Learning a fast transform with a dictionary
Olivier Chabiron, François Malgouyres, Jean-Yves Tourneret, Nicolas Dobigeon

To cite this version:
Olivier Chabiron, François Malgouyres, Jean-Yves Tourneret, Nicolas Dobigeon. Learning a fast transform with a dictionary. international Traveling Workshop on Interactions between Sparse models and Technology - iTWIST’ 2014, Aug 2014, Namur, Belgium. pp. 10-12. hal-01134819

HAL Id: hal-01134819
https://hal.archives-ouvertes.fr/hal-01134819
Submitted on 24 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: http://oatao.univ-toulouse.fr/
Eprints ID: 13244

To cite this version: Chabiron, Olivier and Malgouyres, François and Tourneret, Jean-Yves and Dobigeon, Nicolas Learning a fast transform with a dictionary. (2014) In: international Traveling Workshop on Interactions between Sparse models and Technology - iTWIST' 2014, 27 August 2014 - 29 August 2014 (Namur, Belgium).

Any correspondance concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr
Learning a fast transform with a dictionary

Olivier Chabiron1, François Malgouyres2, Jean-Yves Tourneret1 and Nicolas Dobigeon1.

1 Institut de Recherche en Informatique de Toulouse, IRIT-CNRS UMR 5505, ENSEEIHT, Toulouse, France.
2 Institut de Mathématiques de Toulouse, IMT-CNRS UMR 5219, Université de Toulouse, Toulouse, France.

Abstract—A powerful approach to sparse representation, dictionary learning consists in finding a redundant frame in which the representation of a particular class of images is sparse. In practice, all algorithms performing dictionary learning iteratively estimate the dictionary and a sparse representation of the images using this dictionary. However, the numerical complexity of dictionary learning restricts its use to atoms with a small support.

A way to alleviate these issues is introduced in this paper, consisting in dictionary atoms obtained by translating the composition of $K$ convolutions with $S$-sparse kernels of known support. The dictionary update step associated with this strategy is a non-convex optimization problem, which we study here.

A block-coordinate descent or Gauss-Seidel algorithm is proposed to solve this problem, whose search space is of dimension $K^S$, which is much smaller than the size of the image. Moreover, the complexity of the algorithm is linear with respect to the size of the image, allowing larger atoms to be learned (as opposed to small patches). An experiment is presented that shows the approximation of a large cosine atom with $S$ sparse atoms.

1 Introduction

The problem we introduce in this paper is motivated by the dictionary learning (DL) field. DL was pioneered by [8, 9] and has received a growing attention since then. The principle behind DL is to find a representation for data that makes it simpler, sparser. We invite the reader to consult [4] for more details about sparse representations and DL. The archetype of the DL strategy is to look for a dictionary as the solution of the following optimization problem

$$\text{argmin}_{\mathbf{x},(\mathbf{D}(i)_1 \leq S)} \sum_{i=1}^{f} \| \mathbf{D} \mathbf{x}_i - \mathbf{y}_i \|_2^2 + f(\mathbf{x}_i),$$

where $\mathbf{y}_i$ are the learning database, $\mathbf{D}$ is the dictionary matrix, whose columns are the atoms, and $f$ is a sparsity-inducing function. The resulting problem can be solved (or approximately solved) by many methods including MOD [5] and K-SVD [1]. All these approaches rely on alternatively updating the codes $\mathbf{x}_i$ and the dictionary $\mathbf{D}$.

Our primary motivation for considering the observation model (1) comes from computational issues. Usually, DL is applied to small patches, because of the computational cost of re-modeling. The dictionary update step associated with this strategy is a non-convex optimization problem, which we study here.

We propose a model where the learned atoms are a composition of $K$ convolutions with $S$-sparse kernels. The interest for such a constraint is to provide numerically effective dictionaries and allow to consider larger atoms. Indeed, the search space is only of dimension $K^S$, which is typically smaller than the size of the target atom.

