A FIRST STEP TO CONVOLUTIVE SPARSE REPRESENTATION

H. Firouzi, M. Babaie-Zadeh, A. Ghasemian Sahebi*

Sharif University of Technology
Department Of Electrical Engineering
Tehran, Iran

C. Jutten

Institut National Polytechnique
de Grenoble (INPG), Labratoire
des Images et de Signaux (LIS)
Grenoble, France

ABSTRACT

In this paper an extension of the sparse decomposition problem is considered and an algorithm for solving it is presented. In this extension, it is known that one of the shifted versions of a signal $\mathbf{s}$ (not necessarily the original signal itself) has a sparse representation on an overcomplete dictionary, and we are looking for the sparsest representation among the representations of all the shifted versions of $\mathbf{s}$. Then, the proposed algorithm finds simultaneously the amount of the required shift, and the sparse representation. Experimental results emphasize on the performance of our algorithm.

Index Terms— atomic decomposition, sparse decomposition, sparse representation, overcomplete signal representation, sparse source separation

1. INTRODUCTION

In the classical atomic decomposition problem [1], we have a signal $s(t)$ whose samples are collected in the $n \times 1$ signal vector $\mathbf{s} = [s(1), \dots, s(n)]^T$ and we would like to represent it as a linear combination of m, $n \times 1$ signal vectors $\{\boldsymbol{\varphi}_i\}_{i=1}^m$. After [2], the vectors $\boldsymbol{\varphi}_i$, $1 \le i \le m$ are called *atoms* and they collectively form a *dictionary* over which the signal is to be decomposed. We may write

$$\mathbf{s} = \sum_{i=1}^{m} \alpha_i \boldsymbol{\varphi}_i = \boldsymbol{\Phi}\,\boldsymbol{\alpha}, \tag{1}$$

where $\boldsymbol{\Phi} \triangleq [\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_m]$ is the $n \times m$ dictionary (matrix) and $\boldsymbol{\alpha} \triangleq (\alpha_1, \dots, \alpha_m)^T$ is the $m \times 1$ vector of coefficients. A dictionary with $m > n$ is called *overcomplete*. Although, $m = n$ is sufficient to obtain such a decomposition (like what is done in Discrete Fourier Transform), using overcomplete dictionaries has a lot of advantages in many diverse applications (refer for example to [3] and the references in it). Note that for the overcomplete case, the representation is not unique, but all these applications need a sparse representation, that is, the signal $\mathbf{s}$ should be represented as a linear combination of as small as possible number of atoms of the dictionary. It has been shown [4, 5] that with some mild conditions on the dictionary matrix, if there is a sparse representation with at most $n/2$ non-zero coefficients, then this representation is unique. The main approaches for finding this sparse solution include Matching Pursuit (MP) [2, 6], FOCUSS [5], Basis Pursuit (BP) [1], and Smoothed ℓ^0 (SL0) [7].

In this paper, we introduce a generalization of this classical problem to the case that we call 'convolutive sparse representation'. In this case, it is known that the signal $\mathbf{s}$ has a sparse representation not over the dictionary itself, but over some (unknown) shifted versions of the atoms. To state the problem more clearly, consider a representation of the form:

$$\mathbf{s} = \sum_{i=1}^{m} \alpha_i {\boldsymbol{\varphi}_i}^{(k_i)}, \tag{2}$$

where ${\boldsymbol{\varphi}_i}^{(k_i)}$ stands for the k_i-sample (circularly) shifted version of $\boldsymbol{\varphi}_i$. Then, our problem is to find the sparsest representation in the form of (2) among all the possible values for $k_1, \dots, k_m$.

Note also that the Fourier transform does not convert this problem to the classical sparse representation (1) in the frequency domain: The problem in the transformed domain will be similar to (1), but with *time varying* α_i*'s*.

In this paper, we address only a special case of the general problem (2), that is, where all the shifts k_i are equal. This is equivalent to this simplified problem: an unknown shifted version of $\mathbf{s}$ has a sparse representation over the dictionary, and we would like to find this representation.

