SCATTERING HIDDEN MARKOV TREE

J.B. REGLI, J. D. B. NELSON

UCL, Department of statistical science

ABSTRACT

A Scattering Convolutional Hidden Markov Tree (SCHMT) proposes a new inference mechanism for high-dimensional signals by combining the interesting signal representation created by the Scattering Transform (ST) to a powerful Probabilistic Graphical Model (PGM). A wavelet scattering network computes a signal translation invariant and stable to deformations representation that still preserves the informative content of the signal. Such properties are acquired by cascading wavelet transform convolutions with nonlinear modulus and averaging operators. The network's structure and its distributions are described using a Hidden Markov Tree (HMT). This yield a generative model for high-dimensional inference. It offers a mean for performing several inference tasks among which are predictions. The scattering convolutional hidden Markov tree displays promising results on both classification and segmentation tasks of complex images.

***Index Terms*—** Scattering network, Hidden Markov Model, Classification, Deep network

1 Introduction

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

We proposes a method combining a recently proposed deterministic analytically tractable transformation inspired by deep convolutional to a probabilistic graphical model in order to create a powerful probabilistic tool to handle high dimensional prediction problems. In a similar fashion to the work done by Crouse on wavelet trees [**?**], we propose to describe Mallat's scattering convolutional scattering transform [**?**] using a hidden Markov tree. Doing so we develop a new framework to model high-dimensional inputs. As opposed to the commonly used simple classification method, once trained our model can tackle prediction problems but also other inference tasks —e.g. generation, sensitivity analysis...

In Section 2 we present the requisite background of high dimensional signal classification. Section 3 introduces the Scattering Transform and some of its properties. We fuse these to an hidden Markov Tree concepts in Section 4, propose our Scattering Hidden Markov Tree (SCHMT), and describe the inferential machinery. In Section 5 we perform classification on a selection of standard datasets. We draw conclusions in Section **??**

2 Background

Insist on the lot of training example needed for DL. Present ST has an alternative to lower the number of samples required.

Introduces Hidden Markov tree as an alternative to SVM providing generative model and better low proba perf

3 Scattering networks

Scattering convolutional networks (SCNs) **?** are Convolutional Neural Networks (CNNs) **?** using a fixed filter bank of wavelets. Those filters can hand-crafted to yield descriptors with the desired invariances **???**. For image classification tasks, one is interested in descriptors that are —at least— stable to deformations and invariant to translations. Note that SCNs producing more complexes set of invariances exist but on the remainder of this paper we consider only on descriptors with the previously mentioned properties.

3.1 Scattering transform

Wavelets are localized functions stable to deformations. They are thus well adapted to construct descriptor that would also be translation invariant. A two-dimensional spatial wavelet transform W is obtained by scaling by 2^j and rotating by r_θ a mother wavelet ψ,

$$\psi_\lambda(u) = \psi_{j,\theta}(u) = 2^{-2j}\psi(2^{-j}r_\theta p) \qquad (1)$$

Thanks DSTL/UCL Impact studentship for funding.

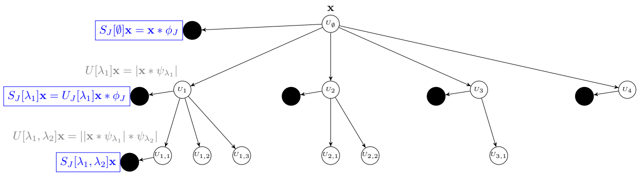

Fig. 1. Scattering networks can be seen as neural networks iterating over wavelet modulus operators $\mathcal{U}_j$. Each layer m outputs the averaged invariants $S[p_m]\mathbf{x}$ and covariant coefficients $U[p_{m+1}]\mathbf{x}$.

In the remainder of this paper we restrict to Morlet wavelet transforms defined on $\Lambda = G \times [\![0, J]\!]$ where G is a finite group of rotations of cardinal L and where the wavelet is taken at scale J,

$$W\mathbf{x} = \{\mathbf{x} * \phi_J(u); \mathbf{x} * \psi_\lambda(u)\}_{p \in \mathbb{R}^2, \lambda \in \Lambda} \quad (2)$$

While the averaging part ϕ_J of the wavelet transform is invariant to translations, the high frequency part ψ_λ is covariant to them **?**. Invariance within a limited range inferior 2^J can be achieved by averaging the positive envelope with a smooth window,

$$S_J[\lambda]\mathbf{x}(u) = |\mathbf{x} * \psi_\lambda| * \phi_J(u) \quad (3)$$

Such non-linearised averaged wavelet coefficients are used in various form in computer vision (SIFT , DAISY), but the scattering transform proposes a new non-linearity as well as a layer based architecture.