The present work focuses on the dictionary update step of one atom. In this context, the code $\mathbf{x}$ is known. Our goals are both to approximate a large target atom $\kappa$ with our model and to obtain target atoms whose manipulation is numerically efficient. This translates into a non-convex optimization problem.

2 Problem formulation

Let consider an observed $d$-dimensional signal $\mathbf{y} \in (\mathbb{R}^N)^d$, assumed to result from the convolution of a known input signal $\mathbf{x} \in (\mathbb{R}^N)^d$ with an unknown target kernel $\kappa \in (\mathbb{R}^N)^d$, contaminated by an additive noise $\mathbf{b}$ following the linear model

$$\mathbf{y} = \kappa * \mathbf{x} + \mathbf{b},$$

where $*$ stands for the circular discrete convolution in dimension $d$. For instance, the unknown target kernel $\kappa \in (\mathbb{R}^N)^d$ may refer to the unknown impulse response of a 1D-filter or, conversely, to the point spread function of a 2D-filtering operator.

The problem addressed in this paper consists of approximating the unknown target kernel $\kappa$ by a composition of convolutions with $K \geq 2$ sparse kernels $((\mathbf{h}_k))_{1 \leq k \leq K} \in ((\mathbb{R}^N)^d)^K$:

$$\kappa \approx \hat{\kappa} \equiv \mathbf{h}_1 \ast \cdots \ast \mathbf{h}_K.$$

The kernels $\mathbf{h}_1, \ldots, \mathbf{h}_K$ are constrained to have less than a fixed number $S$ of non-zero elements, i.e., they are assumed to be at most $S$-sparse. As stated before, this assumption aims at providing a cost-effective dictionary by reducing the computations for $\mathbf{x} \ast \kappa$. Furthermore, the locations of their non-zero elements in $\{0, \ldots, N\}$ are assumed to be known or pre-set. More precisely, the support of the $k$th kernel (i.e., the locations of the non zero elements of $\mathbf{h}_k$), denoted $\text{supp}(\mathbf{h}_k)$, is constrained to a fixed set of discrete indexes $S_k$.

An example of indexes for 1D convolution kernel would be

$$S_k = \{k - 1, 2k - 1, \ldots, Sk - 1\}.$$

Assuming that the noise vector $\mathbf{b}$ is an independent and identically distributed Gaussian sequence, approximating the unknown convolution kernel $\kappa$ from the observed measurements $\mathbf{y}$ can be formulated as the following optimization problem

$$P_k : \begin{cases} \text{argmin}_{\mathbf{h} \in (\mathbb{R}^N)^d} \| \mathbf{y} - \mathbf{h}_1 \ast \cdots \ast \mathbf{h}_K \ast \mathbf{x} \|_2^2, \\ \text{subject to supp}(\mathbf{h}_k) \subset S_k, \forall k \in \{1, \ldots, K\} \end{cases}$$

where $\| \cdot \|_2$ stands for the usual Euclidean norm in $(\mathbb{R}^N)^d$. The problem $(P_k)$ is non convex. Thus, depending on the values of $K$, $(S_k)_{1 \leq k \leq K}$, $\mathbf{x}$ and $\mathbf{y}$, it might be difficult or impossible.

\footnote{All the elements of $(\mathbb{R}^N)^d$ are extended over $2^d$ by periodization.}
to find a good approximation of a global minimizer of \((P_0)\). Our objectives are to study if such a problem bends itself to
global optimization, and to assess the approximation power of the computed compositions of convolutions.