One of the trivial applications of the general problem is to reduce the size of the dictionary in atomic decomposition applications. An example for the applications of the above simplified problem is where our recorded signal, which has to be decomposed as a combination of a small number of atoms of the dictionary, is shifted relative to its underlying atoms that already exist in the dictionary.

The paper organized as follows. In Section 2 the main idea of the algorithm is introduced. The resulting algorithm

* This work has been partially funded by Sharif University of Technology, ISMO and Iran NSF (INSF).

is then stated in Section 3. Finally, simulation results of the algorithm are presented in Section 4.

2. MAIN IDEA

Consider a dictionary with atoms $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \cdots, \boldsymbol{\varphi}_m$. The problem is then to sparsely decompose an $n \times 1$ vector $\mathbf{s}$ as a linear combination of shifted atoms of the dictionary (in this paper, the shifts are assumed to be circular). One trivial solution to the problem is to insert all shifted atoms in the dictionary and then find the sparsest representation of the vector $\mathbf{s}$ for that dictionary using the conventional atomic decomposition methods. However, this direct solution demands a high computational and storage load.

Let also that k_i be a continuous variable (a non-integer shift k_i can be imagined as shifting the hull of signal and then re-sampling it). For handling circular shifts more easily, we take the Discrete Fourier Transform (DFT) of both sides of (2) to obtain:

$$\mathbf{s}^{(\mathcal{F})} = \sum_{i=1}^{m} \alpha_i \mathbf{W}_i \boldsymbol{\varphi}_i^{(\mathcal{F})} \tag{3}$$

in which $\mathbf{s}^{(\mathcal{F})}$ and $\boldsymbol{\varphi}_i^{(\mathcal{F})}$ are the DFTs of the signals $\mathbf{s}$ and $\boldsymbol{\varphi}_i$ repectively, and

$$\mathbf{W}_i \triangleq \begin{pmatrix} {w_i}^0 & 0 & 0 & 0 & 0 \\ 0 & {w_i}^1 & 0 & 0 & 0 \\ 0 & 0 & {w_i}^2 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & {w_i}^{n-1} \end{pmatrix}, w_i \triangleq e^{-j\frac{2\pi}{n}k_i}$$

As stated in the introduction, in this paper we consider only the case in which $w_1 = w_2 = \cdots = w_m$, and we present an iterative algorithm to solve the problem in this case. This case is equivalent to assuming that the atoms of the dictionary are fixed and the signal $\mathbf{s}$ is shifted in opposite direction. In this case:

$$\mathbf{W}_1 = \mathbf{W}_2 = \cdots = \mathbf{W}_m = \mathbf{W},$$

and hence from (3) we have:

$$\mathbf{s}^{(\mathcal{F})} = \mathbf{W} \sum_{i=1}^{m} \alpha_i \boldsymbol{\varphi}_i^{(\mathcal{F})} \tag{4}$$

or:

$$\mathbf{W}'\mathbf{s}^{(\mathcal{F})} = \sum_{i=1}^{m} \alpha_i \boldsymbol{\varphi}_i^{(\mathcal{F})} = \mathbf{\Phi}^{(\mathcal{F})}\boldsymbol{\alpha} \tag{5}$$

where:

$$\mathbf{W}' = \mathbf{W}^{-1} = \begin{pmatrix} {w'}^0 & 0 & 0 & 0 & 0 \\ 0 & {w'}^1 & 0 & 0 & 0 \\ 0 & 0 & {w'}^2 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & {w'}^{n-1} \end{pmatrix}$$

