AN EM-ALGORITHM APPROACH FOR THE DESIGN OF ORTHONORMAL BASES ADAPTED TO SPARSE REPRESENTATIONS

A. Drémeau and C. Herzet

INRIA Centre Rennes - Bretagne Atlantique, Campus universitaire de Beaulieu, 35000 Rennes, France

ABSTRACT

In this paper, we consider the problem of dictionary learning for sparse representations. Several algorithms dealing with this problem can be found in the literature. One of them, introduced by Sezer *et al.* in [1] optimizes a dictionary made up of the union of orthonormal bases. In this paper, we propose a probabilistic interpretation of Sezer's algorithm and suggest a novel optimization procedure based on the EM algorithm. Comparisons of the performance in terms of missed detection rate show a clear superiority of the proposed approach.

Index Terms— Sparse representations, dictionary learning, expectation-maximization algorithm.

1. INTRODUCTION

Sparse representations aim at describing a signal as the combination of a small number of atoms chosen from an overcomplete dictionary. This kind of decomposition has recently been shown to provide a nice solution in a variety of domains including compressed sensing, denoising, inpainting, etc.

Formally, the sparse representation problem can be formulated as follows. Let $\mathbf{D}\in\mathbb{R}^{N\times M}$ be a dictionary with $N\leq M$ and $\mathbf{y}\in\mathbb{R}^N$ an observed signal. We want to find the vector $\mathbf{x}\in\mathbb{R}^M$ such that:

$$\min_{\mathbf{x}} \|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2^2 \quad subject\ to \quad \|\mathbf{x}\|_0 \leq L, \tag{1}$$

where $\|\mathbf{x}\|_0$ denotes the l_0-norm, *i.e.,* the number of nonzero coefficients in $\mathbf{x}$ and L is a given constant. Note that problem (1) is also often expressed in its Lagrangian version:

$$\min_{\mathbf{x}} \|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_0, \tag{2}$$

where λ is a Lagrangian multiplier.

Closely related to the sparse representation problem (1)-(2) is the design of dictionaries adapted to "sparse" representations. Formally, the problem can be expressed as follows: given a training set $\{\mathbf{y}_j\}_{j=1}^K$, find the dictionary $\mathbf{D}^\star$ which leads to the best distortion-sparsity compromise, *i.e.,*

$$\mathbf{D}^\star = \arg\min_{\mathbf{D}} \left\{ \sum_j \min_{\mathbf{x}_j} \|\mathbf{y}_j-\mathbf{D}\mathbf{x}_j\|_2^2 + \lambda\|\mathbf{x}_j\|_0 \right\}. \tag{3}$$

Several algorithms available in the literature deal with this problem. One of the most successful is the K-SVD algorithm proposed by Aharon *et al.* in [2] which sequentially seeks the solution of (3) by using a SVD decomposition of a residual matrix. In another approach by Lesage *et al.* (see [3]) the authors suggested an algorithm to optimize a dictionary made up of the union of P orthonormal bases. In the same spirit as Lesage's work, Sezer *et al.* proposed more recently in [1] a two-step iterative algorithm for solving the same kind of problem: in a first step the training data are classified into P different subsets; then, each subset is used to optimize a particular basis. The optimization of dictionary made up of P orthonormal bases is motivated by results presented in [4]. More precisely, Mallat and Falzon established in [4] that at low bit rates and in the context of orthonormal transforms, the rate-distortion performance depends on the ability of the basis to provide a good approximation of the signal with few coefficients. This result suggests the optimization of bases adapted to different local characteristics, which is the purpose of Sezer's algorithm.

In this paper, we place the problem of learning a dictionary made up of P orthonormal bases into a probabilistic framework. In this context, we show that Sezer's algorithm can be interpreted as a maximum a posteriori (MAP) problem. We then propose an alternative approach for the optimization of the dictionary based on a different MAP criterion. We give a practical implementation of this criterion based on the well-known expectation-maximization (EM) algorithm.

