Parametric dictionary design for sparse coding

Parametric Dictionary Design for Sparse Coding

Mehrdad Yaghoobi, *Member, IEEE,* and Laurent Daudet, *Member, IEEE* and Mike E. Davies, *Member, IEEE*

***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 constraint on the dictionary. Then we briefly explain that equiangular tight frames have minimum coherence. 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. It has been used for coding, source separation, feature extraction and compressive sampling. A basic assumption to apply this model is that the given class of signals can be sparsely represented or approximated in an underdetermined generative model. Often, a linear model has been used as the 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 $\mathbf{y} \approx \mathbf{D}\mathbf{x}$.

Sparse approximation and sparse representation methods have been studied theoretically and practically [3]. 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,

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \ \text{s.t.} \ \|\mathbf{y} - \mathbf{D}\mathbf{x}\|^2 \leq \xi,$$

where $\|.\|_0$ is the sparsity measure that counts the number of the non-zero coefficients and ξ is a small positive constant. This problem in general, like the sparse representation problem ($\xi = 0$), is an NP-hard problem [4] and can not be solved in a reasonable time. Numerous algorithms have been proposed to find an approximate solution. These algorithms are classified as greedy methods, like Matching Pursuit (MP) [5] and its derivations [6], and relaxation methods, like Basis Pursuit Denoising (BPDN) [7] and IRLS-type algorithms [8], [9]. The sparsity of the representation can be increased using an appropriate dictionary for the given class of signals. The common methods for dictionary selection are to concatenate orthogonal bases, see for example [10] and [11] for the possible advantages of using such a dictionary in theory and practice, or to use a tight frame [12]. These dictionaries can be improved using dictionary learning methods [13]–[16]. 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.

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. 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]. In fact, the stimuli responses generate elementary functions which are more related to the analysis dictionary [17]. These elementary functions have also been used for generating the synthesis dictionary $\mathbf{D}$. Here, we assume that the set of elementary functions can be described by using a set of parameters and a parametric function. For example, in the multiscale Gabor functions [5], 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 [18], a sampling scheme was introduced which finds an approximately tight frame, using 2D Gabor functions. Gammatone and Gammachirp filter banks have been shown to approximate the human auditory system. [19] presented two types of filters, which approximate the Gammatone filter banks, and allow a possible fast VLSI implementations. Alternatively, some researchers optimized the parameters based on the closeness to what is observed in the perceptual systems [20], [21], [22]. In practice, [23] showed that the optimal 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 [24]. The coherence is defined as the absolute value of the largest inner-product of two

This research was partially supported by EPSRC grant D000246/1 and EU FP7, FET-Open grant number 225913. M. Yaghoobi and M. Davies are with the Institute for Digital Communication and with the Joint Research Institute for Signal and Image Processing, Edinburgh University, Kings Buildings, Mayfield Road, Edinburgh EH9 3JL, UK (e-mail: yaghoobi@ieee.org, mike.davies@ed.ac.uk).

L. Daudet is with UPMC Univ. Paris 06, LAM / Institut Jean le Rond d'Alembert (UMR 7190), Paris, France (email: daudet@lam.jussieu.fr).

distinct atoms and it has been shown that when μ is smaller than a certain value MP and BPDN can recover the sparse representation of the input signal [10], [25], [26]. It has also been shown that the coherence upper-bounds the residual error decay in MP [27] and OMP [25]. 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 [28]. This normalized dictionary is called an Equiangular Tight Frame (ETF) [29]. Although this type of frame has various nice properties, we here consider its advantages in exact atom recovery [25] and the residual error decay rate [27]. Unfortunately ETF's do not exist for any arbitrary selection of d and N [29]. 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. Therefore, the dictionary design problem can be to find a parametric dictionary whose Gram matrix is close to the Gram matrix of an ETF. This way, domain knowledge is incorporated into the parametric functions used, while the optimization aims at improving the ability of algorithms to find sparse approximations. The given class of signals has a sparse approximation 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 the parameter matrix is much smaller than the size of the corresponding dictionary). Sometimes this type of parametric dictionary can furthermore be multiplied to the coefficient vectors faster than direct matrix-vector multiplication. It then also speeds up most of the currently available sparse coding methods.

The parametric dictionary design, like other dictionary design methods, has some disadvantages. The main disadvantage is that it does not explicitly depend on a given class of signals, but instead on a class of parametric dictionaries. As an example, if the actual data often lies in a subspace of the signal space, the optimal dictionary[1] would have more atoms in that subspace. This might contradict with the minimum coherence constraint. It is hoped that this can be prevented by appropriate choice of the parametric family of functions and the initialization of the algorithm. Another difficulty in the given problem is that the current algorithm stores the Gram matrix explicitly. The current method is thus not tractable for very large dictionaries.

It deserves to be mentioned that there is another way to use parametric dictionaries. In [30], [31] some methods are proposed to sparsely approximate signals using continuous parameter parametric dictionaries. The convergence rate of MP algorithm with this setting is also studied in [31]. In contrast the designed dictionary, using parametric dictionary design, is discretized and can be used by the conventional sparse coding methods.

[1]The optimal dictionary is that by which the given class of signals has the sparsest approximation.

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.

B. Organization of the paper

In the next section we formulate the parametric dictionary design problem. We then present a practical algorithm to find an approximate solution. For a case study we present the parametric dictionary formulation and the update formula derivation. Experiments, in the simulation subsection, show the advantages of the proposed dictionary design. The stability of the algorithm is analyzed after its introduction in Section III, while the convergence of the proposed algorithm is shown in Appendix A.

C. Notation

In this paper we use small and capital bold face characters to indicate vectors and matrices respectively. All the parameters have real values, even though we do not state this explicitly each time. The matrix and vector norm spaces that we use in this paper are defined over the real fields with ℓ_2 and $\|.\|_F$, which is the Frobenius norm, as the corresponding norms respectively.

The tensor product used in this paper is for the multiplication of two three-dimensional arrays. This multiplication uses the first two indices to make a simple matrix-matrix product and the third parameter as the indices of these products. In other words, the third parameter specifies two matices from the three dimensional tensors and simplifies the tensor product to matrix-matrix multiplications. The number of these multiplications is the size of third index.

The terms "ETF" and "Grassmannian Frame" have been used interchangingly for the same concept [32], [28], [29]. In this paper we prefer to use ETF, which is more comprehensive.

