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 :

Our researches have been focused on ***high-dimensional classification problems***. Meaning that throughout this paper we will be working with several -says N- realizations of a signal $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ where,

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

And we are interested in ***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 samled values of the signal needed to find a neighbor to a new realization $\mathbf{x}$ grow exponentially with the number of dimensions.

To over come this issue one could try to reduce the number of dimensions of the problem. In the simple case the signal$\mathbf{x}$ belongs to a subset $\Omega \subset \mathbb{R}^d$ where Ω .

1.2 Image representation :

We are now interested in projecting our signal into a new space where the classification task would be simpler. To do so we need to project the signal into a "good" space, i.e. crate a good representation of our data. We will first provide an intuition of what "good" is and then provide more formal mathematical definitions for our intuition and how this can be achieved.

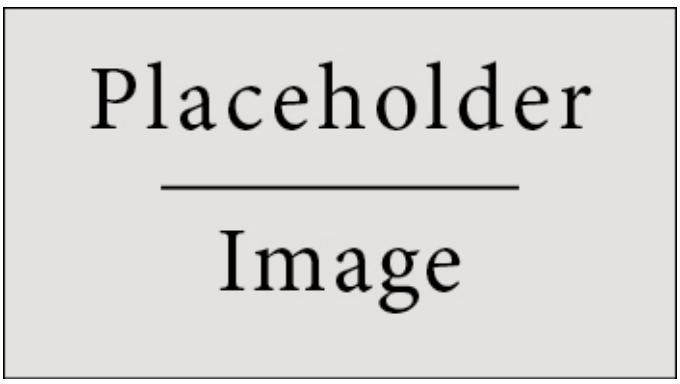

FIGURE 1.1 – (a) Sound waveform (b) Picture

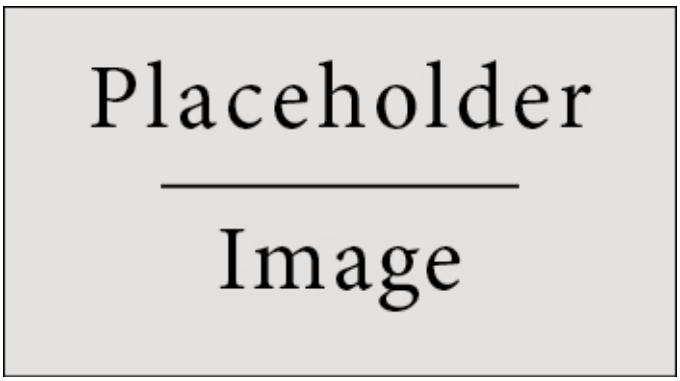

FIGURE 1.2 – A human can easily tell that those two images are from the same class.

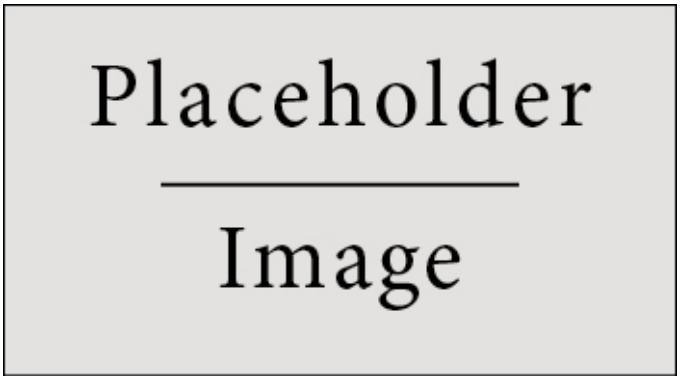

FIGURE 1.3 – A human can easily tell that (a) and (b) are from the same class. (c) can still be recognized even though it is slightly more challenging.

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 the human visual function and what he is able to tell apart.

Based on that, we think our representation should be :
— ***Informative*** enough to permit classification.

— ***Invariant to translations***. Indeed to a human eye there is no difference in the information carried by a signal if it is shifted.

— ***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.

— ***To a certain degree invariant to rotations***. Rotations cannot be handled as easily as translation. Indeed here one is after a local rotational invariance rather than a global one. Solutions exist to develop a scattering transform with such behaviour ***citation*** but this will not be addressed in this review.

