SCATTERING HIDDEN MARKOV TREES

IMAGE REPRESENTATION AND SCATTERING TRANSFORM MODELING

Jean-Baptiste REGLI

2013-2014

RESEARCH REPORT

Academic supervisor : James Nelson
Sponsor : Dstl/UCL Impact studentship

UCL
Department of Statistical Science
London

Contents :

List of figures :

1 Introduction :

1.1 Need for a better signal representation :

This document describes researches focused on ***high-dimensional classification problems***. Meaning cases where several -says N- realizations of a real signal $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are observed. In other words, cases where,

$$\forall i \in [\![1, N]\!] \quad \mathbf{x}_i = (x(1) \ \ldots \ x(d)) \qquad \text{with} \ \ d \sim 10^6 \text{ and } x(.) \in \mathbb{R}.$$

And the task at hand is ***learning the labeling function*** -says f- given N labeled sampled values -training examples- $\{x_i, y_i = f(x_i)\}_{i \leq N}$.

A naive solution to this problem would be to infer the class of a new realization $\mathbf{x}$ by looking at its neighbors, e.g. K-Nearest Neighbors (KNN). This approach is working fine in the case of low dimensional problems (***citation KNN good perf***). However it shows limitations in high dimensional cases (***citation KNN curse of dimensional***), because the number of sampled values of the signal needed to find a neighbor to a new realization $\mathbf{x}^{new}$ grow exponentially with the number of dimensions.

To over come this issue one could try to reduce the number of dimensions of the problem.

1.2 Image representation :

Hence for high-dimensional problem it can be interesting to project the signal into a new space where the classification task would be simpler. But to ensure the simplification of classification task the projection has to be into a "good" representation space. This idea of "good" representation of the data is central to this work. One can first provide an intuition of what "good" is and then formalize the intuited properties with mathematical concepts.

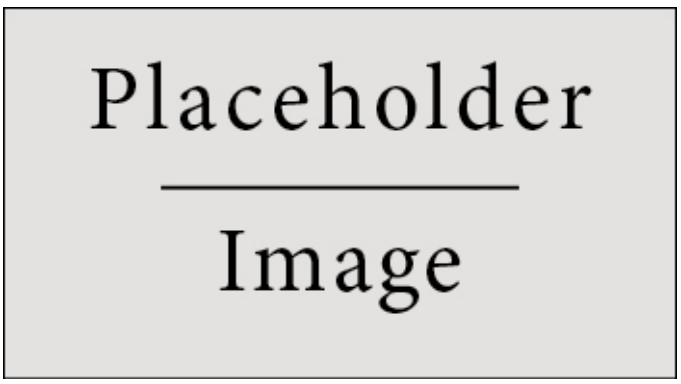

FIGURE 1.1 – (a) Sound waveform (b) Picture

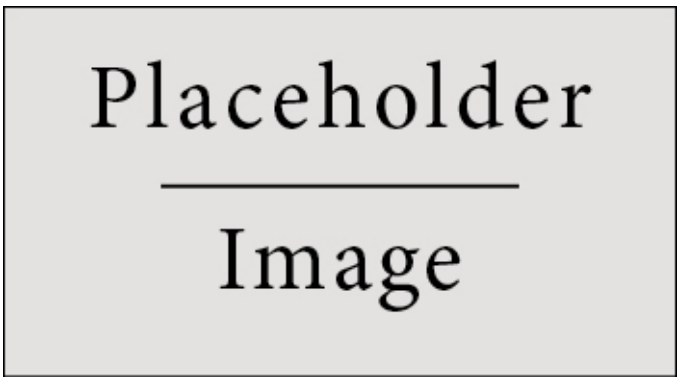

FIGURE 1.2 – A human can easily tell that those two images are from the same class.

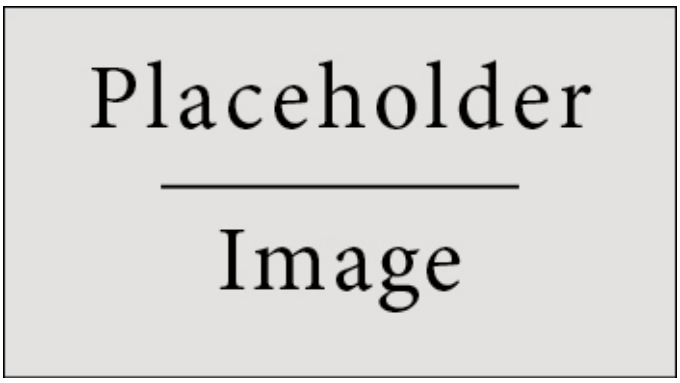

FIGURE 1.3 – A human can easily tell that (a) and (b) are from the same class. (c) can still be recognized even though it is slightly more challenging.

1.2.1 Intuition of a "good" image representation :

One way to develop an intuition on what properties a "good" representation for classification have is to look at how a human handle the classification/labeling of visual stimulus and what are the properties ensuring good generalization capacities. Following this approach one would like the projection to be :

— ***Informative*** enough to permit classification. This is quite obvious but one should ensure that the projection preserve discriminabilty between the different classes.

— ***Invariant to translations***. Indeed to a human eye there is no difference in the information carried by a signal if it is shifted. This means that the projection has to provide the similar -if not equal- outputs for shifted versions of the same signal.

— ***Stable to deformations***. Once again to a human eye, it is still possible to recognize a signal if it has undergone -small- deformations. Yet if the deformations is too important the informational content of the signal is lost. This means that to a certain degree the answer of the projection to morphed realizations of the same signal should be similar. However one need to define a limit to this invariance to ensure that the representation created is still informative enough.

— ***To a certain degree invariant to rotations***. The case of rotations is not as simple as the translations. Indeed one is after a local rotational invariance rather than a global one. This is true because excessive rotation applied to the original signal can be destructive for the information carried. Solutions based on the method described in this document exist ***citation*** but this will not be addressed in this review.

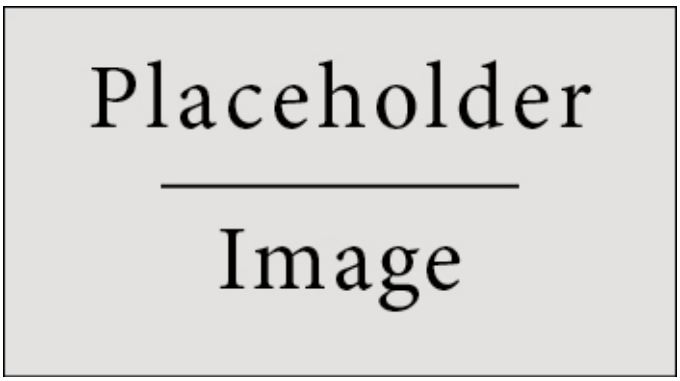

FIGURE 1.4 – A human can easily tell that (a) and (b) are from the same class. (c) could be a '6' slightly rotated or a '9' heavily rotated.

1.2.2 Formalization of a "good" image representation :

Those qualitative intuitions over the properties of a "good" representation are interesting but they need to be formalize to provide leads for the construction of the projection. To do so one first need to define what a signal is. Throughout this document, a signal will be defined as,

Definition 1.2.1. Signal
A signal f is a square-integrable d dimensional real function.

$$f \in \mathcal{L}^2(\mathbb{R}^d).$$

Let also $\mathbf{x}$ be the realisation of the signal for a given input z, i.e. $\mathbf{x} = f(z)$.

To be informative enough, a representation has to preserve separability between elements of different classes. Formally this is,

Proposition 1.2.1. ***Discriminability preservation*** *A representation Φ preserves discriminabilty if all elements of two different classes are distance of margin C in the representation space, i.e. :*

$$\forall (x, x') \in \left(\mathbb{R}^d\right)^2 \ \exists C \in \mathbb{R} \ \mid \ \left|f(x) - f(x')\right| = 1 \ \Rightarrow \ \left\|\Phi(x) - \Phi(x')\right\| \geq C^{-1}$$

The class associated to a representation of a signal appears to be invariant to small shifts. In this document we call $L_{(.)}$ the translation operator for the function in $\mathcal{L}^2(\mathbb{R}^d)$, **i.e.** for $f \in \mathcal{L}^2(\mathbb{R}^d)$ and $(x, c) \in (\mathbb{R}^d)^2 \quad L_c f(x) = f(x - c)$. An operator is translation invariant -resp : canonical translation invariant- if,

Proposition 1.2.2. ***Translation invariant***
An operator $\Phi : \mathcal{L}^2(\mathbb{R}^d) \to \mathcal{H}$ where $\mathcal{H}$ is an Hilbert space is translation invariant if :

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

Proposition 1.2.3. ***Canonical translation invariant***
An operator $\Phi : \mathcal{L}^2(\mathbb{R}^d) \to \mathcal{H}$ where $\mathcal{H}$ is an Hilbert space is canonical translation invariant if :

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

For the usual 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 1.2.2. Non-expensive representation A representation Φ is non-expensive if,

$$\forall (f,h) \in (\mathcal{L}^2(\mathbb{R}^d))^2 \quad \|\Phi(f) - \Phi(h)\| \leq \|f - h\| . \tag{1.1}$$

The stability to deformations of a non-expansive operator can be expressed as its Lipschitz continuity to the action of deformations close to translations ***cite mallat GIS***. Such a diffeormorphism transform can be expressed as,

$$\begin{aligned} L_\tau : \mathcal{L}^2(\mathbb{R}^d) &\to \mathcal{L}^2(\mathbb{R}^d) \\ f \quad &\to f(\mathbb{1} - \tau) \end{aligned}$$

where $\tau(x) \in \mathbb{R}^d$ is a displacement field.

