i|\phi_{i,k}) = \mathcal{N}(s_i|\mu_{k,i},\sigma_{k,i})$
end
// Loop over the leaves $i$ of the tree:
for All the leaves $i$ of the tree $\mathcal{T}$ do
    $\beta_i(k) = \frac{P(s_i|\phi_{i,k})P(H_i=k)}{\sum_{g=1}^{K} P(s_i|\phi_{i,k})P(H_i=g)}$
    $\beta_{i,\rho(i)}(k) = \sum_{g=1}^{K} \frac{\beta_i(g)A_i^{(kg)}}{P(H_i=g)} P(H_{\rho(i)} = k)$
    $l_i = 0$
end
Induction:
// Bottom-Up loop over the nodes of the tree:
for All non-leaf nodes $i$ of the tree $\mathcal{T}$ do
    $M_i = \sum_{k=1}^{K} P(s_i|\phi_{i,k}) \prod_{j \in /mcalC_i} \frac{\beta_{j,i}(k)}{P(H_i=k)^{n_i-1}}$
    $l_i = \log(M_i) + \sum_{j \in \mathcal{C}_i} l_j$
    $\beta_i(k) = \frac{P_{\theta_{k,i}(s_i)} \prod_{j \in /mcalC_i}(\beta_{j,i}(k))}{P(H_i=k)^{n_i-1} M_i}$
    for All the children nodes $j$ of node $i$ do
        $\beta_{i \backslash \mathcal{C}_i}(k) = \frac{\beta_i(k)}{\beta_{i,j}(k)}$
    end
    $\beta_{i,\rho(i)}(k) = \sum_{g=1}^{K} \frac{\beta_i(g)A_i^{(kg)}}{P(H_i=g)} P(H_{\rho(i)} = k)$
end
```

Algorithm 2: Smoothed upward algorithm.

Downward recursion

The downward recursion can either be built on the basis of the quantities $\alpha_i(k)$ defined in Equation 6.5 or using the smoothed probabilities $\xi_i(k) = P(H_i = k|\bar{\mathcal{S}}_i = \bar{s}_i)$. The downward recursion on ξ_i is initialized at the root node with,

$$\xi_0(k) = P(H_0 = k|\bar{\mathcal{S}}_0 = \bar{s}_0) = \beta_0(k).$$

The quantities ξ_i can then be computed recursively for each node of the tree using,

$$\begin{aligned}
\xi_i(k) &= P(H_i = k|\bar{\mathcal{S}}_0 = \bar{s}_0)\\
&= \sum_{g=1}^{K} \frac{P(H_i = k, H_{\rho(i)} = g, \bar{\mathcal{S}}_0 = \bar{s}_0)}{P(H_{\rho(i)} = g, \bar{\mathcal{S}}_0 = \bar{s}_0)} P(H_{\rho(i)} = g|\bar{\mathcal{S}}_0 = \bar{s}_0)\\
&= P(\bar{\mathcal{S}}_i = \bar{s}_i|H_i = k) \sum_{g=1}^{K} \frac{P(H_i = k|H_{\rho(i)} = g)}{P(\bar{\mathcal{S}}_i = \bar{s}_i|H_{\rho(i)} = g)} P(H_{\rho(i)} = g|\bar{\mathcal{S}}_0 = \bar{s}_0)\\
&= \frac{\beta_i(k)}{P(H_i = k)} \sum_{g=1}^{K} \frac{A_i^{(gk)} \xi_{\rho(i)}(g)}{\beta_{\rho(i),i}(g)}.
\end{aligned} \tag{6.6}$$

Using the fact that for all $i \in \mathcal{T}$ $\xi_i(k) = \beta_i(k)\alpha_i(k)$ and the relationship from Equation 6.6, one can express the downward pass as presented in Algorithm 3.

```
Meta-parameters:
$K$
Initialization:
$\alpha_0(k) = 1$
Induction:
// Top-Down loop over the nodes of the tree:
for All nodes i of the tree $\mathcal{T} \backslash \{0\}$ do
    $\alpha_i(k) = \frac{1}{P(H_i=k)} \sum_{g=1}^{K} \alpha_{\rho(i)}(g) A_i^{(gk)} \beta_{\rho(i)\backslash i}(g) P(H_{\rho(i)} = g)$
end
```

Algorithm 3: Smoothed downward algorithm.

Conditional properties

To complete the E-step one needs to compute the conditional probabilities for each node. This is done by noticing that,

$$\forall i \in \mathcal{T} \quad P(H_i = k|\bar{\mathcal{S}}_0 = \bar{s}_0) = \alpha_i(k)\beta_i(k),$$

and

$$\forall i \in \mathcal{T}\backslash\{0\} \quad P(H_{\rho(i)} = g, H_i = k|\bar{\mathcal{S}}_0 = \bar{s}_0) = \frac{\beta_i(k) A_i^{(gk)} \alpha_{\rho(i)}(g) \beta_{\rho(i)}(g)}{P(H_i = k) \beta_{\rho(i)i}(g)}.$$

6.4.2 M-Step

The maximization step of the EM algorithm aims at finding the optimum of the log-likelihood of the observations with regards to the parameters and then use those pseudo-optimal parameters for the next expectation step. In other words at iteration l of the EM process, the M-step carries out the update,

$$\theta^{l+1} = \underset{\theta}{\operatorname{argmax}} \Big(E[\ln f(\mathbf{x}, H|\theta)|\mathbf{x}, \theta^l)] \Big). \tag{6.7}$$

The θ maximizing the log-likelihood in Equation 6.7 can be expressed analytically and this yields Algorithm 4

```
Meta-parameters:
K,
Distribution family for $P_\theta$ ;                // Here Gaussian
$N$ ;       // Number of observed realizations of the
 signal
Initialization:
$\pi_0(k) = \frac{1}{N}\sum_{n=1}^{N} P(H_0^n = m|s_0^n, \theta^l)$
Induction:
// Loop over the nodes of the tree:
for All nodes i of the tree $\mathcal{T}\backslash\{0\}$ do
    $P(H_i = k) = \frac{1}{N}\sum_{n=1}^{N} P(H_i^n = k|\bar{s}_0^n, \theta^l)$,
    $A_i^{gk} = \frac{\sum_{n=1}^{N} P(H_i^n = k, H_{\rho(i)}^n = g|\bar{s}_0^n, \theta^l)}{NP(H_{\rho(i)} = k)}$,
    $\mu_{k,i} = \frac{\sum_{n=1}^{N} s_i^n P(H_i^n = k|\bar{s}_0^n, \theta^l)}{NP(H_i = k)}$,
    $\sigma_{k,i}^2 = \frac{\sum_{n=1}^{N} (s_i^n - \mu_{k,i})^2 P(H_i^n = k|\bar{s}_0^n, \theta^l)}{NP(H_i = k)}$.
end
```

Algorithm 4: M-step of the EM algorithm.

6.4.3 EM algorithm

Finally the EM algorithm iterates over the E-step and the M-step as described by Algorithm 5.

```
Meta-parameters:
$K$;
Distribution family for $P_\theta$;
Convergence criteria ;                // Iteration limit or
  information based
Initialization method for $\theta$ ; // Random or prior knowledge
Initialization:
$l = 0$ ;                              // Iteration counter
Initialize($\theta^0$)
Iteration:
while Not convergence do
    E-step: Calculate $P(\bar{\mathcal{H}}|\bar{\mathcal{H}},\theta^l)$.
    M-step: Set $\theta^{l+1} = \text{argmax}_\theta \left(E[\ln f(\mathbf{x},H|\theta)|\mathbf{x},\theta^l)]\right)$.
    l = l+1
end
```

Algorithm 5: EM algorithm.

6.5 Classification

Let θ_c now be a set of parameters for an SHMT $\mathcal{T}$ learned using the EM algorithm described in Section 6.4 on a training set $\{\bar{S}^n_{0,c}\}_{n\in[\![1,N]\!]} = \{ST_{(\psi,J,M,L)}(\mathbf{x}^n_c)\}_{n\in[\![1,N]\!]}$ composed of the scattering representations of N realizations of a signal of class c . Let also $\mathbf{x}^{new}$ be another realization of this signal, not used for training and $\mathcal{T}^{new}$ be the instance of the SHMT generated by this realisation.

In this context the MAP algorithm aims at finding the optimal hidden tree $\hat{\hat{h}}^{new}_0 = (\hat{h}^{new}_0 \dots \hat{h}^{new}_{I-1})$ maximizing the probability of this sequence given the model's parameters $P(\bar{\mathcal{H}}_0 = \hat{\hat{h}}^{new}_0|\mathcal{T}^{new},\theta_c)$. The MAP framework also provides $\hat{P}$ the value of this maximum.

For SHMT the MAP algorithm has the form described by Algorithm 6.

```
Meta-parameters:
$K$;
Initialization:
for all leaves $i$ of $\mathcal{T}$ do
    $\gamma_i(k) = \beta_i(k)$ ;        // The gamma for all $k$ must be
     computed before the next step
    $\gamma_{i,\rho(i)}(k) = \max_{1\leq g\leq K} \gamma_i(g)\epsilon_i^{kg}$
    $\xi_i(k) = \text{argmax}_{1\leq g\leq K} \gamma_i(g)\epsilon_i^{kg}$
end
Induction:
// Top-Down loop over the nodes of the tree:
for All nodes $i$ of the tree $\mathcal{T}\backslash\{0\}$ do
    $\gamma_i(k) = P_{\theta_{k,i}}(s_i)\prod_{j\in/mcalC_i} \gamma_{j,i}(k)$
    $\gamma_{i,\rho(i)}(k) = \max_{1\leq g\leq K} \gamma_i(g)\epsilon_i^{kg}$ ;   // Except at root node
    $\xi_i(k) = \text{argmax}_{1\leq g\leq K} \gamma_i(g)\epsilon_i^{kg}$
end
Termination:
$\hat{P} = \max_{1\leq g\leq K} \gamma_0(g)$
$\hat{h}_0 = \text{argmax}_{1\leq g\leq K} \gamma_0(g)$
Downward tracking:
// Creation of the hidden tree from the root node
for All nodes $i$ of the tree $\mathcal{T}\backslash\{0\}$ do
    $\hat{h}_i = \xi_i(\hat{h}_{\rho(i)})$
end
```

Algorithm 6: MAP algorithm.

The MAP Algorithm 6 can be used in a multi-class classification problem by training an SHMT model per class and then when presented with a new realization $\mathbf{x}^{new}$ comparing the probability of the MAP hidden tree provided by each model as described by Algorithm 7.

```
Meta-parameters:
$K$; $C$ ;                                   // Number of classes
for All classes $c$ do
  | $\hat{P}_c = \mathrm{MAP}(\mathbf{x}^{new}, \theta_c, K)$
end
$\hat{P} = \max_{0 \leq c < C} \hat{P}_c$
$l = \mathrm{argmax}_{0 \leq c < C} \hat{P}_c$
```

Algorithm 7: MAP algorithm applied to multi-class classification problem.

6.6 Experiments

This section presents some experimental results obtained using scattering convolutional hidden Markov trees for classification tasks. First, in Section 6.6.1, SHMT is used to classify handwritten digits in the complex situation where only a few training examples per class are available. Section 6.6.2 reports the use of SHMTs to classify seabed and ripples in sonar imagery. Finally Section 6.6.3 describes an adaptation of this classifier to perform a very naive segmentation.

6.6.1 Hand written digits

We compare the performance of SHMT to those of a SCN combined with an SVM (SCN+SVM) restricted to a small number of training examples by performing two experiments on the handwritten digit classification dataset MNIST [151].

For all the experiments we use a scattering transform with $M = 3$ orders, $J = 3$ scales, $L = 3$ orientations and a Morlet mother wavelet. The hidden Markov tree has $K = 2$ states and uses a mixture of Gaussian to describe the relationship between the scattering coefficients and the hidden states. For the SVM, the best parameters are selected by cross-validation.

MNIST - "One vs All"

In a similar fashion to Salakhutdinov et al. [101], we first test SHMT on a "One vs All" binary classification task. However we propose this experiment with a more challenging setup. They pre-train their model with 100 samples from each class but one and then provide only limited amount of training examples, say N, for this last class. Instead we propose a framework where all the classes have the same limited amount of training points N. We then test the models on 1000 unseen examples.

Table 6.2 displays the accuracy and the sensitivity for both SHMTs and SCN+SVMs. With $N = 5$ training examples, SVM is not able to discriminate the digit of interest and simply classifies everything as "All". This yields a sensitivity of 0.0% characteristic of an uninformative test. Under the same conditions, SHMT is able to correctly discriminate the digit of interest and provides a very informative test –good sensitivity and accuracy. Some classes, however, —4 and 6— are more challenging than others due to their high intra-class variability.

Note that this experimental setup should be in favor of the SVM classifier since it is effectively provided with $9N$ training points for one class. With this amount training samples,

1 vs All class		MNIST	
		SHMT	SCN+SVM
9	(Acc)	**97.2**%	90.0%
	(Sen)	**94.0**%	0.0%
6	(Acc)	**94.1**%	90.0%
	(Sen)	**57.5**%	0.0%
Average	(Acc)	**93.9**%	90.0%
	(Sen)	**60.2**%	0.0%

Table 6.2: Accuracy and sensitivity on 1000 samples of MNIST trained with 5 training points per class and tested on "One vs All" for digits: "6", "9", and the average over all digits.

MNIST - Full

SHMT and SCN+SVM are tested on the more complex problem of mutli-class labelling. SHMT and SCN+SVM are both trained on a limited number

Table 6.3: Classification score on the complete test set of MNIST (10000 samples) trained using only a limited number N of training points per class

Training samples per class	MNIST	
	SHMT	SCN+SVM
$N = 2$	**28.6**%	18.7%
$N = 5$	**48.0**%	43.2%
$N = 10$	45.2%	**49.9**%

of training examples per class and tested on the full test set.

The best results for each models are displayed in Table 6.3. SHMT displays better generalization and prediction properties than SCN+SVM when trained on a very limited number of training points. For $N = 2$ seen samples per class, SVM beats a random selector—i.e. 10%— by a small margin while SHMT provides a near three folds improvement. With only 5 training examples per class, SHMT does close to five times better than random. As expected, when the number of training samples grow large enough —i.e. 10 and more, SCN+SVM reaches better maximum classification score.

The drop in performance of SHMT for $N = 10$ training examples is explained by the fact that the EM algorithm subroutine is undermined by convergence to local minima issues [39] yielding sometimes to poor learning quality for SHMTs. However when a good minima is found, SHMTs has acceptable generalization performance.

While confirming the superiority of our model in terms of generalisation performance for limited number of training points, this experiment also highlights a potential weakness of it in that sometimes convergence problems occur. However, in the main, SHMT provides good classification score for such a low number of training examples.

6.6.2 Sonar Imagery

In underwater mine detection recovering the largest proportion of the true positives is crucial since missing a target could be very costly. However, recovering a large proportion of the true positives may incur many false positives. Reducing these to a manageable number is an open problem in marine sciences. Indeed by its design an underwater mine can be mistaken with some natural features of the seabed. One of those natural features generating many false alarms is called "ripple". Those regular patterns drawn in the sand by currents can vary greatly in shape and orientation as displayed on the right half of Figure 6.8.

As mentioned in [100] one can relatively engineer an effective mine detector with a matched filter type detector given a clean background. The "ripple" however yield a important number of false alarms. In [100], they also provide a pre-processing step based on Hurst estimation (see Chapter 5) to dim down the "ripple" pattern. This method applied too aggressively, however, can also remove true mines. Reducing the overall performance of the detection pipeline. It is thus interesting to develop a classifier between seabed types to apply different processing pipeline to different classes. This is the task proposed for the SHMT.

The data used are extracted from the *UDRC MCM* sonar imagery dataset [89]. This dataset comprises Synthetic Aperture RADAR (7300 × 2000 pixels SAS images) and meta-data —not used in this experiment. From those images, easier to handle 100 by 100 patches have been extracted and labelled as either seabed or ripple (see respectively Figure 6.7 and Figure 6.8). The classification task at hand is very challenging due to the low informative content of each images and the high intra-class variability.