in which $w' = e^{j\frac{2\pi}{n}k}$. Now the problem is to find the sparsest solution of (5). To do so, we should have some criterion $F(\boldsymbol{\alpha})$ for sparseness of the solution vector $\boldsymbol{\alpha}$ and optimize that criterion subject to the constraint (5) using optimization methods. Note also that one of the unknows, k does not exist in the objective function $F(\boldsymbol{\alpha})$, and appears only in the constraint (5). As their objective functions, two classical sparse decomposition approaches use ℓ^1-norm [1], and smoothed ℓ^0 (SL0) norm [7]. Here we use the second one, because it results in a very fast and accurate algorithm in classical atomic decomposition [7], and also because it is a differentiable measure of the sparsity of $\boldsymbol{\alpha}$. Smoothed ℓ^0-norm of a vector is an approximation to its ℓ^0-norm (number of its non-zero element), and is defined as:

$$F(\boldsymbol{\alpha}) = m - \sum_{i=1}^{m} e^{-{\alpha_i}^2/2\sigma^2} \tag{6}$$

where σ is a parameter which specifies a tradeoff between smoothness and the accuracy of approximation: the smaller σ, the better approximation of the ℓ^0 norm; the larger σ, the smoother objective function.

On the other hand, (5) can be written as:

$$G(\boldsymbol{\alpha}, w') = \|\mathbf{W}'\mathbf{s}^{(\mathcal{F})} - \mathbf{\Phi}^{(\mathcal{F})}\boldsymbol{\alpha}\|^2 = 0 \tag{7}$$

Now we should minimize (6) subject to the (7), for a small value of σ. Note that one of the optimization variables (w'), is not present in $F(\boldsymbol{\alpha})$, and appears only in (7).

Note that for small values of σ, F contains a lot of local minima. Consequently, it is very difficult to directly minimize this function for very small values of σ. The idea of [7] for escaping from local minima is then to decrease the value of σ gradually: for each value of σ the minimization algorithm is initiated with the minimizer of the F for the previous (larger) value of σ. This idea of minimizing a non-convex function is called Graduated Non-Convexity [8], and is also used in simulated annealing methods.

To start the minimization, we should find a proper initial guess for the solution $\boldsymbol{\alpha}_0$, that is, the initial estimation of the sparsest solution of $\mathbf{\Phi}\,\boldsymbol{\alpha} = \mathbf{s}$. It has been shown that for the case of the simple sparse decomposition, the best initial value for $\boldsymbol{\alpha}$ is the minimum ℓ^2-norm solution of $\mathbf{\Phi}\,\boldsymbol{\alpha} = \mathbf{s}$, that is, $\boldsymbol{\alpha}_0 = \mathbf{\Phi}^T(\mathbf{\Phi}\,\mathbf{\Phi}^T)^{-1}\mathbf{s}$ [7]. The reason is that this solution minimizes the function $F(\boldsymbol{\alpha})$ subject to $\mathbf{\Phi}\,\boldsymbol{\alpha} = \mathbf{s}$ where σ goes to infinity. Despite the fact that our method is somehow different with the method presented in [7], we use the same initialization for our algorithm. Since we also have the variable w', we should start from the sparsest $\boldsymbol{\alpha}(k) = \mathbf{\Phi}^T(\mathbf{\Phi}\,\mathbf{\Phi}^T)^{-1}\mathbf{s}^{(k)}$ vector. Let $k_0 = \text{argmin}_k F(\boldsymbol{\alpha}(k))$, for $k = 1, 2, \ldots, n$. Then we choose $\boldsymbol{\alpha}(k_0)$ to be the starting point of our algorithm.

Because of noise, if our algorithm tries to satisfy (7) exactly, it would be very sensitive to noise. Consequently, we try to satisfy this equation approximately. We realize this idea by minimizing the function H defined below with respect to $\boldsymbol{\alpha}$ and w':

$$H(\boldsymbol{\alpha}, w') = \lambda G(\boldsymbol{\alpha}, w') + (1-\lambda)F(\boldsymbol{\alpha}) \tag{8}$$

where $0 < \lambda < 1$ is a constant that specifies the weight which is given to satisfying (7). This equation can be interpreted as a trade-off between the accuracy of the decomposition and maximizing the sparsity.