Proposition 1.2.4. ***Lipschitz continuous*** *A translation invariant operator Φ is said to be Lipschitz continuous to the action of mathcalC^2 diffeomorphisms if for any compact $\Omega \in \mathbb{R}^d$ there exists C such that for all $f \in \mathcal{L}^2(\mathbb{R}^d)$ supported in Ω and all $\tau \in mathcalC^2(\mathbb{R}^d)$,*

$$\|\Phi(f) - \Phi(L_\tau f)\|_{\mathcal{H}} \leq C \|f\| \left(\sup_{x \in \mathbb{R}^d} |\nabla \tau(x)| + \sup_{x \in \mathbb{R}^d} |H\tau(x)| \right) \tag{1.2}$$

where $|\nabla \tau(x)|$ and $|H\tau(x)|$ are respectively the sup-norm and the sup-norm of the Hessian tensor of the matrix $\tau(x)$.

Hence a Lipschitz continuous operator Φ is almost invariant to "local" translations by $\tau(x)$, up to the fist and second order deformations terms. The equation 1.2 also implies that Φ is invariant to global translations.

1.2.3 State of the art in image representation :

Now that we have listed the properties we would like our representation to have, let us have a look at the usual signal representation tools and see if they which of them they fulfil.

The first representation method one can think of is the modulus of the ***Fourier transform***. This operator is informational enough to allow -to a certain extent- discrimination different type of signal ***find a citation for clf with fourier transform***. It is also translation invariant ***find a citation***. However it is well known that those operators present instabilities to deformation at high frequencies ***cite 10 from mallat*** and thus are not Lipschitz continuous to the action of diffeomorphisms.

Wavelet transform is another popular representation method. Again they provide a "good enough" representation to allow classification of different signals ***find citations***. Plus by grouping high frequencies into dyadic packet in $\mathbb{R}^d$, wavelet operators are stable to deformations ***citation mallat's book***.

$$W\mathbf{x} = \begin{pmatrix} \mathbf{x} * \phi \\ \mathbf{x} * \psi_\lambda \end{pmatrix} \begin{matrix} \to \text{averaging part} \\ \to \text{high frequency part} \end{matrix} \tag{1.3}$$

However only the averaging part of a wavelet is invariant to translation ans thus wavelets themselves are known to be non-invariant to translations.

Another signal representation method popular at the moment are the ***convolutional neural networks cite LeCun***. As opposed to the two previously mentioned representation methods, those operators are not fixed but learned from the data ***cite learning method from CNN***. Over the past few year they have provided state of the art results on many standard classification task, such as MNIST ***cite***, CIFAR ***cite***, ImageNet ***cite*** or ***find a example in speech processing***. Those good results are used to advocate that those networks are learning "good" representations. However it seems that in certain cases they learn representation of the data that are -for example- not invariant to deformations ***cite Bruna and Al strange pties of NN***.

1.3 Probabilistic graphical model :

? ? ? - not sure yet

1.4 Outline of the report :

The part 2 of this report will summarize and explain the recent work Stephane Mallat and his group on the ***Scattering Transform*** (ST), a wavelet-based operator fulfilling all the properties of what we have defined as a "good" representation for signal classification. Second (see 3) we will introduce the ***Probabilistic Graphical Models*** (PGMs) as generative models that can be used -among other tasks- for classification. Then in 4 we will describe how the the representation produced by the scattering transform can be modeled by an hidden markov tree, using what we have named ***Scattering Hidden Markov Trees*** (SCHMTs). Finally in **??** we will provide some example of applications.

2 The Scattering transform :

In this section we describe the construction process of a mathematical operator - the scattering transform (ST)- designed to generate what we have considered to be an interesting representation of our data (see 1.2). Therefore a scattering transform builds ***invariant***, ***stable*** and ***informative representation*** of signals through a ***non-linear***, ***unitary transform***. It is an operator delocalizing signal informational content into scattering decomposition path, computed by ***cascading wavelet/modulus operators***. This architecture is similar to a ***convolutional neural network*** (CNN) where the synaptic weights would be given by a wavelet operator instead of learned.

In this section we study a wavelet-based representation method -the scattering transform- having the properties of what we have defined as a "good" representation for signal classification. We do so by first explaining how are built the scattering operators (see 2.2) and review some of their important properties (see ***ref : correct section***). Once more familiar with the theory of the scattering transform we will see how similar in their architecture they are to Convolutional Neural Network (CNN) (see **??**). Finally, in 2.4, we describe how the scattering transform is usually used in classification tasks.

In this section we first introduce a wavelet-based scattering transform built to have interesting properties for classification tasks, meaning being translation invariant and stable to $\mathcal{L}^2$ deformations, while preserving the discriminabilty between classes. Then we explained how those operators can be stacked to create a "deep" scattering transform using a convolutional architecture.

2.1 Scattering wavelets :

A two-dimensional directional wavelet is obtained by scaling and rotating a single band-pass filter ψ. If we 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

$$\psi_{2^j r}(u) = 2^{2j}\psi(2^j r^{-1}u). \tag{2.1}$$

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

A wavelet transform filters the signal x using a family of wavelets $\{x * \psi_\lambda(u)\}_\lambda$. This is computed from a filter bank of dilated and rotated wavelets having no orthogonality property and it creates a multi-scale and orientation representation of the input.

If $u.u'$ and $\|u\|$ define respectively the inner product and the norm in $\mathbb{R}^2$, the Morlet wavelet ψ is an example of wavelet given by,

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

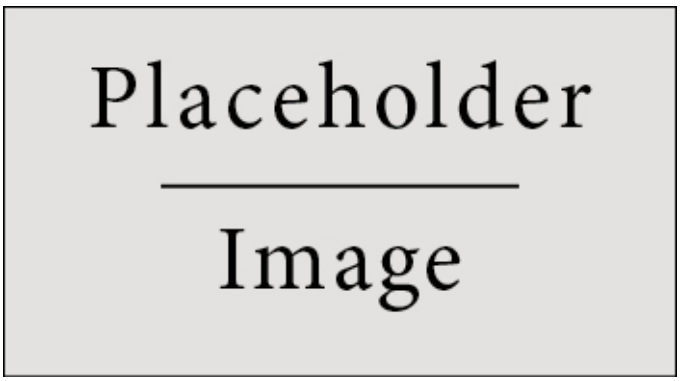

FIGURE 2.1 – Complex Morlet wavelet.

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 2.1 shows a Morlet wavelet for $\xi = 3\pi/4$ and $\sigma = 0.85$.

As opposed to the Fourier sinusoidal waves, wavelets are operators stable to $\mathcal{L}^2$ deformations as they can be expressed as localized waveforms (***citation***). However, wavelet transforms compute convolutions with wavelets, hence they are translation covariant operators (***citation***).

To ensure a translation invariant behavior to an operator commuting with them, one has to introduce a non-linearity. For example if R is a linear or non-linear operator commuting with translations L_c, **i.e.** $R(L_c x) = L_c R(x)$), then the integral $\int R(x(u))du$ is translation invariant. One can apply this to $R(x) = x * \psi_\lambda$ and gets the trivial invariant,

$$\int x * \psi_\lambda(u)du = 0,$$

for all x as $\int \psi_\lambda(u)du = 0$. However to preserve the informative character of the scattering operator, one has to ensure that the integral does not vanish. To do so an operator M such that $R(x) = M(x * \psi_\lambda)$ is introduced. If M is a linear transformation commuting with translation then the integral still vanishes. Hence one has to choose M to be a non-linear.

Keeping in mind that the scattering transform has to be stable to deformations and taking advantages of the wavelet transform stability to small deformations in the input space, we also impose that M commutes with deformations,

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

If a weak differentiability condition is added, one can prove (***ref ISCN 6***) that M must necessarily be a point-wise operator, **i.e.** $Mx(u)$ only depends on the value of $x(u)$. Finally, by adding an $\mathcal{L}^2(\mathbb{R}^2)$ stability constraint,

$$\forall (x, y) \in \mathcal{L}^2(\mathbb{R}^2)^2\ ,\ \|Mx\| = \|x\| \text{ and } \|Mx - My\| \leq \|x - y\|\,,$$

one can show (***ref ISCN 6***) that necessarily $Mx = e^{i\alpha}\,|x|$. For the scattering transform, the simplest solution of setting α to 0 is choosed and therefore the resulting coefficients are the $\mathcal{L}^1(\mathbb{R}^2)$ norms :

$$\|x * \psi_\lambda\|_1 = \int |x * \psi_\lambda|\, du$$

The family of $\mathcal{L}^1(\mathbb{R}^2)$ normed wavelet $\{\|x * \psi_\lambda\|_1\}_\lambda$ generate a crude signal representation which measures the sparsity of the wavelet coefficients. One can prove (***ref ISCN 36***) that

x can be reconstructed from $\{|x * \psi_\lambda(u)|\}_\lambda$ up to a multiplicative constant. Which means that the information loss in $\{\|x * \psi_\lambda\|_1\}_\lambda$ comes from the integration of the absolute value $|x * \psi_\lambda(u)|$ which removes all non-zero frequencies. However those components can be recovered by calculating the wavelet coefficients $|x * \psi_{\lambda_1}| * \psi_{\lambda_2}(u)$. By doing so their $\mathcal{L}^1(\mathbb{R}^2)$ norms define a much larger family of invariants :

$$\forall(\lambda_1, \lambda_2) \, \| |x * \psi_{\lambda_1}| * \lambda_2 \|_1 = \int ||x * \psi_{\lambda_1}(u)| * \psi_{\lambda_2}| \, du$$