2. DICTIONARY OPTIMIZATION BASED ON THE EM ALGORITHM

2.1. A probabilistic framework for the optimization of P orthonormal bases

Let $\{\mathbf{y}_j\}_{j=1}^K$ be a set of training signals for the optimization of an overcomplete dictionary $\mathbf{D}$. We suppose that $\mathbf{D}$ is made up of P orthonormal bases, *i.e.,*

$$\mathbf{D} \triangleq [\mathbf{D}_1,\ldots,\mathbf{D}_i,\ldots,\mathbf{D}_P], \qquad \mathbf{D}_i^T\mathbf{D}_i = \mathbf{I}_N, \tag{4}$$

where $\mathbf{I}_N$ is the N-dimensional identity matrix. We thus have $M = P\times N$. Let finally $\mathbf{x}_{ji}$ denote the vector made up of the components of $\mathbf{x}_j$ which correspond to basis $\mathbf{D}_i$, *i.e.,*

$$\mathbf{x}_j^T \triangleq [\mathbf{x}_{j1}^T,\ldots,\mathbf{x}_{ji}^T,\ldots,\mathbf{x}_{jP}^T]^T. \tag{5}$$

Based on these definitions, we consider the following model for $\mathbf{y}_j$:

$$p(\mathbf{y}_j|\mathbf{D}) = \int_{\mathbb{R}^M} \sum_{c_j=1}^{P} p(\mathbf{y}_j|\mathbf{x}_j,\mathbf{D},c_j)\,p(\mathbf{x}_j|c_j)\,p(c_j)\,d\mathbf{x}_j, \tag{6}$$

with

$$p(\mathbf{y}_j|\mathbf{x}_j,\mathbf{D},c_j=i) = \mathcal{N}(\mathbf{D}_i\mathbf{x}_{ji},\sigma^2\mathbf{I}_N) \tag{7}$$

where $\mathcal{N}(\mu,\Gamma)$ denotes a Gaussian distribution with mean μ and covariance Γ, and

$$p(\mathbf{x}_j|c_j=i) \propto \exp\{-\lambda\|\,\mathbf{x}_{ji}\|_0\}, \tag{8}$$

where $\lambda > 0$ and $\propto$ denotes equality up to a normalization factor [1]. This imposes sparsity on $\mathbf{x}_{ji}$.

The model (6)-(8) can be interpreted as follows: each $\mathbf{y}_j$ is assumed to be a noisy combination of vectors from *one* single basis; the choice of the basis is indexed by c_j. Sparsity is encouraged via prior (8) which penalizes $\mathbf{x}_{ji}$'s with many nonzero elements. $p(\mathbf{y}_j|\mathbf{D})$ can therefore be understood as a mixture of Gaussians $\mathcal{N}(\mathbf{D}_i\mathbf{x}_{ji}, \sigma^2\mathbf{I}_N)$ where each element is weighted by a factor depending on the sparsity of $\mathbf{x}_{ji}$ and the prior probability $p(c_j = i)$.

2.2. Sezer's algorithm revisited

In [1], Sezer *et al.* proposed an iterative algorithm for the "sparse" optimization of dictionary made up of a set of orthonormal bases. The algorithm iterates between two main steps (see Table 1): in a first step, each observation is assigned to a family $\mathcal{S}_i, \quad i \in \{1, \ldots, P\}$; in a second step, the training data associated to family $\mathcal{S}_i$ are used to optimize basis $\mathbf{D}_i$ under a sparsity-distortion criterion. Note that λ' (see Table 1) is a user-defined parameter which allows a tuning between sparsity and distortion.

