Learning Unions of Orthonormal Bases with Thresholded Singular Value Decomposition

Sylvain Lesage, Rémi Gribonval, Frédéric Bimbot, Laurent Benaroya

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 $\mathbf{x}(t) = (x_n(t))_{n=1}^N$, $1 \leq t \leq T$ which are supposed to have been generated following the model:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \boldsymbol{\epsilon}(t) \tag{1}$$

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

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 $\mathbf{A}$:

$$\arg\min_{\mathbf{S}} \|\mathbf{X} - \mathbf{A}\mathbf{S}\|_2^2 + \lambda\|\mathbf{S}\|_1 \tag{2}$$

2. **Dictionary update** given coefficients $\mathbf{S}$:

$$\arg\min_{\mathbf{A}} \|\mathbf{X} - \mathbf{A}\mathbf{S}\|_2^2 \tag{3}$$

under some constraint on $\mathbf{A}$.

⋆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 $\|\mathbf{S}\|_1$ is replaced with $\|\mathbf{S}\|_0^0$, the number of non-zero components in $\mathbf{S}$. Computing the solution to Eq. (2) by Quadratic Programming is rather computationally intensive in the general case where $\mathbf{A}$ has no special structure. However, when $\mathbf{A}$ 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 $\mathbf{A}$ 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 $\mathbf{A}$ 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 $\mathbf{X}$ is the result of a compromise between

- the minimization of the **reconstruction error**:

$$\|\mathbf{X} - \mathbf{A}\mathbf{S}\|_2^2 := \sum_{n=1}^{N}\sum_{t=1}^{T} |x_n(t) - (\mathbf{A}\mathbf{s}(t))_n|^2.$$

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

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

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

This problem is generally difficult. It is indeed NP-hard with the $\|\mathbf{S}\|_0^0$ diversity measure when $\mathbf{A}$ 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

$$\min_{\mathbf{S}} \|\mathbf{X} - \mathbf{AS}\|_2^2 + \lambda \|\mathbf{S}\|_p^p \tag{4}$$

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 $\mathbf{A}$ 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 $\mathbf{A}$ 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 λ).

2.1. Case of an orthonormal basis

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

$$\forall k, t \; \hat{s}_k(t) = \begin{cases} \mathbf{a}_k^T \mathbf{x}(t) - \lambda/2 & \text{if } \mathbf{a}_k^T \mathbf{x}(t) > \lambda/2 \\ 0 & \text{if } |\mathbf{a}_k^T \mathbf{x}(t)| \leq \lambda/2 \\ \mathbf{a}_k^T \mathbf{x}(t) + \lambda/2 & \text{if } \mathbf{a}_k^T \mathbf{x}(t) < -\lambda/2 \end{cases} \tag{5}$$

where $\mathbf{a}_k$ is the k^{th} column of $\mathbf{A}$ (also called *atom* of the dictionary).

2.2. Case of a union of orthonormal bases

When $\mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_L]$ is a union of L orthonormal bases, the coefficients $\mathbf{S}$ are decomposed in L subsets of coefficients $\mathbf{S}_l$ corresponding to the L bases, as $\mathbf{S} = [\mathbf{S}_1^T, \ldots, \mathbf{S}_L^T]^T$. The BCR algorithm, described in Table 1 deals with the difficulty to directly solve (2) for a redundant dictionary $\mathbf{A}$ by successively solving it for its different bases $\mathbf{A}_l$. Then the sub-coefficients of an initial estimate $\mathbf{S}_{init}$ 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 $\mathbf{S}_l$ follows a systematic cycle, or results from an optimal descent rule [6]. Unfortunately, if λ 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 λ_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 λ_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 λ_0 greatly impacts the diversity of the estimated coefficients. An optimal value of λ_0 may be choosen *a posteriori* to minimize (2). Note that the higher the number of iterations, the sparser the coefficients obtained using the optimal threshold.

