Parametric Dictionary Design for Sparse Coding

Mehrdad Yaghoobi, Laurent Daudet, Mike E. Davies

Parametric Dictionary Design for Sparse Coding

Mehrdad Yaghoobi†, and Laurent Daudet‡, and Mike E. Davies†,
†Institute for Digital Communication, The University of Edinburgh, Kings Buildings, Mayfield Road, Edinburgh EH9 3JL, UK. yaghoobi@ieee.org, mike.davies@ed.ac.uk
‡The Universite Pierre et Marie Curie-Paris VI, Institut Jean le Rond dAlembert-LAM, 11 rue de Lourmel, 75015 Paris, France. daudet@lam.jussieu.fr

***Abstract*—This paper introduces a new dictionary design method for sparse coding of a class of signals. It has been shown that one can sparsely approximate some natural signals using an overcomplete set of parametric functions, e.g. [1], [2]. A problem in using these parametric dictionaries is how to choose the parameters. In practice these parameters have been chosen by an expert or through a set of experiments. In the sparse approximation context, it has been shown that an incoherent dictionary is appropriate for the sparse approximation methods. In this paper we first characterize the dictionary design problem, subject to a minimum coherence constraint. Then we briefly explain that equiangular tight frames have minimum coherence. The parametric dictionary design is then to find an admissible dictionary close to being tight frame. The complexity of the problem does not allow it to be solved exactly. We introduce a practical method to approximately solve it. Some experiments show the advantages one gets by using these dictionaries.**

***Index Terms*—Sparse Approximation, Dictionary Design, Incoherent Dictionary, Parametric Dictionary, Gammatone Filter Banks, Exact Sparse Recovery.**

I. INTRODUCTION

SPARSE modeling of signals has recently received much attention as it has shown promising results in different applications. A basic assumption to apply this model is that the given class of signals can be sparsely represented or approximated in an underdetermined linear generative model. In this framework, one can use a matrix $\mathbf{D}_{d\times N} \in \mathbb{R}^{d\times N} : d < N$, called dictionary, to represent the signal approximately using $\mathbf{y} \approx \mathbf{D}\mathbf{x}$. Let $\mathbf{y} \in \mathbb{R}^d$ and $\mathbf{x} \in \mathbb{R}^N$ be the given signal and the coefficient vector respectively. A sparse approximation would be,

$$\widehat{\mathbf{x}} = \arg\min_{x} \|x\|_0 \;\; \mathrm{s.\,t.} \;\; \|\mathbf{y} - \mathbf{D}\mathbf{x}\|^2 \le \xi, \qquad (1)$$

where $\|.\|_0$ is the sparsity measure that counts the number of the non-zero coefficients and ξ is a small positive constant. Because this problem is generally NP-hard, numerous algorithms have been proposed to find an approximate solution. The sparsity of the approximation is increased using an appropriate dictionary for the given class of signals. A dictionary often is selected by concatenating orthogonal bases [3] or using a tight frame [4]. These dictionaries can be improved by dictionary learning methods, see [5] and references therein. These methods adapt an initial dictionary to a set of training samples. Therefore the aim is to *learn* a dictionary for which an input signal, taken from a given class of signals, has a sparse approximation.

This research was fully supported by the UK's EPSRC, grant number D000246/1. MED acknowledges support of his position from the Scottish Funding Council and their support of the Joint Research Institute with the Heriot-Watt University as a component part of the Edinburgh Research Partnership.

There is another dictionary selection method, which is called dictionary *design*. Different methods exist to design a suitable $\mathbf{D}$ for a set of natural signals. One method is based on a generative model of the signals. Alternatively, if these signals are to be received by the human sensory system, a more effective method to design $\mathbf{D}$ is to use a human perception model [1], [2]. Here, we assume that the set of elementary functions, which are generated by the proposed model, can be described by using a set of parameters and a parametric function. For example, in the multiscale Gabor functions, the parameters are scale, time and frequency shifts and the parametric function is Gaussian. In general the parameters are in the continuous domain. To generate a dictionary based on these generative functions, we can sample these continuous parameters. The question is then how best to sample the parameters. Several researchers have introduced different methods to optimize the sampling process. In [6], a sampling scheme was introduced which finds an approximately tight frame, using 2D Gabor functions. Alternatively, some researchers optimized the parameters based on the closeness to what is observed in the perceptual systems. In practice, [7] showed that the optimal Gammatone parameters, found by fitting to the human auditory system, do not match the parameters estimated from English speech signals.

