Random time-frequency Subdictionary design for sparse representation with greedy algorithms
Manuel Moussallam, Laurent Daudet, Gael Richard

To cite this version:
Manuel Moussallam, Laurent Daudet, Gael Richard. Random time-frequency Subdictionary design for sparse representation with greedy algorithms. ICASSP, Mar 2012, Kyoto, Japan. pp.3577-3580. hal-00696187

HAL Id: hal-00696187
https://hal-imt.archives-ouvertes.fr/hal-00696187
Submitted on 11 May 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.
RANDOM TIME-FREQUENCY SUBDICTIOn DESIGN FOR SPARSE REPRESENTATIONS WITH GREEDY ALGORITHMS

Manuel Moussallam\textsuperscript{1,2}, Laurent Daudet\textsuperscript{2}, Gaël Richard\textsuperscript{1}

\textsuperscript{1}Institut Telecom - Telecom ParisTech - CNRS/LTCI
37/39, rue Dareau 75014 Paris, France

\textsuperscript{2}Institut Langevin - ESPCI ParisTech - UMR7587
10 rue Vauquelin, 75005 Paris, France

ABSTRACT

Sparse signal approximation can be used to design efficient low-bitrate coding schemes. It heavily relies on the ability to design appropriate dictionaries and corresponding decomposition algorithms. The size of the dictionary, and therefore its resolution, is a key parameter that handles the tradeoff between sparsity and tractability. This work proposes the use of a non-adaptive random sequence of subdictionaries in a greedy decomposition process, thus browsing a larger dictionary space in a probabilistic fashion with no additional projection cost nor parameter estimation. This technique leads to very sparse decompositions, at a controlled computational complexity. Experimental evaluation is provided as proof of concept for low bit rate compression of audio signals.

\textbf{Index Terms}— Matching Pursuits, Random Subdictionaries, Sparse Audio Coding

1. INTRODUCTION

Randomness has proven surprisingly useful in a wide variety of computational and statistical fields. In communications, spread spectrum techniques, where a signal is modulated by a random binomial sequence before transmission allows for better bandwidth management. Quantization has long taken advantage of the dithering technique that uses randomness to avoid perceptually disturbing artefacts linked to the quantization noise. More generally, stochastic resonance theory has shown how a moderate amount of added noise can increase the average behavior of many non-linear systems. More recently, tremendous work has been achieved on the compressive sampling scheme, making use of random measurement matrices. Behind all these examples lies a common intuition: controlling the random part of a system is better than having to deal with colored measurement noise or transmission errors, or signal-dependent deviations. The key point is to spread the information where it can be efficiently found. Having to guess where discriminant low-level features are hidden in huge-dimensional spaces is too costly or simply not feasible. In such cases, randomness can be used as a powerful sieve by information miners.

At the opposite of uniform random distributions is the concept of sparsity. Sparse coding of digital signals has been the subject of many works in the past few years for audio [9], images [4] or video streams. Low bitrate coders have been designed and proved to be competitive with state of the art industrial solutions. The core idea is to decompose an original signal \( f \) as a combination of (a few) objects from a dictionary \( \Phi \) of indexed elementary waveforms. In a coding framework, the coder has to transmit the set of indexes of the non-zero coefficients, together with their quantized values. Here, a crucial yet often underestimated issue is the choice of the size of \( \Phi \), that always resorts to a tradeoff. If the dictionary is small (slight or no overcompleteness), computations are fast, the index coding cost per coefficient is low, but many coefficients may be needed. If it is large (i.e., dense in the parameter space), we have a fast decay of the approximation error as a function of the approximation order, but the cost of encoding the indexes is higher, and computations get cumbersome. Indeed, the computational complexity associated with these sparse techniques, as opposed to suboptimal but much simpler transform-based coders, is probably the main limitation to their widespread use in practical applications. Strategies have been proposed to lower the computational cost of using large dictionaries, based on local adaptation of the selected atoms [5] or probabilistic approaches [3] where successive runs with random sub-optimal atom selection are performed, then averaging yields a robust sparse approximation. Yet, these approaches are still associated with high index coding costs, if the atoms parameter space is large.

In this work, we propose a different paradigm that mitigates the drawbacks of using a large dictionary while keeping most of the benefits. Based on the algorithm described in [8], a single run is performed using varying subdictionaries. These subdictionaries have limited size, but are designed so as to evenly span a much larger dictionary space. In this work, we use a simplistic audio coding example as a proof of concept to demonstrate the usefulness of randomization for sparse representation problems. The key point in our technique is that the choice of subdictionary is not adaptive, but is parameterized by a fixed pseudo-random sequence, also known by the decoder. In other words, we have the (theoretical) complexity of working with a small dictionary, and the small coding costs, but the whole large dictionary is spanned. It should be emphasized that, unlike a compressive sensing framework, we our goal is to design a "standard" coding scheme with maximal efficiency at the cost of computational complexity at the encoder, and minimal decoding complexity at the decoder.

