Convolutional Trees for Fast Transform Learning

Olivier Chabiron, Jean-Yves Tourneret and Herwig Wendt
IRIT/INP-ENSEEIHT, Université de Toulouse,
Toulouse, France.
Email: {firstname.lastname}@enseeiht.fr

François Malgouyres
Institut de Mathématiques de Toulouse,
IMT-CNRS UMR 5219, Université de Toulouse,
Toulouse, France.
Email: francois.malgouyres@math.univ-toulouse.fr

***Abstract*—Dictionary learning is a powerful approach for sparse representation. However, the numerical complexity of classical dictionary learning methods restricts their use to atoms with small supports such as patches. In a previous work, we introduced a model based on a composition of convolutions with sparse kernels to build large dictionary atoms with a low computational cost. The subject of this work is to consider this model at the next level, i.e., to build a full dictionary of atoms from convolutions of sparse kernels. Moreover, we further reduce the size of the representation space by organizing the convolution kernels used to build atoms into a tree structure. The performance of the method is tested for the construction of a curvelet dictionary with a known code.**

I. Introduction

The dictionary learning (DL) problem has received increasing attention since the pioneer works of Lewicki, Sejnowski and Olshausen [7], [8]. The principle behind DL is to find an adapted data representation promoting improved sparsity properties. The archetype of the DL strategy is to look for a dictionary as the solution of the following optimization problem

$$\text{argmin}_{\mathbf{D},(\mathbf{x}_i)_{1\le i\le I}} \quad \sum_{i=1}^{I} \|\mathbf{D}\mathbf{x}_i - \mathbf{y}_i\|_2^2 + g(\mathbf{x}_i)$$

where $\mathbf{y}_i$ constitute the learning database, $\mathbf{D}$ is the dictionary matrix (whose columns are the atoms), $(\mathbf{x}_i)_{1\le i\le I}$ are the codes and g is a sparsity-inducing function. Many methods including MOD [6] and K-SVD [1] have been proposed to solve this problem. DL is usually applied to small patches because of the computational cost induced by successive computations of matrix-vector products $\mathbf{D}\mathbf{x}_i$, which is at least $\mathcal{O}(N^2)$, where N is the sample size. Moreover, the cost of the dictionary update is usually at least $\mathcal{O}(N^3)$.[1]

We introduced in [4] a DL model which assumes that the learned atoms are compositions of convolutions with kernels whose supports each have a maximum of S elements. The interest of this S-sparse constraint was to design a fast algorithm to compute the forward transform $\mathbf{D}\mathbf{x}$ (and its transpose $\mathbf{D}^T\mathbf{y}$), allowing larger atoms living in a small search space to be considered. Despite its non-convexity, the associated optimization problem was shown to converge to a global optimum for a large class of initializations. Moreover, the complexity of the proposed algorithm is linear with respect to N.

The present work focuses on the dictionary update step associated with a convolutional tree structure. In this context, we assume that the code is known and our goal is to approximate a full dictionary $\mathbf{D}$ of atoms, supposed to be curvelets here without loss of generality.

II. Tree Model

Consider a rooted tree $\mathcal{T}_{\mathcal{N},\mathcal{E}}$ composed of nodes $n \in \mathcal{N}$ and edges $e \in \mathcal{E}$. Following [4], we assume that each edge $e \in \mathcal{E}$ from $\mathcal{T}_{\mathcal{N},\mathcal{E}}$ is the convolution with a kernel $h^e \in \mathbb{R}^N$. Nodes that do not have any children are called leaves and are denoted as $f \in \mathcal{L}$. Note that the number of atoms in the dictionary is equal to $N|\mathcal{L}|$, where $|\mathcal{L}|$ is the number of leaves in the tree. We propose an image formation model where $\mathbf{D}\mathbf{x}$ is the sum of convolutions between codes $(\mathbf{x}^f)_{f\in\mathcal{L}} \in \mathbb{R}^{N\times|\mathcal{L}|}$ and atoms $(\mathbf{H}^f)_{f\in\mathcal{L}} \in \mathbb{R}^{N\times|\mathcal{L}|}$ leading to