When we use an approximate or a relaxed method to find a sparse approximation, having an exact generative model does not guarantee that we find the best sparse approximation. An important parameter of a dictionary, for successful sparse recovery, is its coherence μ [8]. The coherence is defined as the absolute value of the largest inner-product of two distinct atoms and it has been shown that when μ is smaller than a certain threshold MP and BPDN can recover the sparse representation of the input signal [9]. It has also been shown that the coherence upper-bounds the residual error decay in MP [10] and OMP [8]. Therefore a dictionary with small μ is desirable for sparse coding. Let $\mathbf{G} := \mathbf{D}^T\mathbf{D}$ be the Gram matrix of the dictionary. The coherence of $\mathbf{D}$ is the maximum absolute value of the off-diagonal elements of $\mathbf{G}$, whenever the columns of the dictionary are normalized. For such $\mathbf{D}$ if the magnitude of all off-diagonal elements of $\mathbf{G}$ are equal, $\mathbf{D}$ has minimum coherence [11]. This normalized dictionary is called an Equiangular Tight Frame (ETF) [12]. Although this type of frame has various nice properties, we mainly consider its advantages in the exact atom recovery [8] and the residual error decay rate [10]. Unfortunately ETF's do not exist for any arbitrary selection of d and N [12]. Therefore a dictionary design aim can be to find the nearest admissible solution. On the other hand, natural signals do not generally have sparse approximations using an ETF.

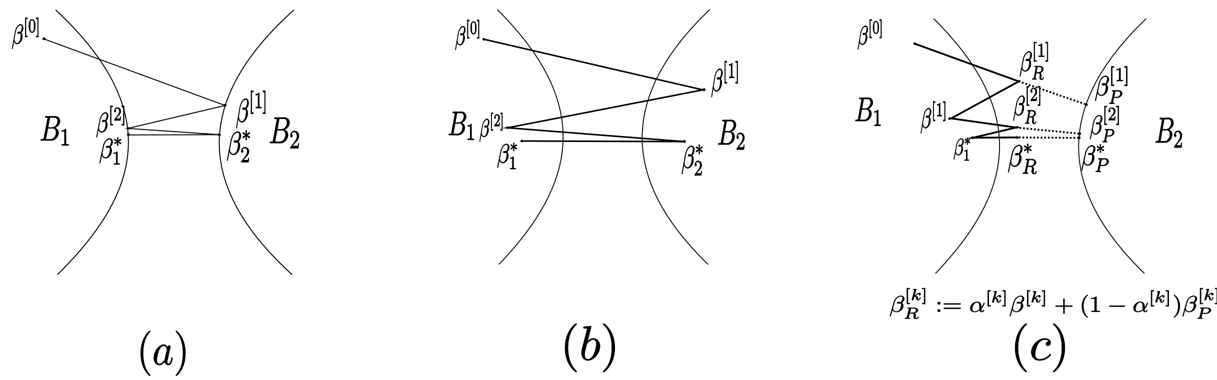

Fig. 1. Different alternating optimization methods: (a) Alternating Projection, (b) Alternating Minimization and (c) Proposed Method.

Therefore, the dictionary design problem can be to find a parametric dictionary whose Gram matrix is close to being the Gram matrix of an ETF. This way, domain knowledge is incorporated into the parametric functions and the initial parameters, while the optimization aims at improving the ability of algorithms to find sparse approximations. We expect to have a sparse approximation for the given class of signals using the proposed dictionary. That is because it is generated by sampling the parameters of generative functions fitted to the signal, whilst the dictionary has nice properties that allow exact atom recovery, because it is close to being an ETF. In practice we show that the designed dictionary indeed gives advantages over the standard dictionary, in terms of efficient sparse approximation. Another advantage of the parametric dictionary is that sparse approximation methods only need to store the parameters, instead of the full dictionary, which offers a huge reduction in memory requirement (the size of parameter matrix is much smaller than the size of the corresponding dictionary).

The parametric dictionary design also has some disadvantages. The method is explicitly not a data dependent method. Another difficulty in the given problem is that the current algorithm stores the Gram matrix explicitly. Therefore for a very large block of signal, the current method is not tractable.

A. Contributions of the paper

In this paper we introduce a new framework for dictionary design. To the authors knowledge, this formulation has not been considered previously. This formulation can be used to design a dictionary when dictionary learning is not possible, or is computationally intractable. We show how we can find an approximate solution using an alternating minimization type method.

The parametric dictionary is represented using a small number of parameters (often less than 5). Therefore we do not need to store the dictionary explicitly. This can save a considerable amount of memory when using sparse approximation algorithms.

Finally we show experimentally that there are sparse approximation benefits in using such a parametric dictionary for audio coding.