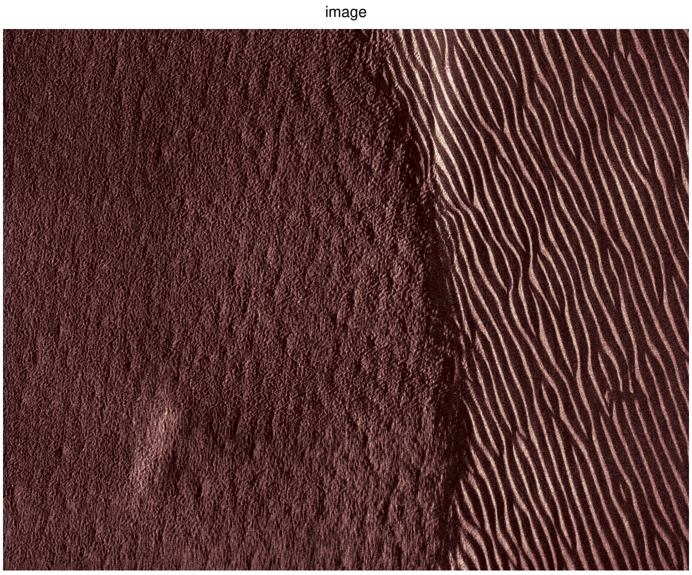

Figure 6.6: SAS sonar raw image: Ripples can be observed on the right half of the image

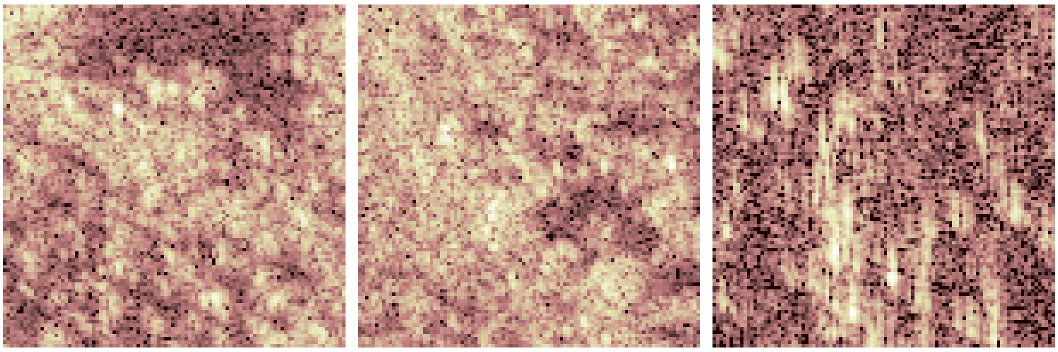

Figure 6.7: Sample of seabed patches.

The scattering transform used has $M = 3$ orders, $J = 5$ scales, $L = 3$ orientations and uses a Morlet wavelet. The hidden Markov tree has $K = 2$ states and is using a mixture of Gaussian to describe the relationship between the scattering coefficients and the hidden states. Two models —one for each class considered— θ_{ripple} and θ_{seabed} are trained on 200 realizations of their class signal. The testing is then realized on 80 images —40 of each classes. The performance of the SHMT are assessed on 100 instances of this experiment and the results are displayed in Table 6.4.

The first row of Table 6.4 displays the results obtained on 100 exper-

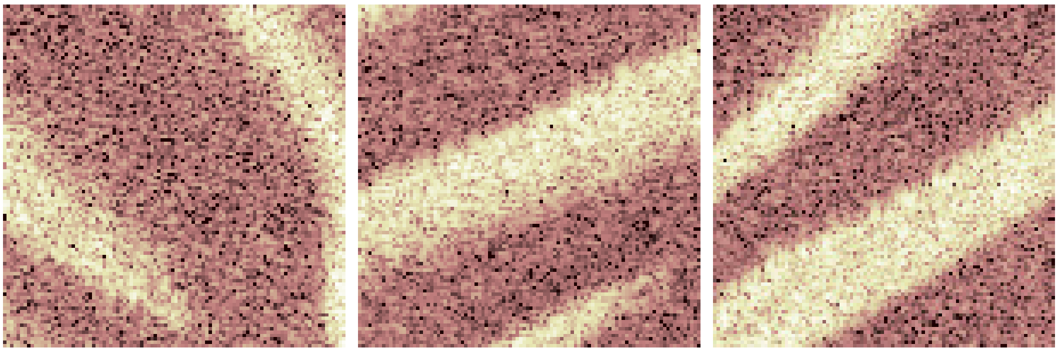

Figure 6.8: Sample of ripple patches.

Classification results					
Classification score:	N	Mean	Variance	Maximum	Minimum
Full:	100	0.74	0.101	0.9	0.5
$\geq$ 60%:	91	0.76	0.079	0.9	0.6
$\geq$ 70%:	73	0.79	0.058	0.9	0.7

Table 6.4: Classification performance over 100 experiments of Ripple/Seabed classification.

iments run. Despite a slightly unsatisfying average classification score of 74%, the best models reach a good accuracy of 90%. The lowest score is 50% accuracy and is obtained because all the testing examples are all associated to the same class. This pathological case can be explained by one class's model having converged to a poor local maximum — or not converged yet. Such a model provides non informative outputs; regardless of the true class of the image, this model will always produce a high (or always low) probability of belonging to the class it is supposed to discriminate. other. Those cases highly a weakness of the current learning method. At the moment the convergence is tested using simple improvement rule which can lead to local maxima. The design of a smarter model selection test based on information criterion could be a way to overcome this issue. One could for example rely on the Akaike Information Criterion (AIC) [22] or the Bayes Information Criterion (BIC) [81] to selected the best model amongst a set of trained model prior to the supervised testing. Overall, this experiment and its very satisfying best model validate the assumptions made since when convergence occurs correctly the discriminative performance are good. The second and third rows in Table 6.4 simulates the results if a validation criterion based on an imposed accuracy score on a validation set was imposed. This validation set would be another way to address the problem of sometimes converging to a poor local maxima.

6.6.3 Segmentation

On its simplest form segmentation can be seen as a set of independent classification tasks on subparts of an image. Hence one can use the models trained in the previous section to realize the segmentation of a full sonar imagery.

One of the 2001 × 7333 image from the *UDRC MCM* is cut into a set 100 × 100 patches —some regions of the original image are not considered. And each of those patches is presented to the classifier. Results of this procedure can be seen in Figure 6.9.

Even though this approach to segmentation is very naive and does not introduce any form of spatial smoothing or correlation between nearby patches to improve accuracy, the SHMT model provides satisfying segmentation of the seabed. Furthermore, as displayed by the bottom figure in Figure 6.9, it provides a probability map for the confidence in our segmentation decision. Those probability maps are very interesting as they show that the misclassified patches do have a high variance on their prediction. On average they are more frequently predicted as one class than the other. The uncertainty on that prediction, however, makes that prediction less trustworthy and call for further investigation. One could then decide to do further analysis on those patches. Such maps are interesting as they could easily be exploited by a huma operator to help in a decision taking process.

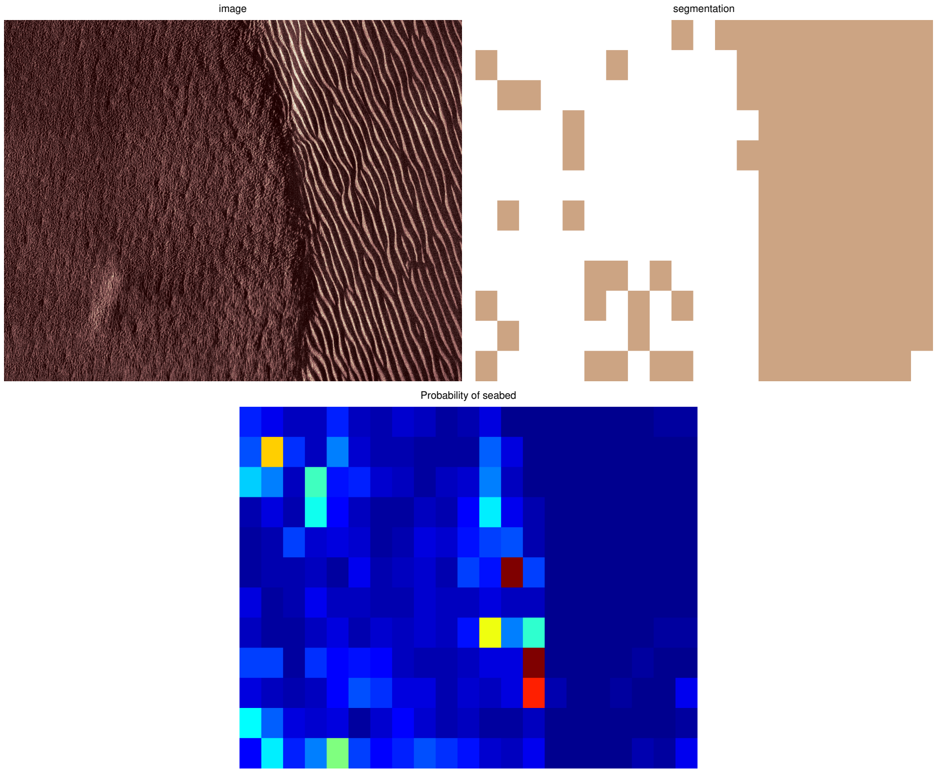

Figure 6.9: Segmentation of a sonar imagery.
Top Left: Original signal.
Top Right: Naive segmentation.
Bottom: Prediction variance over 50 predictions on the same tile. Color scale is ranging from dark blue (low variance) to dark red (high variance).

6.7 Conclusion

A SHMT framework has been proposed which comprises a scattering transform and a hidden Markov tree model. The scattering transform projects the data into a representational space of even higher dimensionality but of reduced volume along the invariants in the data. Then a probabilistic graphical model —hidden Markov tree— was used to fit a generative model to the distribution of the representation of the data. As such, the proposed model takes advantage of the way in which the scattering transform introduces invariances into the representation but also the manner in which hidden Markov models capture dependencies between coefficients. Experiments have demonstrated that the modelled distribution can be used to perform efficient classification tasks even with small training sizes. Even though we only here consider classification, a generative model is much more versatile than a simple —yet efficient— discriminative one. Because they model the full distribution of the data they can express more complex relationships between the observed and the unknown variables than simple discrimination.

To enhance SHMT and especially the chance of converging toward a good minima during the EM learning, Chapter 7 will include development of variational methodology to learn the model parameters [87].

Chapter 7

Variational Scattering Hidden Markov Tree

In this section we detail the quantities and computations used while learning the parameters of a variational hidden Markov tree. First we recall how a scattering network can be modelled as an Hidden Markov tree. Then we introduce the variational approximation of this hidden Markov tree and the necessary equations for the learning algorithm are derived.

Chapter outline

In this chapter we extend the scattering hidden Markov Tree, a powerful statistical model defined in Chapter 6, to use variational approximation and flexible learning objectives. In Section 7.1, we provide more detailed motivation for such a model. Section 7.2 introduces some related work. In Section 7.3, we provide a description of this model under its simplest form as well as detailed computations for the fitting algorithm. In Section 7.4, we combine the variational SHMT with the AB-variational objective defined in Chapter 4. Finally, Section 7.5 displays some experimental results.

7.1 Introduction

We are interested in the posterior distribution of the states tree and parameters given the observations, $p(\mathbf{H},\theta|\mathbf{S})$. As seen in Section 6.4, the MLE can be solved exactly. One can evaluate the marginal likelihood of an observation given the model's parameters $p(\mathbf{S}|\theta)$ and the most probable state sequences given an observation $\text{argmax}_{\mathbf{H}}\, p(\mathbf{H}|\mathbf{S},\theta)$. This is done via upward-downward algorithm where the values of θ and $\mathbf{H}$ are alternately fixed [71].

This Maximum-Likelihood (ML) based approach, however, produces a point estimate of the model's parameters. It is thus unable to capture the variance over them. Furthermore, it also has a known tendency to overfitting the training data and suffers from potential convergence toward local minimum issues [55]. The former can be overcome by setting the problem as a fully Bayesian model where all parameters are given a prior distribution. However those types of model quickly become intractable and will require using approximate inference. We will focus here on variational based method as described in Section Section 3.3 and Chapter 4. We approximate the true posterior $p(\mathbf{H},\theta|\mathbf{S})$ by a variational distribution $q(\mathbf{H},\theta)$ laying within a simpler family of distributions. The variational approximation also takes care, to some extent, of the overfitting and convergence towards local minimum issue. Since the inference problem is now cast as a optimisation problem, we can tap into the optimisation literature and leverage more robust solvers.

Note that in this chapter, we are considering the scattering transform of a p pixels images but to simplify the notations, the indexing on those pixels will be omitted. Throughout "scattering coefficient" S_i will be abusively used to express the set of p values obtained for a given node i. Similar abuse is used for the hidden nodes.

7.2 Background

We first review briefly the usage of variational methods for hidden Markov models and more specifically tree-like models.

Bernardo et al. [66] propose a variational version of the EM algorithm (VBEM), allowing the use of variational methods in the context of graphical models with missing data. In order to fit models to ever increasing size of graphs, refinements specific to the HMMs have been developed on top of the general VBEM algorithm. Ji et al. [80], for example, propose an extension to continuous models. McGrory and Titterington [92] establish model selection methods. And recently, Foti et al. [136] extended the method to perform stochastic variational inference on HMMs; thus allowing the fitting of models to even longer chains/bigger graphs.

As seen in Section 6.2, the scattering hidden Markov tree model shares a lot of similarity with the wavelet Markov tree models. It is thus not surprising to find variational extensions of description of the standard wavelet tree by an hidden Markov tree. Dasgupta and Carin [78] extend the work of Crouse et al. [43] to allow fitting of variational model using the standard upward-downward procedure. Similarly, Olariu et al. [93] propose an extension to Durand et al. [71] to perform smoothed VBEM on hidden Markov trees.

7.3 Structured mean field for HMTs

The idea behind the variational representation of the hidden Markov tree is to introduce a prior distribution to the parameters θ of model described in Section 6.3 and approximate the posterior distribution of interest $p(\mathbf{H}|\mathbf{S},\theta)$. To do so each parameter of the model is described by its own parameterised distribution q. Under certain assumptions, we can derive the exact update formulae for the model parameters. The remainder of this section will describe that case.

As a starting point, we remember the joint likelihood function used for the exact SHMT model,

$$P(\mathbf{H},\mathbf{S}) = \pi_0(H_0)P(S_0|H_0)\prod_{i}^{|\mathcal{T}|}P(H_i|H_{\rho(i)},A_i)p(S_i|H_i,\phi_i). \tag{7.1}$$

where A_i is the transition probability matrix at node i as defined in Equation 6.2. As seen in Section 3.3, the log-marginal probability of an observation can be decomposed as,

$$\ln p(\mathbf{S}) = \mathcal{L}_{KL}(q(\mathbf{H},\theta),p(\mathbf{H},\theta|\mathbf{S})) + KL(q(\mathbf{H},\theta)||p(\mathbf{H},\theta|\mathbf{S})),$$

Where θ represents the model parameters.

$$\theta = \big(\pi_0,\{A_i,\{\phi_{k,i}\}_{k\in[\![1,k]\!]}\}_{i\in\sqcup}\big).$$

Since $KL(q(\mathbf{H},\theta)||p(\mathbf{H},\theta|\mathbf{S})) \geq 0$, minimising the divergence or maximising the lower bound $\mathcal{L}_{KL}(q(\theta),q(\mathbf{H}))$ are equivalent. The lower bound, however, can be made computationally tractable and will be used in that section as an equivalent objective.

The usual approach to simplifying the variational problem and for achieving tractability for hidden Markov models uses a structured mean field approximation [92] such that,

$$q(\mathbf{H},\theta) = q(\pi_0)q(A)q(\phi)q(\mathbf{H}). \tag{7.2}$$

We break the dependencies between each parameters in θ and latent states $\mathbf{H}$. Note that while we do not break the dependencies between the hidden states to preserve possibly crucial dependencies of the tree structure, we make the parameters independent across the tree so that we have,

$$q(A) = \prod_{i=1}^{|T|} q(A_i)$$

and

$$q(\phi) = \prod_{i=1}^{|T|} q(\phi_i)$$

Each factor in Equation (7.2) is endowed with its own variational parameters and is set to be in the same exponential family distribution as its respective complete conditional. This allows the variational parameters to be optimised separately to maximise the evidence lower bound $\mathcal{L}_{KL}$.

$$\begin{aligned} \ln p(\mathbf{S}) &\geq \mathcal{L}_{KL}(q(\theta),q(\mathbf{H})) \\ &= \mathbb{E}_q[\ln p(\theta)] - \mathbb{E}_q[\ln q(\theta)] + \mathbb{E}_q[\ln p(\mathbf{H},\mathbf{S}|\theta)] - \mathbb{E}_q[\ln q(\mathbf{H})] \end{aligned}$$

The evidence-lower bound can be maximised using variational EM [66]. It alternately updates:

- The global parameters θ, i.e. the hidden variables coupled to the entire set of observations.