By letting $w' = e^{j\theta}$, the final objective function $H(\boldsymbol{\alpha}, \theta)$ will be a real-valued function of real-valued variables $\boldsymbol{\alpha}$ and θ. For each σ, this function may be minimized by gradient based algorithms (specifically steepest descent). Direct calculations show:

$$\frac{\partial H}{\partial \boldsymbol{\alpha}} = 2\lambda\Re\{\mathbf{\Phi}^{\,(\mathcal{F})\mathrm{T}}(\mathbf{\Phi}^{\,(\mathcal{F})}\boldsymbol{\alpha} - \mathbf{W}'\mathbf{s}^{(\mathcal{F})})\}+$$
$$(1-\lambda)(1/\sigma^2)[\alpha_1 e^{(-\alpha_1^2/2\sigma^2)}, \ldots, \alpha_m e^{(-\alpha_m^2/2\sigma^2)}]^T \tag{9}$$

$$\frac{\partial H}{\partial \theta} = -2\Re\{\mathbf{s}^{(\mathcal{F})}\mathbf{M}\mathbf{W}'\bar{\mathbf{\Phi}}\} \tag{10}$$

where $\mathbf{M} \triangleq \mathrm{diag}(0, 1j, \ldots, (n-1)j)$.

3. THE FINAL ALGORITHM

The final algorithm of the proposed method is given in Fig. 1. As seen in the algorithm, the final values of the previous estimation are used for initialization of the next steepest descent. As explained in the previous section, the decreasing sequence of σ is used to escape from getting trapped into local minima.

In the minimization part, the steepest descent with variable step-size (μ) has been used: If μ is such that $H(\boldsymbol{\alpha} - \mu\frac{\partial H}{\partial \boldsymbol{\alpha}}, \theta - \mu\frac{\partial H}{\partial \theta}) < H(\boldsymbol{\alpha}, \theta)$ we multiply the value of μ by 1.2 for the next iteration. Otherwise if μ is such that $H(\boldsymbol{\alpha} - \mu\frac{\partial H}{\partial \boldsymbol{\alpha}}, \theta - \mu\frac{\partial H}{\partial \theta}) \geq H(\boldsymbol{\alpha}, \theta)$ we multiply the value of μ by 0.5 for the next iteration.

```
• Initialization:
    1. Let: $F(\boldsymbol{\alpha}) = m - \sum_{i=1}^{m} e^{-{\alpha_i}^2/2\sigma^2}$ and
       $\boldsymbol{\alpha}(k) = \mathbf{\Phi}^{\,T}(\mathbf{\Phi}\,\mathbf{\Phi}^{\,T})^{-1}\mathbf{s}^{(k)}$.
    2. Find the minimum of $F(\boldsymbol{\alpha}(k))$ for $k = 1, 2, \cdots, n$. Assuming this minimum occurs
       for $k = k_0$, let $\hat{\boldsymbol{\alpha}}_0 = \boldsymbol{\alpha}(k_0)$ and $\hat{\theta}_0 = \frac{2\pi}{n}k_0$.
    3. Choose a suitable decreasing sequence for $\sigma = [\sigma_1 \ldots \sigma_R]$. Choose a small value for $\mu$.
    4. Let $\mathbf{M} = \mathrm{diag}(0, 1j, \ldots, (n-1)j)$.
• For $r = 1, \ldots, R$:
    1. Let $\sigma = \sigma_r$.
    2. Minimize (approximately) the function $H(\boldsymbol{\alpha}, \theta)$ using $L$ iterations of the steepest
       descent algorithm:
        – Initialization: $\boldsymbol{\alpha} = \hat{\boldsymbol{\alpha}}_{r-1}$ and $\theta = \hat{\theta}_{r-1}$.
        – for $l = 1 \ldots L$ (loop $L$ times):
            (a) Calculate $\frac{\partial H}{\partial \boldsymbol{\alpha}}$ and $\frac{\partial H}{\partial \theta}$ from (9) and (10), respectively.
            (b) If $H(\boldsymbol{\alpha} - \mu\frac{\partial H}{\partial \boldsymbol{\alpha}}, \theta - \mu\frac{\partial H}{\partial \boldsymbol{\alpha}}) <$
                $H(\boldsymbol{\alpha}, \theta)$ let $\rho = 1.2$ else $\rho = 0.5$.
            (c) Let $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} - \mu\partial H/\partial\boldsymbol{\alpha}$
                and $\theta \leftarrow \theta - \mu\partial H/\partial\theta$.
            (d) Let $\mu \leftarrow \mu \times \rho$.
    3. Set $\hat{\boldsymbol{\alpha}}_r = \boldsymbol{\alpha}$ and $\hat{\theta}_r = \theta$.
• Let $\hat{\boldsymbol{\alpha}} = \hat{\boldsymbol{\alpha}}_R$ and $\hat{\theta} = \hat{\theta}_R$. The final coefficient
  vector is $\hat{\boldsymbol{\alpha}}$ and the final shift value is $n \times \frac{\hat{\theta}}{2\pi}$.
```

