Learning Unions of Orthonormal Bases with Thresholded Singular Value Decomposition
Sylvain Lesage, Rémi Gribonval, Frédéric Bimbot, Laurent Benaroya

To cite this version:

HAL Id: inria-00564483
https://inria.hal.science/inria-00564483
Submitted on 9 Feb 2011

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.
LEARNING UNIONS OF ORTHONORMAL BASES
WITH THRESHOLDED SINGULAR VALUE DECOMPOSITION

LESAGE Sylvain, GRIBONVAL Rémi, BIMBOT Frédéric, BENAROYA Laurent *

IRISA (CNRS & INRIA), campus de Beaulieu, 35042 RENNES cedex, FRANCE
sylvain.lesage@irisa.fr, remi.gribonval@irisa.fr, frederic.bimbot@irisa.fr

ABSTRACT
We propose a new method to learn overcomplete dictionaries for sparse coding structured as unions of orthonormal bases. The interest of such a structure is manifold. Indeed, it seems that many signals or images can be modeled as the superimposition of several layers with sparse decompositions in as many bases. Moreover, in such dictionaries, the efficient Block Coordinate Relaxation (BCR) algorithm can be used to compute sparse decompositions. We show that it is possible to design an iterative learning algorithm that produces a dictionary with the required structure. Each step is based on the coefficients estimation, using a variant of BCR, followed by the update of one chosen basis, using Singular Value Decomposition. We assess experimentally how well the learning algorithm recovers dictionaries that may or may not have the required structure, and to what extent the noise level is a disturbing factor.

1. INTRODUCTION
Sparse coding [1, 2] is a useful tool to analyze and try to explain the structure of series of observed data, such as successive time frames of an audio signal [3] or natural images [4]. Formally, assume that we observe $T$ vectors $x(t) = (x_n(t))_{n=1}^{N}$, $1 \leq t \leq T$ which are supposed to have been generated following the model:

$$x(t) = As(t) + \epsilon(t) \quad (1)$$

$A$ being an overcomplete dictionary (an $N \times K$ matrix), $s(t) \in \mathbb{R}^K$ some “sparse” coefficients and $\epsilon(t) \in \mathbb{R}^N$ a Gaussian noise. Sparse coding can be viewed as a way of estimating $A$ from the only observation of $X = AS + E$ where $X$ is the $N \times T$ matrix containing $T$ signal frames (similar notations being used for $S$ and $E$).

Jointly optimizing the coefficients and the dictionary, under constraints added to enforce the well-posedness of the problem, is a hard task, so we use an alternating optimization strategy:

1. **Coefficient update** given a dictionary $A$:

$$\arg \min_S \|X - AS\|_2^2 + \lambda \|S\|_1 \quad (2)$$

2. **Dictionary update** given coefficients $S$:

$$\arg \min_A \|X - AS\|_2^2 \quad (3)$$

under some constraint on $A$.

*The initial part of this work was done in collaboration with Laurent Benaroya while he was finishing his PhD with IRISA.

The coefficient update step (2) can be justified in a probabilistic framework using a Laplacian prior on the coefficients $s_k(t)$ [5]. Moreover, it is a simpler parent problem of the NP-hard combinatorial problem, where $\|S\|_1$ is replaced with $\|S\|_0^0$, the number of non-zero components in $S$. Computing the solution to Eq. (2) by Quadratic Programming is rather computationally intensive in the general case where $A$ has no special structure. However, when $A$ is a union of orthonormal bases (ONB), the Block Coordinate Relaxation (BCR) methods are efficient [6]. Another motivation to constrain the dictionary to be a union of ONB is that it seems that audio signals [7] and images [8] can indeed be modeled as the superimposition of several layers, each of which having sparse representation in its own adapted ONB. Note that when $A$ is constrained to have this precise structure, the dictionary update step (3) is also made relatively easy. This step, in a probabilistic framework, can be interpreted as a likelihood maximization and solved with an Expectation-Maximization (EM) algorithm [9].

