# Parametric Dictionary Learning Using Steepest Descent

Mahdi Ataee, Hadi Zayyani, Massoud Babaie-Zadeh, Christian Jutten

# PARAMETRIC DICTIONARY LEARNING USING STEEPEST DESCENT

*Mahdi Ataee*[1]*, Hadi Zayyani*[1]*, Massoud Babaie-Zadeh*[1]**, and Christian Jutten*[2]

[1]Department Of Electrical Engineering, Sharif University of Technology, Tehran, Iran
[2]GIPSA-Lab, Grenoble, and Institut Universitaire de France, France.
mahdiataee1367@yahoo.com, zayyani2000@yahoo.com
mbzadeh@yahoo.com, Christian.Jutten@inpg.fr

## ABSTRACT

In this paper, we suggest to use a steepest descent algorithm for learning a parametric dictionary in which the structure or atom functions are known in advance. The structure of the atoms allows us to find a steepest descent direction of parameters instead of the steepest descent direction of the dictionary itself. We also use a thresholded version of Smoothed-$\ell^0$ (SL0) algorithm for sparse representation step in our proposed method. Our simulation results show that using atom structure similar to the Gabor functions and learning the parameters of these Gabor-like atoms yield better representations of our noisy speech signal than non parametric dictionary learning methods like K-SVD, in terms of mean square error of sparse representations.

***Index Terms—*** Dictionary learning, Sparse representation, parametric dictionary, Sparse Component Analysis.

## 1. INTRODUCTION

Sparse representation of signals has found a wide range of applications in signal processing in recent years including Sparse Component Analysis (SCA) [1] and Compressed Sensing [2]. In these applications, the existence of a proper dictionary in which the signals have sparse representations is a preliminary necessity. It means that there should be a dictionary in advance, such that the expansion of the signal based on the columns of the dictionary (called atoms) is sparse.

To choose a dictionary in these applications, one way is to use some predefined analytically constructed dictionaries, e.g., Wavelet Packets (WP) and Discrete Cosine Transform (DCT) in special class of signals such as images, speeches or biomedical signals. They should be designed analytically for each class of signals. This approach could be nominated as dictionary design. The dictionary design can be performed by extensive research on generative model of signals. So, in this method an extreme effort should be done for modeling the signals of interest.

The other way which is more general is to learn a dictionary based on a set of training signals. This approach is called dictionary learning (Refer to [3], [4], [5], [6] and a more recent paper [7]). 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:

$$\mathbf{Y} = \mathbf{DX} + \mathbf{E} \tag{1}$$

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 is to be learned from the training signals $\mathbf{Y}$. $m$ is the number of atoms and $n$ is the length of the signals. $\mathbf{E}$ can be considered as approximation errors.

In dictionary learning methods, there is no information about the dictionaries except some mild constraints such as unit Frobenius norm of the dictionary or unit norm of columns of the dictionary. But, in some applications, the dictionaries for representations may have some known structures. In [8], a Toeplitz structure has been suggested for compressed sensing applications. In addition, [9] suggested a sparse structure for dictionary in sparse representations. Moreover, recently a parametric dictionary design is proposed for sparse modeling of signals [10]. Parametric dictionary design or parametric dictionary learning assumes a known parametric model for atoms. Then, it tries to find better parameters for atoms based on some criteria. It has the advantage of dictionary learning methods which is yielding to better and more adaptive representations of signals. It also gains the benefits of dictionary design approaches which are the simplicity and better matches to the structure of a special class of signals. An important advantage of parametric dictionary learning is that only the parameters of an atom (it is less than 5 parameters in typical applications) should be stored instead of all the samples of the atom. So, it is very well suited to the applications with large matrix dimensions.

In addition to precise modeling of our signals, we can use

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

experimental experience to select a proper parametric dictionary for our applications. For example, it is found that Gabor atoms work sufficiently well, and have fewer adverse effects in reconstruction of speech signals compared to other dictionaries [11]. The parametric dictionary learning methods uses this prior knowledge to select the atom functions properly. But, they also tries to even more improve the performance of representation by learning their parameters. In contrast, [10] uses a Gammatone function[1] for reconstruction of audio signals. It proposes an algorithm similar to alternating-minimization for reducing a cost function which is derived from the coherence of the dictionary. It shows the superior performance of the designed dictionary over the initial Gammatone dictionary in representing the audio signals. In contrast, in this paper, we use the same criterion as dictionary learning methods which is the total Mean Square Error (MSE). We examine Gabor-like atom functions for representing a sample speech signal. We not only show the superior performance of the proposed algorithm over the initial dictionary in terms of reducing MSE, but also over the K-SVD algorithm [12] and THresholded Smoothed-$\ell^0$(THSL0) dictionary learning algorithm proposed in [13].