By further iterating on the wavelet/modulus operators more translation invariant coefficients can be computed. Let us define :

Definition 2.1.1. Scattering Propagator
The scattering operator U for a scale and an orientation $\lambda \in G \times \mathbb{Z}$ is defined as the absolute value of the input convolved with the wavelet operator at this scale and orientation.

$$U[\lambda](x) \overset{d}{=} |x * \psi_\lambda| \tag{2.2}$$

Definition 2.1.2. Path Ordered Scattering Propagators
Any sequence $p = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ where $\forall i \in [\![1, m]\!] \lambda_i \in G \times \mathbb{Z}$ defines a ***path*** of length m, **i.e.** the ordered product of non-linear and non-commuting operators,

$$\begin{aligned} U[p]x &\overset{d}{=} U[\lambda_m] \ldots U[\lambda_2] U[\lambda_1](x) \\ &= ||| |x * \psi_{\lambda_1}| * \psi_{\lambda_2}| \ldots | * \psi_{\lambda_m}| \, . \end{aligned} \tag{2.3}$$

With the convention : $U[\emptyset]x = x$

From there one can provide a first formal definition of the scattering transform :

Definition 2.1.3. Scattering Coefficient
A scattering coefficient along the path p is defined as an integral of the p ordered scattering propagators, normalized by the response of a Dirac :

$$\bar{S}[p](x) \overset{d}{=} \mu_p^{-1} \int U[p]x(u) du \tag{2.4}$$

with,

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

We shall see later (***reference***) that each scattering coefficient $\bar{S}[p](x)$ is -as desired - invariant to translation of the input x and Lipschitz continuous to deformations.

For classification tasks, one might want to compute localized descriptors only invariant to translations smaller than a predefined scale 2^J, while keeping the spatial variability at scales larger than 2^J. One can achieved this by localizing the scattering integral with a scaled spatial window $\phi_{2^J}(u) = 2^{-2J}\phi(2^{-2J}u)$. This yield to the definition of the windowed scattering transform :

Definition 2.1.4. -Windowed- Scattering Coefficient Of Order m
If p is a path of length $m \in \mathbb{N}$, the -windowed- scattering coefficient of order m at scale 2^J is defined as :

$$\begin{aligned} S_J[p](x) &\stackrel{d}{=} U[p]x * \phi_{2^J}(u) \\ &= \int U[p]x(v)\phi_{2^J}(u-v)dv \\ &= ||||x * \psi_{\lambda_1}| * \psi_{\lambda_2}| \ldots | * \psi_{\lambda_m}| * \phi_{2^J}(u) \end{aligned} \tag{2.5}$$

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

2.2 Scattering Convolution Network :

This section introduces the scattering transform as an ***iterative process over a one-step operator*** and creates a parallel with convolutional neural networks ***cite LeCun convNet 11 of mallat long paper***. Let us note denote $U_J[\Omega] \stackrel{d}{=} \{U_J[p]\}_{p\in\Omega}$ and $S_J[\Omega] \stackrel{d}{=} \{S_J[p]\}_{p\in\Omega}$ a family of operators indexed by a path set Ω.

One can compute a windowed scattering transform by iterating on the ***one-step propagator*** U defined by,

Definition 2.2.1. One-step propagator
The one-step propagator U can be defined as,

$$\forall x \in \mathcal{L}^2(\mathbb{R}^d) \quad U_J x = \{A_J x,\ (U[\lambda]x)_{\lambda\in\Lambda_J}\}, \tag{2.6}$$

with $A_J x = x * \phi_{2J}$ and $U[\lambda]x = |x * \psi_\lambda|$.

Indeed after calculating $U_J x$, applying U_J again to each $U\lambda x$ yields a larger infinite family of functions. Furthermore since $U[\lambda]U[p] = U[p+\lambda]$ and $A_J U[p] = S_J[p]$ it holds that,

$$\forall x \in \mathcal{L}^2(\mathbb{R}^d) \quad U_J U[p]x = \{S_J[p]x,\ (U[p+\lambda]x)_{\lambda\in\Lambda_J}\}, \tag{2.7}$$

Let Λ_J^m be a set of path of length m with the convention $\Lambda_J^0 = \{\emptyset\}$, its propagation is,

$$\forall x \in \mathcal{L}^2(\mathbb{R}^d) \quad U_J U[\Lambda_J^m]x = \{S_J[\Lambda_J^m]x,\ (U[\Lambda_J^{m+1}]x)_{\lambda\in\Lambda_J}\}, \tag{2.8}$$

Hence $S_J[\mathcal{P}_J]x$ can be computed from $x = U[\emptyset]x$ by iteratively computing $U_J U[\Lambda_J^m]x$ for m going from 0 to ∞. This iterative process is illustrated in Figure 2.2 and one can notice that the scattering calculation as the same general architecture as the convolutional neural networks introduced by LeCun ***cite LeCun convNet 11 of mallat long paper***. Both CNN and scattering convolutional network (SCN) cascade convolutions and a "pooling" non linearity. However while convolution networks use kernel filters learned from the data with back-propagation algorithm, SCNs use a fixed wavelet filter bank.

2.3 Properties of the scattering transform :

This section provides a formal version of the previously mentioned properties of the scattering transform.

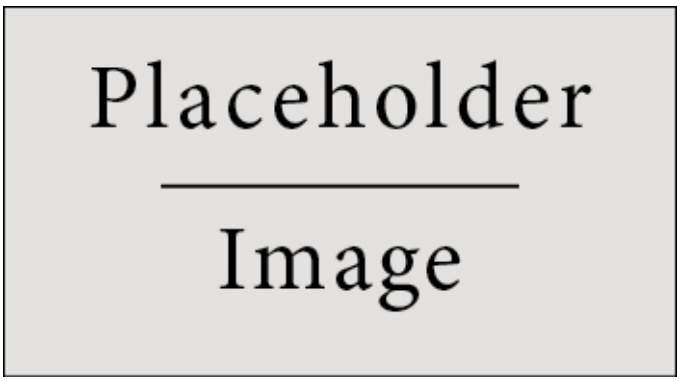

FIGURE 2.2 – see mallat long paper p1341

2.3.1 Non-expansivity :

The scattering propagator $U_J x = \{A_J x,\ (|W_J x|)_{\lambda \in \Lambda_J}\}$ results of the composition of a wavelet transform W_J that is unitary and of a modulus operator that is non-expansive - as $\forall (a,b) \in \mathbb{C}^2\ ||a| - |b|| \leq |a-b|$- and is thus non-expansive. Since $S_J[\mathcal{P}_J]$ iterates on U_J, which is non-expansive, the proposition (***cite mallat's long report 12***) proves that $S_J[\mathcal{P}_J]$ is also non-expansive.

Proposition 2.3.1. ***Non-expansivity***
The scattering transform is non expansive.

$$\forall (x,z) \in \mathcal{L}^2(\mathbb{R}^d)^2 \quad \|S_J[\mathcal{P}_J]x - S_J[\mathcal{P}_J]z\| \leq \|x - z\| \tag{2.9}$$

2.3.2 Energy preservation :

We have seen (***reference to previous section if true or TODO*** that each propagator $U[\lambda]x = |f * \psi_\lambda|$ captures the frequency energy contained in the signal x over a frequency band covered by the Fourier transform $\hat{\psi}_\lambda$ and propagates this energy towards lower frequencies. We can thus prove that under some assumption on the wavelet, 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 first we need to define what an ***admissible scattering wavelet*** is.

Proposition 2.3.2. ***Admissible scattering wavelet***
A scattering wavelet ψ is admissible if there exist $\eta \in \mathbb{R}^d$ and $\rho \geq 0$, with $|\hat{\rho}(\omega)| \leq \left|\hat{\phi}(2\omega)\right|$ and $\hat{\rho}(\omega) = 0$, such that the function,

$$\hat{\Psi}(\omega) = |\hat{\rho}(\omega - \eta)|^2 - \sum_{k=1}^{+\infty} k(1 - \left|\hat{\rho}(2^{-k}(\omega - \eta)\right|^2), \tag{2.10}$$

satisfies,

$$\alpha = \inf_{1 \leq |\omega| \leq 2} \sum_{j=-\infty}^{+\infty} \sum_{r \in G} \hat{\Psi}(2^{-j} r^{-1} \omega) \left|\hat{\psi}(2^{-j} r^{-1} \omega)\right|^2 > 0. \tag{2.11}$$

We can now state,

Theorem 2.3.3. ***Energy conservation***
If the scattering wavelet ψ is admissible, then for all signal $x \in \mathcal{L}^2(\mathbb{R}^d)$,

$$\lim_{m \to +\infty} \|U[\Lambda_J^m]x\|^2 = \lim_{m \to +\infty} \sum_{n=m}^{+\infty} \|S_J[\Lambda_J^n]x\|^2 = 0, \tag{2.12}$$

and

$$\|S_J[P_J]x\|^2 = \|x\|. \tag{2.13}$$

The proof of the theorem 2.3.3 also shows that the scattering energy propagates progressively towards lower frequencies and that ***the energy of*** $U[p]x$ ***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 rest of the paper ***only frequency decreasing path will be considered***.

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 path can be neglected. For signal processing applications, this decay appears to be exponential. And ***for classification applications path of length*** $m = 3$ provides the most interesting results ***cite mallat's long paper 1 3***.