In Section 2, we describe BCR and a variant which we used to solve (2). In Section 3, we introduce our algorithm to learn a union of bases by iteratively optimizing (3) with respect to each basis. In Section 4 we describe and analyze the experiments made with the learning algorithm on data generated following the model (1). We study the influence of the number $T$ of frames of the learning dataset, of the a priori knowledge on the noise level, and of the possible modeling error corresponding to the fact that the true $A$ might not be a union of bases or the number of bases could be wrong.

2. COMPUTATION OF SPARSE COEFFICIENTS
Finding sparse coefficients for the observed data $X$ is the result of a compromise between

- the minimization of the reconstruction error:

$$\|X - AS\|_2^2 := \sum_{n=1}^{N} \sum_{t=1}^{T} |x_n(t) - (As(t))_n|^2.$$  

- the minimization of a diversity measure, the most common ones being:

$$\|S\|_p := \sum_{k=1}^{K} \sum_{t=1}^{T} |s_k(t)|^p$$

for $0 \leq p \leq 1$. The strict diversity, defined by the number of non-zeros coefficients, is given by $\|S\|_0$.

This problem is generally difficult. It is indeed NP-hard with the $\|S\|_0$ diversity measure when $A$ is an arbitrary redundant dictionary. Many sub-optimal algorithms have been proposed such as
Matching Pursuit (MP) [10], Basis Pursuit (BP) [5] and FOCUSS [11]. The latter solves the minimization problem

\[ \min_{S} \| X - AS \|_2^2 + \lambda \| S \|_0^p \]  (4)

and Basis Pursuit solves it for \( p = 1 \) (see Eq. (2)). These algorithms are generally rather computationally intensive. However Basis Pursuit can be implemented more efficiently with a Block Coordinate Relaxation (BCR) method [6] when \( A \) is a union of ONB. Moreover, it has been shown that, under some conditions, the solution of (4) for any \( 0 \leq p \leq 1 \) is close to the solution given by Basis Pursuit [5].

In this section we recall how Basis Pursuit is implemented with soft-thresholding when \( A \) is a single ONB, then we remind the reader about BCR and eventually we describe a variant of BCR that we introduced to deal with low noise levels (small threshold parameter \( \lambda \)).

2.1. Case of an orthonormal basis

When \( A \) is an orthonormal basis, the solution of (2) is given by soft thresholding:

\[ \forall k, t \quad \hat{s}_k(t) = \begin{cases} a_k^T x(t) - \lambda / 2 & \text{if } a_k^T x(t) > \lambda / 2 \\ 0 & \text{if } |a_k^T x(t)| \leq \lambda / 2 \\ a_k^T x(t) + \lambda / 2 & \text{if } a_k^T x(t) < -\lambda / 2 \end{cases} \]  (5)

where \( a_k \) is the \( k^{th} \) column of \( A \) (also called atom of the dictionary).

2.2. Case of a union of orthonormal bases

When \( A = [A_1, \ldots, A_L] \) is a union of \( L \) orthonormal bases, the coefficients \( S \) are decomposed in \( L \) subsets of coefficients \( S_i \) corresponding to the \( L \) bases, as \( S = [S^T_1, \ldots, S^T_L]^T \). The BCR algorithm, described in Table 1 deals with the difficulty to directly solve (2) for a redundant dictionary \( A \) by successively solving it for its different bases \( A_i \). Then the sub-coefficients of an initial estimate \( S_{init} \) are iteratively updated until convergence is reached. The BCR algorithm has been proven to converge to a solution of (2), when, in Step 1, the selection of \( S_i \) follows a systematic cycle, or results from an optimal descent rule [6]. Unfortunately, if \( \lambda \) is very small (which corresponds to looking for a small reconstruction error, namely the low noise assumption), BCR might converge very slowly since almost no thresholding is performed in Step 3. In order to compute sparse coefficients in this low noise case, we propose to start BCR with a large initial threshold \( \lambda_0 \) and decrease it regularly, leading to the algorithm explained in Table 2. During the very first steps, since the threshold is high, sparsity is enforced; when the threshold becomes smaller, the error vanishes.