## 2. PROBLEM FORMULATION FOR LEARNING A GENERAL PARAMETRIC DICTIONARY

Suppose that we have a dictionary $\mathbf{D} = [\mathbf{d}_1|\mathbf{d}_2|...|\mathbf{d}_m]$ which the structure or atom functions are known in advance. Each atom $\mathbf{d}_k$ is determined completely by $P$ deterministic parameters collected in a parameter vector $\mathbf{v}_k = [v_{1k}, v_{2k}, ..., v_{Pk}]^T$ where $v_{ik}$ is the $i$'th parameter of the $k$'th atom. We call this dictionary as a parametric dictionary. In parallel to this original $n \times m$ parametric dictionary, a $P \times m$ parameter dictionary can be defined as $\mathbf{V} = [\mathbf{v}_1|\mathbf{v}_2|...|\mathbf{v}_m]$. This parameter dictionary $\mathbf{V}$ has much fewer rows than the original parametric dictionary $\mathbf{D}$ because $P$ is the degree of freedom of each atom while $n$ is the length of the signals or atoms. So, in parametric dictionary learning, we need only to update the parameter matrix instead of the dictionary itself. It reduces the complexity of the algorithm by a large factor equal to $\frac{n}{P}$. In addition, reducing the number of free variables, decreases the dimensionality of the corresponding optimization problem and hence the larger dimension problems can be solved simpler and more efficient.

In parametric dictionary learning, various criteria may be used for updating or learning parameters. The first measure which is used in [10] is the nearness of the dictionary to an Equiangular Tight Frame (ETF) which has the minimum coherence. But, in this paper, similar to the classical dictionary learning algorithms such as K-SVD [12], we use the total MSE of the representation as a measure to update the parameters. The total MSE is defined as:

$$\text{MSE} = \frac{1}{N}F(\mathbf{D}) = \frac{1}{N}\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_2^2 = \frac{1}{N}\sum_{r=1}^{N}\|\mathbf{y}_r - \mathbf{D}\mathbf{x}_r\|_2^2 \tag{2}$$

where this cost function is explicitly dependent on the dictionary matrix and is implicitly dependent on the parameter matrix. To explore the implicit relation of the cost function to parameters, we need to know the deterministic function of atoms with respect to the parameters. To do so, we define the atom elements as $d_{tk} = g(\mathbf{v}_k, t)$ where $t$ stands for the time index. One of such atom functions are Gabor-like functions[2] which is defined as:

$$g_{\text{Gab}}(s, u, f, t) = \exp(-\pi(\frac{t-u}{s})^2)\cos(2\pi(t-u)f) \tag{3}$$

where $u$ and $f$ are time and frequency shifts and $s$ is the scale factor.

## 3. STEEPEST DESCENT ALGORITHM FOR LEARNING THE PARAMETERS

Most dictionary learning algorithms use two step iterative techniques to solve their problems. 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 the sparse representation step of our algorithm, we use a THSL0 algorithm [13]. In this paper, the thresholding is done on the number of active atoms and a fixed number $K$ of the largest absolute values of coefficients are chosen after performing an Smoothed-$\ell^0$ (SL0) algorithm [14]. For dictionary update, we suggest to use steepest descent method for reducing the MSE cost function defined in (2) on the parameter space instead of the admissible dictionary space.

The steepest descent update formula for the $i$'th parameter of the $k$'th atom ($v_{ik}$) is as follows:

$$v_{ik} \leftarrow v_{ik} - \mu\frac{\partial F(\mathbf{D})}{\partial v_{ik}} \tag{4}$$

where $\mu$ is a small step size. Using (2), we have $\frac{\partial F(\mathbf{D})}{\partial v_{ik}} = \sum_{r=1}^{N}\frac{\partial}{\partial v_{ik}}\|\mathbf{y}_r - \mathbf{D}\mathbf{x}_r\|_2^2$. We have $\|\mathbf{y}_r - \mathbf{D}\mathbf{x}_r\|_2^2 = -2\mathbf{y}_r^T\mathbf{D}\mathbf{x}_r + \mathbf{x}_r^T\mathbf{D}^T\mathbf{D}\mathbf{x}_i + \mathbf{y}_r^T\mathbf{y}_r$. So, the partial derivative will be decomposed into two terms as follows:

$$\frac{\partial}{\partial v_{ik}}\|\mathbf{y}_r - \mathbf{D}\mathbf{x}_r\|_2^2 = \frac{\partial}{\partial v_{ik}}(-2\mathbf{y}_r^T\mathbf{D}\mathbf{x}_r) + \frac{\partial}{\partial v_{ik}}(\mathbf{x}_r^T\mathbf{D}^T\mathbf{D}\mathbf{x}_r) \tag{5}$$

[1] The generative function for a Gammatone dictionary is defined as $g(t) = (t-u)^{\gamma-1}\exp(-2\pi bB(t-u))cos(2\pi(t-u)f)$ where $B = \frac{f}{Q} + b_{\min}$ for some $Q$ and $b_{\min}$, $u$ and $f$ are time and frequency shifts, $\gamma$ and $b$ control the rise time and the width of the atoms in the time domain, respectively [10].

[2] Note that each of the atoms are in the form of (3), a Gabor atom. However, since in the final dictionary, the values of $u$, $s$ and $f$ for different atoms will be learned from the data, and do not follow a specific structure (e.g., uniform distance), the final dictionary is different from a classical Gabor dictionary, and hence throughout the paper, we use the term 'Gabor-like' atoms and dictionary.

The first derivative term $\frac{\partial}{\partial v_{ik}}(-2\mathbf{y}_r^T\mathbf{D}\mathbf{x}_r)$ is equal to:

$$\frac{\partial}{\partial v_{ik}}(\text{trace}(-2\mathbf{y}_r^T\mathbf{D}\mathbf{x}_r)) = \frac{\partial}{\partial v_{ik}}(\text{trace}(-2\mathbf{x}_r\mathbf{y}_r^T\mathbf{D}))$$

$$= \frac{\partial}{\partial v_{ik}}\sum_{j=1}^{m}\sum_{l=1}^{n} a_{jl}d_{lj} = \sum_{j=1}^{m}\sum_{l=1}^{n} a_{jl}\frac{\partial}{\partial v_{ik}}d_{lj} \quad (6)$$

where $\mathbf{A} = [a_{jl}] \triangleq -2\mathbf{x}_r\mathbf{y}_r^T$. Since $v_{ik}$ is the parameter of $k$'th atom, we have $\frac{\partial}{\partial v_{ik}}d_{lj} = 0$ for $j \neq k$. Hence, we have:

$$\frac{\partial}{\partial v_{ik}}(-2\mathbf{y}_r^T\mathbf{D}\mathbf{x}_r) = \sum_{l=1}^{n} a_{kl}\frac{\partial d_{lk}}{\partial v_{ik}} = \mathbf{a}_k(\mathbf{\Delta}_k)_i \quad (7)$$

where $\mathbf{a}_k$ is the $k$'th row of matrix $\mathbf{A}$ and $(\mathbf{\Delta}_k)_i$ is the $i$'th column of matrix $(\mathbf{\Delta}_k)_{n\times P} = [(\Delta_k)_{li}] \triangleq \frac{\partial d_{lk}}{\partial v_{ik}}$.

Similarly, the second derivative term $\frac{\partial}{\partial v_{ik}}(\mathbf{x}_r^T\mathbf{D}^T\mathbf{D}\mathbf{x}_r)$ is equal to:

$$\frac{\partial}{\partial v_{ik}}(\text{trace}(\mathbf{x}_r^T\mathbf{D}^T\mathbf{D}\mathbf{x}_r)) = \frac{\partial}{\partial v_{ik}}(\text{trace}(\mathbf{x}_r\mathbf{x}_r^T\mathbf{D}^T\mathbf{D}))$$

$$= \frac{\partial}{\partial v_{ik}}(\text{trace}(\mathbf{X}\mathbf{G})) \quad (8)$$

where $\mathbf{X} \triangleq \mathbf{x}_r\mathbf{x}_r^T$ and $\mathbf{G} \triangleq \mathbf{D}^T\mathbf{D}$. Since only the elements of $k$'th row or $k$'th column of matrix $\mathbf{G}$ depend on the $k$'th column of matrix $\mathbf{D}$ and hence depend to the parameter $v_{ik}$, we only write these terms for calculating the above partial derivative. Hence, we have:

$$\frac{\partial}{\partial v_{ik}}(\text{trace}(\mathbf{X}\mathbf{G})) = \sum_{l\neq k}\frac{\partial}{\partial v_{ik}}(X_{lk}G_{kl}) + \sum_{j=1}^{m}\frac{\partial}{\partial v_{ik}}(X_{kj}G_{jk}) \quad (9)$$

