MATCHING PURSUIT WITH STOCHASTIC SELECTION

Thomas Peel, Valentin Emiya, Liva Ralaivola

Aix-Marseille Université - CNRS
LIF
39, rue F. Joliot Curie
13453 Marseille Cedex 13, France

Sandrine Anthoine

Aix-Marseille Université - CNRS
LATP
39, rue F. Joliot Curie
13453 Marseille Cedex 13, France

ABSTRACT

In this paper, we propose a Stochastic Selection strategy that accelerates the atom selection step of Matching Pursuit. This strategy consists of selecting randomly a subset of atoms and a subset of rows in the full dictionary at each step of the Matching Pursuit to obtain a sub-optimal but fast atom selection. We study the performance of the proposed algorithm in terms of approximation accuracy (decrease of the residual norm), of exact-sparse recovery and of audio declipping of real data. Numerical experiments show the relevance of the approach. The proposed Stochastic Selection strategy is presented with Matching Pursuit but applies to any pursuit algorithms provided that their selection step is based on the computation of correlations.

***Index Terms*—** Sparsity; Pursuit Algorithm; Stochastic Procedure

1. INTRODUCTION

We are interested in estimating the sparse representation $\mathbf{x} \in \mathbb{R}^K$ of a signal $\mathbf{y} \in \mathbb{R}^M$ in a so-called *dictionary* $\mathbf{\Phi} \in \mathbb{R}^{M \times K}$ such that

$$\mathbf{y} \approx \mathbf{\Phi x} \quad (1)$$

where $M \leq K$, and where the number of non-zero elements $\|\mathbf{x}\|_0$ in $\mathbf{x}$ is small ($\|\mathbf{x}\|_0 \ll M$). The columns of $\mathbf{\Phi}$ are usually called *atoms* and $\|\mathbf{x}\|_0$ is the *sparsity* of $\mathbf{x}$.

Retrieving the sparsest representation from an observed signal and a known dictionary is NP-hard. The Matching Pursuit (MP) algorithm and its descendants [1, 2, 3] have been proposed to estimate such a sparse representation. Those greedy algorithms, which essentially iterate over two steps, a *selection* step and an *update* step, are known to perform well for exact sparse recovery and sparse approximation [4].

Computational aspects of greedy algorithms have motivated a few works over the past years. Their goal is not only to propose fast algorithms but also to make it possible to deal with high-dimensional data, where the dictionary size may be so large that it cannot be stored as a matrix in memory.

Accelerating the update stage is useful when high accuracy is targeted. Indeed, performance is better —regarding the decrease of the *residual* per iteration— with Orthogonal Matching Pursuit (OMP) [1, 2] than with MP, at the price of a computationally-demanding update stage. This has been thoroughly studied in [3]: the authors propose a detailed analysis of the computational complexity of MP and OMP, together with fast implementations of the update stage in OMP; then, their gradient-based pursuit (GP) algorithms use new update stages and achieve better performance than MP with a lower computational cost than OMP.

Accelerating the selection stage consists in avoiding a full and naive matrix-vector product between the dictionary-matrix transpose and the current residual vector. It is particularly useful when a high-dimensional dictionary must be handled. For particular dictionaries, fast transforms [1] can be used. However, they do not apply in general cases; instead, an approximate selection stage can be performed using an approximate nearest neighbor (ANN) search. An approximate search is motivated and supported by the theoretical study of so-called weak greedy algorithms [5, 6], for which good convergence properties are guaranteed using a suboptimal selection [7]. ANN search for sparse representations have been developed using random projections in compressed sensing [8, 9], a tree structure [10], a locality-sensitive hashing sheme [11], or random selections of subsets of atoms [12]. In this paper, we propose a new approach to accelerate the selection stage using a stochastic principle, generalizing a recent work proposed in [12].

The main contributions of the paper are: i) a new family of algorithms that are stochastic variants of existing ones (including MP, OMP, GP); ii) numerical evidence of the accuracy – which is comparable to the accuracy of non-stochastic algorithms – and of the decrease in the computational complexity; iii) a discussion on the issues at stake regarding the proposed method, including the relation with compressed sensing and the analysis of the algorithm.