The two restrictions stated above yields to an easier parametrization of a scattering network. Indeed when only the frequency decreasing paths up until a given order a scattering network is completely defined by :

— M : The maximum path length considered.
— J : The coarsest scale level considered.
— L : The number of orientation considered, which can be define as the cardinality of the previously define G.

Hence for a given set of parameter (M, J, L), one can generate one and only one scattering network -with frequency decreasing paths. Each node of this network generates a -possibly empty- set of of nodes of size,

2.3.3 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 that the scattering distance $\|S_J[\mathcal{P}_J]x - S_J[\mathcal{P}_J]z\|$ converges when J goes to infinity - as it is non-increasing when J increases (see section 2.3.1). From there one can bound the distance between the scattering transform of the signal and the one of its translated version $\|S_J[S_J[\mathcal{P}_J]\mathcal{L}_c x - \mathcal{P}_J]x\|$ and prove that this bound tends to 0 when J goes to infinity. This yields to the following theorem,

Theorem 2.3.4. ***Translation invariance***
For admissible scattering wavelets,

$$\forall x \in \mathcal{L}^2(\mathbb{R}^d),\ \forall c \in \mathbb{R}^d \quad \lim_{J \to \infty} \|S_J[S_J[\mathcal{P}_J]\mathcal{L}_c x - \mathcal{P}_J]x\| = 0 \tag{2.14}$$

Note. A formal proof of 2.3.4 can be found in ***cite Mallat's long paper***.

2.3.4 Lipschitz continuity to the action of diffeomorphisms :

The Lipschitz continuity to the action of diffeomorphisms of $\mathbb{R}^d$ can be proved for those sufficiently close to a translation. Such diffeomorphisms maps u to $u - \tau(u)$ where $\tau(u)$ is a displacement filed such that $\|\nabla\tau\|_\infty < 1$. Let $L_\tau x(u) = x(u - \tau(u))$ denote the action of such diffeomorphisms on the signal 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 $\|S_J[\mathcal{P}_J]\mathcal{L}_\tau x - S_J[\mathcal{P}_J]x\|$. 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 yields to the theorem,

Theorem 2.3.5. ***Lipschitz continuity to the action of diffeomorphisms***
There exists C such that all $x \in \mathcal{L}(\mathbb{R}^d)$ with $\|U[\mathcal{P}_J]x\|_1 < \infty$ and all $\tau \in \mathcal{C}^2(\mathbb{R}^d)$ with $\|\nabla\tau\|_\infty < \frac{1}{2}$ satisfy,

$$\|S_J[\mathcal{P}_J]\mathcal{L}_\tau x - S_J[\mathcal{P}_J]x + \tau.\nabla S_J[\mathcal{P}_J]x\| \leq C\,\|U[\mathcal{P}_J]x\|_1\,K(\tau), \tag{2.15}$$

with

$$K(\tau) = 2^{-2J}\|\tau\|_\infty^2 + \|\nabla\tau\|_\infty\left(\max\left(\log\frac{\|\Delta\tau\|_\infty}{\|\nabla\tau\|_\infty}, 1\right)\right) + \|H\tau\|_\infty \tag{2.16}$$

Note. Again a formal proof of 2.3.4 can be found in ***cite Mallat's long paper***.

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 2.15 under the assumptions mentioned above,

$$\forall p \in \mathcal{P}_J\,\|S_J[p]\mathcal{L}_\tau x - S_J[p]x + \tau.\nabla S_J[p]x\| \approx 0. \tag{2.17}$$

Find reference for Application of displacement field estimation - best if not video

2.4 Application to classification :

The scattering transform has been designed in order to provide a "good" representation for signal classification. The classical way of using it for such a task is to use the features generated by the scattering transform of the dataset considered as inputs for a discriminative classifier (e.g. Support Vector Machine classifier). Using this model provides results able to compare with state of the art CNNs on some standard datasets ***cite mallat***.

The energy contained in the scattering transform is mostly contained in the short frequency decreasing paths (see 2.3.2). Thus for most of the classification path of length $m = 3$ provides the best trade out between classification performance and computational time. Unless stated otherwise this length will be use in all our experiments.

3 Probabilistic graphical models :

Those models use a ***graph based representation of conditional dependence between a set of random variables*** and thus encode a complete distribution over a multi-dimensional space in a compact -or factorized- graph.The field of PGMs can be split into two main families, the ***Bayesian Networks*** (BNs) and the ***Markov models*** (MMs). As they are graphical models, both families encompass the properties of factorization and independences defined by the graph, but differ when it comes to the specificities of the set of independences they can encode as well as the factorization of the distribution that they can induce (***cite : Bishop, Christopher M. (2006). "Chapter 8. Graphical Models" (PDF). Pattern Recognition and Machine Learning. Springer. pp. 359–422. ISBN 0-387-31073-8. MR 2247587***).

Add par on how they will be used.

The aim of this section is not to provide a complete overview of the Probabilistic graphical models but rather to introduce some interesting concepts that have been used in our work. Someone with more interest in PGMs could refer to ***cite A tutorial on learning BN*** or ***cite cours dafney koller***. This chapter introduces those two main classes of probabilistic graphical models. Section 3.1 focuses on the Bayesian networks, while section 3.2 provides more details about the Markov models.

3.1 Bayesian Network :

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). The architecture of the Bayesian Networks is further explained in section 3.1.1. Then section 3.1.2 presents a brief overview of the state of the art in terms of learning algorithm for BNs and section 3.1.3 describes the inference mechanism for those networks.

3.1.1 Architecture :

Bayesian networks can be defined as,

Definition 3.1.1. Bayesian Network
For a set of random variables $\mathbf{X} = \{X_i\}_{i\in[\![1,N]\!]}$ a Bayesian network consists of a ***direct acyclic graph*** $\mathcal{G}$ encoding a set of conditional independence assertions about the random variables in $\mathbf{X}$ and a set P of ***local probability distribution associated with each variable***. Each node of $\mathcal{G}$ represent one of the random variable X_i and each edge $E_{i\to j}$ represents the conditional dependence between the nodes X_i and X_j.

For such networks the following property holds,

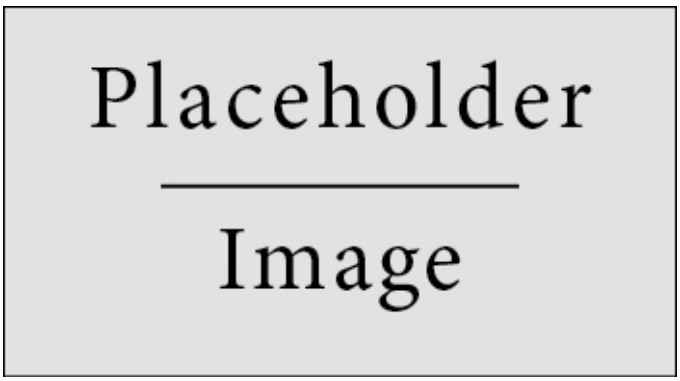

FIGURE 3.1 – Example of Bayesian network.

Proposition 3.1.1. ***Conditional independence for Bayesian networks***
In a Bayesian network each node of the graph are "conditionally independent of any subset of the nodes that are not descendants of itself given its parent".

$$P(\{X_i\}_{i\in[\![1,N]\!]}) = \prod_{i=1}^{N} P(X_i|\rho(X_i)) \tag{3.1}$$

where $\rho(X_i)$ are the parents of the node X_i.

Thus in a Bayesian network a node with no parents is not conditioned on the other random variables. Such a node is called a ***prior probability***.

By their architecture, Bayesian networks allow to simplify the computation of the joint probability distribution. For example for the network defined by figure 3.1, one can obtain $P(X_1, X_2, X_3, X_4, X_5, X_6)$ by the use of chain rules and theory on conditional independent,

$$\begin{aligned} P(X_1, X_2, X_3, X_4, X_5, X_6) &= P(X_6|X_3, X_4, X_5)P(X_1, X_2, X_3, X_4, X_5) \\ &= P(X_6|X_3, X_4, X_5)P(X_3|X_1, X_5)P(X_4|X_2)P(X_5|X_2)P(X_1, X_2) \\ &= P(X_6|X_3, X_4, X_5)P(X_3|X_1, X_5)P(X_4|X_2)P(X_5|X_2)P(X_2|X_1)P(X_1). \end{aligned}$$

3.1.2 Learning :

In some application the structure $\mathcal{G}$ and the set P of local probability distribution associated with each variable of the network are provided. In such a case the BNs are "simply" used for inference. The task is then to infer the most probable values for a subset $\mathbf{Y} \subset \mathbf{X}$, given a partially complete set of realizations $\mathbf{X}_{obs} \subset \mathbf{X}\backslash\mathbf{Y}$.

However, most of the time the full characterization of the BN is not provided. Ignoring the case of missing data, one can split the learning problem into two main categories :

— ***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. Microsoft trouble shooting system ***cite***) or be imposed by some construction rules (e.g. Boltzmann Machine ***cite***, Restricted Boltzmann Machine ***cite*** ...). The task at end is then to learn the parameters governing the local probability distributions of the Bayesian network.

— ***Architecture and local probability distributions :*** In this case the architecture of the network has to be learned along side with the local probability distributions. This problem is not developed in the rest of this paper, but one could refer to ***cite A tutorial on learning BN*** for an introduction to this problem.

Expectation-maximization :

Variational Bayes :

3.1.3 Inference :

`http://www.cse.unsw.edu.au/~cs9417ml/Bayes/Pages/Bayesian_Networks_Inference.html`

3.2 Markov Models :

Quick overview over the main methods for HMMs.