- The local variables $\mathbf{H}^n$, i.e. a set of hidden states per observation $\mathbf{S}^n$

of a signal $\mathbf{x}_n$.

In some sense, this algorithm is the counterpart of the exact EM algorithm introduced in Section 6.4 but using the approximated distribution q during the E-step rather than its exact counter part p. Despite this similarity the VB-EM algorithm is simplified by the fact that q can be made as simple as desired — to the cost of reduced expressiveness. And thus simplify the computational load.

To simplify the update procedures we consider the approximate distributions to be members of the exponential family. In statistic, the exponential family refers to the set of probability distributions that can be expressed in the form of the following equation:

$$f(x|\tau) = h(x).\exp(\eta(\tau)T(x) - C(\eta(\tau))) \tag{7.3}$$

where $T(x)$ is a sufficient statistic, $\eta(\tau)$ is the natural parameter, $h(x)$ is the carrier density and $C(\eta(\tau))$ is the cumulant generating function. Examples of common distributions belonging in the exponential family are the Normal, binomial, or the Poisson distributions.

7.3.1 Priors

We now introduce priors over the exact model parameters defined in Equation (6.3). In the case where the approximation distributions q belong to the exponential family, the analysis is considerably simplified if we use conjugate prior distributions [77]. In that case, the updated posteriors belong to the same families of distributions.

We specify a Dirichlet prior on the initial state probability,

$$p(\pi_0) = \text{Dir}(\pi_0|\alpha^{\pi_0}).$$

Where $\text{Dir}(\pi|\alpha^{\pi_0})$ denotes a $K-$dimensional Dirichlet distribution with concentration parameters α^{π_0}.

Each row of the transition matrix is also given a Dirichlet prior,

$$p(A_i) = \prod_{g=1}^{K} \text{Dir}(A_i^{(g:)}|\alpha_g^{A_i}).$$

Finally, we specify a normal-inverse Wishart (NIW) prior on the generative distribution parameters,

$$\begin{aligned}\phi_{i,k} = (\mu_{i,k}, \sigma_{i,k}) &\sim \text{NIW}(\mu_0^i, \kappa_0^i, \zeta_0^i, \nu_0^i) \\ &\sim \mathcal{N}(\mu_{i,k}|\mu_0^i, \frac{1}{\kappa_0^i}\sigma_{i,k})\mathcal{W}^{-1}(\sigma_{i,k}|\zeta_0^i, \nu_0^i).\end{aligned}$$

It is interesting to note that the NIW prior is slightly over expressive for our case. The NIW prior is the conjugate prior of a mixture when both the mean and the covariance are unknown, but with non diagonal covariance matrix. In the SHMT model the errors are independent — i.e. the covariance matrix is diagonal. In such a case a Normal-Gamma prior can also be used. Experimental results from Section 7.5 shows, however, that this extra expressiveness is not too harmful to the model performance.

Figure 7.1 provides a graphical representation of the variational model used to describe the SHMT.

7.3.2 Global update

During the global update, we optimise distribution of the parameters θ assuming $q(\mathbf{H})$ is known and optimal. This is somehow very similar in principle to the M-step of the EM procedure described in Section 6.4.2. The optimal local parameters $q^*(\mathbf{H})$ can be computed following the procedure

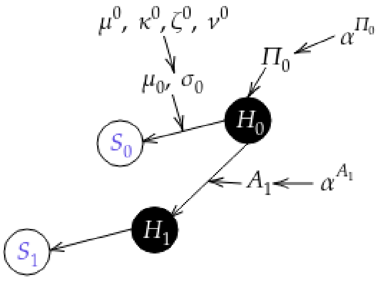

Figure 7.1: The variational approximation for the scattering hidden Markov tree posits a Dirichlet prior on the initial state and the rows of the transition matrices. The generation model is parameterised by a normal-inverse Wishart prior.

detailed in Section 7.3.3.

The variational approximation cast the posterior computation problem as an optimisation problem. Thus, given those local parameters, the global updates can then be obtained by differentiating $\mathcal{L}_{KL}$ with regard to θ. Since we are using distributions from the conjugate-exponential family and the KL-divergence, the update takes the simple form [123],

$$\mathbf{w} = \mathbf{u} + \mathbb{E}_{q^*(\mathbf{H})}[t(\mathbf{H}, \mathbf{S})] \tag{7.4}$$

where $t(\mathbf{H}, \mathbf{S})$ is a vector of sufficient statistics, $\mathbf{w} = (\mathbf{w}^{\pi_0}, \mathbf{w}^A, \mathbf{w}^\theta)$ are the variational parameters in natural form and $\mathbf{u} = (\mathbf{u}^{\pi_0}, \mathbf{u}^A, \mathbf{u}^\theta)$ are the model hyper-parameters also in natural form.

Natural parameters

Equation (7.4) requires using the natural form of the parameters of the variational distributions. The general definition of natural parameter can be sen in Equation 7.3.

The initial state follows a Dirichlet distribution and thus has one natural parameter per state,

$$u_k^{\pi_0} = \alpha_k^{\pi_0} - 1 \quad \text{for} \quad k = 1 \ldots K.$$

Similarly, the transition matrix distribution has the natural parameters,

$$u_{gk}^{A_i} = \alpha_{gk}^{A_i} - 1 \quad \text{for} \quad g, k = 1 \ldots K.$$

The emission parameters of each state are governed by a normal-inverse Wishart distribution and thus have four natural parameters per state,

$$\begin{aligned} u_{k,1}^{\phi_i} &= \kappa_0^i \mu_0^i \quad \text{for} \quad k = 1 \ldots K. \\ u_{k,2}^{\phi_i} &= \kappa_0^i \\ u_{k,3}^{\phi_i} &= \zeta_0^i + \kappa_0^i \mu_0^i \mu_0^{iT} \\ u_{k,4}^{\phi_i} &= \nu_0^i + 2 + p \end{aligned} \tag{7.5}$$

Expected sufficient statistics

To perform the global parameter updates defined by Equation (7.4), we need to compute the expectation of the sufficient statistics with regard to the variational distribution $q^*(\mathbf{H})$.

Since the initial state follows a Dirichlet distribution, the associated sufficient statistics are,

$$t_k^{\pi_0} = \sum_{n=1}^{N} \mathbb{1}(H_0^n = k) \quad \text{for} \quad k = 1 \ldots K.$$

The sufficient statistics for the transition matrices are also those a Dirichlet distribution,

$$t_{gk}^{A_i} = \sum_{n=1}^{N} \mathbb{1}(H_{\rho(i)}^n = g, H_i^n = k) \quad \text{for} \quad g, k = 1 \ldots K.$$

The emission parameters are described by a NIW distribution and the associated sufficient statistics are,

$$
\begin{aligned}
t_{k,1}^{\phi_i} &= \sum_{n=1}^{N} S_i^n \mathbb{1}(H_i^n = k) \quad \text{for} \quad k = 1 \dots K. \\
t_{k,2}^{\phi_i} &= \sum_{n=1}^{N} \mathbb{1}(H_i^n = k) \\
t_{k,3}^{\phi_i} &= \sum_{n=1}^{N} S_i^n S_i^{n'} \mathbb{1}(H_i^n = k) \\
t_{k,4}^{\phi_i} &= \sum_{n=1}^{N} \mathbb{1}(H_i^n = k)
\end{aligned}
$$

Finally, the expectation of the sufficient statistics with regard to the variational distribution $q^*(\mathbf{H})$ can be expressed as,

$$
\begin{aligned}
\mathbb{E}_{q^*(\mathbf{H})}[t_k^{\pi_0}] &= \sum_{n=1}^{N} q^*(H_0^n = k) \\
\mathbb{E}_{q^*(\mathbf{H})}[t_{gk}^{A_i}] &= \sum_{n=1}^{N} q^*(H_{\rho(i)}^n = g, H_i^n = k) \\
\mathbb{E}_{q^*(\mathbf{H})}[t_{k,1}^{\phi_i}] &= \sum_{n=1}^{N} S_i^n q^*(H_i^n = k) \\
\mathbb{E}_{q^*(\mathbf{H})}[t_{k,2}^{\phi_i}] &= \sum_{n=1}^{N} q^*(H_i^n = k) \\
\mathbb{E}_{q^*(\mathbf{H})}[t_{k,3}^{\phi_i}] &= \sum_{n=1}^{N} S_i^n S_i^{n'} q^*(H_i^n = k) \\
\mathbb{E}_{q^*(\mathbf{H})}[t_{k,4}^{\phi_i}] &= \sum_{n=1}^{N} q^*(H_i^n = k)
\end{aligned}
\tag{7.6}
$$

Using the expectations defined in Equations (7.6), we can perform the update of the "global" parameters. Given the hidden states — i.e. the local variable values— for this observation the global updates takes a simple computationally tractable form.

7.3.3 Local update

In a similar fashion to the EM algorithm, we have so far performed the global parameter updates assuming the local parameter to be fixed. The local updates are performed by computing the optimal distribution over the local variables $q^*(\mathbf{H})$. More precisely we need to be able to compute both the marginal-beliefs —i.e. $q^*(H_i = k)$ for $i = 0 \ldots |T|$ and $k = 1 \ldots K$— and the pairwise-beliefs —i.e. $q^*(H_{\rho(i)} = g, H_i = k)$ for $i = 0 \ldots |T|$ and $g, k = 1 \ldots K$. Following Durand et al. [71], we use the smoothed upward-downward procedure to compute them.

Again, the local update formulae can be obtained by differentiating $\mathcal{L}_{KL}$ with regard to the local parameters $\mathbf{H}$. Following [77], we can express the optimal variational distribution over the hidden variables as,

$$q^*(\mathbf{H}) \propto \exp\left(\mathbb{E}_{q(\pi_0)}[\ln \pi(H_0)] + \sum_{i=1}^{|T|} \mathbb{E}_{q(A_i)}[\ln A_i] + \sum_{i=0}^{|T|} \mathbb{E}_{q(\theta_i)}[\ln p(S_i|H_i)]\right). \tag{7.7}$$

Comparing with Equation 7.1, Equation 7.7 takes exponentiated expected log*probabilities under the current variational distribution instead of simple probabilities. In this equation we also define the auxiliary parameters,

$$\tilde{\pi}_0 \overset{d}{=} \exp(\mathbb{E}_{q(\pi_0)}[\ln \pi_0]),$$

$$\tilde{A}_i^{gk} \overset{d}{=} \exp(\mathbb{E}_{q(A_i)}[\ln A_i^{gk}]),$$

and

$$\tilde{p}(S_i|H_i) \overset{d}{=} \exp(\mathbb{E}_{q(\theta_i)}[\ln p(S_i|H_i)]).$$

Note that for the HMT models used here, we can express the expectations and we have,

$$\tilde{\pi}_0 = \exp(\gamma(w_k^{\pi_0}) - \gamma(\sum_{l=1}^{K} w_l^{\pi_0}))$$

where $\gamma(.)$ is the digamma function. Similarly we have,

$$\tilde{A}_i^{gk} = \exp(\gamma(w_{gk}^{A_i}) - \gamma(\sum_{l=1}^{K} w_{gl}^{A_i}))$$

The posterior for the generative model is expected to be in the NIW family and can be written as,

$$q^*(\mu_{i,k}, \sigma_{i,k}) = \mathcal{N}(\mu_{i,k}|\mu_p^i, \frac{1}{\kappa_p^i}\sigma_{i,k})\mathcal{W}^{-1}(\sigma_{i,k}|\zeta_p^i, \nu_p^i),$$

where the posterior parameters are computed combining Equation 7.5 and Equation 7.6 according to Equation 7.4.

Finally, we use those auxiliary parameters to run an upward-downward algorithm producing β and α which allows to compute both $q^*(H_i = k)$ and $q^*(H_{\rho(i)} = g, H_i = k)$,

$$q^*(H_i = k) \propto \alpha_i(k)\beta_i(k) \quad \text{for} \quad k = 1 \ldots K.$$

and

$$q^*(H_{\rho(i)} = g, H_i = k) \propto \frac{\beta_i(k)\tilde{A}_i^{gk}\alpha_{\rho(i)}(g)\beta_{\rho(i)}(g)}{p(H_i = k)\beta_{\rho(i),i}(g)} \quad \text{for} \quad g, k = 1 \ldots K.$$

Iterating over the local and global updates as described in Section 7.3.3 and Section 7.3.2, we can efficiently fit our variational approximation to the scattering hidden Markov tree model using the KL objective. Section 7.5 presents some experimental results using that method.

7.4 AB-variational objective for HMTs

In Section 7.3, we have shown it was possible to fit a variational approximation using the KL-objective to the SHMT. Though efficient, in Part II, we have seen that the KL-divergence could be highly impacted by the presence of outliers in the training set. We have also seen that the KL variational objective could have an undesired tendency to under-estimating the true variance of the posterior. In Chapter 4, we have seen that one could use the scaled AB-divergence as a variational objective and leverage its robustness and mass-covering properties to obtain more sensible posterior estimates.

The SHMT model is a good example of model where outliers can appear in the training data. First of all, as demonstrated in Section 6.3.2, the correlation between the coefficients through the layers is not perfect and can introduce some unexpected/uncommon behaviours. Second, since the VBEM performs a two stage approximation assuming the other set of parameters to be fixed and optimal, it is likely to have some misestimated values for the hidden coefficients.

In this section, we apply the AB-variational objective to the variational approximation model defined in Section 7.3.

As mentioned in Chapter 4, when given a method to estimate both $p(\mathbf{H},\theta|\mathbf{S})$ and $q(\mathbf{H},\theta)$ independently one can easily replace the KL-divergence objective by the AB-objective. In the case of the variational SHMT it simply involves swapping $\mathcal{L}_{KL}$ used in the global update defined in Section 7.3.2 by $\mathcal{L}_{AB}$ as defined in Equation 4.13. The rest of the procedure described in Section 7.3 — i.e. the local updates — remain unchanged. We are thus left with a variational EM procedure where the global updates are performed to minimise the loss,

$$\begin{aligned}
&\mathcal{L}_{AB}(q||p) = \\
&= \frac{1}{\alpha(\alpha+\beta)} \log \mathbb{E}_q \left[\frac{p(\mathbf{H},\theta|\mathbf{S})^{\alpha+\beta}}{q(\mathbf{H},\theta)} \right] \\
&+ \frac{1}{\beta(\alpha+\beta)} \log \mathbb{E}_q \left[q(\mathbf{H},\theta)^{\alpha+\beta-1} \right] \\
&- \frac{1}{\alpha\beta} \log \mathbb{E}_q \left[q(\mathbf{H},\theta)^{\alpha+\beta-1} \left(\frac{p(\mathbf{H},\theta|\mathbf{S})}{q(\mathbf{H},\theta)} \right)^{\beta} \right]
\end{aligned} \tag{7.8}$$

The idea behind optimising the variational SHMT model using the AB-objective is the same as for the KL, and will require a two step iterative process as described in Section 7.3. However due to the higher complexity of the AB variational objective $\mathcal{L}_{AB}$ defined in Equation 7.8 compared to the KL variational objective $\mathcal{L}_{KL}$, we cannot express the parameter update formulae exactly. We can however use automatic differentiation tools [17, 152, 161] to computationally estimate the gradients of the objective and update the parameters.

7.5 Experiments

This section presents some experimental results obtained using the variational scattering hidden Markov trees for classification tasks. For the sake of comparison we reproduce the same set of experiments as in Section 6.6 using both the KL and AB objectives. First, in Section 7.5.1, the variational SHMT is used to classify handwritten digits in the complex situation where only a few training examples per class are available. Section 7.5.2 reports the use of VI-SHMTs to classify seabed and ripples in sonar imagery. Finally Section 7.5.3 describes an adaptation of this classifier to perform a very naive segmentation.

Throughout, unless stated otherwise, we use near uninformative prior for the hidden states with $\mathbf{u}^{\pi_0} = 0.5 + \epsilon$ and $\mathbf{u}^{A_0} = 0.5 + \epsilon$, where $\epsilon \sim \mathcal{N}(0,0.1)$ and $\mathbf{u}$ is normalised such that it sums to one. The priors for the generative distributions are selected to match the observed mean and variance of the studied data (i.e. mean and variance of each scattering coefficient for a given dataset) with again some added Gaussian noise.