The rest of the paper is organized as follows. In Section 2, we motivate and present our Stochastic Selection procedure. Section 3 contains numerical experiments that show the relevance of our approach. This work leads to several theoretical questions discussed in Section 4.

2. ALGORITHM

2.1. Notations

Let $\mathbf{y}$ be a discrete signal living in $\mathbb{R}^M$; $\mathbf{y}$ is a (possibly) noisy observation of a signal $\mathbf{\Phi x}$ sparsely coded in the dictionary $\mathbf{\Phi}$. By this, we mean that we consider the model

$$\mathbf{y} = \mathbf{\Phi x} + \mathbf{n},$$

where:

- $\mathbf{\Phi} \in \mathbb{R}^{M \times K}$ is a dictionary composed by K *atoms* : $\mathbf{\Phi} = [\boldsymbol{\varphi}^1 \dots \boldsymbol{\varphi}^K]$. Each $\boldsymbol{\varphi}^j$ is in $\mathbb{R}^M$ and has a unit l_2-norm.
- $\mathbf{x} \in \mathbb{R}^K$ is sparse, its support is of length $\|x\|_0 = K_0$.
- $\mathbf{n} \in \mathbb{R}^M$ denotes the noise in the measurements.

Note that we will also consider the non-noisy case $\mathbf{y} = \mathbf{\Phi x}$ where $\mathbf{y}$ is exactly K_0-sparse.

In this paper, we will randomly select m rows and k columns of the dictionary. Accordingly, $\mathcal{M} = \{i_1, \dots, i_m\}$ denotes the set of selected rows and $\mathcal{K} = \{j_1, \dots, j_k\}$ the set of selected columns (*i.e.* atoms) of the dictionary. We denote by $\mathbf{y}_\mathcal{M} \in \mathbb{R}^\mathcal{M}$ the restriction of the vector $\mathbf{y}$ to the coordinates indexed by $\mathcal{M}$.

2.2. Motivations

Matching Pursuit [1] is a greedy algorithm developed in the 90's that aims at approximately solving the NP-hard problem of finding the sparsest decomposition of the signal $\mathbf{y}$ with respect to the dictionary $\mathbf{\Phi}$. As depicted in Algorithm 1, Matching Pursuit computes an approximate solution to the problem

$$\arg\min_{\mathbf{x}\in\mathbb{R}^K} \|\mathbf{x}\|_0 \text{ s.t. } \|\mathbf{y} - \mathbf{\Phi x}\|_2 \leq \epsilon, \tag{2}$$

where $\epsilon \geq 0$; $\epsilon = 0$ in the *exact sparse* case and $\epsilon > 0$ in the noisy case. Matching Pursuit is an iterative procedure that improves the estimation of the sparse coefficients vector $\mathbf{x}$ in a greedy fashion. At step t, the current coefficient vector $\mathbf{x}^t$ yields the approximation $\mathbf{\Phi x}^t$ of $\mathbf{y}$. We call $\mathbf{r}^t = \mathbf{y} - \mathbf{\Phi x}^t$ the residual. At this step, Matching Pursuit selects the best atom that may be picked, *i.e.* the atom $\boldsymbol{\varphi}^{\gamma_{t+1}}$ that minimizes the norm of the residual at the next step:

$$\boldsymbol{\varphi}^{\gamma_{t+1}} = \arg\min_{\boldsymbol{\varphi}\in\{\boldsymbol{\varphi}_1,\dots,\boldsymbol{\varphi}_K\}} \left\|\mathbf{r}^t - \langle \mathbf{r}^t, \boldsymbol{\varphi}\rangle\boldsymbol{\varphi}\right\|_2 .$$

The coefficients vector $\mathbf{x}^{t+1}$ and residual $\mathbf{r}^{t+1}$ are updated accordingly and one continues until a stopping criterion is met (see Algorithm 1 for the details). The usual criteria are to stop either when a predefined maximal number of iterations is reached ($t = T$) or when the norm of the residual is below a predefined precision ($\left\|\mathbf{r}^t\right\|_2 \leq \epsilon$).

