# SAGI: Sparsification Algorithm using Greedy Iteration

Scott Rickard, Ruairí de Fréin, Conor Fearon
*Sparse Signal Processing Group*
*University College Dublin*
*Ireland*

### Abstract

We introduce a method, called SAGI (Sparsification Algorithm using Greedy Iteration), for making a representation of a signal more sparse in an over-complete dictionary in a greedy manner. The sparsification is achieved by iteratively increasing the magnitude of the largest signal coefficient and simultaneously reducing the other signal coefficients so as to maximize the sparsity of the representation while maintaining invariant the reconstruction of the signal from the coefficients. Any measure of sparsity can be used with the method. Two versions are presented. The first, the one-pass version, considers each coefficient once in order from largest to smallest. The second, the exhaustive version, only considers the $(n+1)$st largest coefficient if iteratively considering the $n$ largest coefficients results in no increase in sparsity.

## 1 Brief Overview

In general, *frames* provide for a stable and redundant representation of signals and can be thought of as an overcomplete dictionary, a generalization of the concept of a basis [1]. A frame is formally defined as:

**Definition 1.** *$F = \{\vec{f_i}, i \in I\}$ is a frame if $\exists A, B : 0 < A \leq B < \infty$, such that $\forall \vec{x} \in H$, a Hilbert space,*

$$A||\vec{x}||^2 \leq \sum_{i \in I} |\langle \vec{x}, \vec{f_i} \rangle|^2 \leq B||\vec{x}||^2 \tag{1}$$

Given a frame $F$, there exists another (dual) frame $\tilde{F} = \{\vec{\tilde{f}}_i, i \in I\}$ such that for all $\vec{x} \in H$,

$$\vec{x} = \sum_{i \in I} \underbrace{\langle \vec{x}, \vec{f_i} \rangle}_{c_i} \vec{\tilde{f}}_i \tag{2}$$

This property is extremely useful as one can represent the signal without loss of information using the coefficients, $c_i$.

Often, the set of coefficients can be used to determine useful attributes of the signal (eg, when the $\vec{f_i}$ perform Fourier analysis) and thus the $\vec{f_i}$ are called *analysis* functions. Similarly, as the $\vec{\tilde{f}}_i$ are used to re-synthesize the signal from the coefficients, the $\vec{\tilde{f}}_i$ are called *synthesis* functions.

In this work, for simplification of notation and exposition, we consider the Hilbert space of signals to be $N$-dimensional vectors, $\vec{x} = (x_1, x_2, \ldots, x_N)^T$, where $(\,)^T$ denotes transpose. Then the analysis functions $\vec{f_i}$ are also in $\mathbb{C}^N$ and we can stack the $M$ analysis vectors in rows to create an $M$-by-$N$ dimensional matrix $F$ so that the analysis process is,

$$\vec{c} = F\vec{x} \tag{3}$$

where $\vec{c}$ is the resulting $M$ dimensional coefficient vector created by the $M$ analysis vectors. The synthesis process is then

$$\vec{x} = \tilde{F}^T \vec{c} \tag{4}$$

where $\tilde{F}$ is an $M$-by-$N$ dimensional matrix constructed by stacking the $M$ synthesis vectors row by row.

As frames yield, in general, redundant representations, one can choose the coefficients that satisfy Equation 2 with a secondary constraint in mind. In this paper, we investigate the constraint that the coefficients be chosen so as to make the representation as sparse as possible. Sparse representations are desirable in, for example, coding and source separation algorithms[2; 3].

## 2 Other Methods

Several methods have been proposed for constructing the sparsest representation of a signal using the overcomplete dictionary approach, these include Matching Pursuit (MP), Basis Pursuit (BP) and Method Of Frames (MOF). Let us begin by looking at these.

MP as presented by [4], is a method for generating an approximate decomposition of the signal that tackles the sparsity issue directly. An iterative approach is used to build up a representation from an initial approximation $\vec{x}^{(0)} = 0$ and a residue $R^{(0)} = \vec{x}$. The algorithm proceeds as follows: At each iteration $k$, the atom from the analysis frame, that yields the highest inner-product after being projected onto the signal, is taken. The corresponding coefficient $c_k = \langle R^{(k-1)}, \vec{f}_{i_k} \rangle$ is computed. The selected atom is then scaled by its coefficient and added to current approximation of the signal i.e. $\vec{x}^{(k)} = \vec{x}^{(k-1)} + c_k \vec{f}_{i_k}$. The resulting residue is then updated $R^{(k)} = \vec{x} - \vec{x}^{(k)}$ and the next iteration is performed.

MP achieves an optimal decomposition when the dictionary is an orthogonal basis i.e. there is no redundancy and hence there is only one possible representation of the signal. For example if the signal is made up of only $m \ll n$ atoms, the $m^{th}$ iteration of the algorithm will yield the sparse structure exactly. Yet when redundancy is added to the dictionary, it is clear that if an atom is chosen erroneously in one of the initial steps due to the multiple possible reconstructions of the signal, then a less sparse representation of the signal will be the out come as the following steps are spent compensating for the error.