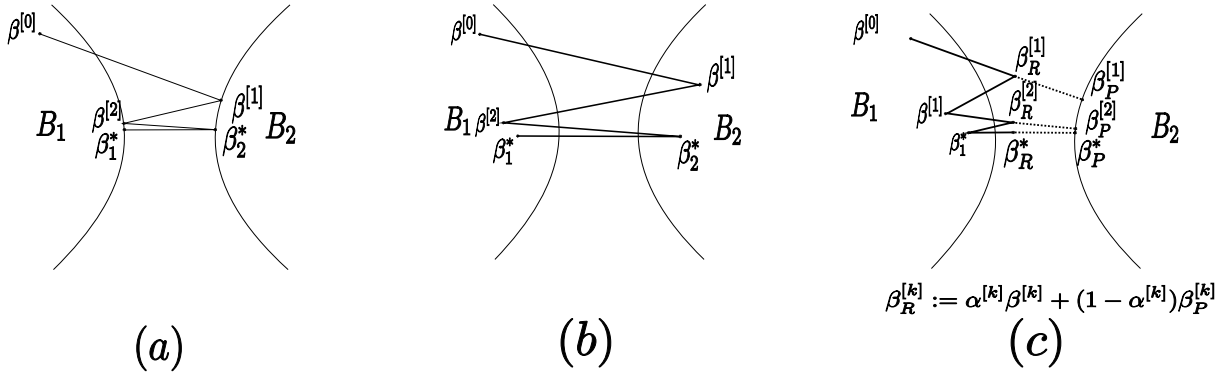

Fig. 1. Different alternating optimization methods: (a) Alternating Projection, (b) Alternating Minimization and (c) Proposed Method.

II. Parametric Dictionary Design: Formulation

In this section we formulate the problem of optimizing $\mathbf{D}$ to be close to an ETF. 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. Each column of $\mathbf{D}_\Gamma$, $\mathbf{d}_i$ (with the associated parameters γ_i), is called an atom. In this paper, by letting $\mathbf{D}_\Gamma$ be a matrix, we implicitly assume that the generative model is discrete. This model can be extended to a continuous model, which is out of our scope. To select a $\Gamma \in \Upsilon$, where Υ is an admissible parameter set, we need to introduce 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,

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

A column normalized dictionary $\mathbf{D}_G$ is called ETF, or Grassmannian frame [32], when there is a $\gamma \ : \ 0 < \gamma < \pi/2$, such that,

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

The authors in [32] showed that if there exists an ETF in $\mathscr{D}$, the set of d by N uniform frames[2], it is the solution of,

$$\arg\inf_{\mathbf{D}\in\mathscr{D}} \{\mu_\mathbf{D}\}.$$

The infimum has been used to guarantee that the problem has at least a solution, when $\mathscr{D}$ is not closed, which is in the closure of $\mathscr{D}$. To study the lower bound of μ_D, the existence of an ETF and its Gram matrix, [32] introduced the following Theorem.

Theorem 1: [32, Theorem 2.3] Let $\mathbf{D}$ be a uniform frame in $\mathbb{R}^{d\times N}$. Then,

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

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

Let Θ_d^N be the set of Gram matrices of all $d \times N$ ETFs. 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 ETFs can be defined as the minimum distance between the Gram matrix of $\mathbf{D}$ and $\mathbf{G}_G \in \Theta_d^N$ [28]. To optimize the distance of a dictionary to an ETF, we can solve,

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

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[3]. An advantage of using ℓ_2 measure in the given problem is that it considers the errors of all elements (and not just 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 [25] [27]. Therefore we use the following formulation,

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

where $\|.\|_F$ is the Frobenius norm. This is a non-convex optimization problem in general. It might have a set of solutions or it may not have any solution (e.g. Θ_d^N is empty as there do not always exist ETF's for the arbitrary N and d). One can extend Θ_d^N to a convex set Λ^N [28], which is non-empty for any N, by

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

Relaxing (2), by replacing Θ_d^N with Λ^N, gives the following optimization problem.

$$\inf_{\Gamma\in\Upsilon, \mathbf{G}\in\Lambda^N} \|\mathbf{D}_\Gamma^T\mathbf{D}_\Gamma - \mathbf{G}\|_F^2 \tag{3}$$

An important difference between (2) and (3) is that the relaxed problem, by using non-empty admissible sets, is guaranteed to have at least one solution. In this work, it is assumed that Υ is closed, which allows us to use the "min" operator instead of "inf" in (3). We therefore use the relaxed formulation from now on. We show experimentally that the approximate solutions of (3), even though the Gram matrix of the dictionary might only be close to Λ^N, show good performances in sparse approximation.

[2]A frame with unit column norms.

[3]Although the matrix space with ℓ_∞ is a well defined Banach space, here, we use ℓ_2 norm Hilbert space to use easy formulation of the optimization process.

Algorithm 1 *Parametric Dictionary Design*

1: **initialization:** $k=1$, $\mathbf{D}_{\Gamma_1} \in \mathscr{D}$, $\{\alpha_i\}_{1\leq i\leq K} : 0<\alpha_i\leq 1$
2: **while** $k \leq K$ **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**

In the next section we introduce a practical method to find an approximate solution to (3). 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[4] due to Lyapunov's second theorem [33]. Also, one can show that the objective value converges. The stability of the algorithm and the convergence of the objective value do not prove the convergence of the algorithm. In Appendix A, it has been show that the algorithm converges to a set of accumulation points under mild conditions.

III. Parametric Dictionary Design: A Practical Algorithm

A standard method to solve (3) is alternating projection, see for example [34], [28] and references therein. In this method we alternatingly project the current solution onto the admissible sets, see Fig.1.a. When the admissible sets are convex, the algorithm converges[5] to a solution in $\mathscr{D}\cap\Lambda^N$ or a pair of solutions in $\mathscr{D}$ and Λ^N, when $\mathscr{D}\cap\Lambda^N = \emptyset$, 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 [36] (or generalized alternating projection [37]), 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 (3).

[4]Here stability means boundedness of the algorithm output.

[5]At least in finite dimensional spaces. There are counter-examples for the lack of convergences in the infinite dimension setting [35].

Algorithm 2 *Parameters Update*

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

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 (3), $\mathbf{G}$ 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 the $\mathbf{G}$ with a similar sign pattern. We know that in a normed space, finding the nearest elements of a set to a point is a projection of that point onto the set. Because Λ^N is convex, the projection is unique. 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 [28].

$$g_{P\,i,j} = \begin{cases} \text{sign}(g_{D\,i,j})\mu_G & i \neq j \\ 1 & otherwise\ , \end{cases} \tag{4}$$

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

B. Parameter update:

Let us assume $\mathbf{D}_\Gamma$ is a differentiable function on Υ and therefore (3) 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 (3) as a minimization problem based on Γ when $\mathbf{G}_{R_{k+1}}$ is fixed.

$$\min_{\Gamma\in\Upsilon}\phi(\Gamma)\ ,\ \phi(\Gamma) := \|\mathbf{D}_\Gamma^T\mathbf{D}_\Gamma - \mathbf{G}_{R_{k+1}}\|_F^2 \tag{5}$$

The gradient of the objective function in (5) can be found by chain rule for the matrix functions [38, D.1.3].

$$\begin{aligned}\nabla_\Gamma\phi &= \nabla_\Gamma\mathbf{D}_\Gamma\ \nabla_{\mathbf{D}_\Gamma}\phi \\ &= 4\nabla_\Gamma\mathbf{D}_\Gamma\ (\mathbf{D}_\Gamma\mathbf{D}_\Gamma^T\mathbf{D}_\Gamma - \mathbf{D}_\Gamma\mathbf{G}_{R_{k+1}})\end{aligned} \tag{6}$$