Algorithm 1 Matching Pursuit

```
inputs: $\mathbf{\Phi} \in \mathbb{R}^{M\times K}, \mathbf{y} \in \mathbb{R}^M$.
outputs: $\mathbf{x} \in \mathbb{R}^K$.

initialization: $\mathbf{x} \leftarrow \mathbf{0}, \quad \mathbf{r} \leftarrow \mathbf{y}$.
repeat
  Selection : $\widehat{j} = \arg\max_{j\in\{1,..,K\}} | < \boldsymbol{\varphi}^j, \mathbf{r} > |$
  Update :  $\mathbf{x}_{\widehat{j}} = \mathbf{x}_{\widehat{j}} + < \boldsymbol{\varphi}_{\widehat{j}}, \mathbf{r} >$
            $\mathbf{r} = \mathbf{r} - < \boldsymbol{\varphi}_{\widehat{j}}, \mathbf{r} > \boldsymbol{\varphi}_{\widehat{j}}$
until a stopping criterion is met.
```

Note that unless the dictionary has a particular structure, the complexity of the Matching Pursuit algorithm lies in the selection step, and more precisely the computation of the inner products $|\langle\boldsymbol{\varphi}^j, \mathbf{r}\rangle|$. Indeed, this step involves computing K inner products of signals of length M, which takes $O(MK)$ operations, and sort them to extract the index of the selected atom ($O(\log K)$), while the update step takes $O(M)$ operations.

To speed up Matching Pursuit, people have essentially focused on three directions. Firstly, when the dictionary has a particular structure, for example when it corresponds to a fast transform such as a Gabor transform, the computation complexity of the dot products decreases from $O(MK)$ to $O(K \log M)$, and it makes sense to ake advantage of such properties of the dictionary. However, in the general case, it might well be the case that the redundant dictionary at stake does not have such properties and it is necessary to envision other acceleration strategies. To this end, a second route has been to focus on the effectiveness of the update step: once the atom has been selected, the goal is to implement an update step such that the norm of the residual decreases faster than in the case of Matching Pursuit. This is the purpose of the Orthogonal Matching Pursuit algorithm (which was proposed along with Matching Pursuit), as well as many others. These updates are more effective to decrease the norm of the residual, i.e. the magnitude of the decrease induced by such updates is larger than that induced the Matching Pursuit update, while the cost of the refined update usually dominates that of the computations of the inner products. A third direction focuses on reducing the complexity of the selection step, while keeping the simple and efficient update step of Matching Pursuit. This is the strategy undertaken by [5, 6, 7, 8, 9, 10, 11, 12], and this is the strategy that we build our algorithm upon.

Note that we make no assumption on the properties of the dictionary at hand and our goal is to keep every step of this modified Matching Pursuit to a very low complexity, so that the computation of each iteration is fast. As we shall see, we propose an approximate selection step, which may result in sub-optimal selections of atoms. Our modified algorithm thus may be less efficient than Matching Pursuit in terms of the residual norm decrease per iteration. However, a gain in computation time per iteration, resulting in an overall faster algorithm, may be hoped for.

Before stating the algorithm, we want to stress out that although our Stochastic Selection procedure is here proposed in the framework of MP, it obviously carries over any other Pursuits algorithm with a selection step based on the computation of inner products, such as Orthogonal Matching Pursuit, Gradient Pursuits and so on.

2.3. Matching Pursuit with a stochastic selection

Our Matching Pursuit with a Stochastic Selection differs from the classical Matching Pursuit only in the selection step. The Stochastic Selection we propose here reduces the cost of the scalar products computations by proceeding to two dimension reductions:

1. We consider only a sub-dictionary of $\mathbf{\Phi}$ consisting of k atoms chosen uniformly at random among the K original ones.
2. We consider an approximation of each scalar product $\langle\boldsymbol{\varphi}^j, r\rangle$ at stake using only m coordinates of the signals. The coordinates are chosen uniformly at random among $\{1, .., M\}$.