For $N_{it} = 100$ and λ_0 in a fairly large range, the diversity of the coefficients obtained by the BCR variant is the same than for MP and FOCUSS. However, the BCR algorithm variant is computationally less costly, taking about 2 seconds instead of about 3 seconds for MP, and 6 seconds for FOCUSS.

1. Select a subset $\mathbf{S}_l$ of the current $\mathbf{S}$ to update;
2. Compute $\mathbf{X}_l = \mathbf{X} - \sum_{i \neq l} \mathbf{A}_i \mathbf{S}_i$;
3. Update $\mathbf{S}_l$ by replacing it by

$$\arg\min_{\mathbf{S}_l} \|\mathbf{X}_l - \mathbf{A}_l \mathbf{S}_l\|_2^2 + \lambda \|\mathbf{S}_l\|_1,$$

   which is computed by soft thresholding (Eq. (5));
4. If the stopping criterion is not reached, go to step 1.

Table 1. Block Coordinate Relaxation algorithm

for $it = 0$ to N_{it}
  use BCR with threshold $\lambda_0(1 - \frac{it}{N_{it}})$ to update $\mathbf{S}$
end

Table 2. Modified BCR algorithm for the low noise case

3. DICTIONARY LEARNING WITH SVD

The algorithm used to learn a union of L orthonormal bases (ONB) is described in Table 3. To understand the rationale behind the use of the SVD in Step 3, we will first analyze the optimization problem (3) when $\mathbf{A}$ is constrained to be a single ONB. Then, we will briefly explain how the algorithm for L bases is derived from the single basis one, and we will discuss in more details how the coefficient update (Step 2) is performed, depending whether we know which value of λ to use in (2) or we want to adapt it to the data. As for the the stopping criterion (4), we simply set *a priori* the number of learning steps. Studying how much the dictionary varies between two steps may help design a better criterion in the future.

1. Choose an initial dictionary $\mathbf{A} = [\mathbf{A}_1 \ldots \mathbf{A}_L]$;
2. Update the coefficients $\mathbf{S} = [\mathbf{S}_1 \ldots \mathbf{S}_L]$ using the current $\mathbf{A}$ (see text);
3. Choose which basis $\mathbf{A}_l$ to update and:
   (a) Compute $\mathbf{X}_l = \mathbf{X} - \sum_{i \neq l} \mathbf{A}_i \mathbf{S}_i$
   (b) Compute the singular value decomposition:

$$\mathbf{X}_l \mathbf{S}_l^T = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

   (c) Update

$$\mathbf{A}_l = \mathbf{U}\mathbf{V}^T$$

4. If the stopping criterion is not reached, go to step 2 (see text).

Table 3. Learning algorithm for L orthonormal bases

3.1. Learning a single orthonormal basis

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

$$\mathcal{L}(\mathbf{A}, \boldsymbol{\mu}) = \|\mathbf{X} - \mathbf{AS}\|_2^2 + \mathrm{Tr}\left[\boldsymbol{\mu}(\mathbf{A}^T\mathbf{A} - \mathbf{Id})\right] \quad (6)$$

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

$$\mathbf{A}^T\mathbf{A} - \mathbf{Id} = 0 \quad (7)$$

$$-2(\mathbf{X} - \mathbf{AS})\mathbf{S}^T + \mathbf{A}(\boldsymbol{\mu} + \boldsymbol{\mu}^T) = 0 \quad (8)$$

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

$$\mathbf{Y}(\boldsymbol{\mu})\mathbf{Z}^{-1}(\mathbf{Z}^{-1})^T\mathbf{Y}^T(\boldsymbol{\mu}) = \mathbf{Id}. \quad (9)$$

Let $\mathbf{Z} = \mathbf{UDV}^T$ be the Singular Value Decomposition (SVD) of $\mathbf{Z}$, that is to say $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\mathbf{D}$ is a diagonal matrix. Condition (9) becomes $(\mathbf{Y}(\boldsymbol{\mu})\mathbf{VD}^{-1})(\mathbf{D}^{-1}\mathbf{V}^T\mathbf{Y}^T(\boldsymbol{\mu})) = \mathbf{Id}$, i.e. the following matrix $\mathbf{W}(\boldsymbol{\mu})$ should be orthogonal:

$$\mathbf{W}(\boldsymbol{\mu}) := \mathbf{Y}(\boldsymbol{\mu})\mathbf{VD}^{-1} \quad (10)$$