Since both the matrices $\mathbf{X}$ and $\mathbf{G}$ are symmetric, we can write the above formula as follows:

$$\frac{\partial}{\partial v_{ik}}(\text{trace}(\mathbf{X}\mathbf{G})) = 2\sum_{l\neq k} X_{lk}\frac{\partial G_{kl}}{\partial v_{ik}} + X_{kk}\frac{\partial G_{kk}}{\partial v_{ik}} \quad (10)$$

where the first term of (10) is equal to:

$$2\sum_{l\neq k} X_{lk}\sum_{j=1}^{n} d_{jl}\frac{\partial d_{jk}}{\partial v_{ik}} = 2\sum_{j=1}^{n}\frac{\partial d_{jk}}{\partial v_{ik}}\sum_{l\neq k} X_{lk}d_{jl} \quad (11)$$

Similarly, the second term of (10) is as follows:

$$X_{kk}\frac{\partial}{\partial v_{ik}}\sum_{j=1}^{n} d_{jk}^2 = 2X_{kk}\sum_{j=1}^{n}(\frac{\partial d_{jk}}{\partial v_{ik}})d_{jk} \quad (12)$$

Hence, from (10), (11), (12) and (8), we can write:

$$\frac{\partial}{\partial v_{ik}}(\mathbf{x}_r^T\mathbf{D}^T\mathbf{D}\mathbf{x}_r) = 2\sum_{j=1}^{n}\frac{\partial d_{jk}}{\partial v_{ik}}\sum_{l=1}^{m} X_{lk}d_{jl} =$$

$$2\sum_{j=1}^{n}\frac{\partial d_{jk}}{\partial v_{ik}}(\mathbf{D}\mathbf{X})_{jk} = 2\sum_{j=1}^{n}(\mathbf{\Delta}_k)_{ji}(\mathbf{D}\mathbf{X})_{jk} \quad (13)$$

Now, from (13), (7), (5) and some simple matrix algebra, we have:

$$\frac{\partial}{\partial v_{ik}}\|\mathbf{y}_r - \mathbf{D}\mathbf{x}_r\|_2^2 = [\mathbf{\Delta}_k^T(\mathbf{A}^T + 2\mathbf{D}\mathbf{X})]_{ik} \quad (14)$$

and if we replace $\mathbf{A} = -2\mathbf{x}_r\mathbf{y}_r^T$ and $\mathbf{X} = \mathbf{x}_r\mathbf{x}_r^T$, the above formula is equal to:

$$\frac{\partial}{\partial v_{ik}}\|\mathbf{y}_r - \mathbf{D}\mathbf{x}_r\|_2^2 = [2\mathbf{\Delta}_k^T(\mathbf{D}\mathbf{x}_r - \mathbf{y}_r)\mathbf{x}_r^T)]_{ik} \quad (15)$$

and finally summing all the above terms on all the training signals, the final formula for steepest descent is derived from (4). The final formula in vector format is as follows:

$$\mathbf{v}_k \leftarrow \mathbf{v}_k - \eta(\mathbf{\Delta}_k^T\mathbf{R})_k \quad (16)$$

where $\mathbf{v}_k$ is the $k$'th parameter vector, $(\mathbf{\Delta}_k^T\mathbf{R})_k$ is the $k$'th column of $\mathbf{\Delta}_k^T\mathbf{R}$ and $\mathbf{R} \triangleq \sum_{r=1}^{N}(\mathbf{D}\mathbf{x}_r - \mathbf{y}_r)\mathbf{x}_r^T$.

## 4. EXPERIMENTS

In this section, we investigate our proposed parametric dictionary learning method in representing a noisy speech signal. The noisy speech signal is divided into $N$ blocks. Each block has a length of $n$. So, training signals are obtained from different segments of a speech signal. Then, these training signals learn a parametric dictionary. We used Gabor-like atom functions defined in (3) for our parametric dictionary. We compared our algorithm with K-SVD algorithm [12] and THSL0 dictionary learning algorithm [13]. For comparison, we repeated our experiment for 20 times and report the average of logarithmic MSE of each algorithm. So, the comparison measure is as follows:

$$\text{MSE}_{\text{ave}} = \frac{1}{20}\sum_{h=1}^{20} 10\log(\frac{1}{N}\sum_{r=1}^{N}\|\mathbf{y}_r - \mathbf{D}\mathbf{x}_r\|_2^2) \quad (17)$$

where $h$ is the index of the experiment.