1.2.2 Formalization of a "good" image representation :

To formalize the intuitions on the representation stated earlier, we first need to define the signal. Throughout this document, a signal will be defined as,

Definition 1.2.1. Signal

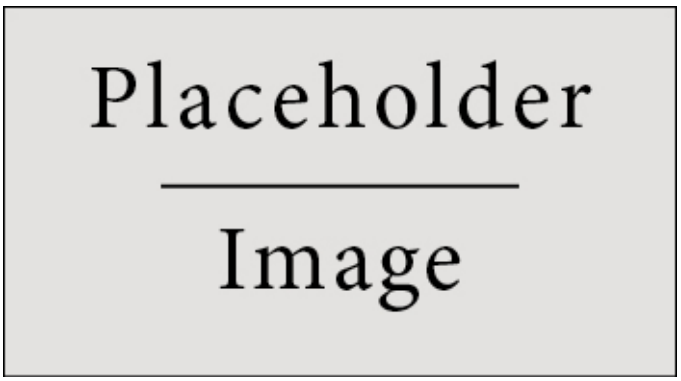

FIGURE 1.4 – A human can easily tell that (a) and (b) are from the same class. (c) could be a '6' slightly rotated or a '9' heavily rotated.

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

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

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,

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

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 5 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,

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

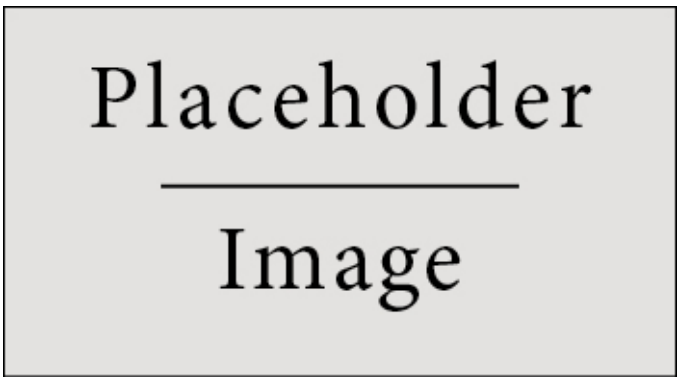

FIGURE 2.1 – Complex Morlet wavelet.

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.

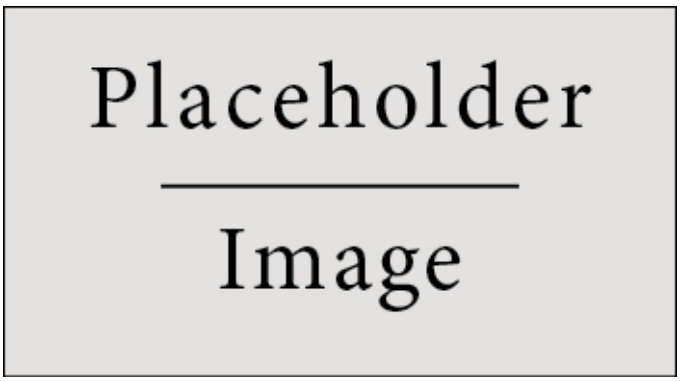

FIGURE 2.2 – see mallat long paper p1341

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***.

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,

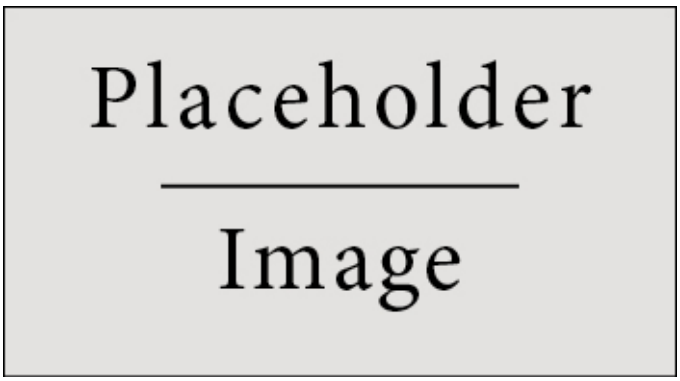

FIGURE 3.1 – Example of Bayesian network.

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 $i \in [\![1, I]\!]$ denotes one of the node $S[p_i]x \in \mathcal{T}$ for a given path $p_i = [\lambda_1 ... \lambda_k]$ ($k \in \mathbb{N}$). Note that for the sake of clarity the signal dependence of the node S_i has been removed from the notation and in the remainder of the paper the short notation $i \in \mathcal{T}$ will be use to denote a path belonging to the tree. 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.