3.2.1 Architecture :

TBD

3.2.2 Learning :

TBD

3.2.3 Inference :

TBD

4 Scattering hidden Markov tree :

Section 2.4 introduced the usage of scattering networks combined with an support vector machine classifier to achieve state of the art classification performances on some problems. However this method provides a boolean answer for each class. Some methods to express the output of an SVM as a probability exists (***cite : Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods (1999)*** but they are just a rescaling of the output and not a true probabilistic approach. If one is interested in a true probabilistic model to describe the scattering coefficient it is quite natural to try to express them as a probabilistic graphical model. Indeed if one hide the propagation step from the scattering transform (see 2.2) the scattering network defines a tree structure.

To simplify the notation in the remainder of this section, let $\mathcal{T}$ denotes the tree structure defines as state above and depicted in figure 4.1. Let I denotes the total number of nodes -i.e. scattering coefficient and let S_i for $i \in [\![0, I-1]\!]$ denotes one of the node $S[p_i]x \in \mathcal{T}$ for a given path $p_i = [\lambda_0 ... \lambda_u]$ ($u \in \mathbb{N}$). Note that S_i represents a node and does not depends on the signal x. For a given signal x, a realization of the node i is denoted by $S_i = s_i = S[\lambda_i]x$ where λ_i denotes the path in the scattering tree to the node i. Note also that in the remainder of the paper the short notation $i \in \mathcal{T}$ will be use to denote λ_i. Let also use the convention $S_0 = S[\emptyset]x$. 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. Note also that a node S_i can have no children, in such a case this node is a leaf of the tree.

The remainder of this section is organized as follows : Section 4.1 introduces related work and provide a description of the SCHMT model. Section 4.2 details the hypothesis needed to develop this model as well as provides some intuition on their validity. Finally section 4.3 and 4.4 respectively describe the learning algorithm for the parameters of the model and the classification method.

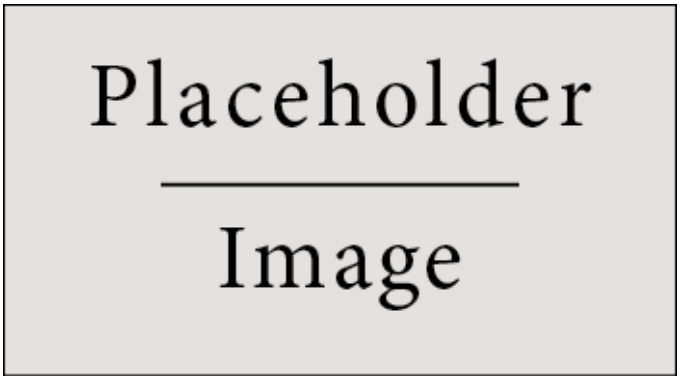

FIGURE 4.1 – Scattering transform tree.

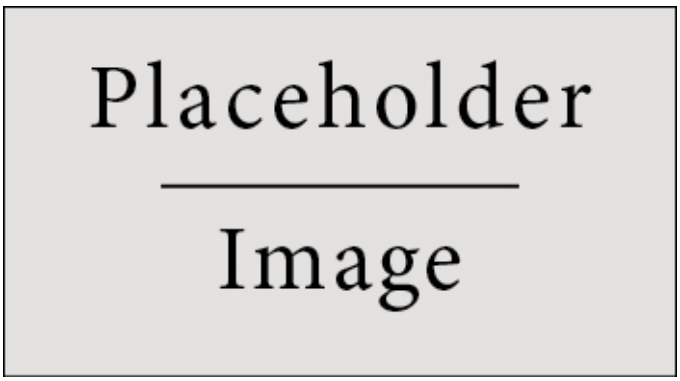

FIGURE 4.2 – Scattering Hidden Markov Tree.

4.1 SCHMT model and related works :

The idea behind the SCHMT model is to say that the more detailed representation of the signal is somehow correlated to the less detailed one from which it is generated. More formally this means that for a signal x, S_i is somehow correlated to $S_{\rho(i)}$. To do so one could model the scattering network by a Markov tree and assumes the following :

$$P(S_i|\mathcal{T}) = P(S_i|S_{\rho(i)}). \tag{4.1}$$

Those independence properties yield to the graph displays in figure 4.1. However models trying to describe directly the correlation across coefficients at different scales have been studied for traditional wavelet transforms ***cite : in Crouse previous work*** and it seems that a simple one-step Markovian assumption across scale is notsufficient to describe the complex relationship between wavelet coefficients.

A common approach when a direct Markovian model does not hold is to introduce hidden states and to assume the Markovian property across those states. The observed nodes are then only dependent on their state. This is the architecture adopted for the SCHMT and its graph is represented in figure 4.2. This model has the following independence properties,

$$P(H_i|\mathcal{T}) = P(H_i|H_{\rho(i)}), \tag{4.2}$$

$$P(S_i|\mathcal{T}) = P(S_i|H_i). \tag{4.3}$$

As the scattering transform is closely related to wavelet transforms it is not surprising to find similar ideas exploited for wavelet trees. MS. Crouse proposed where an Hidden Markov Tree Model is used to model the wavelet coefficients ***cite : Crouse*** of a classic wavelet trees. Later N. Kingsbury ***cite : Kingsbury*** adapted MS. Crouse's model to his Dual wavelet complex trees. The resulting hidden Markov tree models provides better classification performances than MS. Course's WHMT as the wavelet used generate a "better" representation of the signal -in the sense defined in section 1.2.1. Indeed this version can leverage the quasi-translation invariance property of the complex wavelets. The improvement in performances due to the quasi-invariance property provides a good motivation to try the hidden Markov tree modeling on the scattering transform as they have even "better" representational properties (see 2.3. The parameters of the original WHMT were trained using a version of the ***Expectation-Maximization*** adapted

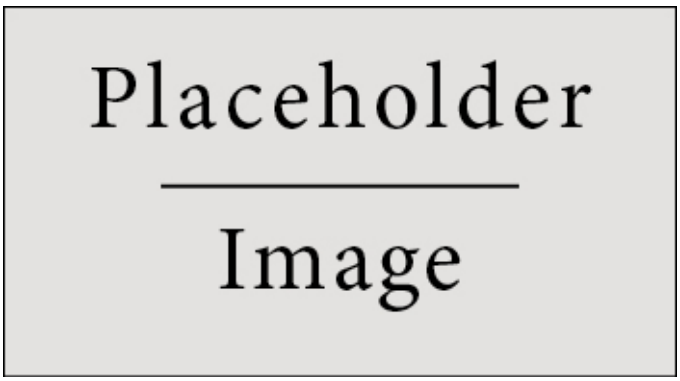

FIGURE 4.3 – Wavelet hidden Markov tree.

to binary hidden Markov trees. However this learning method suffered from overflowing issues ***cite : ? - Durand ?***, JB. Durand ***cite : Durand*** proposed a smoothed version of the training algorithm preventing overflowing.

A scattering hidden Markov tree is composed by a set of visible nodes $\{\mathbf{S}_i\}_{i\in\mathcal{T}}$ and a set of of hidden node $\{\mathbf{H}_i\}_{i\in\mathcal{T}}$. Then the tree is parametrised as follow,

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

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

$$\forall k \in [\![1, K]\!] \quad \pi_0(k) = P(H_0 = k). \tag{4.4}$$

— For any index i of the tree, the ***emission distribution*** describe the probability of the visible node S_i conditional to the hidden state H_i,

$$\forall i \in \mathcal{T} \;\; \forall k \in [\![1, K]\!] \text{ and } \forall s \in \mathbb{R} \qquad P(S_i = s|H_i = k) = P_{\theta_{k,i}}(s) \tag{4.5}$$

Where P_θ belongs to a parametric distribution family and where θ is the vector of ***emission parameters***. In the remainder of the paper the emission distribution is a Gaussian so that,

$$\forall i \in \mathcal{T} \;\; \forall k \in [\![1, K]\!] \text{ and } \forall s \in \mathbb{R} \qquad P(S_i = s|H_i = k) = \mathcal{N}(\mu_{k,i}, \sigma_{k,i}). \tag{4.6}$$

where $\theta_{k,i} = (\mu_{k,i}, \sigma_{k,i})$ where $\mu_{k,i}$ and $\sigma_{k,i}$) are 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 to move from one state to the other in characterized by a ***transition probability***,

$$\forall i \in \mathcal{T}\backslash\{0\} \quad \forall g, k \in [\![1, K]\!]^2 \qquad \epsilon_i^{(gk)} = P(H_i = k|H_{\rho(i)} = g), \tag{4.7}$$

where ϵ_i defines a transition probability matrix such that,

$$\forall i \in \mathcal{T}\backslash\{0\} \quad \forall k \in [\![1, K]\!] \qquad P(H_i = k) = \sum_{g=1}^{K} \epsilon_i^{(gk)} P(H_{\rho(i)} = g). \tag{4.8}$$

And using the chain rule of probability one can express $P(H_i)$ from the root node's initial distribution.

The SCHMT model is thus fully described by,

$$\Theta = \left(\pi_0, \{\epsilon_i, \{\theta_{k,i}\}^{k\in[\![1,K]\!]}\}_{i\in\mathcal{T}}\right). \tag{4.9}$$

The SCHMT model differs from the previous work by the shape of its tree structure. Previous work are based on regular binary tree structures where all the leafs have the same depth the scattering tree is a irregular tree. As seen in section 2.2 each node has $TODO$ children. This yields to an architecture where each node have a different number of children and where leafs can be found at any depth of the tree. However as detailed in section 4.3, the learning algorithm can be adapted to this case. Another difference between SCHMT and the previous works is the non-homogeneity of the transition matrix. Indeed by the nature of the scattering transform one can expect a non homogeneous transmission of the information across the orders and thus to accomodate this the learning algorithm has also be adapted to this case.