In this formulation, one still needs to calculate $\nabla_\Gamma\mathbf{D}_\Gamma$. In Appendix B, we derive this formulation for a special parametric dictionary. We iteratively use the gradient descent method to find a *local* minimum of the problem (5). Let $\Gamma_k^{[0]} = \Gamma_k$, the updating formula is as follows,

$$\Gamma_{k+1}^{[l+1]} = \Gamma_k^{[l]} - \epsilon\nabla_\Gamma\phi\big|_{\Gamma_k^{[l]}}, \tag{7}$$

where ϵ is a small positive value. The parameter ϵ should be chosen such that the update reduces the objective function in

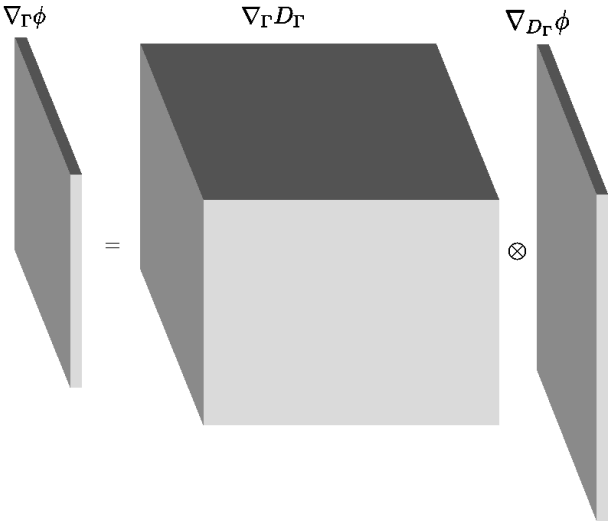

Fig. 2. The chain rule (6) in the tensor form.

(5) [39]. 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 $\nabla_\Gamma \phi|_{\Gamma_k^{[l]}}$ becomes very small. Algorithm 2 summarizes this parameter update algorithm.

Because $\phi(\Gamma)$ is a continuous function, its epigraph [40], for an initial Γ_0[6], is closed. By choosing a bounded set of admissible parameters Υ, the epigraph is a compact set in Euclidean space. To show that the algorithm gets as close as possible to the set of limit points, we need to use the Bolzano-Weierstrass theorem.

Theorem 2: [41, 3.24] Every bounded infinite subset of $\mathbb{R}^N$ has at least one limit point in $\mathbb{R}^N$.

Therefore, when the set of admissible parameters is bounded and ϵ is selected such that moving in the gradient direction with this step size reduces the objective, this gradient descent algorithm has at least one limit point in the admissible set.

Remark 1: The function $\phi(\Gamma)$ is a lower bounded function. Hence, if we reduce ϕ in each iteration, due to Lyapunov's second theorem [33], the algorithm is stable.

Remark 2: Algorithm 1 is an iterative algorithm in which we also used another iterative method for the dictionary update in line 6. The stability and the convergence of the updates mentioned above were related to the inner loop in Algorithm 1. We deal with the convergence of Algorithm 1 in Appendix A.

Remark 3: We draw the readers attention to the formulation (6). The parameters $\nabla_\Gamma \mathbf{D}_\Gamma$, $\nabla_{\mathbf{D}_\Gamma}\phi$ and $\nabla_\Gamma \phi$ are tensors of rank 3, 2 and 2 respectively. If $\Gamma \in \mathbb{R}^{p\times N}$ and $\mathbf{D} \in \mathbb{R}^{d\times N}$ then $\nabla_\Gamma \mathbf{D}_\Gamma \in \mathbb{R}^{p\times d\times N}$, $\nabla_{\mathbf{D}_\Gamma}\phi \in \mathbb{R}^{d\times 1\times N}$ and $\nabla_\Gamma \phi \in \mathbb{R}^{p\times 1\times N}$. A graphical presentation of this formulation is presented in Fig. 2 . Furthermore, to use this directional update in (7), we need to map $\nabla_\Gamma \phi \in \mathbb{R}^{p\times 1\times N}$ into the appropriate matrix in $\mathbb{R}^{p\times N}$. It is easily done by changing the order of indices (1,2,3 to 1,3,2), following by cancelling the third dimension. Because the rank of $\nabla_\Gamma \phi$ is 2, this mapping is injective.

[6]Epigraph of $\phi(\Gamma) : \Upsilon \to \mathbb{R}$ for an initial Γ_0 is defined [40, 3.1.7] by: $\operatorname{epi}\phi = \{\Gamma \ : \ \Gamma \in \Upsilon, \phi(\Gamma) \le \phi(\Gamma_0)\}$

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, we choose a case study. In sparse audio processing, an important question is how to choose the dictionary [42], [43]. Different methods have been introduced to adapt the dictionary to better fit a set of training samples [44], [45], [46]. For example, some researchers used a class of parametric dictionaries based on Gammatone filter banks, which have been shown to have similarities with the human auditory system [23], [47]. We now show that the parametric dictionary design improves the performance of audio sparse approximation and exact recovery based around a Gammatone representation.

A. Gammatone parametric dictionary

The generative function for a Gammatone dictionary is as follows,

$$g(t) = at^{n-1}e^{-2\pi bBt}\cos(2\pi f_c t), \tag{8}$$

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. One can select the parameters such that the generated atoms match the auditory impulse response. The auditory system has been optimized through evolution and may not be optimized for a practical application. Our goal is to optimally select these parameters so that sparse approximation methods can be used. Another difficulty in using the Gammatone filter banks as a dictionary is its large size. A moderate size dictionary can be designed by the proposed method.

The dictionary is generated by sampling the parameters of $g(t - t_c)$, where t_c is the time-shift. In this paper, $\gamma = [t_c \ f_c \ n \ b]^T$ 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 $\{\gamma_i\}_{1\le i\le N}$ 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.

The differentiability of $\mathbf{D}_\Gamma$ with respect to Γ makes the parameter update easier. In this paper we assume the parametric dictionary satisfies this constraint. Letting $n \in \mathbb{R}$, (8) becomes a generative function over a continuous domain Υ. 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 thus guaranteeing that the algorithm converges 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.

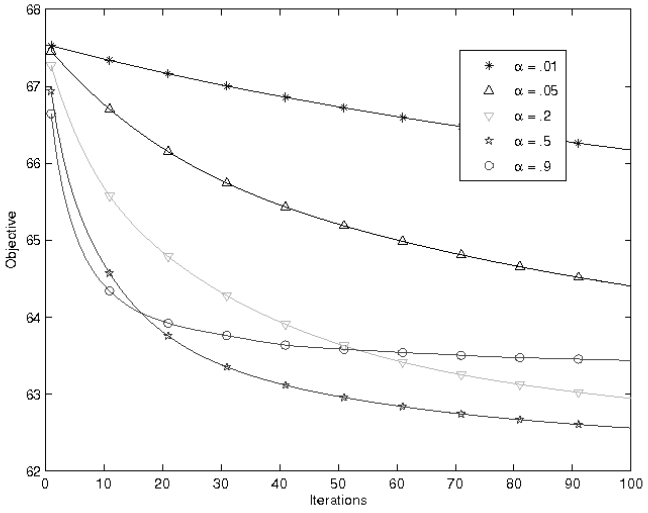

Fig. 3. The objective functions for different $\{\alpha_k\}_{\forall k,\alpha_k=\alpha}$, for a constant α.