3.2 Scattering convolutional network

While providing local translation invariance, the averaging convolution introduced in 3 also removes the spatial variability of the wavelet transform. SCNs cascade this wavelet modulus operator to recover the lost information and compute progressively more invariant descriptors. Let combine the wavelet transform and modulus operations into a single wavelet modulus operator,

$$\mathcal{U}_J\mathbf{x} = \{S[\emptyset]\mathbf{x}; U[\lambda]\mathbf{x}\}_{\lambda \in \Lambda_J} = \{\mathbf{x}*\phi_J; |\mathbf{x} * \psi_\lambda|\}_{\lambda \in \Lambda_J}, \quad (4)$$

A scattering transform can be interpreted as a CNN illustrated in Figure 1 **?** which propagates a signal $\mathbf{x}$ across multiple layers of the network and which outputs at each layer m the scattering invariant coefficients $S[p_m]\mathbf{x}$ where $p_m = (\lambda_1 \ldots \lambda_m)$ is a path of m orientations and scales.

The scattering energy is mainly concentrated along frequency decreasing paths, i.e. for which $|\lambda_{k+1}| \leq |\lambda_k|$ **?**. The energy contained in the other paths is negligible and thus for applications only frequency decreasing paths are considered. Moreover there exist a path length $M > 0$ after which all longer paths can be neglected. For signal processing applications, this decay appears to be exponential. And for classification applications, paths of length $M = 3$ provides the most interesting results [**?**], [**?**].

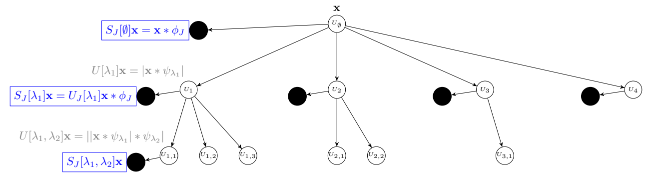

Fig. 2. Frequency decreasing scattering convolution network with $J = 4$, $L = 1$ and $M = 2$. A node i at scale j_i generates $(j_i - 1) \times L$ nodes.

This restrictions yield an easier parametrization of a scattering network. Indeed its now completely defined by the mother wavelet ϕ, the maximum path length considered M, the finest scale level considered J and the number of orientation considered L.

Hence for a given set of parameter (ψ, M, J, L), let $ST_{(\psi,M,J,L)}(\mathbf{x})$ denotes the unique frequency decreasing windowed scattering convolutional network with those parameters evaluated for signal $\mathbf{x}$. Each node i of this network generates a -possibly empty- set of of nodes of size $(j_i - 1) \times L$ where j_i is the scale of node i and L is the number of orientations considered and it has the architecture displayed by Figure 2.

4 The Scattering hidden Markov tree :

? introduced the use of scattering networks combined with a support vector machine classifier to achieve competitive classification performance on some problems **?**. However this method only provides a boolean label for each class. Methods to express the output of an SVM as a probability exists **?** 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 coefficients, it is quite natural to try expressing them as a probabilistic graphical model. Furthermore generative models are known to have better accuracy for inference tasks when the number of training example is low **?**.

4.1 Hidden Markov tree model

We propose an adaptation of the wavelet hidden Markov trees **?** to create the SCHMTs. We model an SCN by a hidden Markov tree composed of a set of visible nodes $\{\mathbf{S}_i\}_{i \in \mathcal{T}}$ and a set of hidden node $\{\mathbf{H}_i\}_{i \in \mathcal{T}}$. Both sets are organized in a tree structure such that for any index i of the tree, $S_i \in \mathbb{R}$ and $H_i \in [\![1, K]\!]$ where K is the number of possible hidden states. The initial state is drawn from a discrete non uniform distribution π_0 such that, $\forall k \in [\![1, K]\!]$ $\pi_0(k) = P(H_0 = k)$. For any index i of the tree, the emission distribution describes the probability of the visible node S_i conditional to the hidden state H_i such that, $\forall i \in \mathcal{T}$, $\forall k \in [\![1, K]\!]$ and $\forall s \in \mathbb{R}$ $P(S_i = s_i | H_i = k) = P_{\theta_{k,i}}(s)$, where $P_{\theta_{k,i}}$ belongs to a pa-

