THRESHOLDED SMOOTHED-ℓ^0(SL0) DICTIONARY LEARNING FOR SPARSE REPRESENTATIONS

H. Zayyani, M. Babaie-Zadeh[*]

Sharif University of Technology,
Department of Electrical Engineering
and Advanced Communication Research
Institute, Tehran, Iran

R. Gribonval

INRIA, Centre Inria Rennes-Bretagne Atlantique,
Campus de Beaulieu,
F-35042 Rennes Cedex, France

ABSTRACT

In this paper, we suggest to use a modified version of Smoothed-ℓ^0 (SL0) algorithm in the sparse representation step of iterative dictionary learning algorithms. In addition, we use a steepest descent for updating the non unit column-norm dictionary instead of unit column-norm dictionary. Moreover, to do the dictionary learning task more blindly, we estimate the average number of active atoms in the sparse representation of the training signals, while previous algorithms assumed that it is known in advance. Our simulation results show the advantages of our method over K-SVD in terms of complexity and performance.

Index Terms— Dictionary learning, Sparse representation, Compressed sensing, Sparse Component Analysis.

1. INTRODUCTION

Sparse representation of signals has a wide range of applications in signal processing including underdetermined Blind Source Separation (BSS) and Sparse Component Analysis (SCA) [1] and Compressed Sensing [2]. In all applications, there should be a dictionary such that the expansion of the signal based on the columns of the dictionary (called atoms) is sparse. So, finding the proper dictionary is a pre-processing task in all the applications. One way is to use some predefined analytically constructed dictionaries, e.g., Wavelet Packets (WP) and Discrete Cosine Transform (DCT). They should be designed analytically for each special class of signals. Another way is to learn a dictionary based on a set of training signals, which is called dictionary learning.

In dictionary learning, we want to find a dictionary such that the representations of all training signals in that dictionary are sparse. Consider the following model:

$$\mathbf{Y} = \mathbf{D}\mathbf{X} + \mathbf{E} \tag{1}$$

where all the training signals are collected in a signal matrix $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N]$ and all the sparse coefficients are collected in a coefficient matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$ and N is the number of training signals. $\mathbf{D}$ is an $n \times m$ overcomplete dictionary ($m > n$) which supposed to be learned from the training signals $\mathbf{Y}$. m is the number of atoms and n is the length of the signals. $\mathbf{E}$ is the collection of observation noises which can also be considered as approximation errors.

Most dictionary learning algorithms use two step iterative techniques to solve the problem. In the first step, they use a sparse representation algorithm to determine the sparse coefficients based on knowing the dictionary. In the second step, they update the dictionary based on some criteria such as maximizing a likelihood probability or minimizing a cost function.

In Method of Optimal Directions (MOD) [3], an optimal updated direction is computed by minimizing the total Mean Square Error (MSE) and then a normalizing operation is done to preserve the norm of the columns of the dictionary. This normalization may increase the MSE. Some other statistical methods assume prior sparse distributions for coefficients and try to update the dictionary optimally in a Bayesian framework [4], [5]. In [4], a Laplacian distribution is used for modeling the sparse sources and update the dictionary based on the Maximum likelihood (ML) of the dictionary by some approximations in computing the likelihood. In [5], some prior information is used for the dictionaries and a Maximum A Posteriori (MAP) dictionary estimate is used for dictionary update. The K-SVD algorithm [6] uses a SVD operation for updating only one column at each iteration based on reducing MSE. It also allows the coefficients to be updated after each column update. Another SVD based method is also used for dictionary learning in the case of unions of orthonormal bases [7]. Recently, a regularized dictionary learning algorithm is proposed to minimize a cost function under different constraints on the dictionary [8].

In our paper, we use smoothed-ℓ^0(SL0) for finding sparse coefficients which is a fast and accurate sparse representation algorithm [9]. For finding exact sparse representation of

[*]This work has been partially funded by Iran NSF (INSF) under contract number 86/994, by Iran Telecom Research Center (ITRC), and also by center for International Research and Collaboration (ISMO) and French embassy in Tehran in the framework of a GundiShapour collaboration program.

the training signals, we suggest to use a thresholded SL0 in the sparse coefficient update stage of dictionary learning.Both schemes are based on statistical estimation techniques. In this paper, we focus only on the unit column-norm constraint on the dictionary. We use a steepest descent method to reduce the MSE by updating the non unit column-norm dictionary elements because re-normalizing the column norms after updating the unit column-norm dictionary may increase the MSE.