Even though the theoretical framework of SCHMT holds for any $K \in \mathbb{N}^*$, in the remainder of the work K is set to 2. This means that the scattering coefficient are described by a mixture of two Gaussians and can be in ***two states,*** (H) ***High or*** (L) ***Low***. This model yields to sparser representations as the number of hidden states is highly constrained.

4.2 Hypothesis :

Expressing the dependencies between the scattering coefficients as an Hidden Markov Tree implies that two modeling assumptions are done. The first assumption reflects the fact that the scattering coefficients can effectively be expressed by two hidden states,

Assumption 1. **Two populations :**
A signal's scattering transform can be described as two clusters of coefficients' value. The smooth regions are represented by small scattering coefficients, while edges, ridges, and other singularities are represented by large coefficients.

As this assumption is common for standard or complex wavelet (***cite : Kingsbury*** and because a scattering coefficient of order m can simply be seen as the modulus of a "new" signal -i.e. the scattering coefficient of order $m-1$, the two-populations assumption for scattering network is reasonable. This intuition can be confirmed by plotting the scattering coefficients obtained for several signals.

Figure 4.4 displays 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 highest values of the scattering coefficient are obtained on highly informational pixels (edges in this case) while the low informational pixels are represented by scattering coefficients near 0. Similar observations can be made for more complex signal -such as the one displays in 4.5. A statistical interpretation of the two-populations implies that scattering coefficients have a non-Gaussian marginal statistics, that is, their marginal probability density functions have a large peak at zero (many small coefficients) and heavy tails (a few large coefficients). Finally since many real-world signals (photograph-like images, for example) consist mostly of smooth regions separated by a few singularities, the two 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 ***cite : Kingsbury - 4, 5]*** for example). Most of the scattering coefficient magnitudes (representing the smooth regions) are small, while a few of them (representing the singularities and encoding the informational content) are large.

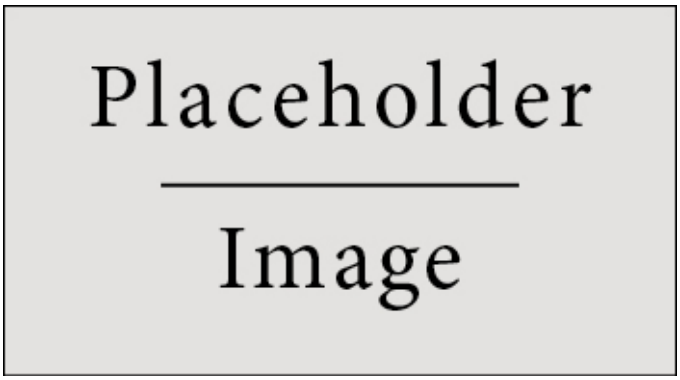

FIGURE 4.4 – Two populations - experiment 1 : The signal is a binary square (0 : background, 1 : square) with noise. The scattering network has $m = 3$ layers, $J = 3$ scales and $L = 2$ orientations.

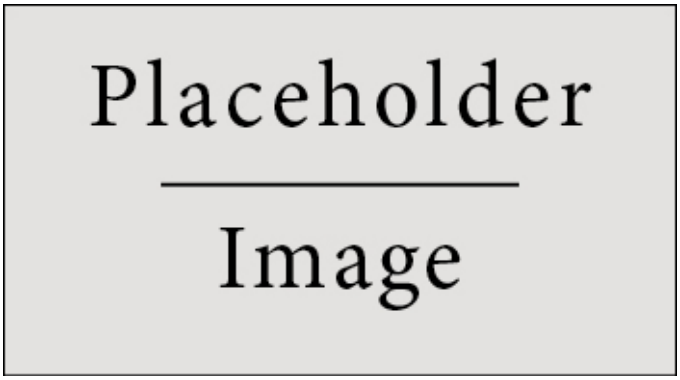

FIGURE 4.5 – Two populations - experiment 1 : The signal is the picture of Lena. The scattering network has $m = 3$ layers, $J = 3$ scales and $L = 2$ orientations.

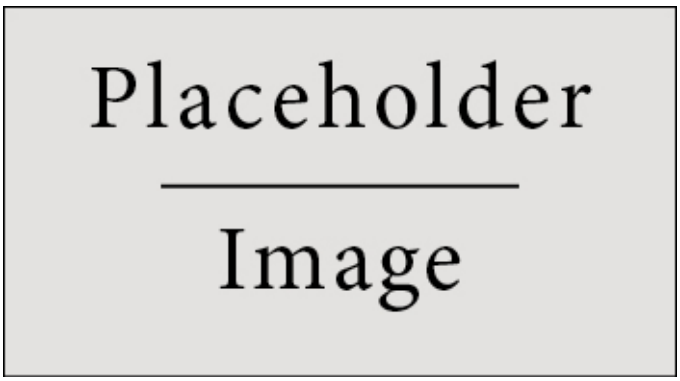

FIGURE 4.6 – Persistence - experiment 1 : Plot the magnitude of the scattering coefficients at a given index i of the tree against those of its father $\rho(i)$.

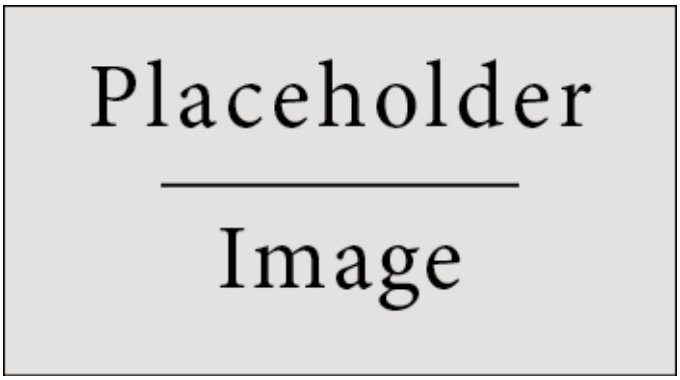

FIGURE 4.7 – Persistence - experiment 2 :.

The second assumptions transcripts the smoothness of the states across the scattering transform and can be stated as,

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 represented by low/high values at every order. Persistence leads to scattering coefficient values that are statistically dependent along the branches of the scattering tree. Figure 4.6 displays the magnitude of the scattering coefficient for a given node i of the tree against those of its father $\rho(i)$. A clear positive correlation can be observed between the magnitude of the father and the son. Furthermore the transitions can be regrouped in two main clusters : Low-Low and High-High. This tends to confirm the validity of the persistence assumption. A second experiment computes the average correlation between each pair father/child of the tree for signal belonging to different classes. Again this experiments advocates in favor of a correlation in between the magnitude of the connected nodes.

4.3 Learning the tree structure :

As seen in 3.2.2 training Markov models can be done using ***Expectation-Maximization*** methods. Hidden Markov chains use a version of the EM algorithm called ***Forward-Backward***

algorithm allowing the propagation of the hidden states along the chain. MS. Crouse ***cite : Crouse*** proposed the ***Upward-Downward algorithm***, an adaptation to the hidden Markov trees of the Forward-Backward algorithm. Both algorithm were suffering from overflowing problems ***cite : Durand 9*** and JB. Durand ***cite : Durand*** adapted ***name + cite : Durand 7*** work to create a smoothed version of the learning algorithm for trees. This section proposes a rewriting of JB. Durand's version of the Upward-Downward algorithm adapted to non-regular non-binary HMTs.

To do so one needs to introduce the following notations,
— $\forall i \in \mathcal{T}$ 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 subtree rooted at node i. By convention $\bar{\mathcal{S}}_0$ denotes the entire observed tree.
— $\forall i \in \mathcal{T}$ let $\bar{\mathcal{S}}_{c(i)} = \bar{s}_{c(i)}$ be the entire -possibly empty collection of observed subtrees rooted at children of node i (i.e. the subtree $\bar{s}_i$ except its root s_i).
— If $\bar{\mathcal{S}}_i$ is a proper subtree of $\bar{\mathcal{S}}_j$, then $\bar{\mathcal{S}}_{j\backslash i} = \bar{s}_{j\backslash i}$ is the subtree rooted at node j except the subtree rooted at node i.
— $\forall i \in \mathcal{T}$ let $\bar{\mathcal{S}}_{0\backslash c(i)} = \bar{s}_{0\backslash c(i)}$ be the entire tree except for the subtrees rooted at children of node i.
Note that those notations transpose to the hidden state and for instance $\bar{\mathcal{H}}_i = \bar{h}_i$ is the state subtree rooted at node i.