In this section we show that Sezer's algorithm can be understood as a particular implementation of a MAP problem within the probabilistic framework exposed in section 2.1. Indeed, let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_j, \ldots, \mathbf{x}_K]$ be a matrix whose columns are sparse vectors $\mathbf{x}_j$'s and $\mathbf{c} = [c_1, \ldots, c_K]^T$ a vector made up of the concatenation of the c_j's. If we make the assumption

$$p(c_j) = \tfrac{1}{P}, \forall\, c_j, \forall\, j \text{ , and } \lambda' = 2\lambda\sigma^2, \tag{9}$$

then recursions (13)–(16) can be reformulated as follows:

$$\mathbf{c}^{(k)} = \arg\max_{\mathbf{c}} \sum_{j=1}^{K} \log p(\mathbf{y}_j, \mathbf{x}_j^{(k-1)}, \mathbf{D}^{(k-1)}), \tag{10}$$

$$(\mathbf{D}^{(k)}, \mathbf{X}^{(k)}) = \arg\max_{(\mathbf{D},\mathbf{X})} \sum_{j=1}^{K} \log p(\mathbf{y}_j, \mathbf{x}_j, \mathbf{D}, c_j^{(k)}). \tag{11}$$

The equivalence between (10)-(11) and (13)-(16) is straightforward by taking model (6)-(8) into account. The detailed derivations are however omitted here due to space limitation.

It is clear from (10)-(11) that Sezer's algorithm is equivalent to a coordinate-ascent implementation of the following MAP problem:

$$(\mathbf{D}^\star, \mathbf{X}^\star, \mathbf{c}^\star) = \arg\max_{(\mathbf{D},\mathbf{X},\mathbf{c})} \sum_{j=1}^{K} \log p(\mathbf{y}_j, \mathbf{x}_j, \mathbf{D}, c_j). \tag{12}$$

Interestingly, the MAP formulation of Sezer's algorithm gives a connection between the user parameter λ' and the physical parameters of the model λ, σ^2.

It is important to note that there is in general *no* guarantee of the convergence of Sezer's algorithm. Indeed, although (10)-(11) increase the goal function $\sum_{j=1}^{K} \log p(\mathbf{y}_j, \mathbf{x}_j, \mathbf{D}, c_j)$ at each iteration, the c_j's can only take on values in a finite set (*i.e.,* $c_j \in \{1, \ldots, P\}$). This prevents us from applying any general convergence results.

[1] Note that (8) is actually improper since the normalization factor is equal to ∞. This technical problem does however not lead to any particular issue in the rest of the paper.

0. Initialization

- Set $\mathbf{D}^{(0)} = \mathbf{D}_0$.
- $\forall i \in \{1, \ldots, P\}, \forall j \in \{1, \ldots, K\}$, set
$\mathbf{x}_{ji}^{(0)} = \arg\min_{\mathbf{x}_{ji}} \left\{ \|\mathbf{y}_j - \mathbf{D}_i^{(0)}\mathbf{x}_{ji}\|_2^2 + \lambda'\|\mathbf{x}_{ji}\|_0 \right\}$.

1. Classification

$\forall\, i \in \{1, \ldots, P\}$, compute

$$\mathcal{S}_i^{(k)} = \left\{ j \in \{1, \ldots, K\} \mid c_j^{(k)} = i \right\}, \tag{13}$$

where

$$c_j^{(k)} = \arg\min_{i \in \{1,\ldots,P\}} \left\{ \|\mathbf{y}_j - \mathbf{D}_i^{(k-1)}\mathbf{x}_{ji}^{(k-1)}\|_2^2 + \lambda'\|\mathbf{x}_{ji}^{(k-1)}\|_0 \right\}. \tag{14}$$

2. Basis update

$\forall i \in \{1, \ldots, P\}, \forall j \in \{1, \ldots, K\}$, update $\mathbf{D}_i$ and $\mathbf{x}_{ji}$ as follows:

$$\mathbf{D}_i^{(k)} = \arg\min_{\mathbf{D}_i} \left\{ \sum_{j \in \mathcal{S}_i^{(k)}} \min_{\mathbf{x}_{ji}} \{\|\mathbf{y}_j - \mathbf{D}_i\mathbf{x}_{ji}\|_2^2 + \lambda'\|\mathbf{x}_{ji}\|_0\} \right\}$$

$$subject\ to\ \ \mathbf{D}_i^T\mathbf{D}_i = \mathbf{I}_N, \tag{15}$$

$$\mathbf{x}_{ji}^{(k)} = \arg\min_{\mathbf{x}_{ji}} \left\{ \|\mathbf{y}_j - \mathbf{D}_i^{(k)}\mathbf{x}_{ji}\|_2^2 + \lambda'\|\mathbf{x}_{ji}\|_0 \right\}. \tag{16}$$

3. Convergence check

If convergence is reached, set $\mathbf{D}^\star = \mathbf{D}^{(k)}$; otherwise go to step 1.