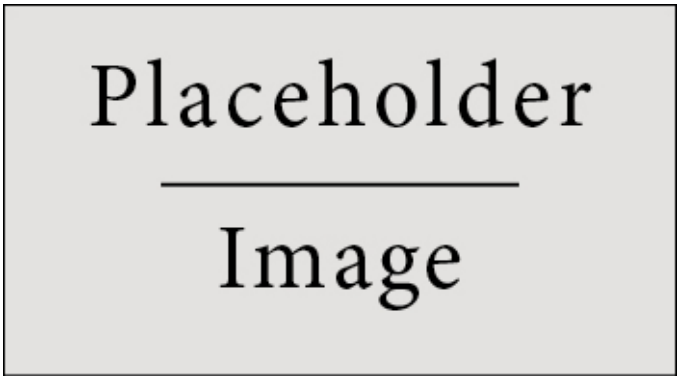

FIGURE 4.1 – Scattering transform tree.

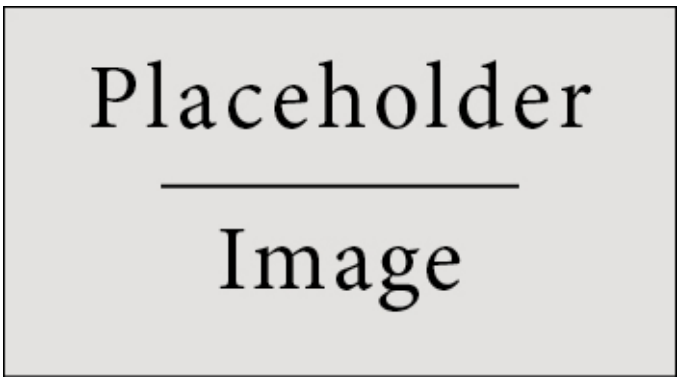

FIGURE 4.2 – Scattering Hidden Markov Tree.

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

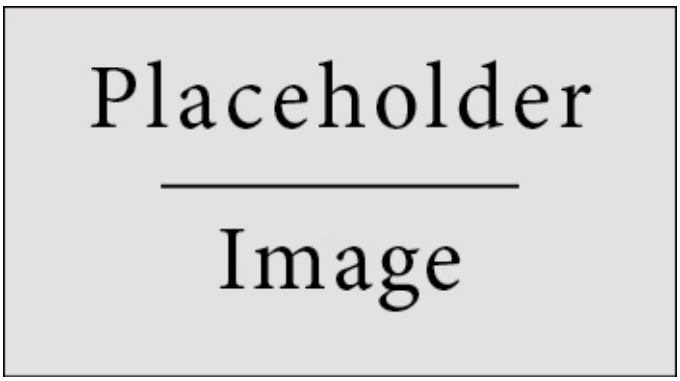

FIGURE 4.3 – Wavelet hidden Markov tree.

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 distribution π_0 such that :

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

— For any index i of the tree, the probability of the visible node S_i is a mixture of Gaussian conditional to the hidden state H_i.

$$\forall i \in \mathcal{T} \quad P(S_i|H_i) = \sum_{k=1}^{K} w_{k,i}\mathcal{N}(\mu_{k,i}, \sigma_{k,i}). \tag{4.5}$$

where $\mu_{k,i}$, $\sigma_{k,i}$ and w_i are respectively the mean, the variance and the weight 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 matrix,

$$\forall i \in \mathcal{T}\backslash\{0\} \quad P(H_i|H_{\rho(i)}) = \pi_i P(H_{\rho(i)}), \tag{4.6}$$

where,

$$\forall i \in \mathcal{T}\backslash\{0\} \quad \pi_i = \left[P(H_i = m|H_{\rho(i)} = n)\right]_{m,n\in[\![1,K]\!]^2}. \tag{4.7}$$

Using the chain rule of probability $P(H_i)$ can be expressed from the root node as,

$$\forall i \in \mathcal{T}\backslash\{0\} \quad P(H_i|H_{\rho(i)}) = \prod \pi_i P(H_{\rho(i)}), \tag{4.8}$$

cite : Durand improved the learning algorithm proposed by MS. Crouse and ***cite : Kingsburry*** adapted this model to the dual complex wavelet trees.

While having the same

4.2 Hypothesis :

(1) 2 populations. (2) Persistence.

4.3 Learning the tree structure :

Hopefully some kind of proof there.

4.4 Classification :

Hopefully some kind of proof there.

5 Experimental results :

TBD

6 Conclusion :

Scattering hidden Markov tree : TBD

Next steps : Variational methods General graphical models Bayesian neural networks

7 Acknowledgements :

Bibliographie