In our experiments, for initialization, we used a semi random dictionary matrix with Gabor-like atom functions. The parameter $u$ which is the time shifts are selected linearly to cover the entire length of the signal. But, we selected parameters $f$ and $s$ randomly. For speech signals, we used approximately $5.5$ seconds of a speech signal with 44100 samples per second.

In the first experiment, the number of nonzero coefficients $K$ is varied. The length of the signal is chosen as $n = 30$. The number of atoms are selected as $m = 50$ and $K$ is varied between 15 to 27. The results of averaged MSE in terms of the number of nonzero coefficients are shown in Fig. 1. It can be seen that our algorithm has the best result. THSL0 dictionary learning method has also better results than K-SVD.

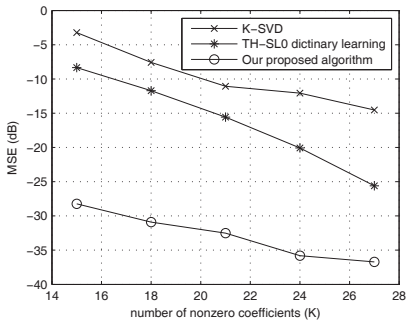


**Fig. 1**. The MSE for various algorithms versus number of non zero coefficients $K$. The parameters are $m = 50$, $n = 30$. 20 iterations are used for all algorithms.

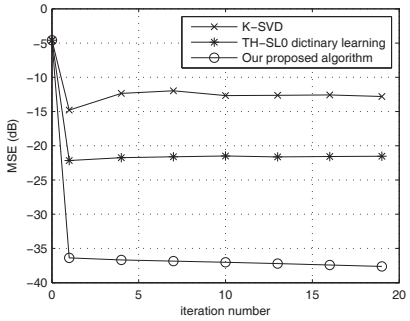


**Fig. 2**. The MSE for various algorithms versus number of iterations. The parameters are $m = 50$, $n = 30$, $K = 25$ and $N = 400$.

In the second experiment, we the number of nonzero coefficients are $K = 25$. Then, we depicted the averaged MSE with respect to iteration in Fig. 2. It is shown that the rate of convergence of our algorithm is higher than K-SVD.

## 5. CONCLUSION

In this paper, we proposed a steepest descent algorithm for a general parametric dictionary which the structure of atom functions are known in advance. We also used a THSL0 algorithm in the sparse representation step of our algorithm which is an efficient while fast method. Experimentally, our proposed algorithm for Gabor-like atom functions outperforms K-SVD and THSL0 dictionary learning algorithms in representing a noisy speech signal in terms of MSE.

## 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.

[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] 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.

[7] R. Rubinstein, A. M. Bruckstei, and M. Elad, "Dictionaries for sparse representation modeling," *Submitted to IEEE Proceedings - Special Issue on Applications of Compressive Sensing and Sparse Representation*, 2009.

[8] W. U. Bajwa, J. D. Haupt, G. M. Raz, S. J. Wright, and R. D. Nowak, "Toeplitz-structures compressed sensing matrices," *in Proc. 14th IEEE/SP Workshop Statistical Signal Processing (SSP'07)*, pp. 294–298, 2007.

[9] R. Robinson, M. Zibulevsky, and M. Elad, "Learning sparse dictionaries for sparse signal approximation," *Submitted to IEEE Trans. Signal. Proc*, 2009.

[10] M. Yaghoobi, L. Daudet, and M. Davies, "Parametric dictionary design for sparse coding," *in Workshop of Signal Processing with Adaptive Sparse Structured Representations (SPARS'09)*, 2009.

[11] B. L. Sturm and J. D. Gibson, "Matching pursuit decompositions of non-noisy speech signals using several dictionmaries," *in Proc. ICASSP 2006*, 2006.

[12] 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.

[13] H. Zayyani and M. Babaie-Zadeh, "Thresholded smoothed-$\ell^0$ (SL0) dictionary learning for sparse representations," *in Proc. ICASSP 2009*, pp. 1825–1828, 2009.

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