Table 1. Sezer's algorithm

2.3. An EM-algorithm approach for dictionary optimization

In the last section, we emphasized that Sezer's algorithm can be interpreted as an iterative algorithm for solving a joint (over $\mathbf{D}, \mathbf{X}$ and $\mathbf{c}$) MAP estimation problem. This formulation suggests alternative approaches for the optimization of the dictionary. In this paper, we consider the following *marginalized* MAP estimation problem:

$$(\mathbf{D}^\star, \mathbf{X}^\star) = \arg\max_{(\mathbf{D},\mathbf{X})} \sum_{j=1}^{K} \log p(\mathbf{y}_j, \mathbf{x}_j, \mathbf{D}), \tag{17}$$

where

$$p(\mathbf{y}_j, \mathbf{x}_j, \mathbf{D}) = \sum_{c_j=1}^{P} p(\mathbf{y}_j, \mathbf{x}_j, \mathbf{D}, c_j). \tag{18}$$

Problem (17) has usually no easy analytical solution. Nevertheless, it can be solved efficiently by means of the expectation-maximization (EM) algorithm [5].

The EM algorithm operates in two steps. First a lower bound on $\sum_j \log p(\mathbf{y}_j, \mathbf{x}_j, \mathbf{D})$ is computed by taking the current value of the parameters of interest into account; this step is usually refered to as *expectation* step (E-step). Then, the value of the parameters is updated by maximizing the lower bound (M-step). In particular, as far as problem (17) is concerned, the E-step and M-step can be formalized as :

E-step:

$$\mathcal{Q}(\mathbf{D}, \mathbf{X}, \mathbf{D}^{(k)}, \mathbf{X}^{(k)}) = \sum_{j=1}^{K} \sum_{i=1}^{P} w_{ji}^{(k)} \log p(\mathbf{y}_j, \mathbf{x}_j, \mathbf{D}, c_j), \tag{19}$$

where $w_{ji}^{(k)} \triangleq p(c_j = i|\mathbf{y}_j, \mathbf{x}_j^{(k-1)}, \mathbf{D}^{(k-1)})$.

0. Initialization

- Set $\mathbf{D}^{(0)} = \mathbf{D}_0$.
- Set $\lambda' \triangleq 2\lambda\sigma^2$.
- $\forall i \in \{1,\dots,P\}, \forall j \in \{1,\dots,K\}$, set
$$\mathbf{x}_{ji}^{(0)} = \arg\min_{\mathbf{x}_{ji}} \left\{ \|\mathbf{y}_j - \mathbf{D}_i^{(0)}\mathbf{x}_{ji}\|_2^2 + \lambda'\|\mathbf{x}_{ji}\|_0 \right\}.$$

1. E-step

$\forall i \in \{1,\dots,P\}, \forall j \in \{1,\dots,K\}$, compute

$$w_{ji}^{(k)} \propto \exp(-\frac{1}{2\sigma^2}\|\mathbf{y}_j - \mathbf{D}_i^{(k-1)}\mathbf{x}_{ji}^{(k-1)}\|_2^2 - \lambda\|\mathbf{x}_{ji}^{(k-1)}\|_0)\, p(c_j). \quad (22)$$

2. M-step

$\forall i \in \{1,\dots,P\}, \forall j \in \{1,\dots,K\}$, update $\mathbf{D}_i$ and $\mathbf{x}_{ji}$ as follows:

$$\mathbf{D}_i^{(k)} = \arg\min_{\mathbf{D}_i} \left\{ \sum_{j=1}^{K} w_{ji}^{(k)} \min_{\mathbf{x}_{ji}} \{\|\mathbf{y}_j - \mathbf{D}_i\mathbf{x}_{ji}\|_2^2 + \lambda'\|\mathbf{x}_{ji}\|_0\} \right\}$$
$$subject\ to\ \ \mathbf{D}_i^T\mathbf{D}_i = \mathbf{I}_N, \quad (23)$$

$$\mathbf{x}_{ji}^{(k)} = \arg\min_{\mathbf{x}_{ji}} \left\{ \|\mathbf{y}_j - \mathbf{D}_i^{(k)}\mathbf{x}_{ji}\|_2^2 + \lambda'\|\mathbf{x}_{ji}\|_0 \right\}. \quad (24)$$