We select the atom in the sub-dictionary yielding the largest approximate magnitude scalar product with the current residual. The update step is exactly the same as in MP: once an atom $\boldsymbol{\varphi}^j$ is selected we use the exact scalar product $\langle\boldsymbol{\varphi}^j, r\rangle$ to update the coefficients vector and the residual. The stopping criteria are also the same as with MP. The algorithm is described in details in Algorithm 2.

Algorithm 2 Matching Pursuit with Stochastic Selection

```
inputs: $\mathbf{\Phi} \in \mathbb{R}^{M\times K}, \mathbf{y} \in \mathbb{R}^M, m \in \{1, .., M\}, k \in \{1, .., K\}$.
outputs: $\mathbf{x} \in \mathbb{R}^K$.

initialization: $\mathbf{x} \leftarrow \mathbf{0}, \quad \mathbf{r} \leftarrow \mathbf{y}$.
repeat
  Randomly pick a set $\mathcal{M}$ of $m$ row indexes.
  Randomly pick a set $\mathcal{K}$ of $k$ column indexes.
  Selection : $\widehat{j} = \arg\max_{j\in\mathcal{K}} | < \boldsymbol{\varphi}^j_\mathcal{M}, \mathbf{r}_\mathcal{M} > |$
  Update :  $\mathbf{x}_{\widehat{j}} = \mathbf{x}_{\widehat{j}} + < \boldsymbol{\varphi}_{\widehat{j}}, \mathbf{r} >$
            $\mathbf{r} = \mathbf{r} - < \boldsymbol{\varphi}_{\widehat{j}}, \mathbf{r} > \boldsymbol{\varphi}_{\widehat{j}}$
until a stopping criterion is met.
```

The algorithm proposed in [12] is a particular case of our algorithm where only a subset of columns is randomly selected. When using only random subset of the atoms in $\mathbf{\Phi}$, one can not ensure we pick the best atom, however, one hopes to find one of the atoms involved in the representation of $\mathbf{y}$. We add here the coordinate selection which implies that we do not at first have access to the correct values of the scalar products even for the selected sub-dictionary. Drawing a new random sub-dictionary and a new random set of coordinates at each iteration however allows us to explore all coordinates and all atoms, thus recover an accurate representation of $\mathbf{y}$.

The selection step of Algorithm 2 takes $O(mk)$ operations instead of the $O(MK)$ operations needed in MP. The gain in computation time is thus of order MK/mk. This gain is to be balanced with the loss of accuracy due to the sub-optimality of the chosen atoms. We show in the numerical experiments section (Section 3) that overall, this Stochastic Selection procedure allows for faster computations time than MP.

3. NUMERICAL EXPERIMENTS

In this section, we present the results of numerical experiments in order to validate our approach. In a first experiment, we generate a random overcomplete dictionary and study the accuracy of our algorithm in terms of the residual energy decrease as a function of iterations and time. We compare our Matching Pursuit with Stochastic Selection approach (MP-S) to the standard Matching Pursuit (MP) algorithm. We then use a DCT dictionary in order to validate the capacity of the MP-S approach to recover the support of a signal (*exact sparse recovery* problem). Declipping of real audio data is performed in a third experiment using the stochastic selection in OMP.

3.1. Data generation

In the two following subsections, we use signals of length $M = 128$ and two types of dictionaries. We generate on the one hand random dictionaries $\mathbf{\Phi} \in \mathbb{R}^{M\times K}$: each entry is an i.i.d random variable drawn from a standard Gaussian distribution and normalized to have unit l_2-norm atoms. On the other hand, we use standard DCT IV dictionaries. In sections 3.2 and 3.3 the vector $\mathbf{x}$ is a K_0-sparse vector where each nonzero entry is an i.i.d Gaussian random variable.