4.3.1 E-Step :

The smoothed version of the E-step requires the computation of the conditional probability distribution $\xi_i(k) = P(H_i = k|\bar{\mathcal{S}}_i = \bar{s}_i)$ (smoothed probability) and $P(H_i = k, H_{\rho(i)} = g|\bar{\mathcal{S}}_i = \bar{s}_i)$ for each node $i \in mcalT$ and state k and g. A decomposition of the smoothed probability adapted to the HMT structure is,

$$\xi_i(k) = \frac{P(\bar{\mathcal{S}}_{0\backslash i} = \bar{s}_{0\backslash i}|H_i = k)}{P(\bar{\mathcal{S}}_{0\backslash i} = \bar{s}_{i\backslash i}|\bar{\mathcal{S}}_1 = \bar{s}_i)} P(H_i = k|\bar{\mathcal{S}}_i = \bar{s}_i) \tag{4.10}$$

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

$$\beta_i(k) = P(H_i = k|\bar{\mathcal{S}}_i = \bar{s}_i) \tag{4.11}$$

$$\beta_{\rho(i)i}(k) = \frac{P(\bar{\mathcal{S}}_i = \bar{s}_i|H_{\rho(i)} = k)}{P(\bar{\mathcal{S}}_i = \bar{s}_i)} \tag{4.12}$$

$$\alpha_i(k) = \frac{P(\bar{\mathcal{S}}_{0\backslash i} = \bar{s}_{0\backslash i}|H_{\rho(i)} = k)}{P(\bar{\mathcal{S}}_{0\backslash i} = \bar{s}_{0\backslash i}|\bar{\mathcal{S}}_i = \bar{s}_i)} \tag{4.13}$$

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 initialized for the root node with $P(H_0 = k) = \pi_0(k)$ and then using the recursive formula 4.8.

Upward recursion :

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

$$\begin{aligned}\beta_i(k) &= P(H_i = k|\bar{\mathcal{S}}_i) = \bar{s}_i) \\ &= \frac{P(S_i = s_i|H_i = k)P(H_i = k)}{P(S_i = s_i)} \\ &= \frac{P_{\theta_{k,i}}(s_i)P(H_i = k)}{N_i},\end{aligned} \tag{4.14}$$

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

$$N_i = P(S_i = s_i) = \sum_{k=1}^{K} P_{\theta_{k,i}}(s_i)P(H_i = k). \tag{4.15}$$

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

$$\begin{aligned}\beta_i(k) &= P(H_i = k|\bar{\mathcal{S}}_i = \bar{s}_i) \\ &= \left[\prod_{j\in c(i)} P(\bar{\mathcal{S}}_j = \bar{s}_j|H_i = k)\right] P(S_i = s_i|H_i = k)\frac{P(H_i = k)}{P(\bar{\mathcal{S}}_i = \bar{s}_i)} \\ &= \left[\frac{\prod_{j\in c(i)} P(\bar{\mathcal{S}}_j = \bar{s}_j|H_i = k)}{P(\bar{\mathcal{S}}_j = \bar{s}_j)}\right] P(S_i = s_i|H_i = k)P(H_i = k)\frac{\prod_{j\in c(i)} P(\bar{\mathcal{S}}_j = \bar{s}_j)}{P(\bar{\mathcal{S}}_i = \bar{s}_i)} \\ &= \frac{\left[\prod_{j\in c(i)} \beta_{ij}(k)\right] P_{\theta_{k,i}}(s_i)P(H_i = k)}{N_i},\end{aligned} \tag{4.16}$$

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

$$\begin{aligned}N_i &= \frac{P(\bar{\mathcal{S}}_i}{\prod_{j\in c(i)} P(\bar{\mathcal{S}}_j = \bar{s}_j)} \\ &= \sum_{k=1}^{K}\left[\prod_{j\in c(i)} \beta_{ij}(k)\right] P_{\theta_{k,i}}(s_i)P(H_i = k).\end{aligned} \tag{4.17}$$

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

$$\begin{aligned}\beta_{\rho(i)i}(k) &= \frac{P(\bar{\mathcal{S}}_i = \bar{s}_i|H_{\rho(i)} = k)}{P(\bar{\mathcal{S}}_i = \bar{s}_i)} \\ &= \frac{\sum_{g=1}^{K} P(\bar{\mathcal{S}}_i = \bar{s}_i|H_i = g)P(H_i = k|H_{\rho(i)} = k)}{P(\bar{\mathcal{S}}_i = \bar{s}_i)} \\ &= \sum_{g=1}^{K} \frac{P(H_i = g|\bar{\mathcal{S}}_i = \bar{s}_i|)}{P(H_i = g)} P(H_i = g|H_{\rho(i)} = k) \\ &= \sum_{g=1}^{K} \frac{\beta_i(g)\epsilon_i^{(kl)}}{P(H_i = g)}.\end{aligned} \tag{4.18}$$

Those relationships one can derive the the upward algorithm 1.

Meta-parameters :

K

Initialization :

// $P_{\theta_{k,i}}(s_i)$:

for *All the node i of the tree $\mathcal{T}$* **do**

$P_{\theta_{k,i}}(s_i) = \mathcal{N}(s_i|\mu_{k,i}, \sigma_{k,i})$

end

// Loop over the leafs i of the tree :

for *All the leaf i of the tree $\mathcal{T}$* **do**

$\beta_i(k) = \frac{P_{\theta_{k,i}}(s_i)P(H_i=k)}{\sum_{g=1}^{K} P_{\theta_{g,i}}(s_i)P(H_i=g)}$

$\beta_{i,\rho(i)}(k) = \sum_{g=1}^{K} \frac{\beta_i(g)\epsilon_i^{(kg)}}{P(H_i=g)}.P(H_{\rho(i)} = k)$

$l_i = 0$

end

Induction :

// Bottom-Up loop over the nodes of the tree :

for *All non-leaf node i of the tree $\mathcal{T}$* **do**

$M_i = \sum_{k=1}^{K} P_{\theta_{k,i}}(s_i) \prod_{j\in c(i)} \frac{\beta_{j,i}(k)}{P(H_i=k)^{n_i-1}}$

$l_i = \log(M_i) + \sum_{j\in c(i)} l_j$

$\beta_i(k) = \frac{P_{\theta_{k,i}(s_i)} \prod_{j\in c(i)}(\beta_{j,i}(k))}{P(H_i=k)^{n_i-1}M_i}$

for *All the children node j of node i* **do**

$\beta_{i\setminus c(i)}(k) = \frac{\beta_i(k)}{\beta_{i,j}(k)}$

end

$\beta_{i,\rho(i)}(k) = \sum_{g=1}^{K} \frac{\beta_i(g)\epsilon_i^{(gk)}}{P(H_i=g)}.P(H_{\rho(i)} = k)$

end

Algorithm 1: Smoothed upward algorithm.

Downward recursion :

The downward recursion can either be built on the basis of the quantities $\alpha_i(k)$ or on the basis of 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,

$$\xi_0(k) = P(H_i = k|\bar{\mathcal{S}}_0 = \bar{s}_0) = \beta_0(k). \tag{4.19}$$

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

$$\begin{aligned}
\xi_i(k) &= P(H_i = k|\bar{\mathcal{S}}_0 = \bar{s}_0) \\
&= \sum_{g=1}^{K} \frac{P(H_i = k, H_{\rho(i)} = g, \bar{\mathcal{S}}_0 = \bar{s}_0)}{P(H_{\rho(i)} = g, \bar{\mathcal{S}}_0 = \bar{s}_0)} P(H_{\rho(i)} = g|\bar{\mathcal{S}}_0 = \bar{s}_0) \\
&= P(\bar{\mathcal{S}}_i = \bar{s}_i|H_i = k) \sum_{g=1}^{K} \frac{P(H_i = k|H_{\rho(i)} = g)}{P(\bar{\mathcal{S}}_i = \bar{s}_i|H_{\rho(i)} = g)} P(H_{\rho(i)} = g|\bar{\mathcal{S}}_0 = \bar{s}_0) \\
&= \frac{\beta_i(k)}{P(H_i = k)} \sum_{g=1}^{K} \frac{\epsilon_i^{(gk)} \xi_{\rho(i)}(g)}{\beta_{\rho(i),i}(g)}.
\end{aligned} \tag{4.20}$$

Using the fact that for all $i \in mcalT \quad \xi_i(k) = \beta_i(k)\alpha_i(k)$ and the relationship from 4.20, one can express the downward pass as presented in 2.

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

Algorithm 2: Smoothed downward algorithm.

Conditional properties :

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

$$\forall i \in \mathcal{T} \quad P(H_i = k|\bar{\mathcal{S}}_0 = \bar{s}_0) = \alpha_i(k)\beta_i(k), \tag{4.21}$$

and

$$\forall i \in \mathcal{T}\backslash\{0\} \quad P(H_{\rho(i)} = g, H_i = k|\bar{\mathcal{S}}_0 = \bar{s}_0) = \frac{\beta_i(k)\epsilon_i^{(gk)}\alpha_{\rho(i)}(g)\beta_{\rho(i)}(g)}{P(H_i = k)}. \tag{4.22}$$

4.3.2 M-Step :

The *Maximization* step of the EM algorithm aims to maximize the log-likelihood of the observation with regards to the parameters and then use those pseudo-optimal parameters for the next *Expectation* step. Meaning that at the iteration m of the EM process, the M-step does,

$$\Theta^{m+1} = \underset{\Theta}{\operatorname{argmax}} \left(E[\ln f(x, H|\Theta)|x, \Theta^l] \right). \tag{4.23}$$

The Θ maximizing the log-likelihood in 4.23 can be express analytically and this yields to the algorithm 3

Meta-parameters :
K,
Distribution family for P_θ ; // 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 node i of the tree* $\mathcal{T}\backslash\{0\}$ **do**
$P(H_i = k) = \frac{1}{N} \sum_{n=1}^{N} P(H_i^n = m|s_i^n, \Theta^l)$,
$\epsilon_i^{gk} = \frac{\sum_{n=1}^{N} P(H_i^n=k, H_{\rho(i)}^n=g|w^n, \Theta^l)}{NP(H_{\rho(i)}=k)}$,
$\mu_{k,i} = \frac{\sum_{n=1}^{N} s_i^n P(H_i^n=k|w^n, \Theta^l)}{NP(H_i=k)}$,
$\sigma_{k,i}^2 = \frac{\sum_{n=1}^{N} (s_i^n - \mu_{k,i})^2 P(H_i^n=k|w^n, \Theta^l)}{NP(H_i=k)}$.
end

Algorithm 3: M-step of the EM algorithm.