II. Parametric Dictionary Design: Formulation

In this section we formulate the parametric dictionary design as an optimization problem. Let $\mathbf{D}_\Gamma \in \mathscr{D}$ be a parametric dictionary. Γ is the parameter matrix, with γ_i as its i^{th} column and $\mathscr{D}$ is the set of admissible parametric dictionaries. In this paper, by letting $\mathbf{D}_\Gamma$ be a matrix with the atoms $\mathbf{d}_i$ (with the associated parameters γ_i), we implicitly assume that the generative model is discrete. To select a $\Gamma \in \Upsilon$, where Υ is an admissible parameter set, we can optimize an objective. In section I we explained that for a better performance in sparse coding, we are interested to design a dictionary which is close to being an ETF. For a given normalized $\mathbf{D}$, the coherence of $\mathbf{D}$, $\mu_{\mathbf{D}}$, is defined by

$$\mu_{\mathbf{D}} = \max_{i,j:j\neq i}\{|\langle \mathbf{d}_i, \mathbf{d}_j\rangle|\}. \tag{2}$$

A column normalized dictionary $\mathbf{D}_G$ is called ETF, when there is a $\gamma \ : \ 0 < \gamma < \pi/2$.

$$|\langle \mathbf{d}_i, \mathbf{d}_j\rangle| = \cos(\gamma) \ : \ \forall i,j \ i \neq j \tag{3}$$

Strohmer et. al. in [13] showed that if there exists an ETF in $\mathscr{D}$, here the set of d by N uniform frames[1], it is the solution of,

$$\arg\min_{\mathbf{D}\in\mathscr{D}}\{\mu_{\mathbf{D}}\}. \tag{4}$$

To study the lower bound of μ_D, the existence of an ETF and its Gram matrix, [13] introduced a theorem which shows that when $\mathbf{D} \in \mathbb{R}^{d\times N}$ is a uniform frame, $\mu_{\mathbf{D}}$ is lower bounded by,

$$\mu_{\mathbf{D}} \geq \mu_G := \sqrt{\frac{N-d}{d(N-1)}}. \tag{5}$$

Equality holds in (5) *if and only if* $\mathbf{D}$ is an ETF. Furthermore, equality in (5) can only hold if $N \leq \frac{d(d+1)}{2}$.

Let Θ_d^N be the set of Gram matrices of all $d \times N$ ETF's. If $\mathbf{G}_G \in \Theta_d^N$ then the diagonal elements and the absolute values of the off-diagonal elements of $\mathbf{G}_G$ are one and μ_G respectively. A nearness measure of $\mathbf{D} \in \mathbb{R}^{d\times N}$ to the set of ETF's can be defined as the minimum distance between

[1] A frame with unit column norms

Algorithm 1 *Parametric Dictionary Design*

1: **initialization:** $k=1$, $\mathbf{D}_{\Gamma_1} \in \mathscr{D}$, $\{\alpha_i\}_{1\le i\le K} : 0<\alpha_i\le 1$
2: **while** $k \le K$ **do**
3: $\quad \mathbf{G}_{\Gamma_k} = \mathbf{D}_{\Gamma_k}^T \mathbf{D}_{\Gamma_k}$
4: $\quad \mathbf{G}_{P_{k+1}} = \min_{\mathbf{G}\in\Lambda^N} \|\mathbf{G}_{\Gamma_k} - \mathbf{G}\|_F$
5: $\quad \mathbf{G}_{R_{k+1}} = \alpha_k \mathbf{G}_{P_{k+1}} + (1-\alpha_k)\mathbf{G}_{\Gamma_k}$
6: $\quad \mathbf{D}_{\Gamma_{k+1}} \in \mathbf{D}_{\Gamma_k} \cup \{\forall \mathbf{D} \in \mathscr{D} : \|\mathbf{D}^T\mathbf{D} - \mathbf{G}_{R_{k+1}}\|_F < \|\mathbf{G}_{\Gamma_k} - \mathbf{G}_{R_{k+1}}\|_F\}$
7: $\quad k = k+1$
8: **end while**

the Gram matrix of $\mathbf{D}$ and $\mathbf{G}_G \in \Theta_d^N$ [11]. To optimize the distance of a dictionary to an ETF, we can solve,

$$\min_{\Gamma\in\Upsilon, \mathbf{G}_G\in\Theta_d^N} \|\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma - \mathbf{G}_G\|_\infty, \tag{6}$$

where the matrix operator $\|.\|_\infty$ is defined as the maximum absolute value of the elements of the matrix. Instead, we would like to use a different norm space which simplifies the problem. An advantage of using ℓ_2 measure in the given problem is that it considers the errors of all elements (and not only the maximum absolute error). In this framework, when there is no ETF in $\mathscr{D}$, we find a dictionary that is close to be quasi-incoherent [8], [10]. Therefore we use the following formulation,

$$\min_{\Gamma\in\Upsilon, \mathbf{G}_G\in\Theta_d^N} \|\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma - \mathbf{G}_G\|_F^2, \tag{7}$$

where $\|.\|_F$ is the Frobenius norm. This is a non-convex optimization problem in general. It might have a set of solutions or not have any solution (e.g. Θ_d^N is empty as there do not always exist ETF's for arbitrary N and d). One can extend Θ_d^N to a convex set Λ^N [11], which is non-empty for any N, by

$$\Lambda^N = \{\mathbf{G} \in \mathbb{R}^{N\times N} : \mathbf{G} = \mathbf{G}^T, \operatorname{diag}\mathbf{G} = 1, \max_{i\neq j} |g_{i,j}| \le \mu_G\}. \tag{8}$$

