SCATTERING CONVOLUTIONAL HIDDEN MARKOV TREE

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

UCL, Department of Statistical Science

ABSTRACT

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

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

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

When only a very small number of training examples are available one-shot learning [**?**] generative classification methods achieve significantly better results than their discriminative counterparts [**?**]. Generative probabilistic graphical models have been successfully constructed for various wavelet transforms; in particular, Hidden Markov trees have been used to model the wavelet to model the dependencies between coefficients [**?**, **?**, **?**].

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

The remainder of this paper introduces the scattering convolutional hidden Markov tree and is organised as follow. In section 2 we review the scattering transform and some of its properties. Section 3 introduces the proposed Scattering Hidden Markov Tree (SCHMT). In Section 4 we perform classification on a selection of standard datasets restricted to only a few training samples. We draw conclusions in Section 5.

J.-B. Regli is funded by a Dstl/UCL Impact studentship. J. D. B. Nelson is partially supported by grants from the Dstl and Innovate UK/EPSRC.

2 Scattering networks

Scattering convolutional networks (SCNs) [**?**] are Convolutional Neural Networks (CNNs) [**?**] that use a fixed filter bank of wavelets. Filters can be hand-crafted to yield descriptors with various desired invariances [**?**]. For image classification tasks, one is interested in descriptors that are at least stable to deformations and invariant to translations. Although SCNs can produce a more complex set of invariances, we here focus attention to deformations and translations only.

2.1 Scattering transform

Wavelets are localized functions stable to deformations and can be adapted to construct descriptors that are 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:

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

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

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

Whilst 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 tp 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.

2.2 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. The wavelet transform and modulus operations are combined into a single wavelet modulus operator, thus

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

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 $p_m = (\lambda_1 \dots \lambda_m)$ 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

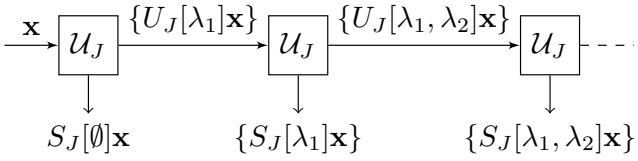

Fig. 1. Scattering networks can be seen as neural networks iterating over wavelet modulus operators $\mathcal{U}_j$. Each layer m outputs the averaged invariants $S[p_m]\mathbf{x}$ and covariant coefficients $U[p_{m+1}]\mathbf{x}$.

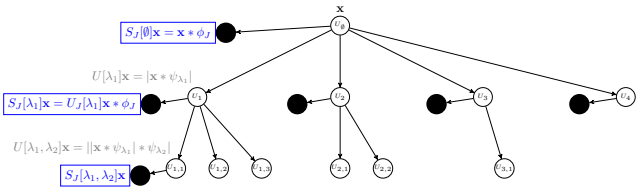

Fig. 2. Frequency decreasing scattering convolution network with $J = 4$, $L = 1$ and $M = 2$. A node i at scale j_i generates $(j_i - 1) \times L$ nodes.

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 $M = 3$, i.e. two convolutions, provide the most interesting results [**?**, **?**].

This restrictions yield a more convenient 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 parameters (ψ, M, J, L), let $ST_{(\psi,M,J,L)}(\mathbf{x})$ denote the unique frequency decreasing windowed scattering convolutional network with those parameters evaluated forsignal $\mathbf{x}$. 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.

2.3 Scattering convolutional classifier

In the original framework [**?**], the scattering network $ST_{(\psi,M,J,L)}(.)$ is used for a classification task using a SVM classifier on the outputs of the network. Performance can be slightly improved by adding a feature selection step to perform PCA on the scattering coefficients and keep only the most informative ones. This classification framework provides results comparable with the state of the art on several datasets [**?**].

3 The Scattering hidden Markov tree

State of the art performance can be achieved using SCNs associated with SVMs. However this approach is not adapted to

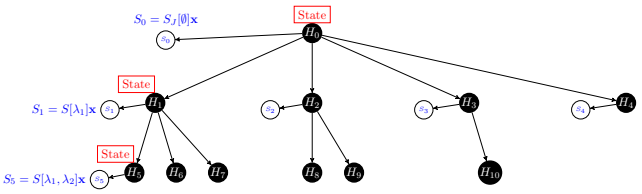

Fig. 3. Scattering convolutional hidden Markov tree.

very small training sets; nor does it deliver a probabilistic output. To overcome these limitations we propose an adaptation of the Crouse [**?**] and Durand [**?**] wavelet hidden Markov trees to the non-regular, non-homogeneous tree structure of SCNs.

3.1 Hidden Markov tree model

The HMT models the marginal distribution of each real ST coefficient S_i as a Gaussian mixture. To each S_i, we associate a discrete hidden state H_i that takes on values in $[\![1, K]\!]$ with probability mass function (pmf) $P(H_i)$. Conditioned on $H_i = k$, S_i is Gaussian with mean $\mu_{i,k}$ and variance $\sigma_{i,k}$. Thus, its overall marginal PDF is given by $P(w_i) = \sum_{k=1}^{K} P(H_i = m)P(S_i|H_i = k)$ with $P(S_i|H_i = k) \sim \mathcal{N}(\mu_{i,k}, \sigma_{i,k})$. While each scattering coefficient S_i is conditionally gaussian given its state variable H_i, overall it has a non-Gaussian density. 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 $\epsilon_i^{(gk)} = P(H_i = k|H_{\rho(i)} = g)$. This yields $P(H_i = k) = \sum_{g=1}^{K} \epsilon_i^{(gk)} P(H_{\rho(i)} = g)$.