3. Convergence check

If convergence is reached, set $\mathbf{D}^\star = \mathbf{D}^{(k)}$; otherwise go to step 1.

Table 2. EM-based learning algorithm

M-step:

$$(\mathbf{D}^{(k+1)}, \mathbf{X}^{(k+1)}) = \arg\max_{(\mathbf{D},\mathbf{X})} \mathcal{Q}(\mathbf{D}, \mathbf{X}, \mathbf{D}^{(k)}, \mathbf{X}^{(k)}). \quad (20)$$

The E-step (19) and M-step (20) equations are particularized to model (6)-(8) in Table 2. Note that the EM algorithm is always ensured to converge (see [6]). This is a key advantage with regard to Sezer's algorithm. The fixed points of the EM algorithm are either saddle points or maxima of $\sum_j \log p(\mathbf{y}_j, \mathbf{x}_j, \mathbf{D})$.

It is quite interesting to compare the operations performed by the proposed algorithm and Sezer's. In particular, the E-step (22) can be regarded as a "soft" version of the classification performed by Sezer's algorithm. Indeed, whereas a hard decision $c_j^{(k)}$ is made about the value of c_j in (14), the EM algorithm rather computes an a posteriori probability of c_j. It is easy to see that

$$c_j^{(k)} = \arg\max_{c_j} p(c_j = i|\mathbf{y}_j, \mathbf{x}_j^{(k-1)}, \mathbf{D}^{(k-1)}). \quad (21)$$

Sezer's algorithm can therefore be interpreted as a thresholded version of the EM algorithm. The M-step is quite similar to the basis update of Sezer's algorithm. In practice, the main difference between both algorithms relies on the fact that in Sezer's algorithm, each basis $\mathbf{D}_i$ is optimized using only the vectors contained in the subset $\mathcal{S}_i$, while in the proposed algorithm, the entire training set $\{\mathbf{y}_j\}_{j=1}^K$ is used by weighting each contribution of the $\mathbf{y}_j$'s by w_{ji} (*i.e.,* the probability of choosing basis $\mathbf{D}_i$ given $\mathbf{x}_{ji}$).

2.4. Algorithm implementation

In this section we discuss the practical implementation of the proposed algorithm.

As emphasized in the last section, the E-step can be seen as an extension of the classification step in Sezer's algorithm. Basically, both algorithms perform similar computations (see Tables 1 and 2). The complexity of these steps is thus of the same order.

We implement the M-step by an iterative conditional method which successively optimizes $\mathbf{X}$ and $\mathbf{D}$:

$$\mathbf{x}_{ji}^{(l)} = \arg\min_{\mathbf{x}_{ji}} \{\|\mathbf{y} - \mathbf{D}_i^{(l-1)}\mathbf{x}_{ji}\|_2^2 + \lambda\|\mathbf{x}_{ji}\|_0\}, \quad (25)$$

$$\mathbf{D}_i^{(l)} = \arg\min_{\mathbf{D}_i} \{\sum_{j=1}^{K} w_{ji}^{(k)}\|\mathbf{y}_j - \mathbf{D}_i\mathbf{x}_{ji}^{(l)}\|_2^2\}$$
$$subject\ to\ \ \mathbf{D}_i^{(l)T}\mathbf{D}_i^{(l)} = \mathbf{I}_N. \quad (26)$$

where k is the EM-algorithm iteration number and l the iteration number in the maximization step.

Problem (25) can be solved by greedy algorithms like Matching Pursuit ([7]) or relaxation algorithms like Basis Pursuit ([8]). In our case, $\mathbf{D}_i$ is orthonormal and the exact solution can be obtained by a simple thresholding operation, see [1].