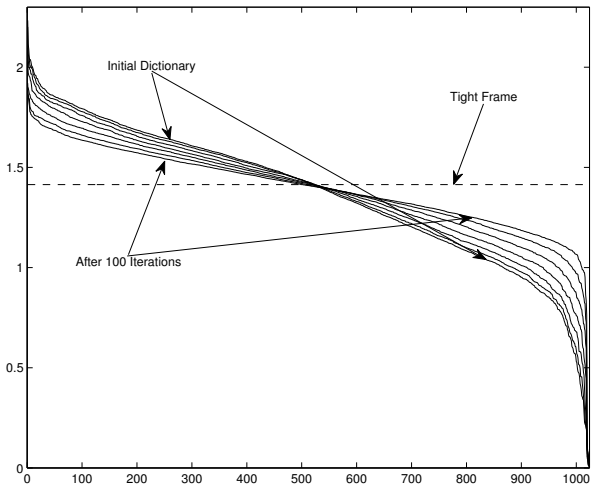

Fig. 4. Eigen values plot of the dictionary.

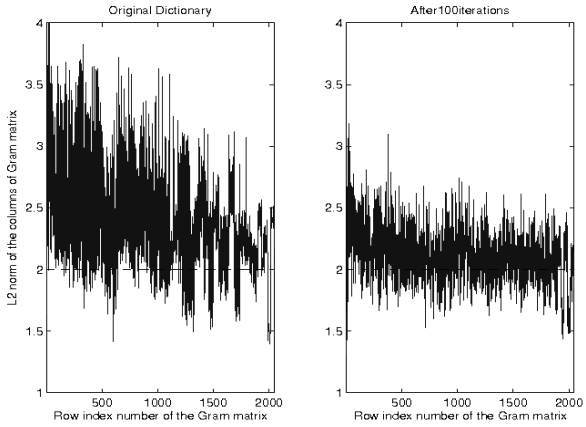

Fig. 5. The column ℓ_2 plots of the Gram matrix of the original (left) and designed (right) dictionaries.

Although the computation of the gradient of a parametric dictionary generated using $g(t)$ is straightforward, we derive it in the Appendix B for completeness.

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. The stability of the algorithm is demonstrated by showing the reduction of the objective function. In the second part of this subsection, we compare the performance of the initial and the optimized dictionaries in terms of sparse approximation and exact sparse recovery. Gammatone type dictionaries have been proposed for sparse approximation of audio and we choose our examples accordingly. In all the simulations we choose two times overcomplete dictionaries and window size 1024.

1) Algorithm Evaluation: In this part, 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 (3) in each iteration. The parameter B, defined after (8), is the bandwidth of the audio filterbank at the center frequency f_c. We use the fixed values $n = 4$, $Q = 9.26449$, $b_{min} = 24.7$, as they have been suggested in [48] and [49], and $b = 0.65$. To generate the initial dictionary, we sample f_c and t_c. We use the method introduced in [50] to generate the filter bank. In this method 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,

$$f_c^k = -Qb_{min} + (f_s/2 + Qb_{min})e^{-k\delta/Q}. \qquad (9)$$

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 (9). 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,

$$t_c^l = t_p + \sigma(l-1)\ t_p\ .$$

σ is set to 0.75 in our simulations. We draw the readers attention to the point that t_c^l is implicitly a function of f_c^k. We therefore generate a set of $\{f_c^k\}_{k\in\mathscr{K}}$ and for each generated atom using f_c^k and $t_c = 0$, we make a set of time-shifted versions using $\{t_c^l\}_{l\in\mathscr{L}}$.

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

$$\mathbf{d}_{\gamma_i,j} = \begin{cases} g_{\gamma_i(j+d)} & 1 \le j < j_{c_i} \\ g_{\gamma_i(j)} & j_{c_i} \le j \le d, \end{cases} \qquad (10)$$

where $j_{c_i} = \lfloor t_{c_i} f_{samp} \rfloor$. As the proposed algorithm is a relaxed version of the alternating minimization, the relaxation

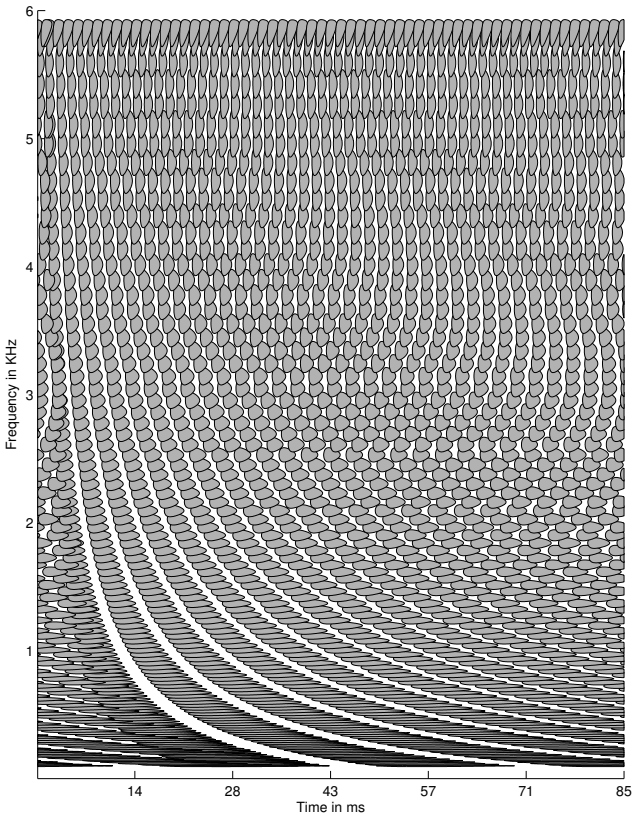

Fig. 6. Wigner-Ville contour plots of the original Gammatone atoms. The WV contour of each atom is calculated at 0.7 times it peak.

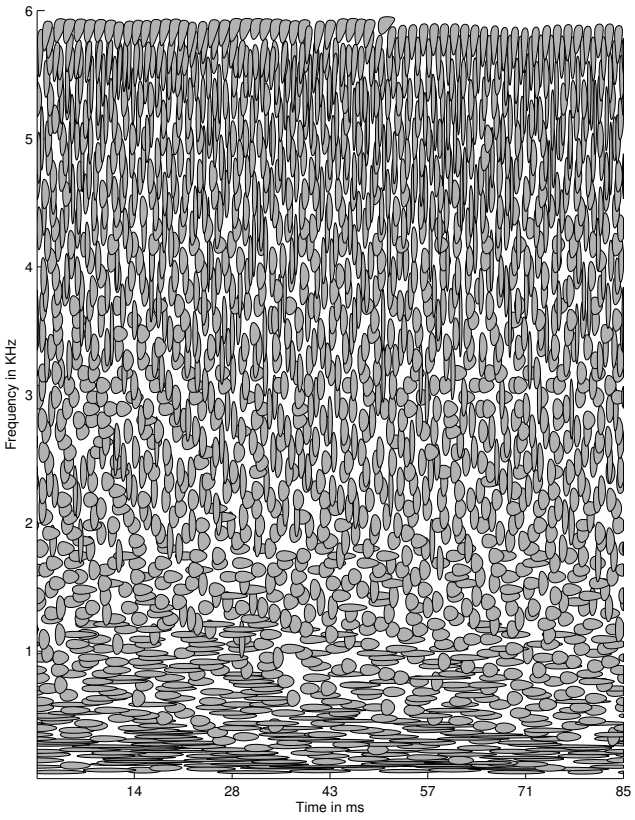

Fig. 7. Wigner-Ville contour plots of the learned Gammatone atoms. The contours are calculated similar to Fig. 6