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,

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

This model implies two assumptions on the scattering transform. We firstly posit K-populations; i.e. that the scattering coefficients of a given signal can be described by K clusters. This is a commonassumptions for standard wavelets [**?**] and it can hence be extended to the scattering transform. The SCHMT also assumes persistence in that the informative character of a coefficient is propagated across layers. This assumption is justified by the fact that scattering coefficients are highly correlated [**?**].

3.2 Learning the tree parameters

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

Let $\bar{\mathcal{S}}_i = \bar{s}_i$ be the observed sub-tree rooted at node i. By convention $\bar{\mathcal{S}}_0$ denotes the entire observed tree. The smoothed version of the E-step requires the computation of the conditional probability distributions $\xi_i(k) = P(H_i = k|\bar{\mathcal{S}}_i = \bar{s}_i)$ (smoothed probability) and $P(H_i = k, H_{\rho(i)} = g|\bar{\mathcal{S}}_i = \bar{s}_i)$ for each node $i \in \mathcal{T}$ and states k and g. This can be achieved through an upward-downward recursion displayed in Algorithm 1 and 2. The output from the downward step are used in the M-step as shown in Algorithm 3.

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

Algorithm 1: Smoothed upward algorithm.

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

3.3 MAP classification

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

In this context the MAP algorithm [**?**] aims to find the optimal hidden tree $\hat{h}_0^{new} = (\hat{h}_0^{new}, \ldots, \hat{h}_{I-1}^{new})$ that maximizes the probability of this sequence given the model's parameters $P(\bar{\mathcal{H}}_0 = \hat{h}_0^{new}|\mathcal{T}^{new}, \Theta_c)$. The MAP framework also provides $\hat{P}$ the value of this maximum.

cell1	cell2	cell3
cell4	cell5	cell6
cell7	cell8	cell9

Fig. 4. Classification score on 200 samples (top row) and on the full testset (bottom row) of MNIST and KTH-TIPS trained using only a limited number of training points per class

4 Classification results

We compare the performance of SCHMT to those of a SCN combined with an SVM (SCN+SVM) restricted to a small number of training examples by performing two experiments on the digit classification dataset MNIST [**?**]. The MNIST dataset contains 60000 training and 10000 test images of ten handwritten digits (zero to nine), with 28×28 pixels. For both experiments we use a scattering transform with $M = 3$ orders, $J = 3$ scales, $L = 3$ orientations and a Morlet mother wavelet. The hidden Markov tree has $K = 2$ states and uses a mixture of Gaussian to describe the relationship between the scattering coefficients and the hidden states. For the SVM, the best parameters are selected by cross-validation.

4.1 MNIST - 1 vs All

In a similar fashion to [**?**], we first test SCHMT on a "one vs all" basis. However we propose this experiment with a more challenging setup. Indeed in [**?**] they pretrain their model with 100 samples from each class but one. They then provide only limited amount of training examples, say N, for this last class. Instead we propose a framework where all the classes have the same limited amount of training points N. Like [**?**] we then test the models on 1000 examples.

Table 4 displays the accuracy and positive likelihood ratio (PLR) belonging to one class versus all the others for both SCHMTs and SCN+SVM. It also displays the averaged accuracy over the 10 classes. Note that this experimental setup should be in favor of the SVM classifier since it is effectively provided with $9N$ training points for one class. However SCHMT outperforms SCN+SVM on both average accuracy and average PLR.

4.2 MNIST - Full

SCHMT and SCN+SVM are then test on the much more complex problem of mutliclass classification on the full MNIST testset. Again SCHMT and SCN+SVM are both trained on a limited number of training examples per class and tested on 200 test samples (20 per class) and on the full testset (10000 images). The best results for each models are displayed in Figure 5.

When the number of training points is extremely limited —i.e. 2 or 5— SVM can reach acceptable score on a small testset. However the model learned does not generalize well and suffers significant degradation in performance when the test set is scaled up. In contrast the SCHMT displays better generalization as its score remains almost constant as the number of test samples are increased. Finally it outperforms SVMs on large testset. As expected, when the number of training samples grow large enough —i.e. 10 and onward— for the SVMs, SCN+SVMs reach better maximum classification score on both the small and the large testset.

Training samples		MNIST		KTH-TIPS	
		SCHMT	SCN+SVM	SCHMT	SCN+SVM
2	(20)	35.0%	**51.5%**	28.9%	**32.2%**
	(Full)	**28.6%**	18.7%	**29.1%**	25.6%
5	(20)	50.0%	**56.5%**	35.1%	**46.5%**
	(Full)	**48.0%**	43.2%	**35.5%**	33.2%
10	(20)	51.3%	**62.0%**	49.1%	**57.7%**
	(Full)	51.2%	**49.9%**	49.4%	**55.3%**
20	(20)	50.0%	**82.3%**	48.3%	**78.0%**
	(Full)	50.0%	**79.8%**	48.7%	**76.1%**

Fig. 5. Classification score on 200 samples (top row) and on the full testset (bottom row) of MNIST and KTH-TIPS trained using only a limited number of training points per class

We can also notice that EM algorithm performances are undermined by convergence to local minima issues [**?**]. This sometimes yields poor learning quality for the SCHMT. However when convergence occurs correctly SCHMTs has better generalization performance than the best SVMs.

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

5 Conclusion

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

To enhance SCHMT and especially the chance of converging toward a good minima during the EM learning, further work will include development of variational methodology learn the model parameters [**?**].