2.3. Experiments

Even though we have no proof of convergence of this modified BCR algorithm, we observed experimentally that if the two parameters \( N_{it} \) and \( \lambda_0 \) are appropriately chosen, it reaches a solution close to those provided at a higher computational cost by MP and FOCUSS.

Our experiments reported in [12] have shown that, for each \( N_{it} \), the initial threshold \( \lambda_0 \) greatly impacts the diversity of the estimated coefficients. An optimal value of \( \lambda_0 \) may be chosen \emph{a posteriori} to minimize (2). Note that the higher the number of

Table 1. Block Coordinate Relaxation algorithm

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Select a subset ( S_i ) of the current ( S ) to update;</td>
</tr>
<tr>
<td>2.</td>
<td>Compute ( X_i = X - \sum_{j \neq i} A_j S_j );</td>
</tr>
<tr>
<td>3.</td>
<td>Update ( S_i ) by replacing it by</td>
</tr>
<tr>
<td></td>
<td>( \arg \min_{S_i} | X_i - A_i S_i |_2^2 + \lambda | S_i |_1 ),</td>
</tr>
<tr>
<td></td>
<td>which is computed by soft thresholding (Eq. (5));</td>
</tr>
<tr>
<td>4.</td>
<td>If the stopping criterion is not reached, go to step 1.</td>
</tr>
</tbody>
</table>

Table 2. Modified BCR algorithm for the low noise case

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Choose an initial dictionary ( A = [A_1 \ldots A_L] );</td>
</tr>
<tr>
<td>2.</td>
<td>Update the coefficients ( S = [S_1 \ldots S_L] ) using the current ( A ) (see text);</td>
</tr>
<tr>
<td>3.</td>
<td>Choose which basis ( A_i ) to update and:</td>
</tr>
<tr>
<td></td>
<td>(a) Compute ( X_i = X - \sum_{j \neq i} A_j S_j );</td>
</tr>
<tr>
<td></td>
<td>(b) Compute the singular value decomposition:</td>
</tr>
<tr>
<td></td>
<td>( X_i S_i^T = UDV^T );</td>
</tr>
<tr>
<td></td>
<td>(c) Update</td>
</tr>
<tr>
<td></td>
<td>( A_i = UV^T );</td>
</tr>
<tr>
<td>4.</td>
<td>If the stopping criterion is not reached, go to step 2 (see text).</td>
</tr>
</tbody>
</table>

Table 3. Learning algorithm for \( L \) orthonormal bases

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Choose an initial dictionary ( A = [A_1 \ldots A_L] );</td>
</tr>
<tr>
<td>2.</td>
<td>Update the coefficients ( S = [S_1 \ldots S_L] ) using the current ( A ) (see text);</td>
</tr>
<tr>
<td>3.</td>
<td>Choose which basis ( A_i ) to update and:</td>
</tr>
<tr>
<td></td>
<td>(a) Compute ( X_i = X - \sum_{j \neq i} A_j S_j );</td>
</tr>
<tr>
<td></td>
<td>(b) Compute the singular value decomposition:</td>
</tr>
<tr>
<td></td>
<td>( X_i S_i^T = UDV^T );</td>
</tr>
<tr>
<td></td>
<td>(c) Update</td>
</tr>
<tr>
<td></td>
<td>( A_i = UV^T );</td>
</tr>
<tr>
<td>4.</td>
<td>If the stopping criterion is not reached, go to step 2 (see text).</td>
</tr>
</tbody>
</table>
3.1. Learning a single orthonormal basis