3.2. Residual energy decrease

In this experiment, we measure the energy of the residual as a function of the iterations of the algorithm and also as a function of runtime of the selection stage. We compare our approach with standard Matching Pursuit and a variant we called MP-S$_0$ of our Matching Pursuit with Stochastic Selection approach. In this variant, a sub-dictionary is extracted at the beginning of the procedure (picking uniformly at random row and column subsets from the entire dictionary). The specificity lies in the fact that we use only this fixed sub-dictionary to identify the best atom at each iteration. Once an atom is selected, we do the same updates as in Algorithm 1 and 2 (i.e. using the entire information contained in the atom selected). This variant, which has the same runtime complexity as MP-S, plays the role of a baseline to demonstrate the usefulness of renewing at each iteration the subsets used for the selection step.

For the MP-S procedure, we denote by μ (resp. κ) the ratio m/M (resp. k/K). We ran experiments with a large set of configurations ($\mu, \kappa \in [0.2, 1.0]$). Large values of these parameters imply a decrease per iteration close to MP while the gain in runtime is low. Small values imply a less efficient decrease per iteration but in general a larger gain in runtime of the algorithm. We only report in Figure 1 and 2 the results for the case $\mu \times \kappa = 0.36$ and $K = 512$. We used three configurations of MP-S with equal computational complexity: one with every atoms and a subset of dimensions of size $m = 0.36 \times M$, one with every dimensions and only $k = 0.36 \times K$ atoms and the last one with $\mu = \kappa = \sqrt{\mu \times \kappa}$. Finally, for each experiment we average the Matching Pursuit with Stochastic Selection procedure over 20 runs.

The first result we want to emphasize here is the failure of the MP-S$_0$ approach and, accordingly, the gain induced by our stochastic procedure. Clearly, MP-S$_0$ is only able to decrease the residual energy in the first iterations and quickly reaches an asymptote. On the contrary, with the same runtime complexity, MP-S takes advantage of renewing the subsets at each iteration and does not suffer this lack of information.

Figure 1 shows the results of the experiment for a support of size $K_0 = 32$. As one can see, with MP-S, the residual energy does not decrease as much as with MP at each iteration (top). This behavior can easily be explained because MP-S does not always find the *best* correlated atom at each iteration. However, since iterations of MP-S involve fewer operations than in MP, the residual energy decreases faster (in term of runtime) with MP-S (bottom). We can notice that MP-S outperforms MP whatever the configuration used, with an advantage for the setting $\mu = 1$ and $\kappa = 0.36$ which performs the selection with the real dot products of a subset of atoms. In this configuration, a 120 dB decrease of the residual energy is reached more than twice faster than with MP. This improvement comes from the fact that we only do 36% of the calculations that MP does. Note that this is precisely the setting used in [12].

The behavior is quite different in Figure 2. Here, we are in a case where the support is very sparse since $K_0 = 2$. In this setting, the version of MP-S using every atom but looking only at 36% of the coordinates is the only one that beats MP. Surprisingly, computing the dot product with only 36% of the information seems to be sufficient to find, at each iteration, an atom as good as the one selected by MP. Moreover, MP-S seems to act exactly the same manner as MP in this setting and reaches again the same performance as MP more than twice faster. Here, the approach of [12] ($\mu = 1$ and $\kappa = 0.36$) experiences difficulties to decrease the residual energy. This configuration seems to be too sensitive to the fact that atoms of the support may often not be selected as candidates. The same behavior appear with the last configuration ($\mu = \kappa = 0.6$).

3.3. Exact support recovery

In this section, we evaluate the capacity of our approach to recover the support of a signal and compare it to the standard Matching Pursuit approach. We only look at the *exact sparse recovery* problem i.e. we assume there is no noise: $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$. We use a DCT dictionary. Phase-transition diagrams are drawn in a standard way. The performance criterion is the normalized correlation between original and estimated vectors of coefficients. We consider that the support is recovered if this correlation is above 99%. We have run experiments for various configurations and plot the recovery success as a function of $\delta = M/K$ and $\rho = K_0/M$. For each couple (δ, ρ), we ran the experiment 50 times and stop the procedure before adding the $(K_0 + 1)$-th nonzero entry to the estimated vector of coefficients.