parameters $\{\alpha_k\}$ should be selected. We choose a simple sequence of $\{\alpha_k\}$ using $\alpha_k = \alpha$ for all k and a fixed α in all simulations. A more complicated sequence might improve the performance of Algorithm 1. However we have not present this here. Here, we intend to show that the designed dictionary is superior to the initial dictionary 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 (3) using selected α's, in Fig. 3. 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. 4. It can be seen that the SV's of the designed dictionary get closer to the SV's of the tight frame at each iteration.

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, for the window size 1024, in Fig. 5. 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. Another observation in Fig. 5 is that the atoms with high total cross-correlations, indicated by the peaks, are adapted.

This parametric dictionary is attempting to tile the time-frequency plane. An ETF is a frame having the minimum total correlation between atoms but it may not be localized in the time-frequency plane. A dictionary which is simultaneously ETF, or close to being an ETF, and localized in time and frequency, tiles time-frequency plane more uniformly. To demonstrate this, we choose the Wigner-Ville (WV) time-frequency representation of the atoms. We show the contour plot of the atoms in the time-frequency plane using a similar method used in [51]. Fig. 6 and 7 show the time-frequency planes found for the original and designed atoms, respectively. Although the algorithm attempts to minimize μ by changing

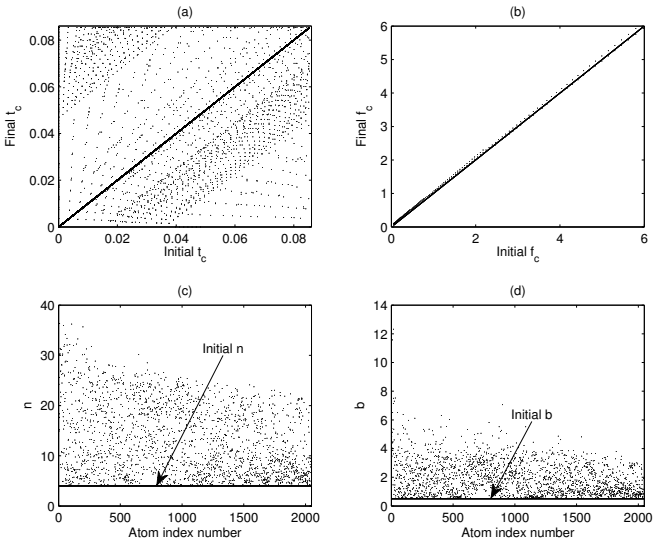

Fig. 8. Parameters of the Gammatone dictionary: the scatter-plots of the parameters t_c and f_c are shown in the top windows (a) and (b) respectively. The initial and final values of n and b are shown in the bottom windows (c) and (d) respectively.

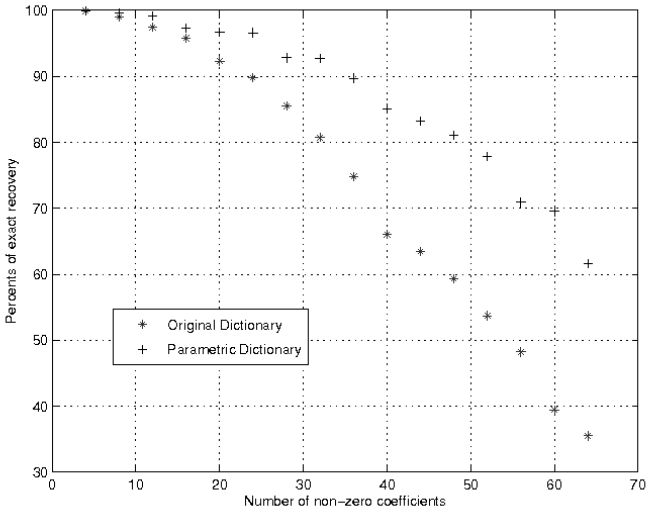

Fig. 9. Exact support recovery of the sparse signals.

the structure of the dictionary, the locations which are not covered by the high energy part of any atom demonstrate its local minimum convergence. It also shows a potential for a more efficient update operator than the gradient descent in Algorithm 2. There exists a shift-invariance structure, with different step size for each frequency band, in the initial parametric dictionary, which disappears in the designed dictionary. If the time-shift is one of the parameters in the parametric dictionary, such a structure can then be preserved. Such a parametric dictionary is not column separable. Designing a structured parametric dictionary, e.g. shift-invariant dictionary, is left for a future work.

The parameter set γ is selected intuitively in this experiment. To show the contribution of each parameter in the dictionary design, we show the initial and the final values of the parameters in Fig.8. The scatter-plots of t_c and f_c are shown in part (a) and (b). Note f_c has not have changed significantly by the dictionary design, so it could be kept fixed to reduce the computational cost. This simulation is also initialized with some fixed values for n and b. The final values of these two parameters are shown in Fig.8.c and Fig.8.d respectively. These plots show significant changes in the values of n and b, which demonstrates the importance of correct selection of n and b in each Gammatone atom.

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 [25] and sparse approximation. The exact recovery condition (ERC) [25] is studied in a worst-case setting. In this setting when a dictionary satisfies ERC, *any* k-sparse representation can exactly be recovered using (O)MP or BP. In practical applications, an average case analysis is more relevant [52], especially when the probability of the failure is very low. Here, by an experiment, it has been shown that the proposed algorithm improves the average exact recovery. We synthetically generate the 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 are selected uniformly at random and the PDF of the magnitudes are selected to be Gaussian with zero mean. The matching pursuit algorithm is 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 index sets to the number of cases in which at least one location of the zero coefficient is set to non-zero. We run the simulations 1000 times. We have shown this ratio as the percentage of exact recovery in Fig. 9. 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 [25], 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 [27]. We use an audio signal taken from more than 8 hours recorded from BBC Radio 3, which mostly plays classical music. We first down-sample by a factor of 4 and sum the stereo channels to make a mono signal with 12K samples per second. We use the original Gammatone and the parametric designed dictionaries to approximate 100 randomly selected blocks, each with the length of 1024 samples. The average decay rate of the residual errors, in logarithmic scales, are shown in Fig. 10. This rate directly influences the performance of sparse approximation methods. That is, we can better approximate the signal with fewer coefficients using a high residual error decay rate dictionary. In Fig. 10, although the curves start with the same slope, after a few iterations, here 10, the designed dictionary shows a clear advantage.