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

$$\mathbf{W}(\boldsymbol{\mu}) = \mathbf{V}\boldsymbol{\Sigma}$$

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

3.2. Learning a union of L orthonormal bases

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

$$\|\mathbf{X} - \mathbf{AS}\|_2^2 + \sum_{l=1}^{L} \mathrm{Tr}\left[\boldsymbol{\mu}_l(\mathbf{A}_l^T\mathbf{A}_l - \mathbf{Id})\right].$$

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 $\mathbf{A}_l$ is optimized.

If we know which parameter λ to use in Eq. (2) – for example, if we know the prior distributions of ϵ and $\mathbf{s}$ 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 λ. If Gaussian noise $\boldsymbol{\epsilon}(t)$ is assumed, with unknown variance, we use the algorithm proposed by Azzalini *et al*, [13]: starting from an initial exact representation $\mathbf{X} = \mathbf{AS}_0$ and the estimate $\mathbf{S} = 0$ we iterate the following steps

1. compute the variance σ^2 of $\mathbf{S}_0 - \mathbf{S}$;
2. update $\mathbf{S}$ by letting it contain all the entries of $\mathbf{S}_0$ that are above the threshold $2\log(N)\sigma^2$.
3. if the last update did not modify $\mathbf{S}$, then stop, else go to 1.

To compute the initial exact decomposition $\mathbf{S}_0$, one can use indifferently the variant of BCR, FOCUSS or MP preferably to $\mathbf{S}_0 = \mathbf{A}^+\mathbf{X}$, because they encourage sparsity of the coefficients. If we try to use the above algorithm with a very small λ – 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 $\mathbf{A}_{ref}$. 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 $\mathbf{a}_{est}$ (i.e. the number of columns of $\mathbf{A}_{est}$, the dictionary estimated using the learning algorithm) that "do not match" any reference atom $\mathbf{a}_{ref}$, and to the relative number of reference atoms that "are not matched" by any estimated atom. Since all atoms have unit norm, $\mathbf{a}_{est}$ and $\mathbf{a}_{ref}$ are considered to match if their inner product $|\mathbf{a}_{est}^T\mathbf{a}_{ref}|$ is close enough to one. So we use a parameter ξ to decide that they match if and only if $|\mathbf{a}_{est}^T\mathbf{a}_{ref}| \geq \xi$. Different values of ξ 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 T of signal frames

First, we study the impact of the number T of frames used to learn the dictionary. Data are generated with $\mathbf{A}_{ref}$ a union of two random ONB in dimension $N = 32$, using the noiseless model (1) with $\mathbf{s}(t)$ (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 λ, small λ and unknown λ. For the known λ case, we chose the parameter λ for which the coefficients computed with were the closer to the original coefficients.

Table 4.2 shows that the algorithms with known λ, and small λ (without prior knowledge on λ), give almost similar results, the first one performing better when there is some noise, while the

Learning algorithm	$+\infty$ dB	$+10$ dB	$+0$ dB
λ_{known}	7%	28%	59%
λ_{small}	6%	30%	63%
$\lambda_{unknown}$	42%	58%	86%

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)

second one giving better estimation in the low-noise case. Unfortunately, the algorithm designed for unknown λ never finds as many atoms.

The three configurations of the algorithm are greatly dependent on the noise level, no one retrieving, among the $N_r = 60$ runs, more than 86% of the dictionary, even for a $+10$dB signal to noise ratio.

4.3. Influence of the model mismatch

We designed experiments to analyse the behavior of the learning algorithm when there is a mismatch between the number of ONB in the reference dictionary and in the estimated one.