Fig. 1. The final algorithm

4. EXPERIMENTAL RESULTS

In order to experimetally evaluate our method, we generated a random dictionary $\mathbf{\Phi}$ which had 80 atoms and each atom was a signal of length 40 (thus we assumed $m = 80$ and $n = 40$ in our simulations). Then we created a synthetic vector $\mathbf{s}$ by generating a sparse coefficient vector $\boldsymbol{\alpha}$ at random, using a Bernoulli-Gaussian model: each coefficient is 'active' with probability p, and is 'inactive' with probability $1-p$. If it is active, its value is modeled by a zero-mean Gaussian random variable with variance σ_{on}^2; if it is not active, its value is modeled by a zero-mean Gaussian random variable with variance σ_{off}^2, where $\sigma_{\text{off}}^2 \ll \sigma_{\text{on}}^2$. Consequently, each α_i is distributed as:

$$\alpha_i \sim p \cdot \mathcal{N}(0, \sigma_{\text{on}}) + (1-p) \cdot \mathcal{N}(0, \sigma_{\text{off}}), \tag{11}$$

where p denotes the probability of activity of the coefficient, and sparsity implies that $p \ll 1$. In our simulations we have fixed $\sigma_{\text{on}} = 1, \sigma_{\text{off}} = 0.01, p = 0.1$, and $\lambda = 0.75$.

Then we created the signal $\mathbf{s}$ by $\mathbf{s} = \mathbf{\Phi}\,\boldsymbol{\alpha} + \mathbf{n}$, where $\mathbf{n}$ is an additive white Gaussian noise with zero mean and standard deviation $\sigma_{\text{n}} = 0.01$. Finally, we shifted the $\mathbf{s}$ vector circularly by k samples where k was a random number from 0 to 39. We applied our algorithm to convolutively decompose this vector $\mathbf{s}$ over the dictionary $\mathbf{\Phi}$.

The simulation was repeated 1000 times with randomly generated coefficients, dictionary and the shift of the signal, and it was seen that in 992 experiment the algorithm could sucessfully estimate the shift value and the coefficient vector $\boldsymbol{\alpha}$. In average, the Signal to Noise Ratio[1] (SNR) was greater than 24dB. Figure 2 shows one of the runs of these experiments. In the other 8 experiments the algorithm felled into local minima, and could not correctly estimated $\boldsymbol{\alpha}$ and θ.

[1] Signal to Noise Ratio is defined as $10\log_{10}\frac{\|\boldsymbol{\alpha}\|^2}{\|\hat{\boldsymbol{\alpha}}-\boldsymbol{\alpha}\|^2}$ where $\hat{\boldsymbol{\alpha}}$ is the estimated coefficient vector.