Relaxing (7), by replacing Θ_d^N with Λ^N, gives the following optimization problem.

$$\min_{\Gamma\in\Upsilon, \mathbf{G}\in\Lambda^N} \|\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma - \mathbf{G}\|_F^2 \tag{9}$$

An important difference between (7) and (9) is that the relaxed problem is guaranteed to have at least one solution. We therefore use the relaxed formulation from now on. We show experimentally that the approximate solutions of (9), even though the Gram matrix of the dictionary might only be close to Λ^N, show good performances in sparse approximation.

In the next section we introduce a practical method to find an approximate solution to (9). Our approach has similarities with alternating minimization. This method is guaranteed not to increase the objective function in each step. Because the objective is non-negative, the algorithm is stable due to Lyapunov's second theorem. One can also show that the objective function converges. The stability of the algorithm and the convergence of the objective function do not prove the convergence of the algorithm. The conditions under which the algorithm converges to a set of accumulation points are discussed in Theorem 1. We present a sketch of proof for this theorem and refer the reader to [14] for further details.

Algorithm 2 *Parameters Update*

initialization: $l=1$, $1\le L$, $\Gamma_k^{[0]} = \Gamma_k$, $\epsilon \in \mathbb{R}^+$, $\phi(\Gamma) = \|\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma - \mathbf{G}\|_F^2$
for all $l \le L$ **do**
$\quad \Gamma_{k+1}^{[l+1]} = \Gamma_k^{[l]} - \epsilon \nabla_\Gamma \phi|_{\Gamma_k^{[l]}}$
$\quad l = l+1$
end for
$\Gamma_{k+1} = \Gamma_{k+1}^{[L]}$

III. Parametric Dictionary Design: A Practical Algorithm

A standard method to solve (9) is alternating projection. In this method we alternatingly project the current solution onto the admissible sets, see Fig.1.a. In a finite dimensional setting when the admissible sets are convex, the algorithm converges to a solution in $\mathscr{D} \cap \Lambda^N$ and when $\mathscr{D} \cap \Lambda^N = \emptyset$ to a pair of solutions in $\mathscr{D}$ and Λ^N respectively. In the following, we derive a formulation for the projection onto Λ^N, but there is no easy formulation for the projection onto the set of admissible dictionaries, in general. Therefore we choose a different method which has similarities with alternating minimization, see Fig.1.b. In the alternating minimization framework, we choose the new solutions in $\mathscr{D}$ and Λ^N alternatingly such that the objective does not increase in each update and is thus stable. If the algorithm converges, the fixed point is either in $\mathscr{D} \cap \Lambda^N$, or is a pair of points in $\mathscr{D}$ and Λ^N respectively.

Although the proposed algorithm has similarities with alternating minimization, it does not follow its steps exactly. The difference is that in the stage in which we update the current solution with respect to Λ^N, we choose a point which is somewhere between the current solution and the projection onto Λ^N. Fig.1.c shows a schematic representation of the proposed method. The reason for this modification is that by projection onto Λ^N, the structure of the Gram matrix changes significantly so that the selection of a new point in $\mathscr{D}$ in the following step is very difficult. We can gradually select a closer point to the projected point on Λ^N, when the current $\mathbf{D}_\Gamma$ is close to Λ^N. In the other step, we update $\mathbf{D}$ such that it does not increase the objective in (9).

The parametric dictionary design is summarized in Algorithm 1. In line 4, the algorithm finds the projection onto Λ^N. In line 6, a point in $\mathscr{D}$ is selected which is closer to $\mathbf{G}_{R_{k+1}}$. In the following we show how we calculate the updates in lines 4 and 6.

A. Projection onto Λ^N:

In the objective function (9), $\mathbf{G}$ is a Hermitian matrix. By sign change of any related off-diagonal pair of elements , i.e. $g_{i,j}$ and $g_{j,i}$, we get a new $\widetilde{\mathbf{G}} \in \Lambda^N$. The closest $\mathbf{G}$ to $\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma$, in a Frobenius norm space, is then the $\mathbf{G}$ with a similar sign pattern. For a given $\mathbf{G}_\mathbf{D} = \mathbf{D}^T\mathbf{D} : \mathbf{D} \in \mathbb{R}^{d\times N}$, the projection of $\mathbf{G}_D$ onto Λ^N can be found by the following operator [11],

$$g_{P\,i,j} = \begin{cases} \text{sign}(g_{D\,i,j})\mu_G & i \neq j \\ 1 & otherwise \ , \end{cases} \tag{10}$$

where μ_G is as defined in (5). This operator can be used to find $\mathbf{G}_{P_{k+1}}$ in line 4 of Algorithm 1, by applying into $\mathbf{G}_{\Gamma_k}$.