7.5.1 Hand written digits

Similarly to the experiments presented in Section 6.6.1, we compare the performance of the various SHMT models — i.e. exact, KL-VI and AB-VI SHMTs— on MNIST [151] limited to a few training samples.

For sake of comparison we use the same scattering network parameters as in Section 6.6.1. We define a SCN with $M = 3$ orders, $J = 3$ scales, $L = 3$ orientations and a Morlet mother wavelet. The hidden Markov tree has $K = 2$ states and uses a mixture of Gaussians to describe the relationship between the scattering coefficients and the hidden states.

MNIST - "One vs All"

We first test the different SHMT models on a "One vs All" binary classification task. Similarly to what is done in Section 6.6.1, we train the model in a setup where all the classes have an equally small amount of training points N. The models are tested on 1000 unseen examples.

Table 7.1 displays the accuracy and the sensitivity for a set of variational SHMTs models trained with different variational objectives (see Chapter 4 for more details) as well as the exact SHMT model and SCN+SVMs used in Section 6.6.1. The meta-parameters (α, β) are selected using a greedy grid search over a limited parameter space $[1.5, 2.5] \times [0.5, 1.5]$.

Even when given a small number of training example, we can see the beneficial influence of the variational approximation for SHMT over the exact version. Furthermore we also see that a setup of the AB-objective enforcing mass covering and mild robustness to outliers out-performs the KL overall.

1 vs All class		MNIST				
		ab-SHMT (KL)	ab-SHMT (1.8,1.2)	ab-SHMT (2.2,0.8)	e-SHMT	SCN+SVM
9	(Acc)	97.9%	**98.3**%	97.3%	97.2%	90.0%
	(Sen)	94.5%	**95.4**%	94.3%	94.0%	0.0%
6	(Acc)	**95.4**%	95.3%	94.2%	94.1%	90.0%
	(Sen)	**59.6**%	59.5%	57.6%	57.5%	0.0%
Avg	(Acc)	94.7%	**95.9**%	94.1%	93.9%	90.0%
	(Sen)	61.5%	**63.2**%	60.3%	60.2%	0.0%

Table 7.1: Accuracy and recall on 1000 samples of MNIST trained with 5 training points per class and tested on "One vs All" for digits: "6", "9", and the average over all digits. The variational SHMT model is trained with different objective. "KL" uses the standard Kullback-Leibler objective. The AB-objective with (1.8,1.2) enforce robustness to outliers and mass-covering. The AB-objective with (2.2,0.8) enforce focus on the outliers and mode-seeking. As baseline we also report the SHMT trained using the exact procedure from Chapter 6, and the SCN coupled with a SVM classifier on the extracted features.

Table 7.2: Classification score on the complete test set of MNIST (10000 samples) trained using only a limited number N of training points per class

Training samples per class	MNIST				
	ab-SHMT (KL)	ab-SHMT (1.8,1.2)	ab-SHMT (2.2,0.8)	e-SHMT	SCN+SVM
$N = 2$	**30.1**%	27.6%	29.5%	28.6%	18.7%
$N = 5$	52.3%	49.0%	**53.1**%	48.0%	43.2%
$N = 10$	55.7%	54.7%	**58.1**%	45.2%	49.9%

MNIST - Full

We also test the variational version of the SHMT on the more complex problem of mutli-class classification. Similarly to what we have done in Section 6.6.1, all the models are trained on a limited number of training examples per class and this time tested on the full test set.

The best results for each models are displayed in Table 7.2. Again we can see the benefit of the variational approach over the exact inference for SHMT. Except when the number of training examples is extremely limited the flexibility of the AB-objective allows us to find a better fit than the simple KL. In that case again the best results are obtained for a model enforcing mild robustness and mass-covering.

Interestingly, with the variational SHMT, we do not observe the drop in perform found in the exact SHMT when the number of example gets bigger. This is due to the fact that we use a more powerful optimisation framework and thus avoid convergence towards local minima [39]. A test with more training points would be interesting to see if the variational SHMT stays competitive with the SVM method. However this is currently prevented as our current implementation of the SHMT becomes prohibitively slow when we increase too much the number of training points.

7.5.2 Sonar Imagery

As mentioned in Section 6.6.2, in order to tackle the task of underwater mine detection, one could use a seabed type classifier. We here test the variational scattering hidden Markov Tree on that task. For the sake of comparison we use the same dataset as in Section 6.6.2 as well as the same scattering network architecture.

The task at hand is a binary image classification problem. The image can either be of the class “ripple” (see Figure 7.3) or “seabed” (see Figure 7.2). The data used are extracted from the *UDRC MCM* sonar imagery dataset [89]. This dataset comprises Synthetic Aperture RADAR (7300 × 2000 pixels SAS images). From those images, easier to handle 100 by 100 patches have been extracted and labelled. The classification task at hand is very challenging due to the low informative content of each images and the high intra-class variability.

The scattering transform used has $M = 3$ orders, $J = 5$ scales, $L = 3$ orientations and uses a Morlet wavelet. The hidden Markov tree has $K = 2$

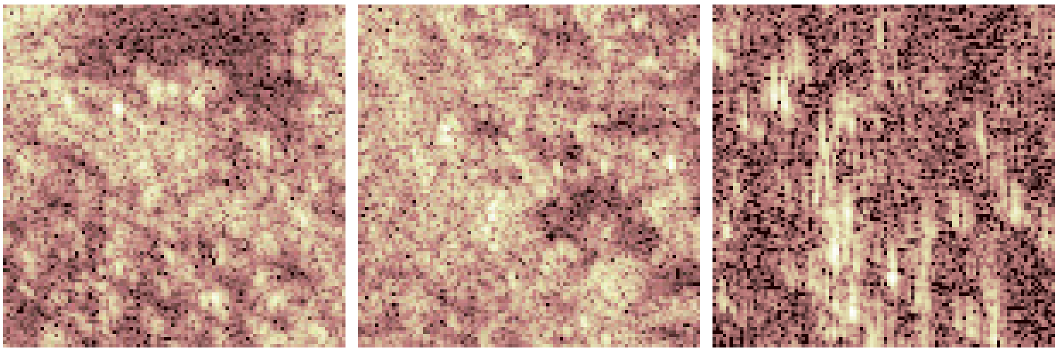

Figure 7.2: Sample of seabed patches.

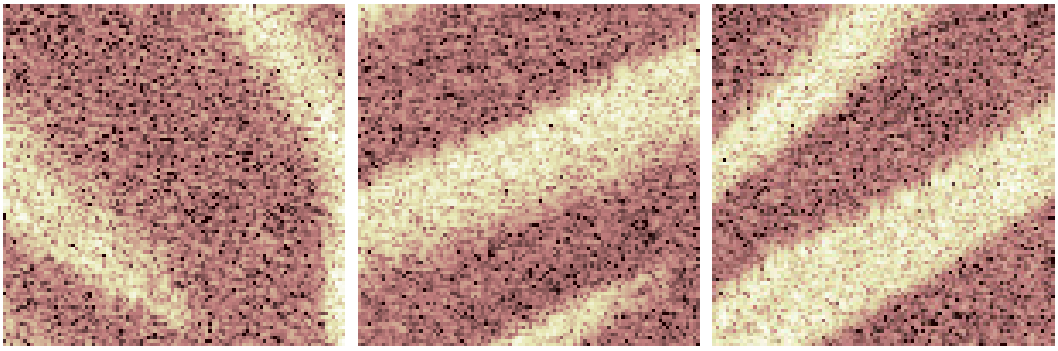

Figure 7.3: Sample of ripple patches.

states and is using a mixture of Gaussians to describe the relationship between the scattering coefficients and the hidden states. The models are trained on 200 realisations of their class signal. The testing is then realised on 80 images —40 of each classes. The performance of the various SHMT models are assessed on 100 instances of this experiment and the results are displayed in Table 7.3.

Classification results		
Classification score	Mean	Variance
e-SHMT	0.74	0.101
ab-SHMT (KL)	0.88	0.072
ab-SHMT $(1.0, 0.5)$	0.85	0.083
ab-SHMT $(1.9, -0.1)$	**0.90**	**0.086**

Table 7.3: Classification performance over 100 experiments of Ripple/Seabed classification.

The task at hand is a binary classification problem with low informative inputs. The normal seabed class can almost be described as white noise. In those condition a variational SHMT fitted with an objective geared toward mode-seeking and mild robustness to outliers — i.e. $(\lambda, \beta) = (1.9, -0.1)$ outperforms all the other methods (at the cost of a slightly higher variance). The variational SHMT systematically outperforms its exact counterparts.

7.5.3 Segmentation

In a similar fashion to Section 6.6.3, we use the best variational SHMT model from the previous section to perform naive image segmentation.

One of the 2001 × 7333 images from the *UDRC MCM* is cut into a set 200 × 200 patches —some regions of the original image are not considered. And each of those patches is presented to the classifier. Results of this procedure can be seen in Figure 7.4.

Again this very naive segmentation method visually shows good results. When compared to the segmentation obtained with the exact shmt model (see Figure 6.9), the variational method provides a better uncertainty estimate for both the very easy tile and the complex ones.

7.6 Conclusion

We have here proposed a variational approximation framework for the SHMT model defined in Chapter 6. This method allows to replace the inference problem by an optimisation one, significantly simplifying the problem.

We develop the framework using two different objectives. We have first used the standard KL divergence which yields “simple” computations but suffer some pitfalls regarding the quality of the approximation.

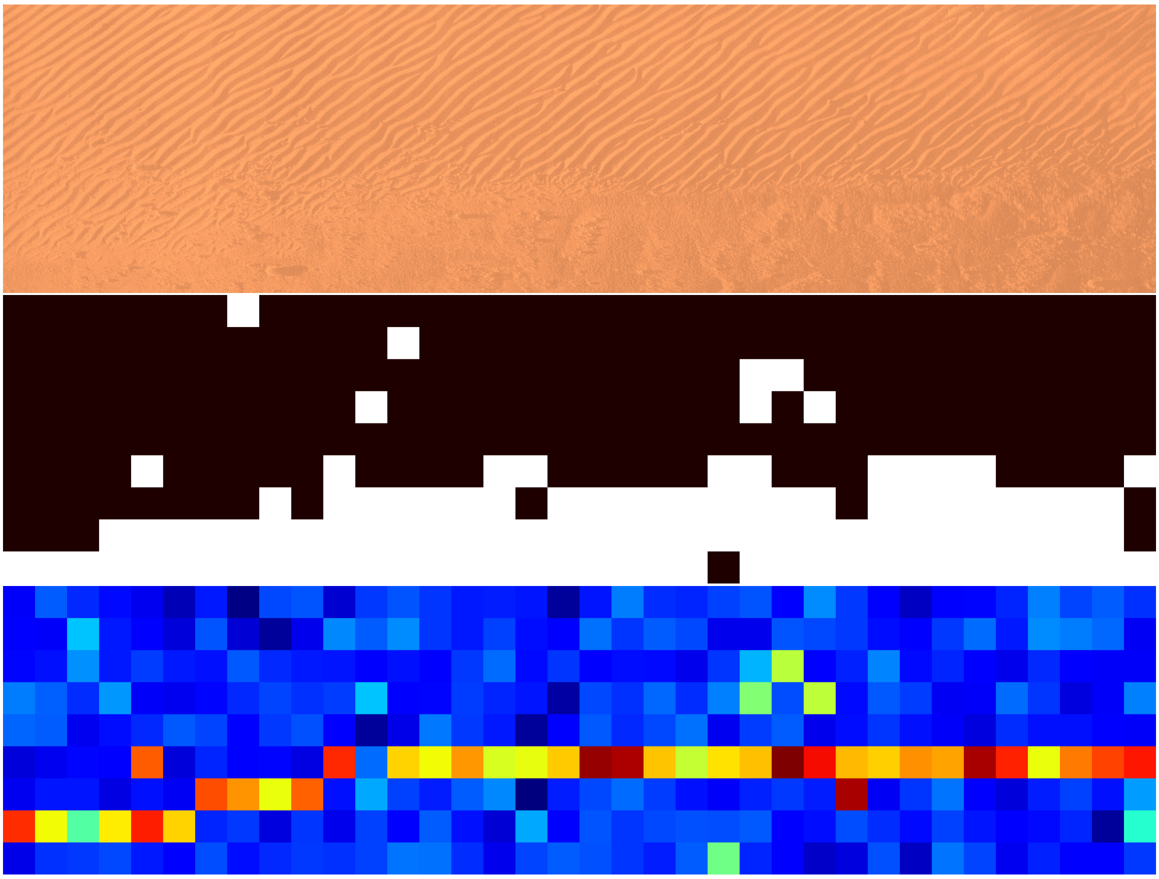

Figure 7.4: Segmentation of a sonar imagery.
Top: Original signal.
Middle: Naive segmentation.
Bottom: Variance map for the class predicted class. Color scale is ranging from dark blue (low variance) to dark red (high variance).

We also leverage the flexibility of the AB variational objective defined in Chapter 4 to fit variational SHMT models with a better control over the approximation properties.

Experimental results demonstrate the effectiveness of both the variational approximation over the exact SHMT model and the AB-objective over the KL objective.

Part V

Conclusions

Alpha-Beta variational inference

In Chapter 4, we have developed a new variational approximation objective based on the scale invariant Alpha-Beta divergence [107]. Chapter 7 leverages this objective to learn the posterior distribution of a custom model developed in that document. We here discuss a number of key points of that framework.

We have seen that using the scale invariant AB-divergence to measure the goodness of fit of the approximated posterior, one can directly optimise the divergence, instead of an equivalent objective. Furthermore we have developed an objective allowing control over both the robustness and the mass-covering properties of the approximation.

Direct optimisation

Though the ELBO provides an equivalent optimisation problem to directly optimising the —usually— intractable divergence between the true and approximated posterior $D(q(\theta)||p(\theta|\mathbf{X},\boldsymbol{\varphi}))$, this method shows limitation. Rainforth et al. [166] show, for example, that a tighter ELBO can in fact be detrimental to learning a good inference model.

Our proposed objective side steps that issue by directly optimising the divergence of interest. Thus allowing full control on the optimisation process on the quantity of interest.

Mass covering control

Minka [75] highlights a weakness of the the KL-divergence used in association to the Mean-Field assumption for variational inference. While the approximation of the mean of the posterior is correct, its variance is not estimated correctly —underestimated or overestimated if using respectively the KL or the reverse KL.

Since then much work has been done to improve on this by using more flexible, parameterised families of divergence to define the variational objective. Doing so the practitioner can smoothly interpolate between the different behaviour —mass-covering/mode-seeking— by tuning a parameter. This idea have been developed for the alpha-divergence in [75, 157]. Li and Turner [158] have developed a similar idea based on the Rényi alpha divergence.

Our proposed variational Alpha-Beta divergence method provides similar level of control over the mass of the posterior approximation. Indeed, the beta parameter of the AB-divergence offers direct control over the mass-covering/mode-seeking property.

Robustness

The KL-divergence also suffers from the presence of outliers in the dataset [42]. When fit on such a dataset using a KL based objective, the posterior tends to be affected by those non statistically representative datapoints [162]. That causes convergence to a sub-optimal posterior in term of generalisation performance. Parameterised divergence measures whose robustness to outliers can be controlled by a meta-parameter have been developed [42, 84]. Their application to VI, however, is only very recent. Futami et al. [162] leverage the properties of the Beta-divergence to perform posterior approximation robust to outliers. However the Beta-divergence does not provide a tractable ELBO and they set aside that difficulty by optimising an approximate objective.

Our proposed variational AB-objective is an extension of the Gamma-divergence and offers control over the robustness of the approximation while maintaining the exact nature of the target optimised.

Future work

Despite its advantages, the variational AB objective also adds complexity for the practitioner. It has two meta-parameters to tune and finding the best values for a specific can prove to be time consuming —despite the intuitive selection rules provided in Section 4.3.4. This issue, however, is not specific to the AB-variational inference but arises as soon as one use a parameterised family of divergence for VI [158, 162]. An interesting extension to our work would be to provide an automatic selection of the optimal (α,β) parameters. To do so one could leverage the link between scale invariant AB-divergence, AB-divergence, Beta-divergence and Tweedie models [107, 118, 132]. Leveraging those equivalences one could express the AB-objective as a distribution over its parameters (α,β). Then use hierarchical VI [159] to jointly optimise the model and also the divergence parameters.