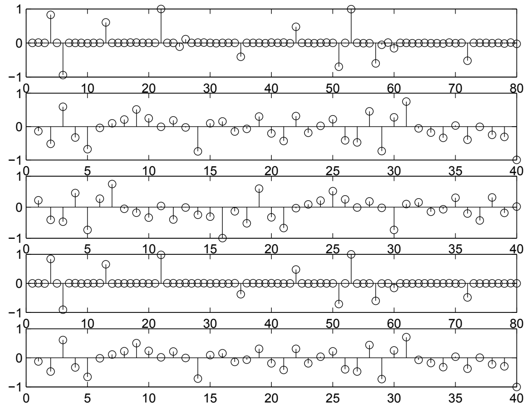

Fig. 2. A sample of our experiments. From top to bottom, first plot represents a randomly generated coefficient vector $\boldsymbol{\alpha}$, second plot is the synthetic vector $\mathbf{s}$ which has the coefficient vector $\boldsymbol{\alpha}$ on the randomly generated dictionary $\mathbf{\Phi}$, third plot is a randomly shifted version of vector $\mathbf{s}$ which is the input of our algorithm, fourth plot is the estimated coefficient vector $\hat{\boldsymbol{\alpha}}$, and the last plot is the vector $\hat{\mathbf{s}}$ which has the coefficient vector $\hat{\boldsymbol{\alpha}}$.

In order to see the effect of λ on the estimation quality, the algorithm was repeated for λ's between 0.3 and 0.9 (outside this interval SNR decreases rapidly). For each value of λ we repeated the algorithm 100 times and the mean SNR for each λ is computed. The mean SNR is plotted versus λ in Fig. 3.

5. CONCLUSION

In this paper, a new method was proposed as the first step for solving the convolutive sparse decomposition problem. The proposed method can be used in the cases in which we know that one of the shifted versions of a signal $\mathbf{s}$ has a sparse representation on an overcomplete dictionary, and we are looking for the sparsest representation among the representations of all the shifted versions of $\mathbf{s}$. We used Discrete Fourier Transform (DFT) to convert the problem to a continuous optimization problem. The proposed method was fast because of using the idea of smoothed ℓ^0-norm [7]. Experimental results emphasized on the performance of the proposed algorithm.

It seems that the proposed algorithm can be generalized for applying to the general convolutive sparse representation problem (in which the shift values k_i are not necessarily equal). However, our simulations show that the main difficulty of such a generalization is that the algorithm very oftenly traps into local minima. Such a generalization is currently under study in our group.

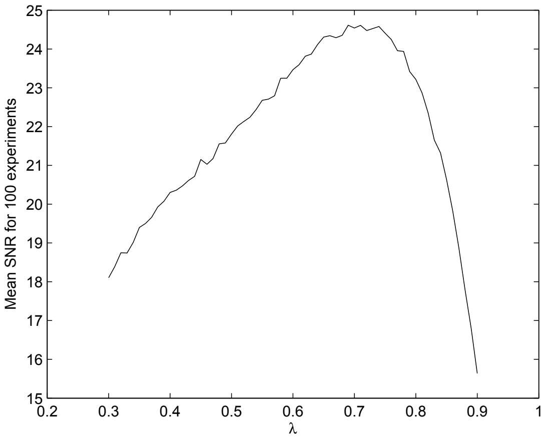

Fig. 3. Output SNR versus λ.

6. REFERENCES

[1] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1999.

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

[3] D. L. Donoho, M. Elad, and V. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. Info. Theory*, vol. 52, no. 1, pp. 6–18, Jan 2006.

[4] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal l^1-norm solution is also the sparsest solution," Tech. Rep., 2004.

[5] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS, a re-weighted minimum norm algorithm," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600–616, March 1997.

[6] S. Krstulovic and R. Gribonval, "MPTK: Matching pursuit made tractable," in *ICASSP'06*, 2006.

[7] G.H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "Fast sparse representation based on smoothed l^0-norm," in *ICA2007*, London, September 2007.

[8] A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press, 1987.