2. THRESHOLDED SL0 FOR SPARSE REPRESENTATION

In SL0 algorithm, ℓ^0 norm which is a discontinuous measure of sparsity, is approximated by the continuous function [9]:

$$||x||_0 \approx m - \sum_{i=1}^{m} \exp(-\frac{x_i^2}{2\sigma^2}) \tag{2}$$

where $\sigma \to 0$. Then, SL0 uses a steepest ascent method, for maximizing $F_\sigma(\mathbf{x}) \triangleq \sum_{i=1}^{m} \exp(-\frac{x_i^2}{2\sigma^2})$ subject to $\mathbf{y} = \mathbf{Dx}$ [9]. This algorithm is very fast and is about 2 or 3 orders of magnitude faster than Basis Pursuit (BP) based on ℓ^1-magic [9].

Our motivation to use this method is to reduce the complexity of dictionary learning algorithms which is a more complicated task than just a sparse representation problem. It can be viewed as many simultaneously sparse representation problem at the same time. The problem, in the dictionary learning application in the noisy case, is that SL0 finds the sparse coefficients which are not exactly sparse. It means that since it uses numerical optimization of a continuous approximation to ℓ^0 norm, all the coefficients have some amplitudes which may be very small. To find sparse coefficients with limited number of nonzero coefficients, we suggest to use two thresholding schemes on the solution of the SL0 algorithm.

2.1. Thresholding on the number of active atoms

In this method, we select the K largest coefficients of the SL0 solution. Similarly, K-SVD algorithm uses K atoms for sparse representation which is done by Orthogonal Matching Pursuit (OMP) [6]. In K-SVD, the number K is assumed to be known in advance. But, we propose an statistical method to estimate the average number of active atoms.

To estimate the number of active atoms in sparse representations of training signals, we assume a statistical model for the coefficients. We use Bernoulli-Gaussian (BG) model for that. So, we assume:

$$p(x_{ij}) = (1-p)\delta(x_{ij}) + pN(0, \sigma_r^2) \tag{3}$$

where $1 \leq i \leq m$ and j is the index of the training signal ($0 \leq j \leq N$). p is the probability of having nonzero coefficient (or activity of atoms) and σ_r^2 is the variance of nonzero coefficients. The probability of having exact k nonzero coefficients has a binomial distribution which is $p(K=k) = \binom{m}{k} p^k (1-p)^{m-k}$. This binomial distribution can be approximated by a continuous Gaussian distribution which is $N(mp, mp(1-p))$. So, the Maximum likelihood estimation of K, the number of active atoms, is mp that is the same as the expected value of the binomial distribution. Therefore, the average value of K is equivalent to its ML estimation and is equal to mp.

To estimate p, we use some similar moment based techniques which was used for parameter estimation in SCA [10]. In [10], the matrix is known in advance, but here we assume a statistical nature for our dictionary matrix. We consider each element of the training signal as a random variable as $y_{ij} = \sum_{r=1}^{m} d_{ir} x_{rj} + e_{ij}$. We use the second and fourth order moment of this random variable for estimating p. We have:

$$E(y_{ij}^2) \approx mE(d_{ir}^2)E(x_{rj}^2) \tag{4}$$

$$E(y_{ij}^4) \approx mE(d_{ir}^4)E(x_{rj}^4) + 6\binom{m}{2} E^2(d_{ir}^2)E^2(x_{rj}^2) \tag{5}$$

where we neglect the noise term and we assume that dictionary elements are zero mean and independent of BG coefficients. Since we assume that the columns of the dictionary have unit norms, we can write $\sum_{i=1}^{n} d_{ir}^2 = 1$. Taking expectation from the both sides of this equation and assuming that all the elements have identical distribution, we will have $E(d_{ir}^2) = \frac{1}{n}$. We also have $(\sum_{i=1}^{n} d_{ir}^2)^2 = 1$. Again, computing expectation of both sides of this equation, we have $nE(d_{ir}^4) + n(n-1)E(d_{ir}^2) = 1$. Then, we can find that $E(d_{ir}^4) = \frac{1}{n^2}$. If we define $m_2 \triangleq E(x_{rj}^2)$ and $m_4 \triangleq E(x_{rj}^4)$, they can be found by (4) and (5), in which $E(d_{ir}^2) = \frac{1}{n}$ and $E(d_{ir}^4) = \frac{1}{n^2}$ and the second and fourth moment of training signal elements y_{ij} are calculated by simple averaging. After finding m_2 and m_4, similar to [10], the parameter p can be estimated as:

$$\hat{p} = \frac{3m_2^2}{m_4} \tag{6}$$

Therefore, in our method, we can find an estimation of $\hat{K} = m\hat{p}$ for the number of active atoms based on (6).