V. Conclusion

The sparse approximation methods successfully approximate a class of signals with a set of sparse coefficient vectors,

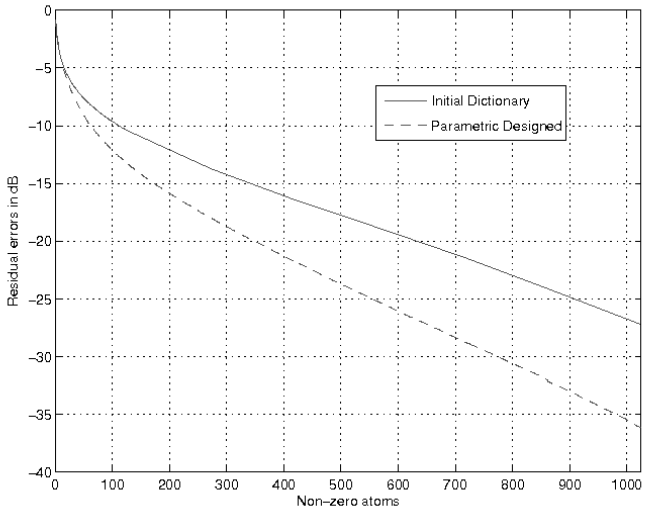

Fig. 10. The residual error using matching pursuit for sparse approximation of the audio signal.

when an appropriate generative model is given. In this paper we have introduced a method to design such a model, which is independent of the signal. A criterion based on an important feature for the success of sparse approximation methods is proposed. A priori knowledge about the signal was included by using parametric functions. 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 of audio signals. What we have shown in this paper is a first step in the design of parametric dictionaries. Extra constraints, such as shift-invariance, quasi-incoherence, data dependence, to have tree structures or structures for fast implementation, could be imposed. However, this has been left for future work.

APPENDIX A
CONVERGENCE STUDY OF THE ALGORITHM

To study the convergence of the algorithm, we first show that Algorithm 1, for any parameter update algorithm (line 6), reduces or keeps the same the objective function. The objective is lower bounded by zero and the algorithm prevents the existence of a continuum of fixed points, which guarantee the stability of Algorithm 1. In the next step we show that when $\mathbf{D}_\Gamma$ is a differentiable function on a compact Υ, the sequence generated by the algorithm becomes as close as possible to a set of fixed points.

A. Definition of a surrogate optimization problem

The objective function in (3) depends on two variables, which makes the convergence analysis more difficult, if we want to use the continuity of the objective in the analysis. Here we define a surrogate objective for (3), which has a single variable, to show the convergence of Algorithm 1 to a set of fixed points. Let $\Gamma^* \in \Upsilon$ and $\mathbf{G}^* \in \Lambda^N$ be a solution pair of (3) and $\mathbf{G}^*_\Gamma = \mathbf{D}^T_{\Gamma^*}\mathbf{D}_{\Gamma^*}$. Then $\mathbf{G}^* = \mathcal{P}_{\Lambda^N}\mathbf{G}^*_\Gamma$, which suggests the optimization problem (3) can be replaced by the following problem based on Γ, as the only parameter,

$$\begin{aligned}&\min_{\Gamma\in\Upsilon} \Phi_S(\Gamma),\\ &\Phi_S(\Gamma) = \|\mathbf{G}_\Gamma - \mathcal{P}_{\Lambda^N}\mathbf{G}_\Gamma\|_F\\ &\qquad = (\sum_{i\neq j}(|\{g_\Gamma\}_{i,j}| - \mu_G)^2 + \sum_{i=j}(\{g_\Gamma\}_{i,j}-1)^2\ )^{1/2},\end{aligned} \tag{11}$$

where $|\{g_\Gamma\}_{i,j}|$ is the absolute value of the (i,j) element of $\mathbf{G}_\Gamma = \mathbf{D}^T_\Gamma\mathbf{D}_\Gamma$. The problems (3) and (11) share common solutions. Therefore one can optimize (11) to find the solution(s) of (3). Although the surrogate objective is a continuous function of Γ ($\Phi_S \in \operatorname{class} C^0$), a difficulty with the optimization of the surrogate objective directly is that it is non-differentiable. We only use the surrogate optimization problem to show the convergence of the proposed algorithm.

B. Convergence analysis of Algorithm 1 using the surrogate optimization problem

We now show that the proposed algorithm reduces the surrogate objective at each parameter update, using the following proposition.

Proposition 1: Let $\mathbf{G}_{\Gamma_k} = \mathbf{D}^T_{\Gamma_k}\mathbf{D}_{\Gamma_k}$ be the Gram matrix of the dictionary at k^{th} iteration. The Algorithm 1 reduces, or keeps the same, $\|\mathbf{G}_{\Gamma_k} - \mathcal{P}_{\Lambda^N}\mathbf{G}_{\Gamma_k}\|_F$ in each update of the parameters ($\Gamma_k \rightarrow \Gamma_{k+1}$), where $\mathcal{P}_{\Lambda^N}$ is the operator of orthogonal projection onto Λ^N.

Proof: Let $\mathbf{G}_{P_{k+1}}$ be an abbreviation for $\mathcal{P}_{\Lambda^N}\mathbf{G}_{\Gamma_k}$, which is found by using (4). Using the parameter update step (line 6) and the fact that $\mathbf{G}_{R_{k+1}} = \alpha_k\mathbf{G}_{P_{k+1}} + (1-\alpha_k)\mathbf{G}_{\Gamma_k}$,

$$\begin{aligned}&\alpha_k\|\mathbf{G}_{\Gamma_k} - \mathbf{G}_{P_{k+1}}\|_F\\ &\quad= \|\mathbf{G}_{\Gamma_k} - \mathbf{G}_{R_{k+1}}\|_F\\ &\quad\geq \|\mathbf{G}_{\Gamma_{k+1}} - \mathbf{G}_{R_{k+1}}\|_F\\ &\quad= \|\mathbf{G}_{\Gamma_{k+1}} - \alpha_k\mathbf{G}_{P_{k+1}} - (1-\alpha_k)\mathbf{G}_{\Gamma_k}\|_F\\ &\quad= \|(\mathbf{G}_{\Gamma_{k+1}} - \mathbf{G}_{P_{k+1}}) - (1-\alpha_k)(\mathbf{G}_{\Gamma_k} - \mathbf{G}_{P_{k+1}})\|_F\\ &\quad\geq \|\mathbf{G}_{\Gamma_{k+1}} - \mathbf{G}_{P_{k+1}}\|_F - (1-\alpha_k)\|\mathbf{G}_{\Gamma_k} - \mathbf{G}_{P_{k+1}}\|_F\ ,\end{aligned}$$

where we used the triangular inequality to derive the last inequality. This provides us the following inequalities,

$$\begin{aligned}\|\mathbf{G}_{\Gamma_k} - \mathcal{P}_{\Lambda^N}\mathbf{G}_{\Gamma_k}\|_F &\geq \|\mathbf{G}_{\Gamma_{k+1}} - \mathcal{P}_{\Lambda^N}\mathbf{G}_{\Gamma_k}\|_F\\ &\geq \|\mathbf{G}_{\Gamma_{k+1}} - \mathcal{P}_{\Lambda^N}\mathbf{G}_{\Gamma_{k+1}}\|_F,\end{aligned} \tag{12}$$

where the last inequality is easily derived by using the definition of the projection in Hilbert space. ■

Prop. 1, with the facts that the objective is lower bounded by zero and there exists no continuum of fixed points, guarantees stability of Algorithm 1, due to Lyapunov's second theorem.