Figure 3 shows the phase-transition diagrams for various values of $\mu \times \kappa$. Recovery is displayed from white (recovery success for all 50 runs) to black (failure for all of them). Here, we don't tweak the parameters and only use the median configuration where $\mu = \kappa$.

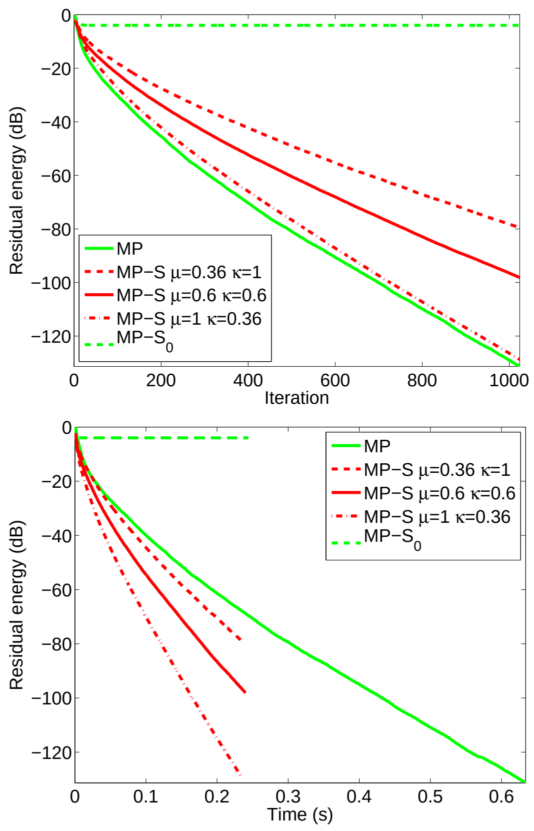


Fig. 1. Residual energy decrease (random dictionary, $K_0 = 32$).

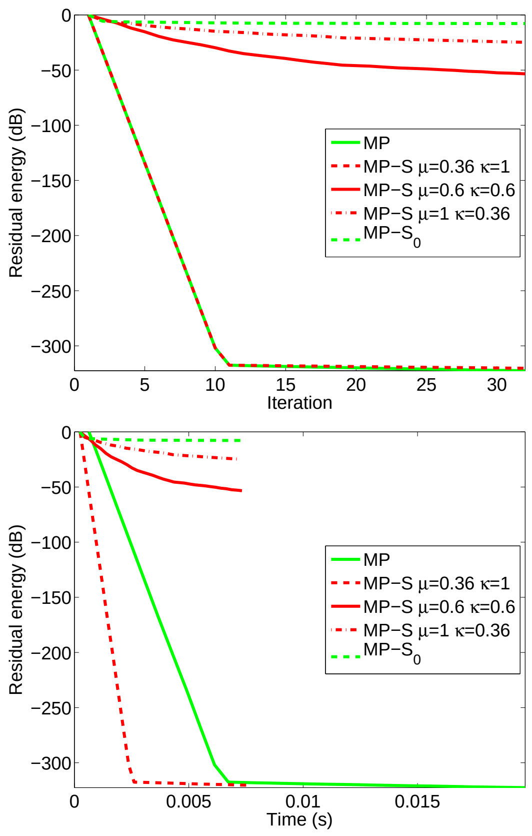


Fig. 2. Residual energy decrease (random dictionary, $K_0 = 2$).

Firstly, the difference is very slight between MP and MP-S, when $\mu \times \kappa = 0.81$ (see (a)) i.e. with subsets of rows and columns close to the ones considered by MP. Secondly, as one decreases the size of the randomly selected subsets, one can notice that the redundancy of the dictionary seems to be a crucial parameter (see (c) and (d)). A theoretical analysis of the algorithm would be needed here to explain this sensitivity.