4.3.3 EM algorithm :

Finally the EM algorithm iterates over the *E-step* and the *M-step* as describe by the algorithm 4.

4.4 Classification :

Let Θ_c now be a set of parameters for an SCHMT $\mathcal{T}$ learned using the algorithm described in section 4.3 and a training set X_c composed of N realization of a signal of class c . Let also x_{new} be another realization of this signal, not used for training and $\mathcal{T}^{(new)}$ be the instance of the SCHMT generated by this realization.

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

For SCHMT the MAP algorithm has the form describe by the algorithm 5.

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

Algorithm 4: EM algorithm.

Meta-parameters :
K ;
Initialization :
for *all leaf* 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 node* i *of the tree* $\mathcal{T}\backslash\{0\}$ **do**
  $\gamma_i(k) = P_{\theta_{k,i}}(s_i) \prod_{j\in c(i)} \gamma_{j,i}(k)$
  $\gamma_{i,\rho i}(k) = \max_{1\leq g\leq K} \gamma_i(g)\epsilon_i^{kg}$; // 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 node* i *of the tree* $\mathcal{T}\backslash\{0\}$ **do**
  $\hat{h}_i = \xi_i(\hat{h}_{\rho(i)})$
end

Algorithm 5: MAP algorithm.

The MAP algorithm 5 can be used in a multi-class classification problem simply by training a SCHMT model per class and then for a new realization x_{new} simply compare the MAP provided by each model as describe by the algorithm 6.

Meta-parameters :
K ; C ; // Number of class
for *All the class c* **do**
  $\hat{P}_c = \text{MAP}(x_{new}, \Theta_c, K)$
end
$\hat{P} = \max_{0 \leq c < C} \hat{P}_c$
$l = \text{argmax}_{0 \leq c < C} \hat{P}_c$

Algorithm 6: MAP algorithm appplied to multiclass classification problem.

5 Experimental results :

This section presents some experimental results obtained using scattering hidden Markov trees for classification tasks. First a binary classification task is performed on simple simulated shapes. Then SCHMT are applied to classify seabed and ripples in sonar imagery.

5.1 Shapes :

The SCHMT performances on classification tasks are at first assessed on telling appart different geometrical shapes. The data are noisy and translated realizations of two signals. The first signal is an 100 by 100 binary image picturing a centered black square on a white background. Examples of this class can be seen in figure 5.1. The second class is a 100 by 100 binary image picturing a centered black triangle on a white background. Examples of this class can be seen in figure 5.2.

5.2 Sonar Imagery :

The SCHMT is now tested on a more complex set of images. The data used are extracted from the ***UDRC MCM sonar dataset*** `https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/326061/DefenceReporter_Winter2009.pdf`. This dataset, designed initially for underwater mines detection, contains targets comparable to those faced by coalition forces in current operational engagements and comprises of Synthetic Aperture RADAR (SAS imagery) and Hyperspectral data. From those very high dimensional images, 100 by 100 patches have been extracted and labeled as either ***seabed*** or ***ripple*** (see respectively figure 5.3 and figure 5.4).

The task at hand is the classification of those realizations of those two signals. The scattering transform used has $m = 3$ orders, $J = 5$ scales, $M = 3$ orientations and uses a Morlet

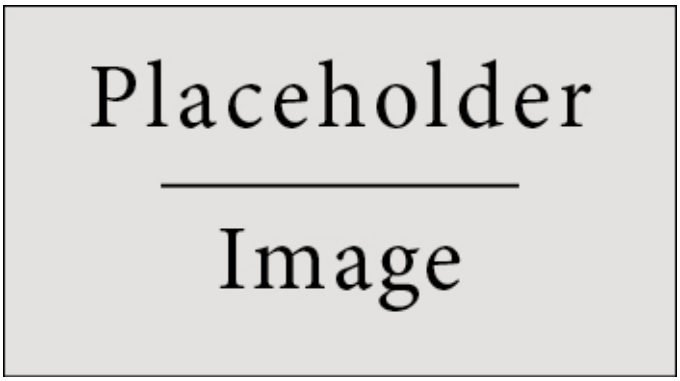

FIGURE 5.1 – (a) Orignial signal (b) realization

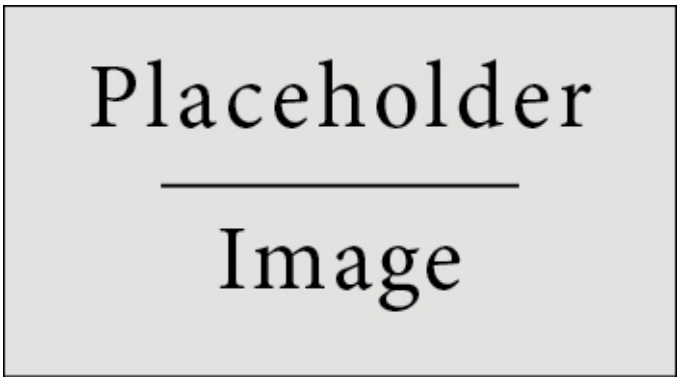

FIGURE 5.2 – (a) Orignial signal (b) realization

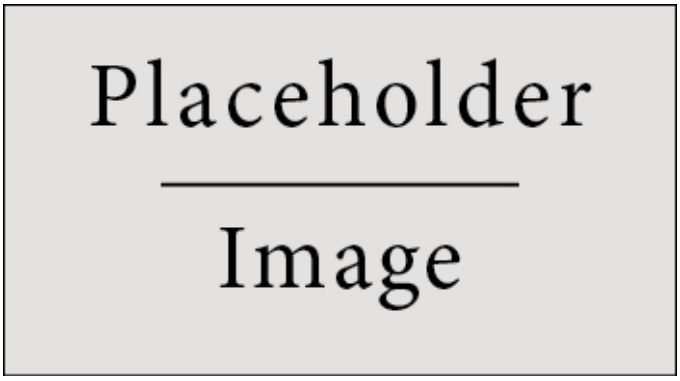

FIGURE 5.3 – Seabed patchs

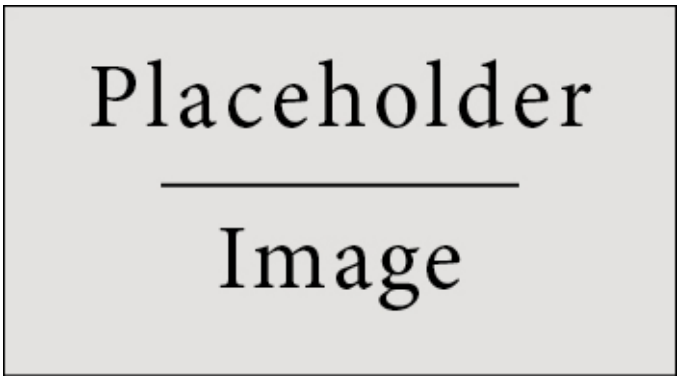

FIGURE 5.4 – Ripple patchs

Classification results				
Classification score :	Mean	Variance	Maximum	Minimum
With outliers :	0	0	0	0
Without outliers :	0	0	0	0

TABLE 5.1 – Classification performances over 100 experiments of Ripple/Seabed classification.

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- 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 performances of the SCHMT are assessed on 100 instances of this experiment and the results are displayed in table **??**.

The results with best scores at 90% accuracy shows encouraging results as per the classification performances of the SCHMTs. However one can notice that some experiment provides scores as low as 50% accuracy. Those scores are obtained when one model has shown a better convergence than the other, resulting to classifying all the instances of the test set to one class. This problem could be addressed by using a validation set to discard those models before testing. Another leads would be to work on the EM algorithm to make it more robust.

6 Conclusion :

Scattering hidden Markov tree : The SCHMT appears to be a powerful model combining the interesting properties of the scattering transform for signal representation and the representational power of the hidden Markov models. It provides with analytically tractable interesting representation of high-dimensional signal as well as a probabilistic classification method. Experimental results confirm this intuition but also reveal some weakness of our model.

Next steps : Despite showing good performances when the learning phase has converged properly, SCHMT's performances are undermined by a sometime poor learning quality. This observation is one of the main drivers for the upcoming work. The other main driver of for the next step is to keep a well define probabilist framework.

At this point the version of the EM algorithm considered in this report is using full Bayesian inference. However this methods is computationally expensive and sometimes yields to poor learning. Furthermore this version of the EM algorithm provides only a point-wise estimate of the model's parameters. It would be interesting to apply ***variational methods*** the this problem (*cite : Graphical Models, Exponential Families, and Variational Inference*. Beside a potential improvement of the performance variational inference would also provide a estimated distribution for the model's parameter, thus allowing us to have access to a ***measure of uncertainty*** for our learning.

Another interesting direction to follow is the works on ***hierarchical graphical models cite : The Hierarchical Hidden Markov Model : Analysis and Applications***. Those models would allow us to use the SCHMT model as a node of a wider probabilistic graphical model. Using such architecture, one could model a network of sensors each providing information on a targeted scene.

Finally another interesting lead would be to consider other architecture for the graphical model and make it includes the representation learning step. This would be possible using ***Bayesian neural networks cite : Bayes backprop and PBP***.

7 Acknowledgements :

Jean-Baptiste Regli is funded by a Dstl/UCL Impact studentship and James Nelson is partially supported by grants from the Dstl and Innovate UK/EPSRC.Fpap

Bibliographie