2.2. Thresholding on the amplitude of the coefficients

Another method for thresholding the solution of the SL0 algorithm is to use a threshold on the absolute value of the amplitude of the SL0 coefficients. To find some optimal threshold, again we use another a bit different BG model for the coefficients. Now, we assign a small variance σ_{off} for the small coefficients of the SL0 solution and a larger variance σ_{on} for active coefficients. If $\tilde{x}_{ij}$ is the SL0 coefficient, we assume that $p(\tilde{x}_{ij}) = (1-p)N(0, \sigma_{\text{off}}^2) + pN(0, \sigma_{\text{on}}^2)$, where $\sigma_{\text{off}} \ll \sigma_{\text{on}}$. Then, a hypothesis testing is used to decide the coefficient is active or inactive. Using Bayes rule, the posterior probabilities are $p(\text{Activity}|\tilde{x}_{ij}) \propto p(\text{Activity})p(\tilde{x}_{ij}|\text{Activity})$ and

$p(\text{Inactivity}|\tilde{x}_{ij}) \propto p(\text{Inactivity})p(\tilde{x}_{ij}|\text{Inactivity})$ where $p(\text{Activity}) = p$ and $p(\text{Inactivity}) = (1-p)$. The likelihood are also Gaussian. If we do some simple calculations for hypothesis testing, we will find that the decision for the activity is as follows:

$$\text{Activity} : |\tilde{x}_{ij}| > \sqrt{\frac{2\log\frac{1-p}{p}\frac{\sigma_{\text{on}}}{\sigma_{\text{off}}}}{\frac{1}{\sigma_{\text{off}}^2}-\frac{1}{\sigma_{\text{on}}^2}}} \triangleq \text{Th}_{\text{opt}} \qquad (7)$$

where the optimal threshold can be approximated with $\text{Th}_{\text{opt}} \approx \alpha\sigma_{\text{off}}$ where $\alpha = \sqrt{2\log(\frac{1-p}{p}\frac{\sigma_{\text{on}}}{\sigma_{\text{off}}})}$. Because of the sparsity of the coefficients, we can slightly overestimate σ_{off} by the variance of all the coefficients which is $\sigma_{\text{off}}^2 \gtrsim E(\tilde{x}_{ij}^2)$. For typical parameters $p = .9$ and $\frac{\sigma_{\text{on}}}{\sigma_{\text{off}}} = 100$, the value of α is equal to 3.69.

3. STEEPEST DESCENT DICTIONARY UPDATE WITH UNIT NORM COLUMN CONSTRAINT

In this paper, we focus on the unit column-norm constraint rather than Frobeniuos-norm constraint. The authors of [8] convert this constraint to a term in their cost function by a Lagrangian method and because determining the optimum Lagrangian multipliers was difficult, they used an iterative method by adding some terms to their cost function to optimize it. Here, we solve the problem of preserving the norm of the columns by considering a model that automatically have unit norms for columns. If **C** is the non unit column-norm dictionary and **D** is the unit column-norm dictionary, then the relation between this two dictionaries is $d_{ij} = \frac{c_{ij}}{\sqrt{\sum_{i=1}^{m} d_{ij}^2}}$. We use this relation to directly optimize the non unit column-norm dictionary **C** by considering the true MSE which is defined with respect to unit column-norm dictionary **D**. The MSE, which we want to minimize or at least to decrease, is $F \triangleq \text{MSE} = ||\mathbf{Y} - \mathbf{DX}||$. In K-SVD, an Singular Value Decomposition (SVD) which is a complex operation is done to update only one atom. But, we use a simple steepest descent method to decrease MSE with respect to **C**. The steepest descent for the elements of matrix **C** is $c_{ij} \leftarrow c_{ij} - \mu\frac{\partial F}{\partial c_{ij}}$. We have $\frac{\partial F}{\partial c_{ij}} = \frac{\partial F}{\partial d_{ij}}\frac{\partial d_{ij}}{\partial c_{ij}}$. By writing $F = \sum_{r=1}^{N}||\mathbf{y}_r - \mathbf{Dx}_r||_2^2 = \sum_{r=1}^{N}(\mathbf{y}_r - \mathbf{Dx}_r)^T(\mathbf{y}_r - \mathbf{Dx}_r)$ and by some further calculations, we have:

$$\frac{\partial F}{\partial d_{ij}} = \sum_{r=1}^{N}(-2\mathbf{y}_r\mathbf{x}_r^T + 2\mathbf{Dx}_r\mathbf{x}_r^T) \qquad (8)$$

The partial derivative $\frac{\partial d_{ij}}{\partial c_{ij}}$ can also be computed easily as $g_{ij} \triangleq \frac{\partial d_{ij}}{\partial c_{ij}} = \frac{||\mathbf{c}_j||_2^2 - c_{ij}^2}{||\mathbf{c}_j||_2^3}$, where $\mathbf{c}_j$ is the j'th column of the matrix **C**. Finally, the overall steepest descent for dictionary update is:

$$\mathbf{C} \leftarrow \mathbf{C} - \eta(\sum_{r=1}^{N}(\mathbf{Dx}_i - \mathbf{y}_i)\mathbf{x}_i^T) \odot \mathbf{G} \qquad (9)$$

where μ has been replaced by η just for scaling and the matrix **G** is a correction matrix due to using the steepest descent for matrix **C** (which has not any constraint) instead of the unit column-norm dictionary **D**. The notation $\odot$ is the Hadamard product which means the element-wise multiplication.

4. EXPERIMENTS

In this section, we investigate our proposed method which uses a new steepest descent formulation for dictionary update. We used two thresholding schemes. One is thresholding on the amplitude and the other is thresholding on the number of active atoms. For abbreviation, we nominate them ATH-SL0 (Amplitude THresholded SL0) and KTH-SL0 (K-THresholded SL0), respectively. For performance comparison, we used the success definition as [6] with the difference that we state the success in percent (number of all successes divided by the number of columns). We repeated our experiment for 20 times and report the averaged success of each algorithm. We also use the CPU time as a measure of complexity. Our simulations were performed in MATLAB7.0 environment using an Intel 2.80 GHz processor with 1024 MB of RAM and under Linux operating system.

In our experiments, we used a random dictionary matrix with uniformly distributed elements on $[-1, 1]$, and then normalized its columns. The dictionary size was selected as $m = 40$ and $n = 20$. The sparse coefficients were generated from a BG model with parameters $p = \frac{K}{m}$ and $\sigma_r = 1$ based on model (3). K which is the average number of active atoms was selected between 3 and 8. The training signals were generated from model (1). Number of training signals was selected as 1000 and the variance of the Gaussian noise was selected as $\sigma_n = .01$.

Our algorithm with its different versions are compared with K-SVD algorithm in various conditions. In ATH-SL0, we used $\alpha = 3$ and $\alpha = 5$. For KTH-SL0, we used two values for choosing the number of atoms. Firstly, we assumed that K was known in advance. Secondly, we used our estimated value $\hat{K} = m\hat{p}$. For K-SVD, we considered three cases. Firstly, we assumed that the value of K was known as a prior information. Secondly, we used an underestimate $K = 3$. Thirdly, we used an overestimate value $K = 7$.

We first examine our parameter estimation approach stated in Sec. 2.1. Table 1 shows the average estimated value of $\hat{K} = m\hat{p}$ versus $K = mp$ averaged over 100 different experiments and the variance of these estimations. It can be seen that the accuracy of estimations is good. Secondly, the performance of all algorithms are compared with each other in Fig. 1. It can be seen that ATH-SL0 has the best performance. The parameter $\alpha = 3$ is better than $\alpha = 5$ for wider range of values for K. KTH-SL0 with estimated parameter is worse than KTH-SL0 with true parameter (less than 10%). It also can be seen that the performance of K-SVD is decreased when the exact value of K is not known in advance in both

Table 1. Result of estimating the averaged number of active atoms for the case $m = 40$, $n = 20$, $p = \frac{K}{m}$, $\sigma_r = 1$ and $\sigma_n = .01$.