$$\mathbf{D}\mathbf{x} \quad = \quad \sum_{f\in\mathcal{L}} \mathbf{x}^f * \mathbf{H}^f$$

where for all $f \in \mathcal{L}$, each atom $\mathbf{H}^f$ is computed as

$$\mathbf{H}^f = *_{e\in\mathcal{C}(f)}\, h^e = \underbrace{h^r * \cdots * h^f}_{\text{from root } r \text{ to leaf} f}$$

where $\mathcal{C}(f)$ denotes the path going from the root of the tree to the leaf f. Kernels $(h^e)_{e\in\mathcal{E}}$ have restricted, known supports which contain only S elements (or pixels) in $\mathcal{S}^e \subset \{1,\ldots,N\}$.

From the above model, we build the following minimization problem

$$(FTL): \quad \underset{(h^e)_{e\in\mathcal{E}}\in\mathbb{R}^{\mathcal{P}\times|\mathcal{N}|}}{\text{argmin}} \quad \|\sum_{f\in\mathcal{L}} \mathbf{x}^f * (*_{e\in\mathcal{C}(f)}\, h^e) - \mathbf{y}\|_2^2$$

$$\text{subject to, } \forall e \in \mathcal{E}\ ,\ \ \text{supp}\,(h^e) \subset \mathcal{S}^e\ ,\ \ \|h^e\|_2 \le 1\ .$$

III. Resolution with the PALM Algorithm

The *proximal alternating linearized minimization* (PALM) algorithm [2] has been designed to solve non-convex optimization problems with block-variables. It can be viewed as alternating the steps of a proximal forward-backward scheme with respect to several block-variables. Since the (FTL) problem is non-convex and the dictionary is composed of many convolution kernels, the PALM algorithm seems very relevant to solve this problem. Moreover, according to [2], the PALM algorithm offers convergence guarantees to a critical point of the cost function to be optimized.

IV. Curvelet Dictionary Approximation

In order to illustrate the performance of the proposed method for fast transform learning, we build a tree based on a curvelet frequency tiling. This experiment synthesizes target atoms using a fast discrete curvelet transform [3], and solves the (FTL) problem to estimate these curvelet atoms.

Figure 1 shows the compositions of convolutions obtained along branches of the proposed tree whose leaves are the learned atoms, illustrating the dictionary structure considered in this paper. We observe a good recovery of the target curvelets, both for low and high frequency atoms.

V. Conclusion

This paper introduces a new structure of convolutional trees for dictionary design. The use of sparse convolution kernels allows larger images to be considered with a reduced computational cost. Future work will be devoted to include an additional sparse coding step in the proposed algorithm for dictionary learning.

[1]We invite the reader to consult [5] for more details about sparse representations and DL.