The BP approach as described in [5], differs from MP in that the algorithm starts with a set of atoms that form a feasible basis for the signal and then it iteratively improves the basis and hence the sparsity by swapping one atom in the dictionary for another which is not in the dictionary . The criterion for selecting the "swapped-in" atoms is that a minimally sparse representation of the signal is constructed whose coefficients have minimal $\ell^1$-norm.i.e. $||\vec{c}||_1$ subject to $\Phi\vec{c} = s$.

The Moore-Penrose inverse is employed to find the coefficient vector among all solutions of $\tilde{F}\vec{c} = \vec{x}$, with the minimum $\ell^2$-norm in MOF discussed in [6]. The following equation is solved for a unique solution denoted $\vec{c}^{\dagger}$:

$$\vec{c}^{\dagger} = F^{\dagger}\vec{x} = (F^T F)^{-1} F^T \vec{x} \tag{5}$$

where $\vec{c}^{\dagger}$ is minimized with respect to the $\ell^2$-norm subject to $\tilde{F}\vec{c}^{\dagger} = \vec{x}$.

The pseudo-inverse $\tilde{F}$ is computed for a given analysis frame $F$. This is a one step calculation minimized with respect to the $\ell^2$-norm as opposed to the $\ell^1$-norm used in BP. One of the major disadvantages of this method is the fact that $F^T(FF^T)^{-1}$ may become computationally infeasible depending on the size of the given analysis frame $F$. Secondly, due to the fact that each atom in the dictionary that is correlated with the signal, it is usually a member of the solution, the MOF can yield a sub-maximally sparse representation, due to the redundancy in the dictionary.

## 3 SAGI

Explicitly, our goal is to:

**Problem 1.** *For a given $\vec{x}$, find the $\vec{c}$ such that $\vec{x} = \tilde{F}\vec{c}$ and sparse measure $s(\vec{c})$ is minimal.*

The typical choice for $s$ is the $\ell^p$-norm, which we will denote $s_p$,

$$s_p(\vec{c}) = \left( \sum_{m=1}^{M} |c_m|^p \right)^{1/p}. \tag{6}$$

When $0 < p < 1$, $s_p$ is a quasi-norm as it fails to satisfy the triangle inequality, and we will refer to it as the $\ell^p$-score. When $p = 0$, $s_0(\vec{c})$ counts the number of nonzero entries of $\vec{c}$, and $s_p$ is not even linear with respect to scalar multiplication.

An alternative measure of sparsity is the Gini index, which was originally proposed as a measure of the inequity of wealth distribution in a population[7]. Given coefficient data, $\vec{c} = \{c_1, c_2, \dots c_N\}$, we order from smallest to largest, $|c_{(1)}| \leq |c_{(2)}| \leq \cdots \leq |c_{(N)}|$ where $(1), (2), \dots, (N)$ are the indices of the sorting operation. We define the parameterized-Lorenz curve as,

$$L_p\left(\frac{i}{N}\right) = \sum_{j=1}^{i} \frac{|c_{(i)}|^p}{\sum_{k=1}^{N} |c_k|^p}, \quad \text{for } i = 0, \dots, N. \tag{7}$$

$L_p$ is piecewise linear with $N+1$ points with support $(0,1)$ and $L_p(0) = 0$ and $L_p(1) = 1$. With $p = 2$, each point on the Lorenz curve $(x = a_0, y = b_0)$ has the interpretation that $100 \times a_0$ percent of the sorted signal coefficients captures $100 \times b_0$ percent of the total signal power and thus the slower the curve rises to 1, the fewer coefficients are needed to accurately represent the signal. The area between the Lorenz curve and the 45 degree line will increase as the sparsity of the signal increases. The area beneath the Lorenz curve is,

$$A(\vec{c}) = \frac{1}{2N} \sum_{n=1}^{N} \left( L\left(\frac{n-1}{N}\right) + L\left(\frac{n}{N}\right) \right) \tag{8}$$

and twice the area between the Lorenz curve and the 45 degree, which is known as the Gini index [7], is then simply,

$$G_p(\vec{c}) = 1 - 2A(\vec{c}), \tag{9}$$

which has several nice properties including $G_p(\vec{c}) = 0$ for $\vec{c} = (x, x, \dots, x)$ for constant $x \neq 0$ and $G_p(\vec{c}) \rightarrow 1$ for $\vec{c} = (x, 0, \dots, 0)$ as the number of zeros goes to infinity for constant $x \neq 0$. As the $\ell^p$-score decreases for more sparse solutions, we, in practice, minimize $1 - G_p(\vec{c})$ for the Gini index measure of sparsity so that we can still use the general minimization problem as formulated above.

The sparsification algorithm which we propose below, is independent of the sparsity measure $s$ and the results presented in the next section compare the performance of the algorithm when using either $\ell^{0.1}$-score or the Gini index.