B. Parameter update:

Let us assume $\mathbf{D}_\Gamma$ is a differentiable function on Υ and therefore (9) is a differentiable function on Υ. An easy way to find Γ_{k+1}, such that it satisfies line 6 of the Algorithm 1, is to use the gradient descent method. We rewrite (9) as a minimization problem based on Γ when $\mathbf{G}_{R_{k+1}}$ is fixed.

$$\min_{\Gamma \in \Upsilon} \phi(\Gamma) \ , \ \phi(\Gamma) := \|\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma - \mathbf{G}_{R_{k+1}}\|_F^2 \tag{11}$$

The gradient of the objective function in (11) can be found by chain rule for the matrix functions [15, D.1.3].

$$\begin{aligned} \nabla_\Gamma \phi &= \nabla_\Gamma \mathbf{D}_\Gamma \ \nabla_{\mathbf{D}_\Gamma} \phi \\ &= 4\nabla_\Gamma \mathbf{D}_\Gamma \ (\mathbf{D}_\Gamma \mathbf{D}_\Gamma^T \mathbf{D}_\Gamma - \mathbf{D}_\Gamma \mathbf{G}_{R_{k+1}}) \end{aligned} \tag{12}$$

We iteratively use the gradient descent method to find a *local* minimum of the problem (11). Let $\Gamma_k^{[0]} = \Gamma_k$, the updating formula is as follows,

$$\Gamma_{k+1}^{[l+1]} = \Gamma_k^{[l]} - \epsilon \nabla_\Gamma \phi|_{\Gamma_k^{[l]}}, \tag{13}$$

where ϵ is a small positive value. In this framework, $\Gamma_{k+1} = \lim_{l\to\infty} \Gamma_{k+1}^{[l]}$. In practice we stop after a given number of iterations or when $\epsilon \nabla_\Gamma \phi|_{\Gamma_k^{[l]}}$ becomes very small. Algorithm 2 summarizes this parameter update algorithm.

The convergence of Algorithm 1 is guaranteed by the following theorem.

Theorem 1: [14, Theorem 3] Let $\mathbf{D}_\Gamma$ be differentiable. The Algorithm 1 converges to a set of fixed points by starting from $\Gamma_0 \in \Upsilon$, where Υ is a compact set.

We only present a sketch of proof in this paper. We first show that the algorithm reduces the distance of $\mathbf{G}_{\Gamma_k}$ to $\mathbf{G}_k$ in each parameter update. We then show that the objective function of (9) is a continuous function of Γ, which implies the compactness of the solution space. The proof of Theorem 1 is completed by applying Bolzano-Weierstrass theorem, which guarantees existence of at least one accumulation point for the sequences of dictionaries $\{\mathbf{D}_{\Gamma_k}\}_{k\in\mathbb{N}}$. Line 6 of Algorithm 1 prevents the existence of a continuum of accumulation points. Therefore, the accumulation points are fixed points.

IV. Case study

The problem we formulated in this paper is developed in a general form. To show the advantages of using parametric dictionary design in practice, we choose a case study. In sparse audio processing, an important question is how to choose the dictionary [16], [17]. We show that the parametric dictionary design improves the performance of audio sparse approximation and exact recovery based around a Gammatone representation.

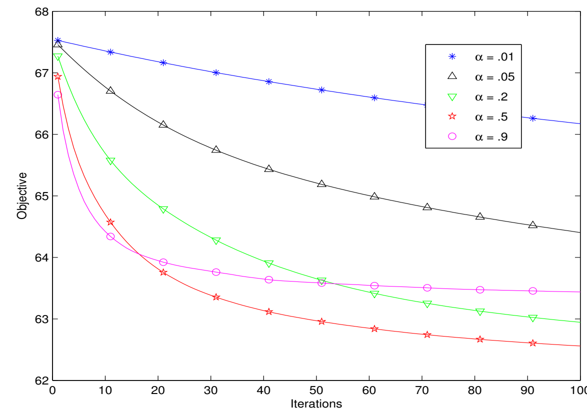

Fig. 2. The objective functions for different $\{\alpha_k\}_{\forall k, \alpha_k = \alpha}$, for a constant α.