Let $\operatorname{class} C^1$ consist of all continuously differentiable functions. The following two Lemmata are needed to show the convergence of Algorithm 1 to a set of fixed points.

Lemma 1: Let $\mathbf{D}_\Gamma : \Upsilon \rightarrow \mathbb{R}^{d\times N} \in \operatorname{class} C^1$ and Υ be compact. The epigraph of the objective (11) at an admissible Γ_0 is compact.

Proof: When the parametric dictionary $\mathbf{D}_\Gamma$ is differentiable on Υ, the objective function in (11) is continuous. The continuity of the surrogate objective function and the compactness of Υ prove the compactness of $\operatorname{epi}\Phi_S$ at an admissible point Γ_0 [40]. ∎

Due to the Bolzano-Weierstrass theorem, Algorithm 1 has a non-empty set of accumulation points. We now reformulate Lemma 1 in [46] for a more general (including asymptotically non-regular[7]) sequence. Although the proof is the same, the set of accumulation points can be dis-connected, when the sequence is not asymptotically regular.

Lemma 2: Let $\{\Gamma_n\}_{n\in\mathbb{N}}$ be an infinite sequence in a compact set Σ and T be the set of its accumulation points then, $\forall\epsilon > 0, \exists N \in \mathbb{N}$ such that for all $n > N, \exists\Gamma^\ddagger \in T, \|\Gamma_n - \Gamma^\ddagger\|_F < \epsilon$

Proof: Let S be an ϵ-neighborhood of T and S_c be its complement in Σ. Σ is compact, thus S_c is also compact. Because S is a neighborhood of T, there is no accumulation point Γ in S_c. If $\{\Gamma_n\}$ has infinite many points in S_c, then it has a converging subsequence and at least one accumulation point in S_c. This contradicts the fact that there is no accumulation point in S_c. Therefore $\exists N : \Gamma_n \in S,\ \forall n > N$. On the other hand the fact that S being an ϵ-neighborhood implies that for all $n > N$, $\exists\Gamma^\ddagger \in T : \|\Gamma_n - \Gamma^\ddagger\|_F < \epsilon$. ∎

Theorem 3: Let $\mathbf{D}_\Gamma : \Upsilon \rightarrow \mathbb{R}^{d\times N} \in \operatorname{class} C^1$. The Algorithm 1 converges to a set of fixed points by starting from $\Gamma_0 \in \Upsilon$, where Υ is a compact set.

Proof: Due to Lemma 1 the epigraph of the surrogate objective at Γ_0 ($\operatorname{epi}\Phi_S(\Gamma_0)$) is compact. The Proposition 1 shows that the sequence $\{\Gamma_n\}_{n\in\mathbb{N}}$ is in $\operatorname{epi}\Phi_S(\Gamma_0)$. The convergence of the algorithm to a non-empty set of accumulation points is guaranteed using Lemma 2. Line 6 of Algorithm 1 prevents the existence of a continuum of accumulation points. Therefore the accumulation points are fixed points. ∎

APPENDIX B
GRADIENT OF THE GAMMATONE DICTIONARY

We calculate the gradient of the parametric Gammatone dictionary with the generative function (8) in this appendix. Let $\mathbf{D}_\Gamma \in \mathbb{R}^{d\times N}$ and $\Gamma \in \mathbb{R}^{4\times N}$. The i^{th} column of $\mathbf{D}_\Gamma$ is a function of the i^{th} column of Γ, $\mathbf{d}_{\gamma_i}$. The rank of $\nabla_\Gamma\mathbf{D}_\Gamma$ is 4 and we represent it by a tensor in $\mathbb{R}^{4\times d\times N}$. Each submatrix of this tensor (fixing the third index) is the gradient of the corresponding atom in $\mathbf{D}_\Gamma$. Therefore we only need to calculate the gradient of $\mathbf{d}_{\gamma_i}$ based on γ_i. Because $\mathbf{d}_\gamma$ is calculated using (10), we only need to derive a formulaton for the gradients of $g_\gamma(t)$ based on t_c, f_c, n and b, followed by sampling t.

$$\frac{\partial g_\gamma}{\partial t_c} = -a((n-1)t_s^{n-2}\cos 2\pi f_c t_s + 2\pi bBt_s^{n-1}\cos 2\pi f_c t_s + 2\pi f_c t_s^{n-1}\sin(2\pi f_c t_s))e^{-2\pi bBt_s}$$

$$\frac{\partial g_\gamma}{\partial f_c} = at_s^{n-1}(-2\pi t_s\frac{dB}{df_c}\cos(2\pi f_c t_s) - 2\pi t_s\sin(2\pi f_c t_s))e^{-2\pi bBt_s}$$

$$\frac{\partial g_\gamma}{\partial n} = a\ln(t_s)\ t_s^{n-1}e^{-2\pi bBt_s}\cos(2\pi f_c t_s)$$

$$\frac{\partial g_\gamma}{\partial b} = -2\pi aBt_s^n e^{2\pi bBt_s}\cos(2\pi f_c t_s)$$

where $t_s = t - t_c$ and $\frac{dB}{df_c} = 1/Q$. Some researchers have proposed more complex formulations for B. In this case, one can substitute B and $\frac{dB}{df_c}$ in the above formulas to find the gradient.

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers for their constructive comments and L. Jacques for bringing their attention to [30] and [31]. MY acknowledges the hospitality and support of E. Ravelli at the Institut Jean le Rond d'Alembert-LAM, during his visit, at the start of this research. 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.

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] J. Tropp, "Topics in sparse approximation," Ph.D. dissertation, University of Texas at Austin, 2004.
[4] G. Davis, "Adaptive nonlinear approximations," Ph.D. dissertation, New York University, 1994.
[5] S. Mallat and Z. Zhang, "Matching pursuits with time frequency dictionaries," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.
[6] T. Blumensath and M. Davies, "Gradient pursuits," *IEEE Trans. on Signal Processing*, vol. 56, no. 6, pp. 2370–2382, 2008.
[7] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.
[8] I. Gorodnitsky and B. Rao, "Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm," *IEEE Trans. on Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.
[9] B. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," *IEEE Trans. on Signal Processing*, vol. 47, no. 1, pp. 187–200, 1999.
[10] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Trans. Information Theory*, vol. 49, no. 12, pp. 3320 – 3325, 2003.
[11] S. Sardy, A. Bruce, and P. Tseng, "Block coordinate relaxation methods for nonparametric wavelet denoising," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 361–379, 2000.
[12] 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.
[13] B. Olshausen and D. Field, "Sparse coding with an overcomplete basis set: a strategy employed by V1?" *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1997.
[14] M. Lewicki and T. Sejnowski, "Learning overcomplete representations," *Neural Comp*, vol. 12, no. 2, pp. 337–365, 2000.
[15] K. Kreutz-Delgado, J. Murray, B. Rao, K. Engan, T. Lee, and T. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Comp.*, vol. 15, pp. 349–396, 2003.
[16] M. Aharon, E. Elad, and A. Bruckstein, "K-SVD: an algorithm for designining of overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[7] A sequence $\{a_k\}_{k\in\mathbb{N}}$ in a normed space is called asymptotically regular when $\lim_{k\rightarrow\infty}\|a_k - a_{k-1}\| = 0$