Another weakness of using the AB divergence for VI is the potentially high variance of the Monte-Carlo estimator used as optimisation objective. An interesting addition to our work would be to provide an analysis of the bias and variance of the MC estimator along the lines of the one done for the Rényi variational objective [158]. Another interesting extension would be to leverage variance reduction methods such as those developed by Ranganath et al. [143] and AUEB and Lázaro-Gredilla [147]. This could improve the quality of the posterior as well as offer a finer control over its properties.

Semi localised Hurst estimation

In Chapter 5, we have developed a method for semi-local Hurst estimation. This method builds upon the global Hurst estimation method developed by Nelson and Kingsbury [99] and extends it to make it spatially localised. We do so by incorporating a Markov random Field on top of the Hurst estimate to cope with smooth variation and jumps in the coefficient value.

Future work

The work described in Chapter 5 is sequential. We first perform a pointwise estimation of the Hurst coefficient and then apply a graphical model on top of those estimate to spatially regulate them. A direct improvement to that method would be to jointly optimise the estimate and the regularisation factor in a one step procedure. This would mean defining an objective encompassing both the regression loss and the MRF roof-edge loss. The MRF roof-edge loss would then act as a regularisation objective.

Scattering hidden Markov Tree

We discuss a number of points that are shared by both the variational and the exact version of the SHMT model proposed, respectively, in Chapter 7 and Chapter 6.

We have seen that we can build a probabilistic graphical model on top of a fixed filter convolutional network like signal representation [121]. Leveraging the quality of the representation [168] and the fact that generative models are known to perform better than discriminative counter parts when provided only limited number of training points [61], we develop models allowing to achieve satisfying classification accuracy despite being provided with an extremely low amount of training points.

Graphical models for high dimensional signal inference

The SHMT model develops a graphical model encoding both the features obtained in the data representation step and the architecture of the data projection pipeline of the scattering transform. Doing so prevents the loss of information due to only encoding the features [121], potentially harmful in terms of inference accuracy. This idea has been developed for other types of signal representation pipelines [43, 57] and provides useful insights.

Variational inference

Though possible, the exact optimisation of the SHMT parameters proves to suffer from convergence towards poor local minima as well as underflowing issues (see Chapter 6). The variational version of this model reduces the effects of those issues (see Chapter 7). The variational setup also allows us to use more complex objectives than the KL-divergence to better control the properties of the posterior approximation. We successfully use, for example, the AB-variational objective defined in Chapter 4.

Future work

The architecture used for the scattering network in that work can be easily swapped to integrate the latest development in that field to the SHMT. Recent development includes the introduction of limited rotation invariant layers [130], and rigid-motion —i.e. combination of translation and rotation invariance— [145]. Since those network have the same general properties as the SCN used for the SHMT, one could easily extend this framework to work with those more complex transforms. Singh and Kingsbury [165] adapt the concept of the scattering transform to the Dual Tree Complex Wavelet (DTCW) [76]. Leveraging the exact invertibility of the DTCW, they create an exactly invertible SCN called “scatternet”. Though slightly different in terms of architecture, this scatternet could also be represented by an hidden Markov tree, in a similar fashion to what we have done for the SHMT.

It would also be interesting to use the generative properties of the SHMT models combined with the exact invertibility of the scatternet to sample from the feature space and perform data generation.

Another path for improving the SHMT is to improve directly on the graphical model. One could try, for example, to reduce the number of free parameters. One way to do so would be to develop a concept similar to stationarity [60] for hidden Markov chains for the trees. One could, for example, use the same transition matrix for all transition with the same scale difference.

Part VI

Annexes

The appendix is organised as follows. Section .1, we review why it was not possible to use the AB-divergence for VI. Section .2 develops the computations to extend the sAB-divergence by continuity to $(\alpha,\beta) \in \mathbb{R}^2$. Section .4 provides the mathematical details fo the computation of the influence of each parameter. Section .3 lists and decribes all the divergences encompassed within the sAB-divergence. Section .5, we provide a more detailled derivation of the sAB-variational objective. Finally, Section .6 details the experimental setups used in Chapter 4.

.1 AB variational Inference:

In Chapter 4, we use the scale invariant version of the AB-divergence (sAB-divergence) to derive the variational objective. We here show why the simple AB-divergence cannot be used for this.

In [107] the AB-divergence is defined as,

$$D_{AB}^{\alpha,\beta}(p||q) = -\frac{1}{\alpha\beta}\int\left(p(\theta)^{\alpha}q(\theta)^{\beta} - \frac{\alpha}{\alpha+\beta}p(\theta)^{\alpha+\beta} - \frac{\beta}{\alpha+\beta}q(\theta)^{\alpha+\beta}\right)d\theta.$$

Let us try to derive the ELBO associated with this divergence,

$$\begin{aligned}
&D_{AB}^{\alpha,\beta}(q(\theta)||p(\theta|\mathbf{X}))\\
&= -\frac{1}{\alpha\beta}\int\left(q(\theta)^{\alpha}p(\theta|\mathbf{X})^{\beta} - \frac{\alpha}{\alpha+\beta}q(\theta)^{\alpha+\beta} - \frac{\beta}{\alpha+\beta}p(\theta|\mathbf{X})^{\alpha+\beta}\right)d\theta\\
&= -\frac{1}{\alpha\beta}\int\left(q(\theta)^{\alpha}\left(\frac{p(\theta,\mathbf{X})}{p(\mathbf{X})}\right)^{\beta} - \frac{\alpha}{\alpha+\beta}q(\theta)^{\alpha+\beta} - \frac{\beta}{\alpha+\beta}\left(\frac{p(\theta,\mathbf{X})}{p(\mathbf{X})}\right)^{\alpha+\beta}\right)d\theta\\
&= -\frac{1}{\alpha\beta}\left(p(\mathbf{X})^{-\beta}\int q(\theta)^{\alpha}p(\theta,\mathbf{X})^{\beta}d\theta - \frac{\alpha}{\alpha+\beta}\int q(\theta)^{\alpha+\beta}d\theta\right.\\
&\qquad\left.-\frac{\beta}{\alpha+\beta}p(\mathbf{X})^{-(\alpha+\beta)}\int p(\theta,\mathbf{X})^{\alpha+\beta}d\theta\right)
\end{aligned}$$

At that step for the KL-divergence or the Renyi-divergence, one can use the log term to separate the products in sums and isolate the likelihood of the data $p(\mathbf{X})$ from the rest of the equation (i.e. the ELBO). For the

AB-divergence, however, we cannot apply this and isolate the intractable terms. This makes using the AB-divergence for variational inference impossible. We will see in section .5 that this is not the case for the scale invariant AB-divergence.

.2 Extension by continuity of the sAB-divergence

We here provide details of the extension by continuity of the sAB-divergence.

In [107] they define the scale invariant AB-divergence as,

$$\begin{aligned} D_{sAB}^{\alpha,\beta}(p||q) = & \frac{1}{\beta(\alpha+\beta)} \log \int p(\theta)^{\alpha+\beta} d\theta \\ & + \frac{1}{\alpha(\alpha+\beta)} \log \int q(\theta)^{\alpha+\beta} d\theta - \frac{1}{\alpha\beta} \log \int p(\theta)^{\alpha} q(\theta)^{\beta} d\theta, \end{aligned} \tag{9}$$

for $(\alpha, \beta) \in \mathbb{R}^2$ such that $\alpha \neq 0$, $\beta \neq 0$ and $\alpha + \beta \neq 0$.

We here provide detailed computation of the extension of the domain of definition to $\mathbb{R}^2$. For simplicity we authorize ourselves to use some shortcuts in the notations of undetermined forms.

.2.1 $\alpha + \beta = 0$

In that case $\beta \to -\alpha$ and Equation 9 becomes,

$$\begin{aligned}
&D_{sAB}^{\alpha+\beta\to 0}(p||q) \\
&= \frac{1}{\beta(\alpha+\beta)} \log \int (1 + (\alpha+\beta) \log p(\theta))\, d\theta \\
&\qquad + \frac{1}{\alpha(\alpha+\beta)} \log \int (1 + (\alpha+\beta) \log q(\theta))\, d\theta \\
&\qquad - \frac{1}{\alpha\beta} \log \int p(\theta)^{\alpha} q(\theta)^{\beta} d\theta \\
&= \frac{1}{\beta(\alpha+\beta)} \int (\alpha+\beta) \log p(\theta) d\theta + \frac{1}{\alpha(\alpha+\beta)} \int (\alpha+\beta) \log q(\theta) d\theta \\
&\qquad - \frac{1}{\alpha\beta} \log \int p(\theta)^{\alpha} q(\theta)^{\beta} d\theta \\
&= -\frac{1}{\alpha} \int \log p(\theta) d\theta + \frac{1}{\alpha} \int \log q(\theta) d\theta \\
&\qquad + \frac{1}{\alpha^2} \log \int \left(\frac{p(\theta)}{q(\theta)} \right)^{\alpha} d\theta.
\end{aligned}$$

The first approximation uses $x^a = 1 + a \log x$ when $a \approx 0$, the second uses $\log x \approx x - 1$ when $x \to 1$.

So finally we get

$$D_{sAB}^{\alpha+\beta=0}(p||q) = \frac{1}{\alpha^2} \left(\log \int \left(\frac{p(\theta)}{q(\theta)} \right)^{\alpha} d\theta - \int \log \left(\frac{p(\theta)}{q(\theta)} \right)^{\alpha} d\theta \right)$$

.2.2 $\alpha = 0$ and $\beta \neq 0$

In that case Equation 9 becomes,

$$
\begin{aligned}
&D_{sAB}^{\alpha\to 0,\beta}(p||q)\\
&= \frac{1}{\beta^2}\log\int p(\theta)^{\beta}d\theta + \frac{1}{\alpha(\alpha+\beta)}\log\int q(\theta)^{\beta}\left(1+\alpha\log q(\theta)\right)d\theta\\
&\qquad - \frac{1}{\alpha\beta}\log\int q(\theta)^{\beta}\left(1+\alpha\log p(\theta)\right)d\theta\\
&= \frac{1}{\beta^2}\log\int p(\theta)^{\beta}d\theta + \frac{1}{\alpha(\alpha+\beta)}\log\int q(\theta)^{\beta}d\theta + \frac{1}{(\alpha+\beta)}\int q(\theta)^{\beta}\log q(\theta)d\theta\\
&\qquad - \frac{1}{\alpha\beta}\log\int q(\theta)^{\beta}d\theta - \frac{1}{\beta}\int q(\theta)^{\beta}\log p(\theta)d\theta\\
&= \frac{1}{\beta^2}\log\int p(\theta)^{\beta}d\theta - \frac{1}{\beta(\alpha+\beta)}\log\int q(\theta)^{\beta}d\theta + \frac{1}{(\alpha+\beta)}\int q(\theta)^{\beta}\log q(\theta)d\theta\\
&\qquad - \frac{1}{\beta}\int q(\theta)^{\beta}\log p(\theta)d\theta
\end{aligned}
$$

The first approximation uses $x^a = 1 + a\log x$ when $a \approx 0$, the second uses $\log x \approx x - 1$ when $x \to 1$.

So finally we get

$$
D_{sAB}^{0,\beta}(p||q) = \frac{1}{\beta^2}\left(\log\int p(\theta)^{\beta}d\theta - \log\int q(\theta)^{\beta}d\theta - \beta\log\int q(\theta)^{\beta}\log\frac{p(\theta)}{q(\theta)}d\theta\right)
$$

.2.3 $\alpha \neq 0$ and $\beta = 0$

In that case Equation 9 becomes,

$$\begin{aligned}
& D_{sAB}^{\alpha,\beta\to 0}(p||q) \\
&= \frac{1}{\beta(\alpha+\beta)} \log \int p(\theta)^{\alpha} \left(1+\beta \log p(\theta)\right) d\theta + \frac{1}{\alpha^2} \log \int q(\theta)^{\alpha} d\theta \\
&\qquad - \frac{1}{\alpha\beta} \log \int p(\theta)^{\alpha} \left(1+\beta \log q(\theta)\right) d\theta \\
&= \frac{1}{\beta(\alpha+\beta)} \log \int p(\theta)^{\alpha} d\theta + \frac{1}{(\alpha+\beta)} \int p(\theta)^{\alpha} \log p(\theta) d\theta + \frac{1}{\alpha^2} \log \int q(\theta)^{\alpha} d\theta \\
&\qquad - \frac{1}{\alpha\beta} \log \int p(\theta)^{\alpha} d\theta - \frac{1}{\alpha} \int p(\theta)^{\alpha} \log q(\theta) d\theta \\
&= -\frac{1}{\alpha(\alpha+\beta)} \log \int p(\theta)^{\alpha} d\theta + \frac{1}{(\alpha+\beta)} \int p(\theta)^{\alpha} \log(\theta) d\theta + \frac{1}{\alpha^2} \log \int q(\theta)^{\alpha} d\theta \\
&\qquad - \frac{1}{\alpha} \int p(\theta)^{\alpha} \log q(\theta) d\theta
\end{aligned}$$

The first approximation uses $x^a = 1 + a \log x$ when $a \approx 0$, the second uses $\log x \approx x - 1$ when $x \to 1$.

So finally we get

$$D_{sAB}^{\alpha,0}(p||q) = \frac{1}{\alpha^2} \left(\log \int q(\theta)^{\alpha} d\theta - \log \int p(\theta)^{\alpha} d\theta - \alpha \log \int pq(\theta)^{\alpha} \log \frac{q(\theta)}{p(\theta)} d\theta \right)$$

.2.4 $\alpha = 0$ and $\beta = 0$

In that case Equation 9 becomes,

$$\begin{aligned}
&D_{sAB}^{\alpha\to 0,\beta\to 0}(p||q)\\
&= \frac{1}{\beta(\alpha+\beta)}\log\int(1+(\alpha+\beta)\log p(\theta))d\theta + \frac{1}{\alpha(\alpha+\beta)}\log\int(1+(\alpha+\beta)\log q(\theta))d\theta\\
&\qquad - \frac{1}{\alpha\beta}\log\int(1+\alpha\log p(\theta))(1+\beta\log q(\theta))d\theta\\
&= \frac{1}{\beta(\alpha+\beta)}\log\int(1+(\alpha+\beta)\log p(\theta))d\theta + \frac{1}{\alpha(\alpha+\beta)}\log\int(1+(\alpha+\beta)\log q(\theta))d\theta\\
&\qquad - \frac{1}{\alpha\beta}\log\int(1+\alpha\log p(\theta)+\beta\log q(\theta)+\alpha\beta\log p(\theta)\log q(\theta))d\theta\\
&= \frac{1}{\beta(\alpha+\beta)}\int(\alpha+\beta)\log p(\theta)d\theta + \frac{1}{\alpha(\alpha+\beta)}\int(\alpha+\beta)\log q(\theta)d\theta\\
&\qquad - \frac{1}{\alpha\beta}\int(\alpha\log p(\theta)+\beta\log q(\theta)+\alpha\beta\log p(\theta)\log q(\theta))d\theta\\
&= -\int\log p(\theta)\log q(\theta)d\theta
\end{aligned}$$

The first approximation uses $x^a = 1 + a\log x$ when $a \approx 0$, the second uses $\log x \approx x - 1$ when $x \to 1$.

So finally we get

$$D_{sAB}^{0,0}(p||q) = \frac{1}{2}\int(\log p(\theta) - \log q(\theta))^2 d\theta$$

.3 Special cases of the sAB-divergence

We here provide a more complete list of the known divergences included in the sAB-divergence.