A. Gammatone parametric dictionary

The generative function for a Gammatone dictionary is as follows,

$$g(t) = at^{n-1}e^{-2\pi bBt}\cos(2\pi f_c t) \tag{14}$$

where $B = f_c/Q + b_{min}$, f_c is the center frequency and $n \in \mathbb{N}$, a, b, Q, b_{min} are some constants. The optimal parameter selection is not easy. The dictionary is often generated by sampling the parameters of $g(t - t_c)$, where t_c is the time-shift. Here, $\gamma = [t_c \ f_c \ n \ b]^T$ are the optimization parameters. The parameters t_c and f_c change the center of the atoms in the time-frequency plane. n and b control the rise time and the width of the atoms in the time domain, respectively. The parameter a is chosen to normalize the atom to unit length. Let $\{\gamma_i\}_{1\leq i \leq N}$ be a set of the parameters and $g_{\gamma_i}(t)$ be the atom generated using γ_i. The parameter matrix Γ and the parametric dictionary $\mathbf{D}_\Gamma$ are generated using γ_i and $g_{\gamma_i}(\lfloor t f_{samp} \rfloor)$ as the columns respectively, where f_{samp} is the sampling frequency.

To use the gradient discent method for parameter update, $\mathbf{D}_\Gamma$ should be differentiable with respect to Γ. We can extend (14) to a more general function using $n \in \mathbb{R}$. This function is differentiable with respect to Γ. We can choose an upper bound for the magnitude of each parameter to generate a bounded admissible set. By including the boundary values, Υ is a compact set that guarantees convergence of the algorithm to a set of fixed points. A necessary modification in Algorithm 1 is to use a mapping to Υ, when at least one parameter goes out of Υ, and comparing to the previous solution (to make sure that we do not increase the objective by the parameter update). A simple mapping operator is the thresholding operator, where it chooses the closest admissible parameter.

B. Simulations results

We study the proposed dictionary design method using the Gammatone dictionary discussed in IV. We first investigate the characteristics of the dictionaries throughout the design iterations. We then compare the performance of the initial and the optimized dictionaries in terms of sparse approximation and exact sparse recovery. In all the simulations we choose two times overcomplete dictionaries and window size 1024.

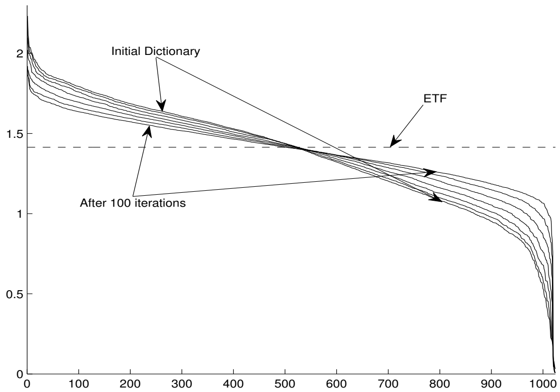

Fig. 3. Eigen values plot of the dictionary.

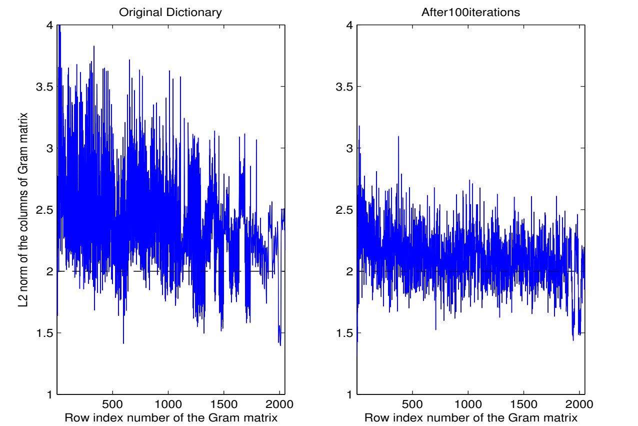

Fig. 4. The column ℓ_2 plots of the Gram matrix of the original (left) and designed (right) dictionaries.

1) Algorithm Evaluation: We evaluate the given algorithm in three different areas. In the first step we show that the algorithm reduces, (or at least keep the same) the objective (9) in each iteration. The parameter B, defined after (14), is the bandwidth of the audio filterbank at the center frequency f_c. We used the fixed values $n = 4$, $Q = 9.26449$, $b_{min} = 24.7$, as they have been suggested in [18], and $b = 0.65$. To generate the initial dictionary, we sampled f_c and t_c. In the method introduced in [19], an extra parameter δ, called step factor, is introduced to indicate the amount of frequency overlap. In this framework the k^{th} frequency center is calculate using the following formula.

$$f_c^k = -Qb_{min} + (f_s/2 + Qb_{min})e^{-k\delta/Q} \tag{15}$$

f_s is the maximum allowed frequency, which is half of the Nyquist frequency. In our simulations, we choose $\delta = 0.45$. We have chosen a similar method to sample t_c. This time sampling is linear, in contrast with the logarithmic sampling in (15). Let the peak of the envelope of the impulse response of the filter be at t_p and σ indicate the amount of time overlap. The l^{th} time center is found using,

$$t_c^l = t_p + \sigma(l-1)\ t_p\ , \tag{16}$$

where $\sigma = 0.75$.