3.4. Audio declipping application

In order to assess the performance on real data, we have used the stochastic selection in OMP for audio declipping. We have reproduced the declipping experiments on speech sampled at 8kHz proposed in [13] using the code released by the authors. The exact same experimental conditions have been used and are not reported here due to space issues. We used the original OMP and its declipping version (see [13]). Their variants with stochastic selection have been used for $\mu = \kappa = 0.8$ and $\mu = \kappa = 0.6$.

Signal-to-noise ratio (SNR) performance is reported in Table 1 together with measured time complexity. Here, we measure the total time needed by the declipping algorithm (not the time for atom selections only). We observe a very low performance decrease but a significant time saving when using stochastic selections. As a consequence, we see that 1/ the proposed method can deal with real, noisy data; 2/ stochastic selection can be used with OMP; 3/ even with OMP, time saving is significant; and 4/ as SNR is computed on missing data, we measure a quantity that is akin to a generalization performance, as encountered in machine learning, while previous experiments assess the approximation and recovery performance.

4. DISCUSSION AND OUTLOOKS

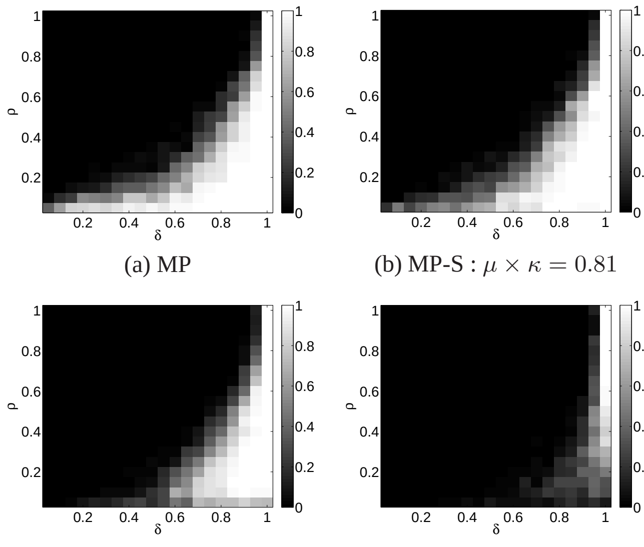
In the present paper, we have proposed a stochastic sampling strategy to accelerate the atom selection step of Matching Pursuit. At each iteration, our approach consists in randomly selecting a set of rows and columns from the dictionary. The set of columns corresponds to the candidate atoms that will be considered for their addition in the signal representation. This means that instead of scanning through all available atoms, we propose to restrain the search to few stochastically selected atoms as in [12]. We generalize this approach with an additional feature: in lieu of computing the correlations between the selected set of atoms and the current residual, we use an estimate of the correlations based on a stochastic downsampling of the atoms and of the residual. In essence, the procedure that we propose performs a random selection of columns and a random selection of rows from the dictionary and only uses the reduced dictionary to perform the selection step. Several numerical simulations illustrate the relevance of our approach both in terms of CPU time gained with respect to classical MP and in terms of the quality of the computed solution. We observe a very advantageous tradeoff between a limited performance decrease and a large time complexity reduction.

Our approach raises a number of questions, which we plan to investigate in the near future.

First —more of an observation than a question— it is important to understand the wild applicability of our approach: even though we have presented it in the context of Matching Pursuit, it obviously

Clipping	OMP	OMP-S$_{0.8}$	OMP-S$_{0.6}$	d-OMP	d-OMP-S$_{0.8}$	d-OMP-S$_{0.6}$
0.4	8.3	8.2	8.2	14.5	13.5	14.0
0.6	13.2	12.7	12.5	18.2	18.2	18.0
0.8	18.5	18.1	17.4	23.9	23.9	23.1
Time (s)	293	238	193	321	264	220

Table 1. Audio declipping performance (SNR in dB) for several clipping levels (rows) and for several algorithms (columns): OMP, declipping OMP (d-OMP) and their variants with stochastic selection (OMP-S$_*$, d-OMP-S$_*$). Last row: average time for processing a sound.