For $(\alpha,\beta) = (1,0)$, the sAB-divergence reduces down to the KL-divergence [2],

$$D_{sAB}^{(1,0)}(q||p) = \int q(\theta)\log\left(\frac{q(\theta)}{p(\theta)}\right)d\theta.$$

For $(\alpha,\beta) = (0,1)$, the sAB-divergence reduces down to the reverse KL-divergence,

$$D_{sAB}^{(1,0)}(q||p) = \int p(\theta)\log\left(\frac{p(\theta)}{q(\theta)}\right)d\theta.$$

For $(\alpha,\beta) = (0.5,0.5)$, the sAB-divergence is a function of the Hellinger-distance [31],

$$\begin{aligned}D_{sAB}^{(0.5,0.5)}(q||p) &= -4\log\int\sqrt{p(\theta)}.\sqrt{q(\theta)}d\theta\\ &= -4\log\int\left(1-\frac{1}{2}\left(\sqrt{p(\theta)}-\sqrt{q(\theta)}\right)^2\right)d\theta\\ &= -4\log(1-D_H(p||q))\end{aligned}$$

For $(\alpha,\beta) = (2,-1)$, the sAB-divergence is a function of the χ^2-divergence [141],

$$\begin{aligned}D_{sAB}^{(2,-1)}(q||p) &= \frac{1}{2}\log\int\frac{p(\theta)^2}{q(\theta)}d\theta\\ &= \frac{1}{2}\log(1-D_{\chi^2}(p||q))\end{aligned}$$

For $(\alpha,\beta) = (0,0)$, the sAB-divergence is equal to the log-euclidean divergence D_E [149],

$$D_{sAB}^{0,0}(p||q) = \frac{1}{2}\int(\log p(\theta)-\log q(\theta))^2 d\theta$$

When $\alpha+\beta = 1$, the sAB-divergence is proportional to the Rényi-divergence [6]

$$D_{sAB}^{\alpha+\beta=1}(p||q) = \frac{1}{\alpha(\alpha-1)}\log\int p(\theta)^{\alpha}q(\theta)^{1-\alpha}d\theta.$$

When $\alpha = 1$ and $\beta \in \mathbb{R}$, the sAB-divergence is equivalent to gamma-divergence [84],

$$D_{sAB}^{\alpha=1,\beta}(p||q) = \frac{1}{\beta(\beta+1)}\log\int p(\theta)^{\beta+1}d\theta + \frac{1}{\beta+1}\log\int q(\theta)^{\beta+1}d\theta - \frac{1}{\beta}\log\int p(\theta)q(\theta)^{\beta}d\theta.$$

.4 Robustness the sAB-divergence

We here provide detailed computation of the derivative of various divergences with regard to the governing parameters of the approximation. Let us here assume we approximate the distribution p by q a function of the vector of parameters $\boldsymbol{\varphi}$.

.4.1 Kullback-Leibler divergence

For the Kullback-Leibler divergence, we get the following results,

$$\begin{aligned}\frac{d}{d\boldsymbol{\varphi}}D_{KL}(q||p) &= -\frac{d}{d\boldsymbol{\varphi}}\left(\int q(\theta)\log\frac{p(\theta)}{q(\theta)}d\theta\right)\\ &= -\int\left(\frac{dq(\theta)}{d\boldsymbol{\varphi}}\log\frac{p(\theta)}{q(\theta)} + q(\theta)\frac{d}{d\boldsymbol{\varphi}}\log\frac{p(\theta)}{q(\theta)}\right)d\theta\\ &= -\int\frac{dq(\theta)}{d\boldsymbol{\varphi}}\left(\log\frac{p(\theta)}{q(\theta)} - 1\right)d\theta.\end{aligned}$$

.4.2 Rényi-divergence

For the Rényi-divergence, we get the following results,

$$\begin{aligned}\frac{d}{d\boldsymbol{\varphi}}D_R^{\alpha}(q||p) &= -\frac{d}{d\boldsymbol{\varphi}}\left(\frac{1}{\alpha-1}\log\int q(\theta)^{\alpha}p(\theta)^{1-\alpha}d\theta\right)\\ &= -\frac{1}{\alpha-1}\frac{\int\frac{dq(\theta)}{d\boldsymbol{\varphi}}\alpha q(\theta)^{\alpha-1}p(\theta)^{1-\alpha}}{\int q(\theta)^{\alpha}p(\theta)^{1-\alpha}d\theta}\\ &= -\frac{\alpha}{1-\alpha}\frac{\int\frac{dq(\theta)}{d\boldsymbol{\varphi}}\left(\frac{p(\theta)}{q(\theta)}\right)^{1-\alpha}d\theta}{\int q(\theta)^{\alpha}p(\theta)^{1-\alpha}d\theta}.\end{aligned}$$

.4.3 Gamma-divergence

For the Gamma-divergence, we get the following results,

$$\begin{aligned}\frac{d}{d\boldsymbol{\varphi}} D_{\gamma}^{\beta}(q||p) = \frac{d}{d\boldsymbol{\varphi}} \Bigg(& \frac{1}{1+\beta} \log \int q(\theta)^{\beta+1} d\theta + \frac{1}{\beta(1+\beta)} \log \int p(\theta)^{\beta+1} d\theta \\ & - \log \int q(\theta)^{\beta} p(\theta) d\theta \Bigg) \\ = & \frac{1}{1+\beta} \frac{\frac{d}{d\boldsymbol{\varphi}} \int q(\theta)^{\beta+1} d\theta}{\int q(\theta)^{\beta+1} d\theta} - \frac{\frac{d}{d\boldsymbol{\varphi}} \int q(\theta)^{\beta} p(\theta) d\theta}{\int q(\theta)^{\beta} p(\theta) d\theta} \\ = & \frac{\int \frac{dq(\theta)}{d\boldsymbol{\varphi}} q(\theta)^{\beta} d\theta}{\int q(\theta)^{\beta+1} d\theta} - \beta \frac{\int \frac{dq(\theta)}{d\boldsymbol{\varphi}} q(\theta)^{\beta-1} p(\theta) d\theta}{\int q(\theta)^{\beta} p(\theta) d\theta} \\ = & -\frac{1}{\beta} \left(\frac{\int \frac{dq(\theta)}{d\boldsymbol{\varphi}} q(\theta)^{\beta} \frac{p(\theta)}{q(\theta)} d\theta}{\int q(\theta)^{\beta} p(\theta) d\theta} - \beta \frac{\int \frac{dq(\theta)}{d\boldsymbol{\varphi}} q(\theta)^{\beta} d\theta}{\int q(\theta)^{\beta+1} d\theta} \right).\end{aligned}$$

.4.4 sAB-divergence

For the sAB-divergence, we get the following results,

$$\begin{aligned}\frac{d}{d\boldsymbol{\varphi}} D_{sAB}^{\alpha,\beta}(q||p) = \frac{d}{d\boldsymbol{\varphi}} \Bigg(& \frac{1}{\beta(\alpha+\beta)} \log \int q(\theta)^{\alpha+\beta} d\theta + \frac{1}{\alpha(\alpha+\beta)} \log \int p(\theta)^{\alpha+\beta} d\theta \\ & - \log \int q(\theta)^{\alpha} p(\theta)^{\beta} d\theta \Bigg) \\ = & \frac{1}{\beta(\alpha+\beta)} \frac{\frac{d}{d\boldsymbol{\varphi}} \int q(\theta)^{\alpha+\beta} d\theta}{\int q(\theta)^{\alpha+\beta} d\theta} - \frac{\frac{d}{d\boldsymbol{\varphi}} \int q(\theta)^{\alpha} p(\theta)^{\beta} d\theta}{\int q(\theta)^{\alpha} p(\theta)^{\beta} d\theta} \\ = & \frac{1}{\beta} \frac{\int \frac{dq(\theta)}{d\boldsymbol{\varphi}} q(\theta)^{\alpha+\beta-1} d\theta}{\int q(\theta)^{\alpha+\beta} d\theta} - \alpha \frac{\int \frac{dq(\theta)}{d\boldsymbol{\varphi}} q(\theta)^{\alpha-1} p(\theta)^{\beta} d\theta}{\int q(\theta)^{\alpha} p(\theta)^{\beta} d\theta} \\ = & -\frac{1}{\beta} \left(\frac{\int \frac{dq(\theta)}{d\boldsymbol{\varphi}} q(\theta)^{\alpha+\beta-1} \left(\frac{p(\theta)}{q(\theta)}\right)^{\beta} d\theta}{\int q(\theta)^{\alpha} p(\theta)^{\beta} d\theta} - \alpha\beta \frac{\int \frac{dq(\theta)}{d\boldsymbol{\varphi}} q(\theta)^{\alpha+\beta-1} d\theta}{\int q(\theta)^{\alpha+\beta} d\theta} \right).\end{aligned}$$

.5 sAB-divergence Variational Inference

We here provide detailed computation of the variational objective using the sAB-divergence. We also detail the extension of this objective to the complete domain of definition.

.5.1 sAB variational objective

We are interested in minimizing the divergence $D_{sAB}^{\alpha,\beta}(q(\theta)||p(\theta|\mathbf{X}))$, this yields,

$$
\begin{aligned}
&D_{sAB}^{\alpha,\beta}(q(\theta)||p(\theta|\mathbf{X}))\\
&=\frac{1}{\alpha\beta}\log\frac{\left(\int q(\theta)^{\alpha+\beta}d\theta\right)^{\frac{\alpha}{\alpha+\beta}}.\left(\int p(\theta|\mathbf{X})^{\alpha+\beta}d\theta\right)^{\frac{\beta}{\alpha+\beta}}}{\int q(\theta)^{\alpha}p(\theta|\mathbf{X})^{\beta}d\theta}.\\
&=\frac{1}{\alpha\beta}\Bigg[\log\left(\int q(\theta)^{\alpha+\beta}d\theta\right)^{\frac{\alpha}{\alpha+\beta}}+\log\left(\int p(\theta|\mathbf{X})^{\alpha+\beta}d\theta\right)^{\frac{\beta}{\alpha+\beta}}\\
&\qquad-\log\left(\int q(\theta)^{\alpha}p(\theta|\mathbf{X})^{\beta}d\theta\right)\Bigg]\\
&=\frac{1}{\alpha\beta}\Bigg[\log\left(\int q(\theta)^{\alpha+\beta}d\theta\right)^{\frac{\alpha}{\alpha+\beta}}+\log\left(\int\left(\frac{p(\theta,\mathbf{X})}{p(\mathbf{X})}\right)^{\alpha+\beta}d\theta\right)^{\frac{\beta}{\alpha+\beta}}\\
&\qquad-\log\left(\int q(\theta)^{\alpha}\left(\frac{p(\theta,\mathbf{X})}{p(\mathbf{X})}\right)^{\beta}d\theta\right)\Bigg]\\
&=\frac{1}{\alpha\beta}\Bigg[\log\left(\int q(\theta)^{\alpha+\beta}d\theta\right)^{\frac{\alpha}{\alpha+\beta}}+\log\left(p(\mathbf{X})^{-(\alpha+\beta)}\int p(\theta,\mathbf{X})^{\alpha+\beta}d\theta\right)^{\frac{\beta}{\alpha+\beta}}\\
&\qquad-\log\left(p(\mathbf{X})^{-\beta}\int q(\theta)^{\alpha}p(\theta,\mathbf{X})^{\beta}d\theta\right)\Bigg]\\
&=\frac{1}{\alpha\beta}\Bigg[\log\left(\int q(\theta)^{\alpha+\beta}d\theta\right)^{\frac{\alpha}{\alpha+\beta}}+\log\left(\int p(\theta,\mathbf{X})^{\alpha+\beta}d\theta\right)^{\frac{\beta}{\alpha+\beta}}\\
&\qquad-\beta\log p(\mathbf{X})+\beta\log p(\mathbf{X})\;-\log\left(\int q(\theta)^{\alpha}p(\theta,\mathbf{X})^{\beta}d\theta\right)\Bigg]\\
&=\frac{1}{\beta(\alpha+\beta)}\log\int q(\theta)^{\alpha+\beta}d\theta+\frac{1}{\alpha(\alpha+\beta)}\log\int p(\theta,\mathbf{X})^{\alpha+\beta}d\theta\\
&\qquad-\frac{1}{\alpha\beta}\log\int q(\theta)^{\alpha}p(\theta,\mathbf{X})^{\beta}d\theta
\end{aligned}
$$

Finally rewriting this expression to make expectations over $q(\theta)$ appears yields,

$$\begin{aligned}
&D^{\alpha,\beta}_{sAB}(q(\theta)||p(\theta|\mathbf{X})) \\
&= \frac{1}{\alpha(\alpha+\beta)}\log\mathbb{E}_q\left[\frac{p(\theta,\mathbf{X})^{\alpha+\beta}}{q(\theta)}\right] + \frac{1}{\beta(\alpha+\beta)}\log\mathbb{E}_q\left[q(\theta)^{\alpha+\beta-1}\right] \\
&\qquad - \frac{1}{\alpha\beta}\log\mathbb{E}_q\left[\frac{p(\theta,\mathbf{X})^{\beta}}{q(\theta)^{1-\alpha}}\right]
\end{aligned}$$

.5.2 Extension by continuity

Computation very similar to those in Section .2 yields,

$$D^{\alpha,\beta}_{sAB}(q(\theta)||p(\theta|\mathbf{X})) =$$

$$\begin{cases}
\frac{1}{\beta(\alpha+\beta)}\log\int q(\theta)^{\alpha+\beta}d\theta + \frac{1}{\alpha(\alpha+\beta)}\log\int p(\theta,\mathbf{X})^{\alpha+\beta}d\theta - \frac{1}{\alpha\beta}\log\int q(\theta)^{\alpha}p(\theta,\mathbf{X})^{\beta}d\theta \\
\qquad\qquad\qquad\qquad \text{for } \alpha,\beta,\alpha+\beta \neq 0 \\
\frac{1}{\alpha^2}\left(\log\int\left(\frac{q(\theta)}{p(\theta,\mathbf{X})}\right)^{\alpha}d\theta - \int\log\left(\frac{q(\theta)}{p(\theta,\mathbf{X})}\right)^{\alpha}d\theta\right) \qquad \text{for } \alpha = -\beta \neq 0 \\
\frac{1}{\alpha^2}\left(\log\int p(\theta,\mathbf{X})^{\alpha}d\theta - \log\int q(\theta)^{\alpha}d\theta - \alpha\log\int p(\theta,\mathbf{X})^{\alpha}\log\frac{p(\theta,\mathbf{X})}{q(\theta)}d\theta\right) \\
\qquad\qquad\qquad\qquad \text{for } \alpha \neq 0, \beta = 0 \\
\frac{1}{\beta^2}\left(\log\int q(\theta)^{\beta}d\theta - \log\int p(\theta,\mathbf{X})^{\beta}d\theta - \beta\log\int q(\theta)^{\beta}\log\frac{q(\theta)}{p(\theta,\mathbf{X})}d\theta\right) \\
\qquad\qquad\qquad\qquad \text{for } \alpha = 0, \beta \neq 0 \\
\frac{1}{2}\int(\log q(\theta) - \log p(\theta,\mathbf{X}))^2 d\theta, \qquad \text{for } \alpha,\beta = 0
\end{cases}$$

.5.3 Monte Carlo approximation

$$
\begin{aligned}
&\mathbb{E}_{\{\mathbf{h}_k\}_1^K}\left[\hat{D}_{sAB}^{\alpha,\beta,K}(q(.)||p(.|\mathbf{x}))\right] \\
&= \frac{1}{\alpha(\alpha+\beta)}\mathbb{E}_{\mathbf{h}_k}\left[\log\frac{1}{K}\sum_{k=1}^{K}\frac{p(\mathbf{h}_k,\mathbf{x})^{\alpha+\beta}}{q(\mathbf{h}_k|\mathbf{x})}\right] \\
&\qquad + \frac{1}{\beta(\alpha+\beta)}\mathbb{E}_{\mathbf{h}_k}\left[\log\frac{1}{K}\sum_{k=1}^{K}q(\mathbf{h}_k|\mathbf{x})^{\alpha+\beta-1}\right] \\
&\qquad - \frac{1}{\alpha\beta}\mathbb{E}_{\mathbf{h}_k}\left[\log\frac{1}{K}\sum_{k=1}^{K}\left[q(\mathbf{h}_k|\mathbf{x})^{\alpha+\beta-1}\left(\frac{p(\mathbf{h}_k,\mathbf{x})}{q(\mathbf{h}_k|\mathbf{x})}\right)^{\beta}\right]\right] \\
&= \frac{1}{\alpha(\alpha+\beta)}\mathbb{E}_{\mathbf{h}_k}\left[\log\mathbb{E}_{\mathcal{I}\in\{1\ldots K\}}\left[\frac{1}{K'}\sum_{k=1}^{K'}\frac{p(\mathbf{h}_k,\mathbf{x})^{\alpha+\beta}}{q(\mathbf{h}_k|\mathbf{x})}\right]\right] \\
&\qquad + \frac{1}{\beta(\alpha+\beta)}\mathbb{E}_{\mathbf{h}_k}\left[\log\mathbb{E}_{\mathcal{I}\in\{1\ldots K\}}\left[\frac{1}{K'}\sum_{k=1}^{K'}q(\mathbf{h}_k|\mathbf{x})^{\alpha+\beta-1}\right]\right] \\
&\qquad - \frac{1}{\alpha\beta}\mathbb{E}_{\mathbf{h}_k}\left[\log\mathbb{E}_{\mathcal{I}\in\{1\ldots K\}}\left[\frac{1}{K'}\sum_{k=1}^{K'}\left[q(\mathbf{h}_k|\mathbf{x})^{\alpha+\beta-1}\left(\frac{p(\mathbf{h}_k,\mathbf{x})}{q(\mathbf{h}_k|\mathbf{x})}\right)^{\beta}\right]\right]\right] \\
&\geq \frac{1}{\alpha(\alpha+\beta)}\mathbb{E}_{\mathbf{h}_k}\left[\mathbb{E}_{\mathcal{I}\in\{1\ldots K\}}\left[\log\frac{1}{K'}\sum_{k=1}^{K'}\frac{p(\mathbf{h}_k,\mathbf{x})^{\alpha+\beta}}{q(\mathbf{h}_k|\mathbf{x})}\right]\right] \\
&\qquad + \frac{1}{\beta(\alpha+\beta)}\mathbb{E}_{\mathbf{h}_k}\left[\mathbb{E}_{\mathcal{I}\in\{1\ldots K\}}\left[\log\frac{1}{K'}\sum_{k=1}^{K'}q(\mathbf{h}_k|\mathbf{x})^{\alpha+\beta-1}\right]\right] \\
&\qquad - \frac{1}{\alpha\beta}\mathbb{E}_{\mathbf{h}_k}\left[\mathbb{E}_{\mathcal{I}\in\{1\ldots K\}}\left[\log\frac{1}{K'}\sum_{k=1}^{K'}\left[q(\mathbf{h}_k|\mathbf{x})^{\alpha+\beta-1}\left(\frac{p(\mathbf{h}_k,\mathbf{x})}{q(\mathbf{h}_k|\mathbf{x})}\right)^{\beta}\right]\right]\right] \\
&= \frac{1}{\alpha(\alpha+\beta)}\mathbb{E}_{\mathbf{h}_k}\left[\log\frac{1}{K'}\sum_{k=1}^{K'}\frac{p(\mathbf{h}_k,\mathbf{x})^{\alpha+\beta}}{q(\mathbf{h}_k|\mathbf{x})}\right] \\
&\qquad + \frac{1}{\beta(\alpha+\beta)}\mathbb{E}_{\mathbf{h}_k}\left[\log\frac{1}{K'}\sum_{k=1}^{K'}q(\mathbf{h}_k|\mathbf{x})^{\alpha+\beta-1}\right] \\
&\qquad - \frac{1}{\alpha\beta}\mathbb{E}_{\mathbf{h}_k}\left[\log\frac{1}{K'}\sum_{k=1}^{K'}\left[q(\mathbf{h}_k|\mathbf{x})^{\alpha+\beta-1}\left(\frac{p(\mathbf{h}_k,\mathbf{x})}{q(\mathbf{h}_k|\mathbf{x})}\right)^{\beta}\right]\right] \\
&= \mathbb{E}_{\{\mathbf{h}_k\}_1^{K'}}\left[\hat{D}_{sAB}^{\alpha,\beta,K'}(q(.)||p(.|\mathbf{x}))\right]
\end{aligned}
$$

This uses Jensen's inequality of logarithm as well as uses the fact that both $\hat{D}_{sAB}^{\alpha,\beta,K}(q(.)||p(.|\mathbf{x}))$ and $\hat{D}_{sAB}^{\alpha,\beta,K'}(q(.)||p(.|\mathbf{x}))$ are positive.