The rest of this paper is organized as follows. Section 2 recalls the Matching Pursuit framework for compression. In Section 3, the novel approach is presented along with considerations on the random sequence design, and in Section 4 trivial audio compression serves as a proof of concept for the suitability of the proposed method.

2. SIGNAL COMPRESSION WITH GREEDY ALGORITHMS

Let \( f \) be a discrete signal living in \( \mathbb{R}^N \). Greedy algorithms iteratively decompose \( f \) using \( m \) elements from a dictionary \( \Phi = \{ \phi_k \} \) of elementary objects called atoms by alternating two steps 1: Select an atom in the dictionary and 2: Update approximant and residual.
of scale

as mentioned in the introduction, the central problem of choosing

This criterion can also be modified to take perceptual models into ac-

(plain Matching Pursuit (MP) [7]), the subspace spanned by all pre-

seen as a descent in a direction defined by the newly selected atom

between consecutive analysis windows. The finer this analysis grid,

frequency resolutions are constrained by the scale of the chosen trans-

based dictionaries are well suited for audio signals. Time and fre-

2.1. Pursuits in time-frequency dictionaries

exactly the one where

has the sparsest representation. [2]: the chosen representation space is not

basis mismatch problem [2]: the chosen representation space is not

mizes a correlation function, usually an energy criterion:

where \( \Phi \) is the sampling frequency and \( \Delta_s \xi \) is the frequency resolution. The

subdictionary \( \Phi_{\gamma_s} \) can be an orthonormal base (e.g an MDCT with

50 \% overlap [9]), in this case \( \Delta_s u = s/2 \) and \( \Delta_s \xi = F_s/s \) and

then \( T_s = N \). It can also be overcomplete (Gabor dictionaries with

more than 50\% overlap: \( \Delta_s u < s/2, T_s > N \) ) or span only a

limited subspace (\( \Delta_s u > s/2, T_s < N \)). \( \gamma_s \) defines a two di-

mensional lattice in the time-frequency plane that can be seen as a

quantization of the underlying continuous time-frequency param-

eters. Indeed, the choice of a suboptimal atom can be understood

as a quantization error artefact. Figure 1 shows how different lat-

tices can fit different components of a signals, here transients and

harmonics of a glockenspiel signal. The quantization error is greatly

lowered by the concatenation of all the bases in the dictionary [9].

Nonetheless, traditional Matching Pursuit-based strategies are using

the same dictionary during the whole decomposition process. By

doing so, the a priori choice of lattices introduces a bias in the de-

composition as explained in Dörk’s work [3]. While their solution

would be to run multiple decompositions with different lattices and

averaging the results in a Monte Carlo fashion, we propose an novel

approach inspired by the dithered quantization technique.

3. Pursuits with a random sequence of subdictionaries

Instead of choosing an analysis grid once and for all, the lattices are

chosen so as to span the largest possible space in an ergodic fash-

ion during the decomposition. For each scale, a sequence of lattices

\( \Phi^*_s = \{ \phi_{s,u,\xi} \} \) is used and at iteration \( i \) an atom is selected in

\( \Phi_i = \bigcup_{s \in S} \Phi_{\gamma_s} \). By doing so, we expect the equivalent of the

quantization error to be evenly spread among the selected atoms,

thus removing the bias. This technique is conceptually equivalent to

adding a uniform noise in the time-frequency domain before quanti-

zing it. In the overall, we hope to promote the selection of more

salient features than with a fixed lattice. This technique is completely

non-adaptive. The sequences \( \Delta_s \gamma \) are known in advance and inde-

pendent from the signal. They are also known by the decoder and

therefore there is no need to encode them. The virtual cost of the

index of atom \( i \) is then down to \( O(\log_2(\sum_{s \in S} \gamma_s)) \).