3 Block-coordinate descent

The problem \((P_0)\), as formulated in the previous section, is unhandy for global optimization. As detailed in [2], it has irrelevant stationary points and is non-convex (though infinitely differentiable). To adress these issues, a scalar weight \(\lambda\) is introduced and kernels are constrained to have a unit norm. Moreover, we elect a block-coordinate formulation in order to solve the problem with a Gauss-Seidel type algorithm (called Alternate Least Squares, sharing many similarities with the one used in [7]).

\[
(P_k) : \left\{ \begin{array}{l}
\text{argmin}_{\lambda \in \mathbb{R}, h \in \mathbb{R}^N} \| y - \lambda h \ast x^k \|_2^2, \\
\text{subject to supp}(h) \subset S_k \text{ and } \|h\|_2 = 1
\end{array} \right.
\]

with

\[
x^k = h^1 \ast \ldots \ast h^{k-1} \ast h^{k+1} \ast \ldots \ast h^K \ast x,
\]

where the kernels \(h^{k'}\) are fixed \(\forall k' \neq k\). The problem \((P_k)\) is linear and can be expressed as a matrix-vector product considering only the elements of \(h\) that belong to its support: The idea is to alternatively solve \((P_k)\) by iterating on \(k\). The support constraint significantly reduces the search space of the problem, and thus the amount of calculations needed to solve it for a stationary point. Algorithm 1 shows an overview of the resolution of \((P_k)\). The computational complexity associated with a passage in the while loop is \(O((K+S)KSN^2d)\), i.e., it is linear with respect to the size \(N^d\) of the signal. The detailed steps to solving \((P_k)\) are given in [2].

Algorithm 1: ALS algorithm

Input:
y: target measurements;
x: known coefficients;
\((S_k)1 \leq k \leq K\): supports of the kernels \((h^k)1 \leq k \leq K\).

Output:
\((h^k)1 \leq k \leq K\): convolution kernels such that
\[h^1 \ast \ldots \ast h^K \approx \kappa.\]

begin
Initialize the kernels \((h^k_p)1 \leq k \leq K\);
while not converged do
\[\text{for } k = 1, \ldots, K \text{ do}
\]
\[\text{Update } h^k \text{ and } \lambda \text{ with a minimizer of } (P_k) ;
\]
end

4 Synthetic example

In this section, we show an experiment consisting of approximating a 2D cosine atom \(\kappa\) in an image \(y\) of size \(64 \times 64\) (i.e., \(d = 2\) and \(N = 64\)). Such an atom can be seen as a large local cosine or a Fourier atom, both widely used in image processing. The interest of this atom is that it covers the whole image and is of a rather large support, making it difficult to handle with existing dictionary learning strategies.

\[\kappa_p = \cos \left(2\pi \frac{p (2, 5)}{N} \right), \forall p \in \{0, \ldots, 63\}^2.\]

The code \(x\) is a sparse image whose elements are chosen independent and identically distributed according to a Bernoulli-Gaussian distribution, widely used in sparse signal and image deconvolution [3, 6, 10]. Therefore, \(y\) contains a few weighted translations of the cosine atom \(\kappa^2\). The target \(y\) is built with additive Gaussian noise of variance \(\sigma^2 = 0.5\). Kernel supports have been set to a simple \(5 \times 5\) square, linearly dilated with \(k\), similar to the 1-D example given in (4).

Figures 1 and 2 show the cosine image \(y\), its approximation \(\lambda h^1 \ast \ldots \ast h^K\), the actual atom \(\kappa\) and \(\lambda h^1 \ast \ldots \ast h^K\), for \(K = 7\) and \(S = 25\). The results obtained here are quite accurate even though the cosine image was corrupted by additive noise.

5 Conclusion

This work shows that simple atoms can be accurately approximated with a composition of convolutions. The kernels used in the approximation are constrained to be sparse (i.e., with sparse supports), leading to a computationally efficient algorithm, despite the non-convexity of the function to optimize. This efficiency was illustrated on a 2D-cosine function, but similar experiments conducted with archetypal kernels (e.g., wavelets or curvelets) show similar performances [2].

The proposed modeling and algorithmic schemes open new perspectives on the general problem of dictionary learning. More specifically, it seems reasonable to derive a DL technique which recovers large structured dictionary whose atoms consist of compositions of convolutions.

Finally, how to choose, set or draw the kernel supports remains a large and yet unexplored issue, that may have significant impact on the method performances.
References