We run the same algorithm to estimate a pair of ONB on three datasets generated as above with the noiseless model (1) with

- $\mathbf{A}_{ref,1}$ a single random ONB
- $\mathbf{A}_{ref,2}$ a union of two ONB
- $\mathbf{A}_{ref,3}$ a union of three random ONB.

Note that with $\mathbf{A}_{ref,3}$, if the three bases are sufficiently different one from another, one cannot expect to get less than 33% missed detection, because only $2N$ atoms are estimated while there are $3N$ reference atoms. With $\mathbf{A}_{ref,1}$, as soon as there is no more than 50% false alarm we are sure to have recovered the atoms of the reference dictionary, but they may be split in the two learned bases.

Reference dictionary	$\mathbf{A}_{ref,1}$	$\mathbf{A}_{ref,2}$	$\mathbf{A}_{ref,3}$
Average missed detection rate	0.5%	7%	99.5%
Average false alarm rate	44%	7%	99%

Table 5. missed detection rate and false alarm rate ($\xi = 0.99$) depending on the number of bases in $\mathbf{A}_{ref}$, when the estimated dictionary owes two bases (average over $N_r = 200$ runs)

The results of two hundred runs are summarized in Table 4.3. More precisely:

- $\mathbf{A}_{ref,1}$: almost all reference atoms are retrieved. In 55 cases out of 100, one of the estimated basis is exactly $\mathbf{A}_{est,1}$. In the 45 other cases out of 100, the retrieved atoms are shared between the two bases, 82% in the first basis, and 17% in the second, while 1% are not detected.
- $\mathbf{A}_{ref,2}$: the average performance values hide two distinct behaviours. In 92% of the experiments, the dictionary is perfectly estimated, while in 8% of the cases, learning totally failed, without any well estimated atom at all.
- $\mathbf{A}_{ref,3}$: never more than 15% of the atoms were retrieved, the average being only 0.5%.

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 *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 relevent underlying dictionary provided that it knows *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

[1] M.S. Lewicki and T.J. Sejnowski, "Learning overcomplete representations," *Neural computation*, vol. 12, no. 2, pp. 337–365, 2000.

[2] M.S. Lewicki and B. Olshausen, "A probabilistic framework for the adaptation and comparison of image codes," *Journal of the Optical Society of America*, 1999.

[3] S.A. Abdallah and M.D. Plumbley, "If edges are the independent components of natural images, what are the independent components of natural sounds?," in *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation*, december 2001, pp. 534–539.

[4] A.J. Bell and T.J. Sejnowski, "The 'independent components' of natural scenes are edge filters," *Vision research*, vol. 37, no. 23, pp. 3327–3338, 1997.

[5] S.S. Chen, D.L. Donoho, and M.A. Saunder, "Atomic decomposition by basis pursuit," *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1998.

[6] S. Sardy, A.G. Bruce, and P. Tseng, "Block coordinate relaxation methods for nonparametric signal denoising with wavelet dictionaries," *Journal of computational and graphical statistics*, vol. 9, pp. 361–379, 2000.

[7] S. Molla and B. Torresani, "Determining local transientness in audio signals," *IEEE signal processing letters*, vol. 11, no. 7, pp. 625–628, july 2004.

[8] J.L. Starck, M. Elad, and D.L. Donoho, "Image decomposition via the combination of sparse representations and a variational approach," *IEEE transactions on image processing*, february 2004.

[9] T.K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, pp. 47–60, november 1996.

[10] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[11] K. Kreutz-Delgado, J.F. Murray, B.D. Rao, K. Engan, T.W. Lee, and T.J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Computation*, vol. 15, pp. 349–396, 2003.

[12] S. Lesage, R. Gribonval, F. Bimbot, and L. Benaroya, "Learning unions of orthonormal bases with thresholded svd," Tech. Rep., IRISA, 2004.

[13] A. Azzalini, M. Farge, and K. Schneider, "A recursive algorithm for nonlinear wavelet thresholding : Application to signal and image processing," Tech. Rep. 41, Institut Pierre Simon Laplace, 2004.