The optimization problem (3) with the constraint that $A$ is an ONB, can be written as the minimization of a Lagrangian,

$$\mathcal{L}(A, \mu) = \|X - AS\|_2^2 + \text{Tr}[\mu(A^T A - \text{Id})]$$

(6)

where $\mu$ is an $N \times N$ matrix of Lagrange multipliers. Setting the gradients $\nabla_\mu \mathcal{L}$ and $\nabla_A \mathcal{L}$ to zero yields:

$$A^T A - \text{Id} = 0$$

(7)

$$-2(X - AS)S^T + A(\mu + \mu^T) = 0$$

(8)

Let $Z := XS^T$ and $Y(\mu) := SS^T + (\mu + \mu^T)/2$. While $Z$ can be explicitly computed, $Y(\mu)$ is unknown since it depends on the unknown multipliers $\mu$. Solving the equations (7) and (8) is equivalent to finding $\mu$ such that $A^{-1} = Y(\mu)Z^{-1}$ is orthogonal, or in other words such that:

$$Y(\mu)Z^{-1}(Z^{-1})^T Y(\mu) = \text{Id}.$$  

(9)

Let $Z = UDV^T$ be the Singular Value Decomposition (SVD) of $Z$, that is to say $U$ and $V$ are orthogonal matrices and $D$ is a diagonal matrix. Condition (9) becomes 

$$Y(\mu)VD^{-1}(D^{-1}V^T Y(\mu)) = \text{Id},$$

i.e. the following matrix $W(\mu)$ should be orthogonal:

$$W(\mu) := Y(\mu)VD^{-1}$$

(10)

Noting that $Y(\mu) = W(\mu)DV^T$ must be symmetric, it can be shown [12] that the only solutions are of the form:

$$W(\mu) = V\Sigma$$

$\Sigma$ being a diagonal matrix with $\pm 1$ diagonal entries. Among the $2^N$ candidate solutions $A = U\Sigma V^T$, the one which minimizes $\mathcal{L}$ is $A = UV^T$, i.e. $\Sigma = \text{Id}$. Note that, even if the SVD of $Z$ is not unique, the product $UV^T$ is.

3.2. Learning a union of $L$ orthonormal bases

Ideally, when $A$ is constrained to be the union of $L$ orthonormal bases, one would like to perform the dictionary update step (3) by minimizing the Lagrangian:

$$\|X - AS\|_2^2 + \sum_{l=1}^{L} \text{tr} \left[\mu_l (A_l^T A_l - \text{Id})\right].$$

However, this optimization problem does not have an explicit solution as in the case of a single ONB. The principle behind the algorithm described in Table 3 is that, at each iteration, only one of the bases $A_l$ is optimized.

If we know which parameter $\lambda$ to use in Eq. (2) – for example, if we know the prior distributions of $e$ and $S$ in the probabilistic model (1) – then we perform Step 2 with the regular BCR algorithm. In many practical cases however, it is difficult to have an idea of a relevant value for $\lambda$. If Gaussian noise $\epsilon(t)$ is assumed, with unknown variance, we use the algorithm proposed by Azzalini et al. [13]: starting from an initial exact representation $X = AS_0$ and the estimate $S = 0$ we iterate the following steps:

1. compute the variance $\sigma^2$ of $S_0 - S$;
2. update $S$ by letting it contain all the entries of $S_0$ that are above the threshold $2 \log(N) \sigma^2$.

3. if the last update did not modify $S$, then stop, else go to 1.

To compute the initial exact decomposition $S_0$, one can use differently the variant of BCR, FOCUSS or MP preferably to $S_0 = A^T X$, because they encourage sparsity of the coefficients. If we try to use the above algorithm with a very small $\lambda$ – for example in the low noise limit – the above strategy for Step 2 needs to be modified since, similarly to what we explained for the variant of BCR, the dictionary update step would almost not change the dictionary. We refer the reader to report [12] for more details.