K	3	4	5	6	7	8
Mean($\hat{K}$)	3.35	4.47	5.50	6.72	7.87	9.06
Var($\hat{K}$)	0.13	0.26	0.29	0.50	0.78	1.22

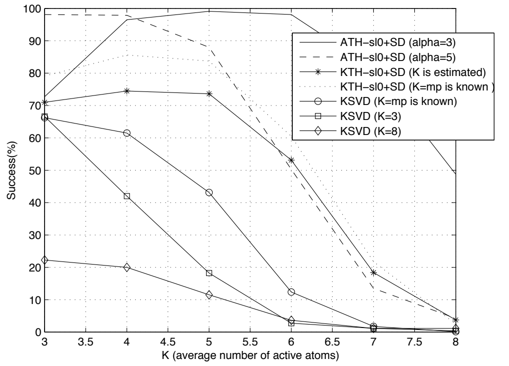

Fig. 1. The success rate for various algorithms versus K. The parameters are $m = 40$, $n = 20$, $p = \frac{K}{m}$, $\sigma_r = 1$, $\sigma_n = .01$, $\eta = 10^{-2}$ and $N = 1000$. 20 iterations are used for all algorithms.

$K = 3$ and $K = 8$ cases. Figure 2 shows the simulation time versus the value of K. It can be seen that our algorithm is approximately one order of magnitude faster than K-SVD because both sparse coefficient recovery stage (using SL0 instead of OMP) and the dictionary update stage (steepest descent instead of SVD) is simpler than K-SVD.

5. CONCLUSION

In this paper, to reduce the complexity of dictionary learning algorithms, we suggested to use the SL0 method. We proposed two modified version of this algorithm to be used with dictionary learning. We also estimated the average number of active atoms in dictionary learning with a moment based approach. We also used a modified steepest descent for updating the direction of the non unit column-norm dictionary instead of unit column-norm dictionary. In the simulations, our algorithm shows better results than K-SVD both in terms of complexity and performance.

6. REFERENCES

[1] R. Gribonval and S. Lesage, "A survey of sparse component analysis for blind source separation: principles, perspectives, and new challanges," *in proceeding ESSAN'06*, pp. 323–330, 2006.

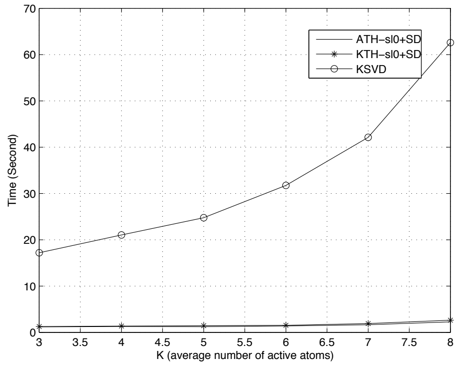

Fig. 2. The simulation time for various algorithms versus K. The parameters are $m = 40$, $n = 20$, $p = \frac{K}{m}$, $\sigma_r = 1$, $\sigma_n = .01$, $\eta = 10^{-2}$ and $N = 1000$. 20 iterations are used for all algorithms.

[2] D.L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, pp. 1289–1306, April 2006.

[3] K. Engan, S.O. Aase, and J.H. Hakon-Husoy, "Method of optimal directions for frame design," *in ICASSP 1999*, pp. 2443–2446, 1999.

[4] M.S. Lewicki and T.J. Sejnowski, "Learning overcomplete representations," *Neural Comp.*, vol. 12, pp. 337–365, 2000.

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

[6] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal. Proc*, vol. 54, pp. 4311–4322, November 2006.

[7] S. Lesage, R. Gribonval, F. Bimbot, and L. Benaroya, "Learning unions of orthonormal bases with thresholding singular value decomposition," *in ICASSP 2005*, pp. 293–296, 2005.

[8] M. Yaghoobi, T. Blumensath, and M. Davies, "Regularized dictionary learning for sparse approximation," *in EUSIPCO 2008*, 2008.

[9] H. Mohimani, M. Babaie-zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed ℓ^0-norm," *IEEE Trans. Signal. Processing*, vol. 57, pp. 289–301, January 2009.

[10] H. Zayyani, M. Babaie-zadeh, and C. Jutten, "Source estimation in noisy sparse component analysis," *in proceeding DSP 2007*, pp. 219–222, 2007.