To generate a dictionary of $g_{\gamma_i}(t)$, we windowed it to a size equal to the signal length d and made it periodic such that one period is selected as an atom by using the following formula,

$$\mathbf{d}_{\gamma_i,j} = \begin{cases} g_{\gamma_i(j+d)} & 1 \le j < j_{c_i} \\ g_{\gamma_i(j)} & j_{c_i} \le j \le d, \end{cases} \tag{17}$$

where $j_{c_i} = \lfloor t_{c_i} f_{samp} \rfloor$. We choose a simple sequence of $\{\alpha_k\}$ using $\alpha_k = \alpha$ for all k and a constant α in all simulations. A more complicated sequence might improve the performance of Algorithm 1. However we have not present this here. Instead, we intend to show that the proposed algorithm works in practice, even with a simple $\{\alpha_k\}$. In the first experiment we want to investigate the effect of α. We have plotted the objective function (9) using selected α's, in Fig. 2. As we expect, simulations show reduction of the proposed objectives in each iteration. It is also demonstrated that if α is small, the algorithm converges very slowly. Although using a large α is desireable for a fast convergence, the solution is not as good as the solution found by using a medium range α. For other simulations we use $\alpha = 0.5$ to find a good solution after an acceptable number of iterations.

The proposed algorithm searches for an equiangular *tight frame*. Therefore one way to show the performance of the proposed algorithm is to compare the singular values (SV) of the designed dictionary and the tight frame. A tight frame in $\mathbb{R}^{d\times N}$ has d non-zero SV equal to $\sqrt{N/d}$. We have plotted the sorted SV's of the dictionaries at selected iterations in Fig. 3. It can be seen that the SV's of the designed dictionary get closer to the SV's of the tight frame after each selected number of iterations.

Given that the algorithm is based on distances in the Gram matrix domain, another way to evaluate the algorithm is to show the Gram matrix of the dictionary. We have plotted the ℓ_2 norm of each row of the Gram matrix in Fig. 4. The Gram matrix of the original dictionary and the designed dictionary, after 100 iterations, are shown in the left and right windows respectively. We have shown the ℓ_2 norm of a possible ETF with a dashed line as reference. It can be seen that the Gram matrix of the designed dictionary is closer to the desired Gram matrix.

2) Exact sparse recovery and sparse approximation: In this part we demonstrate the advantages of the parametric dictionary design in terms of exact sparse recovery [8] and sparse approximation. In the first experiment we generate sparse coefficient vectors, with different sparsity, and plot the percentages of the exact recovery for those sparse vectors.

The location of the non-zero coefficients were selected uniformly at random and the PDF of the magnitudes were selected to be Gaussian with zero mean. The Matching Pursuit (MP) algorithm was used to find the sparse approximation. The rate of exact support recovery is calculated by the ratio of the number of correctly found non-zero coefficient places to the number of cases in which at least one location of the zero coefficient was set to be non-zero. We ran the simulations 1000 times. We have shown this ratio as the percentage of

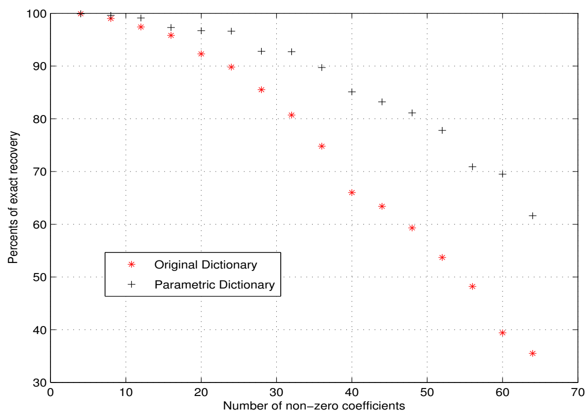

Fig. 5. Exact support recovery of the sparse signals.

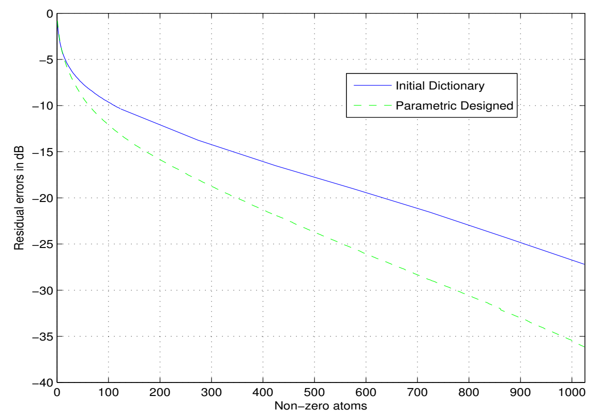

Fig. 6. The residual error using matching pursuit for sparse approximation of the audio signal.

exact recovery in Fig. 5. It is clear that the design method has improved the exact recovery ratio.