(a) MP (b) MP-S : $\mu \times \kappa = 0.81$

(c) MP-S : $\mu \times \kappa = 0.36$ (d) MP-S : $\mu \times \kappa = 0.09$

Fig. 3. Support recovery as a function of the dictionary redundancy and the support sparsity for different values of the product $\mu \times \kappa$.

applies to many kinds of pursuit algorithms provided that their selection step is based on the computation of correlations.

Second, there is the pressing question of bringing theoretical support to our algorithm. Fueled by the very positive empirical results presented here, we are now to devote our efforts to this aspect of our study. Among the questions that we want to address are the following. Is it possible to state conditions on the dictionary, such as coherence-related conditions, that guarantee i) the convergence of our algorithm and its connections to some form of weak-Matching Pursuit procedures, ii) the exponential decrease of the norm of the residual and iii) the retrieval of atoms that are indeed in the support of the signal considered (in case of *exact-sparse recovery*) ? Is it possible to devise non-uniform selection strategies that would accelerate even more? Answering these questions would require us to formally show that, in addition to (memory) space-related matters, there is a gain (or at least, no loss) in resampling different sets of coordinates at each iteration compared to sampling a random set of coordinates beforehand (as in MP-S$_0$).

Third, we are naturally brought to the question of the connections between our work and the contributions making use of random projections to reduce the sizes of the vectors and matrices they are based on. Randomly selecting coordinates (i.e. rows) of atoms might indeed be viewed as performing a random projection, and viewed this way, the uniform random selection strategy induces certain properties on the corresponding projections (regarding the orthogonality of the projection space): those have to be clearly stated. Down the road, it also leads us to compare our approach with compressed sensing-based Matching Pursuit [14]. A critical difference is again how the random projections are computed: at each iteration for our procedure, and once for all in the latter approach.

5. REFERENCES

[1] S.G. Mallat and Zhifeng Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[2] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad, “Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition,” in *Proc. Asilomar Conf. Signals, Systems, and Computers*, Nov. 1993, pp. 40 –44 vol.1.

[3] T. Blumensath and M.E. Davies, “Gradient pursuits,” *IEEE Trans. Sig. Proc.*, vol. 56, no. 6, pp. 2370 –2382, june 2008.

[4] J.A. Tropp, “Greed is good: algorithmic results for sparse approximation,” *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[5] V.N. Temlyakov, “Weak greedy algorithms,” *Adv. Comput. Math.*, vol. 12, no. 2-3, pp. 213–227, 2000.

[6] R. Gribonval and M. Nielsen, “Approximate weak greedy algorithms,” *Advances in Computational Mathematics*, vol. 14, pp. 361–378, 2001, 10.1023/A:1012255021470.

[7] R. Gribonval and P. Vandergheynst, “On the exponential convergence of matching pursuits in quasi-incoherent dictionaries,” *Information Theory, IEEE Transactions on*, vol. 52, no. 1, pp. 255 – 261, jan. 2006.

[8] E.J. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[9] D.L. Donoho, “Compressed sensing,” *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[10] P. Jost, P. Vandergheynst, and P. Frossard, “Tree-based pursuit: Algorithm and properties,” *Signal Processing, IEEE Transactions on*, vol. 54, no. 12, pp. 4685–4697, 2006.

[11] S. K. Tjoa and K. J. Ray Liu, “Factorization of overlapping harmonic sounds using approximate matching pursuit,” in *Proc. of ISMIR*, 2011.

[12] M. Moussallam, L. Daudet, and G. Richard, “Matching pursuits with random sequential subdictionaries,” *CoRR*, vol. abs/1107.2509, 2011.

[13] A. Adler, V. Emiya, M. G. Jafari, M. Elad, R. Gribonval, and M. D. Plumbley, “Audio inpainting,” *IEEE Trans. Audio, Speech and Lang. Proc.*, vol. 20, no. 3, pp. 922 –932, 2012.

[14] D. Needell and J. A. Tropp, “Cosamp: Iterative signal recovery from incomplete and inaccurate samples,” Tech. Rep., California Institute of Technology, Pasadena, 2008.