3.1. Learning the sequence

Since we want the subset sequence to be independant of the signal,

we can try to estimate its desirable properties. When a signal model

is available (e.g sinusoidal+transient modelling of audio, edges and

textures modelling of images), one can try to manually design a se-

quence that will minimize the quantization error (i.e the suboptimal-

ity factor) under a dictionary size constraint. In this work however,

we have no signal model and we are interested in designing a univer-

sal sequence for audio signals. Figure 2 shows a decomposition of

a short glockenspiel signal with MP over a full temporal-resolution

multiscale discrete Gabor dictionary. Frequency resolution is con-

strained in each scale, the full temporal resolution is achieved by

performing Short Time Fourier Transforms with high overlapping

between consecutive analysis frames (i.e \( \Delta_s u = 1 \) sample). Atoms

are clearly not uniformly distributed in the time-frequency plane,

which would be the case for white noise. Most real life signals are
structured, a complete randomness in the choice of an analysis lattice would not give optimal performance.

3.2. Time resolution subsampling

In particular, for audio signals, the frequency resolution problem seems to be efficiently addressed by the use of multiple scales. The time resolution, however, presents a more interesting challenge. Let us consider a reference coarse lattice $\mathcal{L}$ with overlap of $\Delta_s u = s/2$ between consecutive frames and define the local time shift $\tau$ of an atom relative to this coarse grid. Then the distribution of $\tau$ in the interval $[-\frac{s}{2}, \frac{s}{2}]$ resembles a uniform distribution. To verify this statement, we decomposed short audio signals from the MPEG SQAM test database up to the first 1000 atoms in a full resolution multiscale Gabor dictionary and calculated the Gini index of $\tau$. This index quantifies how far from the uniform distribution a candidate distribution stands and has been demonstrated [6] to be a suitable sparsity measure.

Figure 3 shows that distribution for the decomposition of the orchestra signal resembles the one for the decomposition of white noise: atoms become more and more uniformly spread. The glockenspiel signal’s atoms are less uniformly distributed, but their distribution can not be considered sparse after a few hundreds iterations. From this observation, we state that an efficient and simple way to simulate a pursuit in a large dictionary is to use orthonormal basis functions at $\tau$.

4. SCALABLE SPARSE CODING OF AUDIO SIGNALS

4.1. Performances with a simple encoding model

In order to demonstrate the potential benefits of this technique, we compared traditional approaches to the new one in a simplified audio coding task. We considered 3 cases:

MP with a union of 8 MDCT: $\forall s, \Delta_s u = s/2, T_s = N$ and $T = SN$.

MP with a union of 8 MDCT with full temporal resolution (for computational reasons, an approximate base on local optimization after a coarse grid search is used as in [8]) $T_s = Ns/2, T = \sum_{s \in S} Ns/2$.

MP with Random Sequence of Subdictionaries that are 8 MDCT randomly shifted in time. The subdictionary size is constant $T_s = Ns/2$.

4.2. Controlling computational complexity

Although the theoretical complexity of the novel algorithm is equivalent to the one of the fixed small dictionary case, in practice it can get much higher. Inner products computation, actually, are usually performed using Fast transforms, and in the fixed dictionary case, most of these products remains unchanged from one iteration to the next, thus greatly reducing the cost of all but the first iteration. In our case, however, this trick cannot be applied anymore since the central point is to change all the projections at every iteration. There

$N$.

The distortion measure that we used was the Signal to Noise Ratio (SNR), and bitrates are estimated using the upper bound $C(f_m) = m(\log_2(T) + Q)$, where we assume a simple uniform mid-tread quantizer with $Q$ bits per coefficient. Note that using an entropy coder instead has also been tested, with similar results. Figure 4 summarizes the coding scheme. Figure 5 shows that the novel algorithm gives better performances in all cases but at very low bit-rates. This can be explained by the fact that, as exposed in 3.2, in the first iterations, the full resolution dictionary successfully locates the most prominent features and it actually compensates the additional costs. At bigger rates, however, atoms are more evenly spread and the lower index cost favors the randomized method. To summarize our results, this randomized greedy pursuit allows to have the costs of the small dictionary with a decay rate close to the one on the large dictionary.

5.0.2
The proposed algorithm appears to be suitable for sparse approximation of complex signals. The potential benefits are in low bitrate compression, and we exhibit several sound examples where these advantages show off. The unsupervised nature of the algorithm and the randomness introduced in the atom selection makes it very easy to design worst-case scenarios for which the algorithm would converge slower than a pursuit over a fixed dictionary. However, on average, and with a small empirical variance, the proposed scheme appears to have the coding costs of the small dictionary with a decay rate close to the one on the large dictionary.

In conclusion, adding randomness to the parameter space within a greedy sparse decomposition process can be highly beneficial. Although on different paradigms, this is reminiscent of both dithered quantization techniques and compressive sampling strategies. Whether such random sub-dictionary techniques can be combined with other sparse decomposition schemes (e.g., iterated thresholding) is still an open issue.

6. REFERENCES