For sparse approximation applications, we are more interested to have a dictionary that, if it fails to satisfy exact recovery condition [8], still gives a sparse approximation for a given class of signals. Therefore as the second experiment, we compare the decay rates of the residual error when the MP is used for sparse approximation [10]. We used an audio signal taken from more than 8 hours recorded from BBC Radio 3, which mostly plays classic music. We first down-sampled by a factor of 4 and summed the stereo channels to make a mono signal with 12K samples per second. We used the original Gammatone and the parametric designed dictionaries for 100 blocks, each with the length of 1024 samples. The average decay rate of the residual errors, in logarithmic scales, are shown in Fig. 6. This rate directly influences the performance of sparse approximation methods. That is, we can better approximate the signals with fewer coefficients using a high residual error decay rate dictionary. In Fig. 6, although the curves start with the same slope, after a few iterations, here 10, the designed dictionary shows a clear advantage.

V. Conclusion

We have introduced a signal independent dictionary design method. A parametric function, which is closely related to the given class of signals, was used to design a minimal coherence dictionary. In this framework we have shown that the dictionary design problem is to find an optimal set of parameters. This problem can in general not be solved exactly. Fortunately an approximate solution can be found using the proposed method. In some simulations we showed that A) the given method can find an appropriate set of parameters for the given case study and B) the designed dictionary showed promising performance advantages in terms of exact recovery and sparse approximation. The proposed framework can be extended to include extra constraints, such as to be shift-invariance, quasi-incoherence, data dependence, to have tree structures or structures for fast implementations. That has been left for future work.

References

[1] R. Patterson, M. Allerhand, and C. Giguere, "Time-domain modeling of peripheral auditory processing: a modular architecture and a software platform," *Journal of the Acoustical Society of America*, vol. 98, no. 4, pp. 1890–1894, 1995.
[2] J. Daugman, "Two-dimensional spectral analysis of cortical receptive field profile," *Vision Research*, vol. 20, pp. 847–856, 1980.
[3] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Trans. Information Theory*, vol. 49, no. 12, pp. 3320 – 3325, 2003.
[4] E. Candes and L. Demanet, "The curvelet representation of wave propagators is optimally sparse," *Communications on Pure and Applied Mathematics*, vol. 58, no. 11, pp. 1472–1528, 2005.
[5] M. Yaghoobi, T. Blumensath, and M. Davies, "Dictionary learning for sparse approximations with the majorization method," accepted for publication in IEEE Trans. on Signal Processing.
[6] T. Lee, "Image representation using 2d gabor wavelets," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, 1996.
[7] S. Strahl and A. Mertins, "Sparse gammatone signal model optimized for english speech does not match the human auditory filters," *Brain Research*, vol. 1220, pp. 224–233, 2008.
[8] J. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Trans. on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.
[9] A. Gilbert, S. Muthukrishnan, and M. Strauss, "Approximation of functions over redundant dictionaries using coherence," in *Proc. 14th Annu. ACM-SIAM Symp. on Discrete algorithms, Baltimore, MD*, 2003.
[10] R. Gribonval and P. Vandergheynst, "On the exponential convergence of matching pursuits in quasi-incoherent dictionaries," *IEEE Trans. on Information Theory*, vol. 52, no. 1, pp. 255–261, 2006.
[11] J. Tropp, I. Dhillon, R. Heath Jr., and T. Strohmer, "Designing structural tight frames via an alternating projection method," *IEEE Trans. on Information Theory*, vol. 51, no. 1, pp. 188–209, 2005.
[12] M. Sustik, J. Tropp, I. Dhillon, and R. Heath, "On the existence of equiangular tight frames," *Linear Algebra and Its Applications*, vol. 426, no. 2-3, pp. 619–635, 2007.
[13] T. Strohmer and R. Heath, "Grassmannian frames with applications to coding and communication," *Applied and Computational Harmonic Analysis*, vol. 14, no. 3, pp. 257–275, 2003.
[14] M. Yaghoobi, L. Daudet, and M. Davies, "Parametric dictionary design for sparse coding," 2009, submitted.
[15] J. Dattorro, *Convex Optimization and Euclidean Distance Geometry*. Meboo Publishing, 2005 (v2007.09.17).
[16] M. Davies and L. Daudet, "Sparse audio representations using the MCLT," *Signal Processing*, vol. 86, no. 3, pp. 457–470, 2006.
[17] R. Pichevar, H. Najaf-Zadeh, and L. Thibault, "A biologically-inspired low-bit-rate universal audio coder," in *Audio Engineering Society Convention, Vienna, Austria*, 2007.
[18] M. Slaney, "Lyon's cochlear model," Apple Computer, Tech. Rep., 1988.
[19] ——, "An efficient implementation of the Patterson-Holdsworth auditory filter bank," Apple Computer, Tech. Rep., 1993.