With a development similar to the one in [1] and first proposed in [3], we can show that minimization (26) is achieved by computing:

$$\forall\, i \in \{1,\dots,P\},\ \ \mathbf{D}_i^{(l)} = \mathbf{V}\mathbf{U}^T, \quad (27)$$

where $\mathbf{U}\Delta^{1/2}\mathbf{V}^T$ is the singular value decomposition (SVD) of $\sum_{j=1}^K w_{ji}^{(k)}\mathbf{x}_{ji}^{(l)}\mathbf{y}_j^T$. The complexity of the proposed EM approach and Sezer's algorithm is thus similar.

2.5. Estimation of the noise variance

A key advantage of the probabilistic formulation introduced in this paper is that it offers a general framework for the estimation of the model parameters. In this section, we focus on the estimation of the noise variance σ^2. The estimation of this parameter can be made by including σ^2 as a new unknown variable in the MAP problem (17), *i.e.,*

$$(\mathbf{D}^\star, \mathbf{X}^\star, (\sigma^2)^\star) = \arg\max_{(\mathbf{D},\mathbf{X},\sigma^2)} \sum_{j=1}^{K} \log p(\mathbf{y}_j, \mathbf{x}_j, \mathbf{D}). \quad (28)$$

The equations of the EM algorithm are adapted to this new problem by adding the following update in the M-step:

$$(\sigma^2)^{(k)} = \frac{1}{NK}\sum_{j=1}^{K}\sum_{i=1}^{P} w_{ji}^{(k)}\|\mathbf{y}_j - \mathbf{D}_i^{(k)}\mathbf{x}_{ji}^{(k)}\|_2^2. \quad (29)$$

In the next section, we will see that the estimation of the noise variance is crucial for the convergence of the algorithms (whether the actual noise variance is known or not).

3. SYNTHETIC EXPERIMENTS

In this section, we evaluate and compare the performance of three algorithms:

- "Sezer": learning algorithm proposed in [1] and defined in Table 1.
- "EM": algorithm defined in Table 2 where the noise variance estimation (29) is also implemented.
- "EM thresholded": similar to "EM" where the E-step is approximated by a thresholded decision (21).

Note that "Sezer" and "EM thresholded" are similar but distinct since the latter implements a noise variance estimation which is not present in the deterministic formulation of [1].

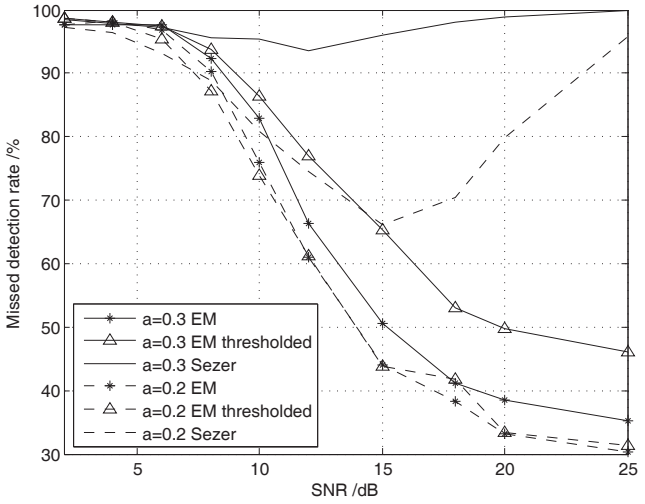

Fig. 1. Comparison between Sezer's, EM and EM-thresholded algorithms for different dictionary initializations (dashed line $a=0.2$, full line $a=0.3$).

3.1. Generation of the training data

We use synthetic signals to test whether the algorithms recover the original dictionary that generated the data. 200 training signals $\mathbf{y}_j$ are generated according to model (6)-(8). We consider a dictionary made up of six 8×8 random orthonormal matrices $\mathbf{D}=[\mathbf{D}_1,\ldots,\mathbf{D}_6]$ generated with a uniform law. Each basis is selected with probability $p(c_j)=1/6$. Vectors $\mathbf{x}_j$'s contain $L=2$ nonzero coefficients at random locations. The amplitude of the nonzero coefficients are drawn from a zero-mean Gaussian distribution with variance $\sigma_a^2=16$.