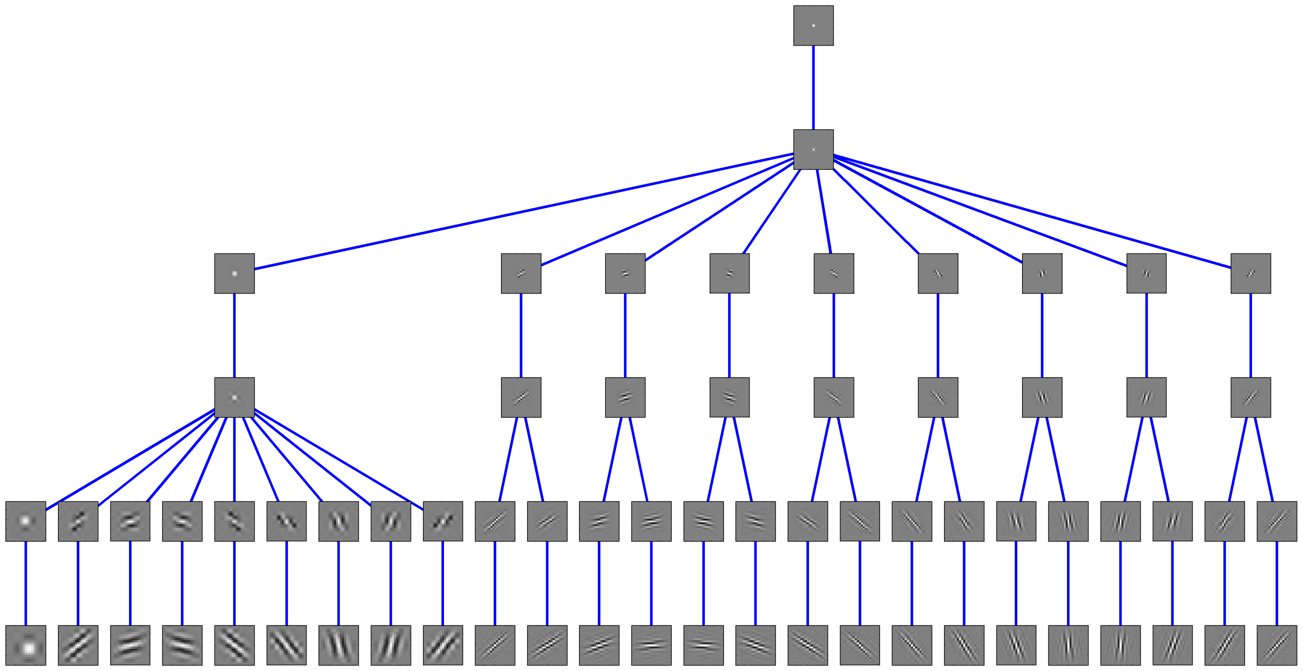

Fig. 1. Tree corresponding to a level 3 curvelet tiling. The sequence of convolutions obtained with the estimated kernels is shown at the nodes of the tree, starting from the root, down to the leaves representing the atoms.

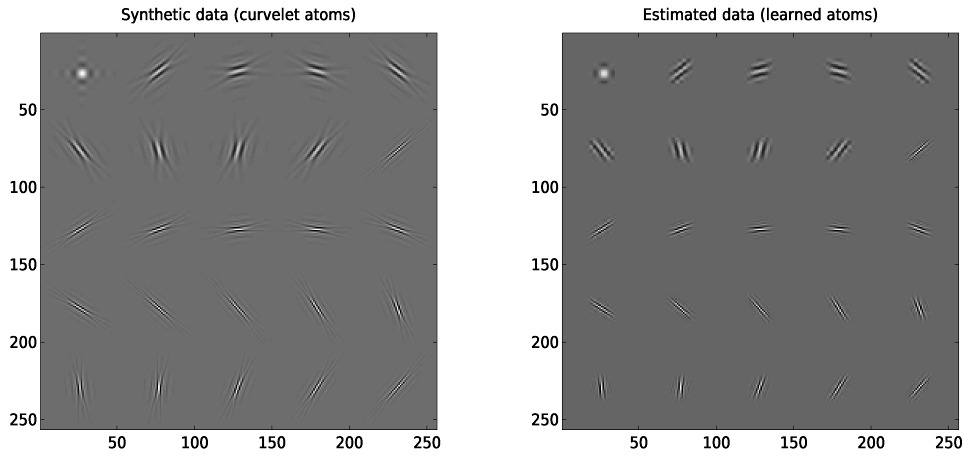

Fig. 2. Ground truth $\mathbf{y}$ (left) and estimated data (right).

References

[1] M. Aharon, M. Elad, and A. M. Bruckstein. The K-SVD, an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.*, 54(11):4311–4322, 2006.

[2] J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.

[3] E. Candes, L. Demanet, D. Donoho, and L. Ying. Fast discrete curvelet transforms. *Multiscale Modeling & Simulation*, 5(3):861–899, 2006.

[4] O. Chabiron, F. Malgouyres, J.-Y. Tourneret, and N. Dobigeon. Toward fast transform learning. *Int. J. Comput. Vision*, pages 1–22, 2014.

[5] M. Elad. *Sparse and redundant representations: From theory to applications in signal and image processing*. Springer, 2010.

[6] K. Engan, S. O. Aase, and J. Hakon Husoy. Method of optimal directions for frame design. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP)*, pages 2443–2446, Washington, DC, USA, 1999.

[7] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000.

[8] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311 – 3325, 1997.