Next we prove that the Monte-Carlo approximation converge towards

the exact divergence.
i.e. when $K \to +\infty$, $\mathbb{E}_{\{\mathbf{h}_k\}_1^K}\left[\hat{D}_{sAB}^{\alpha,\beta,K}(q(.)||p(.|\mathbf{x}))\right] \to D_{sAB}^{\alpha,\beta}(q(.)||p(.))$

.6 Experiments

We here provide a more detailed description of our experimental setups. The following experiments have been implemented using *tensorflow* [152] and *Edward* [161].

.6.1 Regression on synthetic dataset

In this experiment we create a toy dataset to showcase the strength of the sAB variational objective.

The non-corrupted data are generated by the following process,

$$y = \mathbf{w}\mathbf{X} + \mathcal{N}(0,0.1)$$

with $\mathbf{w} = [1/2...1/2]$ a D-dimensional vector and bfX a set of points randomly distributed between $[-1,1]^D$.
A given percentage $p_{outliers}$ of the data are corrupted and follows the process,

$$y = 5 + \mathbf{w}\mathbf{X} + \mathcal{N}(0,0.1)$$

with $\mathbf{w} = [1/2...1/2]$ and $\mathbf{X}$ is sampled from $\mathcal{N}(0,0.2)$.

For N such data points $[(\mathbf{x}_n, y_n)]_{n\in[1,N]}$, we uses the following distributions,

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid \mathbf{0}, \sigma_w^2 \mathcal{I}_D),$$
$$p(b) = \mathcal{N}(b \mid 0, \sigma_b^2),$$

and

$$p(y \mid \mathbf{w}, b, \mathbf{X}) = \prod_{n=1}^{N} \mathcal{N}(y_n \mid \mathbf{x}_n^\top \mathbf{w} + b, \sigma_y^2).$$

We define the variational model to be a fully factorized normal across the weights.

For the experiments presented in Section 4.5 we use $N = 1000$, $D = 4$ and $p_{outliers} = 5\%$.

We train the model using ADAM [139] with learning rate of 0.01 for 1000 steps. We use 5 MC samples to evaluate the divergence.

.6.2 UCI datasets regression

We use here a Bayesian neural network regression model with Gaussian likelihood on datasets collected from the UCI dataset repository [127]. We also artificially corrupt part of the outputs in the training data to test the influence of outliers. The corruption is achieved by randomly adding 5 standard deviation to $p_{outliers}\%$ of the points after normalization.

For all the experiments, we use a two-layers neural network with 50 hidden units with ReLUs activation functions. We use a fully factorized Gaussian approximation to the true posterior $q(\theta)$. Independent standard Gaussian priors are given to each of the network weights. The model is optimized using ADAM [139] with learning rate of 0.01 and the standard settings for the other parameters for 500 epochs. We perform nested cross-validations [95] where the inner validation is used to select the optimal parameters α and β within the $[-0.5, 2.5] \times [-1.5, 1.5]$ (with step 0.25). The best model selected from the inner loop is then re-trained on the complete outer split. We use 25 MC samples to evaluate the divergence. The outer cross validation used $K_1) = 10$ folds and the inner one uses $K_2) = 2$ folds.

Bibliography

[1] P. C. Mahalanobis. “On the generalized distance in statistics”. In: National Institute of Science of India. 1936.

[2] S. Kullback and R. A. Leibler. “On information and sufficiency”. In: *The annals of mathematical statistics* 22.1 (1951), pp. 79–86.

[3] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. “Equation of state calculations by fast computing machines”. In: *The journal of chemical physics* 21.6 (1953), pp. 1087–1092.

[4] B. P. Adhikari and D. D. Joshi. *Distance, discrimination et résumé exhaustif*. 1956.

[5] J. G. Kemeny and J. L. Snell. *Finite Markov chains*. Vol. 356. van Nostrand Princeton, NJ, 1960.

[6] A. Rényi et al. “On measures of entropy and information”. In: *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California. 1961.

[7] R. Bracewell. “The Fourier Transform and its Applications”. In: *New York* (1965).

[8] L. E. Baum and T. Petrie. “Statistical inference for probabilistic functions of finite state Markov chains”. In: *The annals of mathematical statistics* 37.6 (1966), pp. 1554–1563.

[9] T. M. Cover and P. E. Hart. “Nearest neighbor pattern classification”. In: *Information Theory, IEEE Transactions on* 13.1 (1967), pp. 21–27.

[10] A. J. Viterbi. "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". In: *Information Theory, IEEE Transactions on* 13.2 (1967), pp. 260–269.

[11] J. Omura. "On the Viterbi decoding algorithm". In: *IEEE Transactions on Information Theory* 15.1 (1969), pp. 177–179.

[12] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains". In: *The annals of mathematical statistics* (1970), pp. 164–171.

[13] W. K. Hastings. "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika* 57.1 (1970), pp. 97–109.

[14] L. Hörmander. "Fourier integral operators. I". In: *Acta mathematica* 127.1 (1971), pp. 79–183.

[15] G. D. Forney Jr. "The viterbi algorithm". In: *Proceedings of the IEEE* 61.3 (1973), pp. 268–278.

[16] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the royal statistical society. Series B (methodological)* (1977), pp. 1–38.

[17] L. B. Rall. "Automatic differentiation: Techniques and applications". In: (1981).

[18] S. Geman and D. Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images". In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1984), pp. 721–741.

[19] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. "A learning algorithm for Boltzmann machines*". In: *Cognitive science* 9.1 (1985), pp. 147–169.

[20] P. A. Devijver. "Baum's forward-backward algorithm revisited". In: *Pattern Recognition Letters* 3.6 (1985), pp. 369–373.

[21] R. N. Bracewell and R. N. Bracewell. *The Fourier transform and its applications*. Vol. 31999. McGraw-Hill New York, 1986.

[22] Y. Sakamoto, M. Ishiguro, and G. Kitagawa. "Akaike information criterion statistics". In: *Dordrecht, The Netherlands: D. Reidel* 81 (1986).

[23] P. Smolensky. "Information processing in dynamical systems: Foundations of harmony theory". In: (1986).

[24] A. E. Gelfand and A. F. Smith. "Sampling-based approaches to calculating marginal densities". In: *Journal of the American statistical association* 85.410 (1990), pp. 398–409.

[25] Y. Yuan. "A modified BFGS algorithm for unconstrained optimization". In: *IMA Journal of Numerical Analysis* 11.3 (1991), pp. 325–332.

[26] N. S. Altman. "An introduction to kernel and nearest-neighbor nonparametric regression". In: *The American Statistician* 46.3 (1992), pp. 175–185.

[27] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. "Image coding using wavelet transform". In: *IEEE Transactions on image processing* 1.2 (1992), pp. 205–220.

[28] R. A. DeVore, B. Jawerth, and B. J. Lucier. "Image compression through wavelet transform coding". In: *Information Theory, IEEE Transactions on* 38.2 (1992), pp. 719–746.

[29] P. P. Shenoy. "Valuation-based systems for Bayesian decision analysis". In: *Operations research* 40.3 (1992), pp. 463–484.

[30] D. L. Donoho. "Unconditional bases are optimal bases for data compression and for statistical estimation". In: *Applied and computational harmonic analysis* 1.1 (1993), pp. 100–115.

[31] B. G. Lindsay. "Efficiency versus robustness: the case for minimum Hellinger distance and related methods". In: *The annals of statistics* (1994), pp. 1081–1114.

[32] R. D. Shachter, S. K. Andersen, and P. Szolovits. "Global conditioning for probabilistic inference in belief networks". In: *Uncertainty Proceedings 1994*. Elsevier, 1994, pp. 514–522.

[33] Y. LeCun and Y. Bengio. "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361.10 (1995).

[34] S. Z. Li. "On discontinuity-adaptive smoothness priors in computer vision". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.6 (1995), pp. 576–586.

[35] J. C. Brailean and A. K. Katsaggelos. "Recursive map displacement field estimation and its applications". In: *Image Processing, 1996. Proceedings., International Conference on*. Vol. 1. IEEE. 1996, pp. 917–920.

[36] F. V. Jensen. *An introduction to Bayesian networks*. Vol. 210. UCL press London, 1996.

[37] L. M. Kaplan and C. .-.-C. J. Kuo. "An Improved Method for 2-D Self-Similar Image Synthesis". In: *IEEE Transactions on Image Processing* 5.5 (1996), pp. 754–761.

[38] N. Lee, Q. Huynh, and S. Schwartz. "New method of linear time-frequency analysis for signal detection". In: *Time-Frequency and Time-Scale Analysis, 1996., Proceedings of the IEEE-SP International Symposium on*. IEEE. 1996, pp. 13–16.

[39] T. Moon. "The expectation-maximization algorithm". In: *Signal processing magazine, IEEE* 13.6 (1996), pp. 47–60.

[40] C. E. Kahn Jr, L. M. Roberts, K. A. Shaffer, and P. Haddawy. "Construction of a Bayesian network for mammographic diagnosis of breast cancer". In: *Computers in biology and medicine* 27.1 (1997), pp. 19–29.

[41] B. Schölkopf, K.-K. Sung, C. J. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik. "Comparing support vector machines with Gaussian kernels to radial basis function classifiers". In: *Signal Processing, IEEE Transactions on* 45.11 (1997), pp. 2758–2765.

[42] A. Basu, I. R. Harris, N. L. Hjort, and M. Jones. "Robust and efficient estimation by minimising a density power divergence". In: *Biometrika* 85.3 (1998), pp. 549–559.

[43] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk. "Wavelet-based statistical signal processing using hidden Markov models". In: *Signal Processing, IEEE Transactions on* 46.4 (1998), pp. 886–902.

[44] D. Heckerman. *A tutorial on learning with Bayesian networks*. Springer, 1998.

[45] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[46] W. Lohmiller and J.-J. E. Slotine. "On contraction analysis for nonlinear systems". In: *Automatica* 34.6 (1998), pp. 683–696.

[47] N. J. Nilsson. *Artificial intelligence: a new synthesis*. Morgan Kaufmann, 1998.

[48] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. "When is "nearest neighbor" meaningful?" In: *Database Theory—ICDT'99*. Springer, 1999, pp. 217–235.

[49] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. "An introduction to variational methods for graphical models". In: *Machine learning* 37.2 (1999), pp. 183–233.

[50] S. Mallat. *A wavelet tour of signal processing*. Academic press, 1999.

[51] J. Platt et al. “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods”. In: *Advances in large margin classifiers* 10.3 (1999), pp. 61–74.

[52] P. De Chazal, B. Celler, and R. Reilly. “Using wavelet coefficients for the classification of the electrocardiogram”. In: *Engineering in Medicine and Biology Society, 2000. Proceedings of the 22nd Annual International Conference of the IEEE*. Vol. 1. IEEE. 2000, pp. 64–67.

[53] S. Deguy, C. Debain, and A. Benassi. “Classification of Texture Images using Multi-scale Statistical Estimators of Fractal Parameters”. In: *British Machine Vision Conference* (2000).

[54] T. E. Duncan, Y. Hu, and B. Pasik-Duncan. “Stochastic calculus for fractional Brownian motion I. Theory”. In: *SIAM Journal on Control and Optimization* 38.2 (2000), pp. 582–612.

[55] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*. Vol. 1. Springer series in statistics New York, 2001.

[56] S.-T. Hsiang. “Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling”. In: *Data Compression Conference, 2001. Proceedings. DCC 2001.* IEEE. 2001, pp. 83–92.

[57] N. Kingsbury. “Complex wavelets for shift invariant analysis and filtering of signals”. In: *Applied and computational harmonic analysis* 10.3 (2001), pp. 234–253.

[58] P. Abry, R. Baraniuk, P. Flandrin, R. Riedi, and D. Veitch. “Multiscale nature of network traffic”. In: *IEEE Transactions on Signal Processing Magazine* 19 (2002), pp. 28–46.

[59] P. Abry, P. Flandrin, M. S. Taqqu, and D. Veitch. “Self-similarity and long-range dependence through the wavelet lens”. In: ed. by P. Doukhan, G. Oppenheim, and M. S. Taqqu. Birkhäuser, 2002, pp. 527–556.

[60] Y. Ephraim and N. Merhav. "Hidden Markov processes". In: *Information Theory, IEEE Transactions on* 48.6 (2002), pp. 1518–1569.

[61] A. Jordan. "On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes". In: *Advances in neural information processing systems* 14 (2002), p. 841.

[62] S. Z. Li. "Roof-edge preserving image smoothing based on MRFs". In: *IEEE Transactions on Image Processing* 9 (6 2002), pp. 1134–1138.

[63] B. Pesquet-Popescu and J. L. Véhel. "Stochastic fractal models for image processing". In: *IEEE Signal Processing Magazine* 19.5 (2002), pp. 48–62.

[64] A. Pižurica, W. Philips, I. Lemahieu, and M. Acheroy. "A Joint Inter- and Intrascale Statistical Model for Bayesian Wavelet Based Image Denoising". In: *IEEE Trans. Image Processing* 11.5 (2002), pp. 545–557.