3.2. Initialization of the algorithms

The dictionary was initialized from the original dictionary as follows:

$$\forall i \in \{1,\ldots,P\} \quad \mathbf{D}_i^{(0)} = \mathbf{D}_i\mathbf{M}^T \tag{30}$$

where $\mathbf{M}=GS(\mathbf{I}_8+N(a))$, GS represents the Gram-Schmidt orthogonalization process and $N(a)$ represents a 8×8-matrix whose elements are *i.i.d.* realizations of a uniform law on $[-a,a]$. This formulation allows for controlling the deviation of $\mathbf{D}^{(0)}$ to $\mathbf{D}$. We initialize the $\mathbf{x}_{ji}^{(0)}$'s by solving problem (25) with $\mathbf{D}^{(0)}$. The noise variance is initialized as $(\sigma^2)^{(0)}=\sigma_x^2$ where $\sigma_x^2 \triangleq (L/N)\sigma_a^2$.

3.3. Performance evaluation

The performance of the algorithms is evaluated via the missed-detection rate (MDR) corresponding to the relative number of original atoms that "are not matched" by any estimated atom. Since all the atoms have unit norm, two atoms $\mathbf{d}_1$ and $\mathbf{d}_2$ are considered to match if and only if $|\mathbf{d}_1^T\mathbf{d}_2|\geq\xi$, where ξ is fixed to 0.99. The MDR is evaluated versus the signal-to-noise ratio (SNR) which is defined as $SNR \triangleq 10\log(\sigma_x^2/\sigma^2)$.

All three algorithms are initialized in the same way and applied on the same data set. The algorithms are run for 50 iterations. The M-step is implemented by iterating 10 times between (25) and (26).

Figure 1 represents the MDR achieved by the different algorithms for $a=0.2$ and $a=0.3$. We can notice that the proposed probabilistic approach leads to a clear improvement of the performance. When $a=0.2$, Sezer's algorithm is slightly better than the EM and EM-thresholded algorithms for low SNR's. For dictionary initializations close to the original dictionary, using the "real" noise covariance is more advantageous than estimating it. However, for higher SNR's, Sezer's algorithm leads to very poor performance due to its classification step: with a small noise covariance, the sparsity constraint is relaxed and increases the potential classification errors. When the initialization becomes coarser ($a=0.3$), EM and EM-thresholded approaches lead to a clear improvement of the MDR. The EM performance is slightly better than the EM-thresholded one.

4. CONCLUSION

In this paper, we address the problem of learning a dictionary made up of P orthonormal bases. This problem is placed in a probabilistic framework by considering the training data as realizations of a mixture of Gaussians. The learning task is then reformulated as a MAP estimation problem and an EM-algorithm procedure is derived to solve it. The proposed algorithm is shown to give enhanced performance with regard to a previously-proposed algorithm.

5. REFERENCES

[1] O. G. Sezer, O. Harmanci, and O. G. Guleryuz, "Sparse orthonormal transforms for image compression," in *Proc. IEEE Int'l Conference on Image Processing (ICIP), San Diego, CA.*, October 2008.

[2] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, November 2006.

[3] S. Lesage, R. Gribonval, F. Bimbot, and L. Benaroya, "Learning unions of orthonormal bases with thresholded singular value decomposition," in *Proc. IEEE Int'l Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 18-23 March 2005, vol. 5, pp. v293–v296.

[4] S. Mallat and F. Falzon, "Analysis of low bit rate image transform coding," *IEEE Trans. On Signal Processing*, vol. 46, no. 4, pp. 1027–1042, April 1998.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, pp. 1–38, 1977.

[6] C. F. J. Wu, "On the convergence properties of the em algorithm," *Ann. Statistics*, vol. 11, no. 1, pp. 95–103, 1983.

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

[8] S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comp*, vol. 20, no. 1, pp. 33–61, 1999.