[17] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Problems*, vol. 23, no. 3, pp. 947–968, 2007.
[18] T. Lee, "Image representation using 2D Gabor wavelets," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, 1996.
[19] A. Katsiamis, E. Drakakis, and R. Lyon, "Practical Gammatone-like filters for auditory processing," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2007, 2007, Article ID 63685.
[20] T. Irino and R. Patterson, "A time domain, level dependent auditory filter: the Gammachirp," *Journal of the Acoustical Society of America*, vol. 101, pp. 412–419, 1997.
[21] R. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice, "An efficient auditory filterbank based on the Gammatone function," APU Report, Tech. Rep., 1988.
[22] M. Turner, G. Gerstein, and R. Bajcsy, "Underestimation of visual texture slant by human observers: a model," *Biological Cybernetics*, vol. 65, no. 4, pp. 215–226, 2004.
[23] 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.
[24] D. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. on Information Theory*, vol. 47, no. 7, pp. 2845–2862, 2001.
[25] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.
[26] ——, "Just relax: Convex programming methods for identifying sparse signals," *IEEE Trans. on Information Theory*, vol. 52, no. 3, pp. 1030–1051, 2006.
[27] 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.
[28] 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.
[29] 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.
[30] R. Gribonval, "Fast matching pursuit with a multiscale dictionary of Gaussian chirps," *IEEE Trans. on Signal Processing*, vol. 49, no. 5, pp. 994–1001, 2001.
[31] L. Jacques and C. De Vleeschouwer, "A geometrical study of matching pursuit parametrization," *IEEE Trans. Signal Processing*, vol. 56, no. 7, pp. 2835–2848, 2008.
[32] 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.
[33] A. Lyapunov, *Stability of motion*. Academic Press, 1966.
[34] H. Stark and Y. Yang, *Vector Space Projections: A Numerical Approach to Signal and Image Processing, Neural Nets and Optics*. John Wiley & Sons, Inc, 1998.
[35] H. Hein and S. Hundal, "An alternating projection that does not converge in norm," *Nonlinear Analysis*, vol. 57, no. 1, pp. 35–61, 2004.
[36] I. Csiszar and G. Tusnady, "Information geometry and alternating minimization procedures," *Statistics and Decisions, Supplementary no. 1*, pp. 205–237, 1984.
[37] A. Gunawardana and W. Byrne, "Convergence theorems for generalized alternating minimization procedures," *The Journal of Machine Learning Research*, vol. 6, pp. 2049 – 2073, 2005.
[38] J. Dattorro, *Convex Optimization and Euclidean Distance Geometry*. Meboo Publishing, 2005 (v2007.09.17), Palo Alto, CA.
[39] R. Fletcher, *Practical Methods of Optimization*. John Wiley and Sons: Chichester and New York, 1987.
[40] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
[41] T. Apostol, *Mathematical Analysis: A Modern Approach to Advanced Calculus*. Addison-Wesley, 1974.
[42] M. Davies and L. Daudet, "Sparse audio representations using the MCLT," *Signal Processing*, vol. 86, no. 3, pp. 457–470, 2006.
[43] E. Ravelli, G. Richard, and L. Daudet, "Union of MDCT bases for audio coding," 2008, accepted for publication in IEEE Trans. on Audio, Speech and Language Processing.
[44] E. Smith and M. S. Lewicki, "Efficient coding of time-relative structure using spikes," *Neural Computation*, vol. 17, no. 1, pp. 19–45, 2005.
[45] M. Lewicki, "Efficient coding of natural sounds," *Nature Neuroscience*, vol. 5, no. 4, pp. 356–363, 2002.
[46] M. Yaghoobi, T. Blumensath, and M. Davies, "Dictionary learning for sparse approximations with the majorization method," *IEEE Trans. on Signal Processing*, vol. 57, no. 6, pp. 2178–2191, 2009.
[47] 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.
[48] B. R. Glasberg and B. C. J. Moore, "Derivative of auditory filter shapes from notched-noise data," *Hearing Research*, vol. 47, pp. 103–108, 1990.
[49] M. Slaney, "Lyon's cochlear model," Apple Computer, Tech. Rep., 1988.
[50] ——, "An efficient implementation of the Patterson-Holdsworth auditory filter bank," Apple Computer, Tech. Rep., 1993.
[51] S. Abellah and M. Plumbley, "If the independent components of natural images are edges, what are the independent components of natural sounds?" in *Proc. of the International Conference on Independent Component Analysis and Blind Source Separation (ICA)*, 2001.
[52] R. Gribonval and K. Schnass, "Some recovery conditions for basis learning by L1-minimization," in *International Symposium on Communications, Control and Signal Processing, ISCCSP*, 2008.

Mehrdad Yaghoobi (S'98-M'09) received the BSc. and MSc. in electrical and biomedical engineering in 1999 and 2002 from the University of Tehran and Sharif University of Technology, respectively. He started his PhD at Queen Mary University of London in December 2005, before he moved to the University of Edinburgh to accompany his supervisor in April 2006. He is now persuing the PhD degree in the Institute for Digital Communications (IDCom) at the University of Edinburgh. His current research interests include sparse approximation, dictionary selection, compressed sensing and audio modelling/coding.

Laurent Daudet (M'04) received the M.S. degree in statistical and nonlinear physics from the cole Normale Superieure, Paris, France, in 1997 and the Ph.D. degree in mathematical modeling from the Universite de Provence, Marseille, France, in audio signal representations, in 2000. In 2001 and 2002, he was a Marie Curie Post-doctoral Fellow with the Centre for Digital Music at Queen Mary, University of London, London, U.K. Since 2002, he has been working as an Assistant Professor at the UPMC Univ. Paris 06, Paris, France, where he joined the Laboratoire d'Acoustique Musicale, now part of Institut Jean Le Rond d'Alembert. His research interests include audio coding, time-frequency and time-scale transforms, and sparse representations for audio.

Mike E. Davies (M'00) received the B.A. (Hons.) degree in engineering from Cambridge University, Cambridge, U.K., in 1989 and the Ph.D. degree in nonlinear dynamics and signal processing from University College London, London (UCL), U.K., in 1993. Mike Davies was awarded a Royal Society Research Fellowship in 1993 and was an Associate Editor for IEEE Transactions in Speech, Language and Audio Processing, 2003-2007.

He currently holds the Jeffrey Collins SHEFC funded chair in Signal and Image Processing at the University of Edinburgh. His current research interests include: sparse approximation, compressed sensing and their applications.