[65] I. Rish, M. Brodie, and S. Ma. "Efficient fault diagnosis using probing". In: *AAAI Spring Symposium on Information Refinement and Revision for Decision Making*. 2002.

[66] J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, M. West, et al. "The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures". In: *Bayesian statistics* 7 (2003), pp. 453–464.

[67] T. Haveliwala and S. Kamvar. "The second eigenvalue of the Google matrix". In: *Stanford University Technical Report* (2003).

[68] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski. "Dictionary learning algorithms for sparse representation". In: *Neural computation* 15.2 (2003), pp. 349–396.

[69] D. Margaritis. "Learning Bayesian network model structure from data". PhD thesis. US Army, 2003.

[70] P. Y. Simard, D. Steinkraus, and J. C. Platt. "Best practices for convolutional neural networks applied to visual document analysis". In: *null*. IEEE. 2003, p. 958.

[71] J.-B. Durand, P. Goncalves, and Y. Guédon. "Computational methods for hidden Markov tree models-An application to wavelet trees". In: *Signal Processing, IEEE Transactions on* 52.9 (2004), pp. 2551–2560.

[72] D. G. Lowe. "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[73] C. P. Robert and G. Casella. "Monte Carlo Optimization". In: *Monte Carlo Statistical Methods*. Springer, 2004, pp. 157–204.

[74] W. R. Gilks. *Markov chain monte carlo*. Wiley Online Library, 2005.

[75] T. Minka. *Divergence measures and message passing*. Tech. rep. Technical report, Microsoft Research, 2005.

[76] I. W. Selesnick, R. G. Baraniuk, and N. G. Kingsbury. "The Dual-Tree Complex Wavelet Transform". In: *IEEE Signal Processing Magazine* 22.6 (2005), pp. 123–151.

[77] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006. Chap. 8, pp. 359–422.

[78] N. Dasgupta and L. Carin. "Texture analysis with variational hidden Markov trees". In: *IEEE transactions on signal processing* 54.6 (2006), pp. 2353–2356.

[79] L. Fei-Fei, R. Fergus, and P. Perona. "One-shot learning of object categories". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28.4 (2006), pp. 594–611.

[80] S. Ji, B. Krishnapuram, and L. Carin. "Variational Bayes for continuous hidden Markov models and its application to active learning". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.4 (2006), pp. 522–532.

[81] T. Ando. "Bayesian predictive information criterion for the evaluation of hierarchical Bayesian and empirical Bayes models". In: *Biometrika* 94.2 (2007), pp. 443–458.

[82] S. Jaffard, B. Lashermes, and P. Abry. "Wavelet Leaders in Multifractal Analysis". In: *Wavelet Analysis and Applications*. Ed. by M. I. V. T. Qian and X. Yuesheng. Applied and Numerical Harmonic Analysis. Birkhäuser, 2007, pp. 201–246.

[83] A. Echelard and J. L. Véhel. "Wavelet denoising based on local regularity information". In: *Proceedings of the European Signal Processing Program* (2008).

[84] H. Fujisawa and S. Eguchi. "Robust parameter estimation with a small bias against heavy contamination". In: *Journal of Multivariate Analysis* 99.9 (2008), pp. 2053–2081.

[85] P. Kestener and A. Arneodo. "A multifractal formalism for vector-valued random fields based on wavelet analysis: application to turbulent velocity and vorticity 3D numerical data". In: *Stochastic Environmental Research and Risk Assessment* 22.3 (2008), pp. 421–435.

[86] T. Lin and H. Zha. "Riemannian manifold learning". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 30.5 (2008), pp. 796–809.

[87] M. J. Wainwright and M. I. Jordan. "Graphical models, exponential families, and variational inference". In: *Foundations and Trends® in Machine Learning* 1.1-2 (2008), pp. 1–305.

[88] P. Abry, P. Gonçalvès, and J. L. Véhel. "Scaling Fractals and wavelets". In: Wiley, 2009.

[89] Dstl. *DSTL datasets*. Accessed: 09-02-2016. 2009.

[90] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. "What is the best multi-stage architecture for object recognition?" In: *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE. 2009, pp. 2146–2153.

[91] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[92] C. A. McGrory and D. Titterington. "Variational Bayesian analysis for hidden Markov models". In: *Australian & New Zealand Journal of Statistics* 51.2 (2009), pp. 227–244.

[93] V. Olariu, D. Coca, S. A. Billings, P. Tonge, P. Gokhale, P. W. Andrews, and V. Kadirkamanathan. "Modified variational Bayes EM estimation of hidden Markov tree model of cell lineages". In: *Bioinformatics* 25.21 (2009), pp. 2824–2830.

[94] J. Bruna and S. Mallat. "Classification with scattering operators". In: *arXiv preprint arXiv:1011.3023* (2010).

[95] G. C. Cawley and N. L. Talbot. "On over-fitting in model selection and subsequent selection bias in performance evaluation". In: *Journal of Machine Learning Research* 11.Jul (2010), pp. 2079–2107.

[96] A. Cichocki and S.-i. Amari. "Families of alpha-beta-and gamma-divergences: Flexible and robust measures of similarities". In: *Entropy* 12.6 (2010), pp. 1532–1568.

[97] X. Glorot and Y. Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 2010, pp. 249–256.

[98] Y. LeCun, K. Kavukcuoglu, and C. Farabet. "Convolutional networks and applications in vision". In: *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*. IEEE. 2010, pp. 253–256.

[99] J. D. B. Nelson and N. G. Kingsbury. "Dual-tree wavelets for estimation of locally varying and anisotropic fractal dimension". In: *IEEE International Conference on Image Processing* (2010), pp. 341–344.

[100] J. Nelson and N. Kingsbury. "Fractal dimension based sand ripple suppression for mine hunting with sidescan sonar". In: *International conference on synthetic aperture sonar and synthetic aperture radar*. 2010.

[101] R. Salakhutdinov, J. Tenenbaum, and A. Torralba. "One-shot learning with a hierarchical nonparametric Bayesian model". In: (2010).

[102] R. Salakhutdinov and H. Larochelle. "Efficient learning of deep Boltzmann machines". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 2010, pp. 693–700.

[103] E. B. Sudderth, A. T. Ihler, M. Isard, W. T. Freeman, and A. S. Willsky. "Nonparametric belief propagation". In: *Communications of the ACM* 53.10 (2010), pp. 95–103.

[104] E. Tola. "DAISY: A Fast Descriptor for Dense Wide Baseline Stereo and Multiview Reconstruction". PhD thesis. Citeseer, 2010.

[105] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion". In: *Journal of Machine Learning Research* 11.Dec (2010), pp. 3371–3408.

[106] J. Andén and S. Mallat. "Multiscale Scattering for Audio Classification." In: *ISMIR*. 2011, pp. 657–662.

[107] A. Cichocki, S. Cruces, and S.-i. Amari. "Generalized alpha-beta divergences and their application to robust nonnegative matrix factorization". In: *Entropy* 13.1 (2011), pp. 134–170.

[108] C. Févotte and J. Idier. "Algorithms for nonnegative matrix factorization with the β-divergence". In: *Neural computation* 23.9 (2011), pp. 2421–2456.

[109] V. Franc, A. Zien, and B. Schölkopf. "Support Vector Machines as Probabilistic Models." In: *ICML*. 2011, pp. 665–672.

[110] R. E. Turner and M. Sahani. "Two problems with variational expectation maximisation for time-serie models". In: *Bayesian Time series models* (2011), pp. 115–138.

[111] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn. "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition". In: *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE. 2012, pp. 4277–4280.

[112] S.-i. Amari. *Differential-geometrical methods in statistics*. Vol. 28. Springer Science & Business Media, 2012.

[113] J. Bruna. "Operators commuting with diffeomorphisms". In: *CMAP Tech. Report, Ecole Polytechnique* (2012).

[114] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. "Improving neural networks by preventing co-adaptation of feature detectors". In: *arXiv preprint arXiv:1207.0580* (2012).

[115] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[116] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[117] R. M. Neal. *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media, 2012.

[118] Y. K. Yilmaz and A. T. Cemgil. “Alpha/beta divergences and tweedie models”. In: *arXiv preprint arXiv:1209.4280* (2012).

[119] Z. Zhang, J. Wang, and H. Zha. “Adaptive manifold learning”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34.2 (2012), pp. 253–265.

[120] R. Bellman. *Dynamic programming*. Courier Corporation, 2013.

[121] J. Bruna and S. Mallat. “Invariant scattering convolution networks”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.8 (2013), pp. 1872–1886.

[122] M. Chen and J. Strobi. “Multispectral textured image segmentation using a multi-resolution fuzzy Markov random field model on variable scales in the wavelet domain”. In: *International journal of remote sensing* 34 (13 2013), pp. 4550–4569.

[123] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. “Stochastic variational inference”. In: *The Journal of Machine Learning Research* 14.1 (2013), pp. 1303–1347.

[124] M. Julian, R. Alcaraz, and J. Rieta. “Study on the optimal use of Generalized Hurst Exponents for noninvasive estimation of atrial fibrillation organization”. In: *Computing in Cardiology Conference* (2013), pp. 1039–1042.

[125] D. P. Kingma and M. Welling. “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114* (2013).

[126] L. Kristoufek and M. Vosvrda. “Measuring capital market efficiency: Global and local correlations structure”. In: *Physica A: Statistical Mechanics and its Applications* 392 (1 2013), pp. 184–193.

[127] M. Lichman. *UCI Machine Learning Repository*. 2013. URL: `http://archive.ics.uci.edu/ml`.

[128] C. Nafornita and A. Isar. “Estimating directional smoothness of images with the aid of the hyperanalytic wavelet packet transform”. In: *International Symposium on Signals, Circuits, and Systems* (2013).

[129] E. Oyallon, S. Mallat, and L. Sifre. “Generic deep networks with wavelet scattering”. In: *arXiv preprint arXiv:1312.5940* (2013).

[130] L. Sifre and S. Mallat. *Rotation, Scaling and Deformation Invariant Scattering for Texture Discrimination*. 2013.

[131] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. “Intriguing properties of neural networks”. In: *arXiv preprint arXiv:1312.6199* (2013).

[132] Y. K. Yilmaz. “Generalized Beta Divergence”. In: *arXiv preprint arXiv:1306.3530* (2013).

[133] M. J. Baker et al. “Using Fourier transform IR spectroscopy to analyze biological materials”. In: *Nature protocols* 9.8 (2014), pp. 1771–1791.

[134] V. Chudáček, J. Andén, S. Mallat, P. Abry, and M. Doret. “Scattering Transform for Intrapartum Fetal Heart Rate Variability Fractal Analysis: A Case-Control Study”. In: *IEEE Transactions on Biomedical Engineering* (2014), pp. 1100–1108.

[135] V. Chudacek, R. Talmon, J. Anden, S. Mallat, R. Coifman, P. Abry, and M. Doret. “Low dimensional manifold embedding for scattering coefficients of intrapartum fetale heart rate variability”. In: *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*. IEEE. 2014, pp. 6373–6376.

[136] N. Foti, J. Xu, D. Laird, and E. Fox. “Stochastic variational inference for hidden Markov models”. In: *Advances in neural information processing systems*. 2014, pp. 3599–3607.

[137] I. J. Goodfellow, J. Shlens, and C. Szegedy. “Explaining and harnessing adversarial examples”. In: *arXiv preprint arXiv:1412.6572* (2014).

[138] M. D. Hoffman and A. Gelman. “The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo.” In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1593–1623.

[139] D. P. Kingma and J. Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[140] C. Nafornita, A. Isar, and J. D. B. Nelson. “Regularised, semi-local Hurst estimation via generalised lasso and dual-tree complex wavelets”. In: *IEEE International Conference on Image Processing* (2014), pp. 2689–2693.

[141] F. Nielsen and R. Nock. “On the chi square and higher-order chi distances for approximating f-divergences”. In: *IEEE Signal Processing Letters* 21.1 (2014), pp. 10–13.

[142] E. Oyallon and S. Mallat. “Deep roto-translation scattering for object classification”. In: *arXiv preprint arXiv:1412.8659* (2014).

[143] R. Ranganath, S. Gerrish, and D. Blei. “Black box variational inference”. In: *Artificial Intelligence and Statistics*. 2014, pp. 814–822.

[144] I. S. C. d. P. Regularity Team. *Fraclab Toolbox*. `http://fraclab.saclay.inria.fr`. 2014.

[145] L. Sifre and S. Mallat. “Rigid-motion scattering for image classification”. PhD thesis. Citeseer, 2014.

[146] T. Van Erven and P. Harremos. “Rényi divergence and Kullback-Leibler divergence”. In: *IEEE Transactions on Information Theory* 60.7 (2014), pp. 3797–3820.

[147] M. T. R. AUEB and M. Lázaro-Gredilla. “Local expectation gradients for black box variational inference”. In: *Advances in neural information processing systems*. 2015, pp. 2638–2646.

[148] J. Bruna, S. Mallat, E. Bacry, J.-F. Muzy, et al. "Intermittent process analysis with scattering moments". In: *The Annals of Statistics* 43.1 (2015), pp. 323–351.

[149] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen. "Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification". In: *International conference on machine learning*. 2015, pp. 720–729.

[150] B. Huval et al. "An empirical evaluation of deep learning on highway driving". In: *arXiv preprint arXiv:1504.01716* (2015).

[151] Y. LeCun. *Personal webpage: State of the art on MNIST*. Accessed: 09-02-2016. 2015.

[152] Martın Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[153] J.-B. Regli and J. Nelson. "Piecewise parameterised Markov random fields for semi-local Hurst estimation". In: *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE. 2015, pp. 1626–1630.

[154] I. Waldspurger, A. d'Aspremont, and S. Mallat. "Phase recovery, maxcut and complex semidefinite programming". In: *Mathematical Programming* 149.1-2 (2015), pp. 47–81.

[155] S. Depeweg, J. M. Hernández-Lobato, F. Doshi-Velez, and S. Udluft. "Learning and policy search in stochastic dynamical systems with bayesian neural networks". In: *arXiv preprint arXiv:1605.07127* (2016).

[156] A. Ghosh and A. Basu. "Robust Bayes estimation using the density power divergence". In: *Annals of the Institute of Statistical Mathematics* 68.2 (2016), pp. 413–437.

[157] J. M. Hernández-Lobato, Y. Li, M. Rowland, D. Hernández-Lobato, T. D. Bui, and R. E. Turner. “Black-box α-divergence minimization”. In: (2016).

[158] Y. Li and R. E. Turner. “Rényi divergence variational inference”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 1073–1081.

[159] R. Ranganath, D. Tran, and D. Blei. “Hierarchical variational models”. In: *International Conference on Machine Learning*. 2016, pp. 324–333.

[160] J.-B. Regli and J. Nelson. “Scattering convolutional hidden Markov trees”. In: *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE. 2016, pp. 1883–1887.

[161] D. Tran, A. Kucukelbir, A. B. Dieng, M. Rudolph, D. Liang, and D. M. Blei. “Edward: A library for probabilistic modeling, inference, and criticism”. In: *arXiv preprint arXiv:1610.09787* (2016).

[162] F. Futami, I. Sato, and M. Sugiyama. “Variational Inference based on Robust Divergences”. In: *arXiv preprint arXiv:1710.06595* (2017).

[163] A. Ghosh, I. R. Harris, A. Maji, A. Basu, L. Pardo, et al. “A generalized divergence for statistical inference”. In: *Bernoulli* 23.4A (2017), pp. 2746–2783.

[164] A. Singh and N. Kingsbury. “Dual-tree wavelet scattering network with parametric log transformation for object classification”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE. 2017, pp. 2622–2626.

[165] A. Singh and N. Kingsbury. “Scatternet hybrid deep learning (shdl) network for object classification”. In: *Machine Learning for Signal Processing (MLSP), 2017 IEEE 27th International Workshop on*. IEEE. 2017, pp. 1–6.

[166] T. Rainforth, A. R. Kosiorek, T. A. Le, C. J. Maddison, M. Igl, F. Wood, and Y. W. Teh. "Tighter variational bounds are not necessarily better". In: *arXiv preprint arXiv:1802.04537* (2018).

[167] J.-B. Regli and R. Silva. "Alpha-Beta Divergence For Variational Inference". In: *arXiv preprint arXiv:1805.01045* (2018).

[168] S. Mallat. "Group Invariant Scattering". In: *Communications in Pure and Applied Mathematics* 65.10 (Oct. 2012), pp. 1331–1398.