4. EXPERIMENTS

The following experiments have been designed with synthetic data generated with the model (1) using a known reference dictionary $A_{\text{ref}}$. The goal is to study the influence of various parameters on the performance, and to see how modeling error could also impact the results. We use two relevant and complementary performance measures: the false alarm rate and the missed detection rate, corresponding respectively to the relative number of estimated atoms $a_{est}$ (i.e. the number of columns of $A_{est}$, the dictionary estimated using the learning algorithm) that “do not match” any reference atom $a_{ref}$, and to the relative number of reference atoms that “are not matched” by any estimated atom. Since all atoms have unit norm, $a_{est}$ and $a_{ref}$ are considered to match if their inner product $|a_{est}^T a_{ref}|$ is close enough to one. So we use a parameter $\xi$ to decide that they match if and only if $|a_{est}^T a_{ref}| \geq \xi$. Different values of $\xi$ yield different but related performance measures.

In order to evaluate the performance on a wide range of conditions, each experiment is run with $N_{r}$ different dictionaries, and the performance measures are averaged over these runs.

4.1. Influence of the number of signal frames

First, we study the impact of the number $T$ of frames used to learn the dictionary. Data are generated with $A_{\text{ref}}$ a union of two random ONB in dimension $N = 32$, using the noiseless model (1) with $n(t)$ (of dimension $2N$) containing between 0 and 6, randomly located, non-zero coefficients, that follow a standard Gaussian law. For $T \leq 10N$, the dictionary estimation yields poor performance, the false alarm rate and the missed detection rate both reach 80%, for $\xi = 0.99$. On the contrary, for $T = 20N$, about eight out of ten atoms are correctly estimated, and for $T \geq 50N$, all atoms are retrieved (rates are 0% with $\xi = 0.99$). Noting that the computing time linearly increases with $T$, we set $T = 50N$ in the rest of the experiments.

4.2. Influence of the noise level

To understand the effect of the noise level on the performance, we repeat the above experiments with the same data to which we add noise at various signal to noise levels: $+\infty$ dB, 10 dB, 0 dB. At each noise level we run the three configurations of the learning algorithm designed respectively for known $\lambda$, small $\lambda$ and unknown $\lambda$. For the known $\lambda$ case, we chose the parameter $\lambda$ for which the coefficients computed with were the closer to the original coefficients.

Table 4.2 shows that the algorithms with known $\lambda$, and small $\lambda$ (without prior knowledge on $\lambda$), give almost similar results, the first one performing better when there is some noise, while the
Table 4. Missed Detection Rate depending on the noise level on data, and on the learning method, with \( \xi = 0.99 \) (average over sixty experiments)

<table>
<thead>
<tr>
<th>Learning algorithm</th>
<th>+(\infty) dB</th>
<th>+10 dB</th>
<th>+0 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \lambda_{\text{known}} )</td>
<td>7%</td>
<td>28%</td>
<td>59%</td>
</tr>
<tr>
<td>( \lambda_{\text{small}} )</td>
<td>6%</td>
<td>30%</td>
<td>63%</td>
</tr>
<tr>
<td>( \lambda_{\text{unknown}} )</td>
<td>42%</td>
<td>58%</td>
<td>86%</td>
</tr>
</tbody>
</table>

The algorithm seems to be efficient only when estimating at least as many bases as there are in the reference dictionary. A good strategy could therefore be to learn a lot of bases, and to estimate \textit{a posteriori} the number of interesting ones.

5. CONCLUSION

We have presented a new method for learning from a set of observed data vectors, a dictionary structured as a union of orthonormal bases, with the objective that the decomposition of the data on this dictionary would be sparse. We have demonstrated on synthetically generated data that this method is able to recover a relevant underlying dictionary provided that it knows \textit{a priori} the structure (i.e. the number of ONB in the dictionary). The approach seems to behave reasonably well even when the number of bases is overestimated. We are now considering several remaining practical problems, namely estimating the number of bases, studying how the algorithm scales when the dimension \( N \) of the data becomes large and extending experiments to real audio signals or images. Last but not least, we are looking for conditions where we can prove that the algorithm converges to the underlying dictionary.

6. REFERENCES