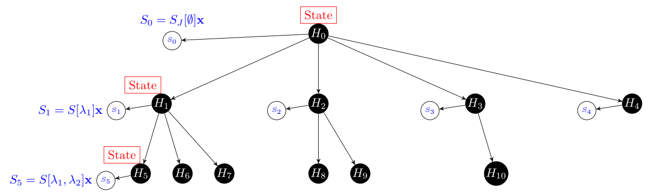

Fig. 3. Scattering convolutional hidden Markov tree.

rametric distribution family and $\theta_{k,i}$ is the vector of emission parameters for the state k and node i. In the remainder of the paper the emission distribution is Gaussian so that $P(S_i = s|H_i = k) = \mathcal{N}(\mu_{k,i}, \sigma_{k,i})$, where $\theta_{k,i} = (\mu_{k,i}, \sigma_{k,i})$ with $\mu_{k,i}$ and $\sigma_{k,i}$ being respectively the mean and the variance of the Gaussian for the k-th value of the mixture and the node i. Finally the probability for the hidden node H_i to be in a state k given its father's state g is characterized by a transition probability such that $\forall i \in \mathcal{T}\backslash\{0\}\ \forall g, k \in [\![1, K]\!]^2\ \epsilon_i^{(gk)} = P(H_i = k|H_{\rho(i)} = g)$ where ϵ_i defines a transition probability matrix such that $P(H_i = k) = \sum_{g=1}^{K} \epsilon_i^{(gk)} P(H_{\rho(i)} = g)$.

Such a model is pictured in Figure 3 and for a given scattering architecture —i.e. fixed M, J and L— the SCHMT model is fully parametrized by,

$$\Theta = \big(\pi_0, \{\epsilon_i, \{\theta_{k,i}\}_{k\in[\![1,K]\!]}\}_{i\in\mathcal{T}}\big). \tag{5}$$

This model implies to do two assumptions on the scattering transform. First one need to assume — K-populations— that a signal's scattering coefficients can be described by K clusters. This is a common assumptions for standard wavelets [?] and hence it can be extended to the scattering transform. The SCHMT also assumed —persistence— that the informative character of a coefficients is propagated across layers. This assumption is sound since ...

4.2 Learning the tree parameters

The SCHMT is trained using the smoothed version of the Expectation-Maximisation algorithm (**?**) for hidden Markov trees proposed by [**?**] and adapted to non-homogeneous and non-binary trees.

5 Classification results

6 Conclusion

// Initialization :
for *All the nodes i of the tree $\mathcal{T}$* **do**
  $P_{\theta_{k,i}}(s_i) = \mathcal{N}(s_i|\mu_{k,i}, \sigma_{k,i})$
end
for *All the leaves 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 :
for *All non-leaf nodes i of the tree $\mathcal{T}$ (Bottom-up)* **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 nodes j of node i* **do**
    $\beta_{i\backslash c(i)}(k) = \frac{\beta_i(k)}{\beta_{i,j}(k)}$
  end
  $\beta_{i,\rho(i)}(k) = \sum_{g=1}^{K} \frac{\beta_i(g)\epsilon_i^{(kg)}}{P(H_i=g)}.P(H_{\rho(i)} = k)$
end

Algorithm 1: Smoothed upward algorithm.

// Initialization :
$\alpha_0(k) = 1$
// Induction :
for *All nodes i of the tree $\mathcal{T}\backslash\{0\}$ (Top-Down)* **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.

// Initialization :
$\pi_0(k) = \frac{1}{N}\sum_{n=1}^{N} P(H_0^n = m|s_0^n, \Theta^l)$
// Induction :
for *All nodes i of the tree $\mathcal{T}\backslash\{0\}$* **do**
  $P(H_i = k) = \frac{1}{N}\sum_{n=1}^{N} P(H_i^n = k|\bar{s}_0^n, \Theta^l)$,
  $\epsilon_i^{gk} = \frac{\sum_{n=1}^{N} P(H_i^n=k, H_{\rho(i)}^n=g|\bar{s}_0^n, \Theta^l)}{NP(H_{\rho(i)}=k)}$,
  $\mu_{k,i} = \frac{\sum_{n=1}^{N} s_i^n P(H_i^n=k|\bar{s}_0^n, \Theta^l)}{NP(H_i=k)}$,
  $\sigma_{k,i}^2 = \frac{\sum_{n=1}^{N} (s_i^n-\mu_{k,i})^2 P(H_i^n=k|\bar{s}_0^n, \Theta^l)}{NP(H_i=k)}$.
end

Algorithm 3: M-step of the EM algorithm.

7 COPYRIGHT FORMS

łlabelsec :copyright

You must include your fully completed, signed IEEE copyright release form when form when you submit your paper. We **must** have this form before your paper can be published in the proceedings.

8 REFERENCES