The Sparsification Algorithm using Greedy Iteration (SAGI) method is based on the following Theorem.

**Theorem 1.** *Given $\vec{x}, F, \tilde{F}, \vec{c}$ such that $\vec{x} = \tilde{F}^T\vec{c}$, then vector $\vec{d}$,*

$$\vec{d}_i(j,\alpha) = \begin{cases} c_i - \alpha\langle \vec{\tilde{f}}_j, \vec{f}_i\rangle & i \neq j \\ c_i - \alpha\langle \vec{\tilde{f}}_j, \vec{f}_i\rangle + \alpha & i = j \end{cases} \quad (10)$$

*satisfies the reconstruction equation $\vec{x} = \tilde{F}^T\vec{d}_i(j,\alpha)$ for all $\alpha \in \mathbb{R}$ and $j \in I$.*

*Proof.* We can write,

$$\vec{d}_i(j,\alpha) = c_i - \alpha\langle \vec{\tilde{f}}_j, \vec{f}_i\rangle + \alpha\delta_{i-j} \quad (11)$$

where $\delta_{i-j}$ is the Kronecker function, $\delta_0 = 1$ and $\delta_i = 1$ for $i \neq 0$. We have,

$$\begin{aligned} \tilde{F}^T\vec{d}_i(j,\alpha) &= \tilde{F}^T(c_i - \alpha\langle \vec{\tilde{f}}_j, \vec{f}_i\rangle + \alpha\delta_{i-j}) \\ &= \tilde{F}^T c_i - \tilde{F}^T\alpha\langle \vec{\tilde{f}}_j, \vec{f}_i\rangle + \tilde{F}^T\alpha\delta_{i-j} \\ &= \vec{x} - \alpha\vec{\tilde{f}}_j + \alpha\vec{\tilde{f}}_j \\ &= \vec{x} \end{aligned}$$

and we are done. □

The theorem states that altering the magnitude of the $j$th component of the reconstruction can be compensated for by altering the magnitude of all the components that are correlated with it. The SAGI method proceeds as follows, given $\vec{x}$, $\tilde{F}$, sparsity measure $s$, and threshold $\alpha_0$, set initially $J = I$ (the indices of all synthesis vectors) and $\vec{c} = \tilde{F}(\tilde{F}^T\tilde{F})^{-1}\vec{x}$. Then:

1. While there are remaining components to be considered $J$, select,
$$i_m = \text{argmax}_{i\in J}(|c_i|). \quad (12)$$
2. Determine the $\alpha$ maximizes sparsity,
$$\alpha_m = \text{argmax}_\alpha s(\vec{d}_i(i_m, \alpha)). \quad (13)$$
3. Update $\vec{c}$,
$$\vec{c} = \vec{d}_i(i_m, \alpha_m). \quad (14)$$
4. Update the remaining components to be considered,
$$J = \begin{cases} J - i_m & \alpha \leq \alpha_0 \\ I - i_m & \alpha > \alpha_0 \end{cases} \quad (15)$$
5. Goto 1.

When the threshold $\alpha_0 = \infty$, the one-pass version of the algorithm results. When $\alpha_0 = 0$, the exhaustive version of the algorithm results. In both versions, the method chooses the largest component that has not been eliminated from consideration and the magnitude of this component is then altered to maximize sparsity while maintaining the reconstruction property of the coefficients by altering the values of the other coefficients to compensate. Then this component is eliminated from consideration. In the one-pass version, each component is considered once, in the order of magnitude. In the exhaustive version, the $j$th largest component is only considered if the previous $j-1$ component passes each resulted in $\alpha_m = 0$. Thus, the method 'iteratively' and 'greedily' chooses the largest component not yet considered and tries to alter it to maximize sparsity.

For some measures of sparsity, step 3 may have a closed form solution. However, in general, we determine $\alpha_m$ using Nelder-Mead unconstrained nonlinear minimization, as implemented in MATLAB (fminsearch).

## 4 Results and Discussion

An initial investigation into SAGI was performed in $2D$ space to provide an illustration of the operation of the algorithm. Four synthesis atoms were randomly generated by picking random unit norm vectors. A subset of synthesis vectors (1 or 2) were then 'turned on' by setting the corresponding coefficients to 1, and a signal constructed from the coefficients. This signal, was passed into SAGI along with the synthesis matrix $\tilde{F}$ and the algorithm was run with both the $\ell^{0.1}$-score and the Gini index measures of sparsity. For these 2D tests, the simple and exhaustive versions of SAGI yielded identical results. The sparsity of the SAGI solution was compared to the sparsity of the Moore-Penrose solution and set of coefficients used to generate the signal (the 'cheat' solution). As one or two synthesis vectors were 'on' for each test, the cheat solution was a four dimensional vector with 1 or 2 ones and the remaining values zero. In order to test the stability of the SAGI solution, the initial coefficients fed into SAGI were set to a linear combination of the cheat and Moore-Penrose solution,

$$\vec{c} = (1-\beta)\vec{c}_{\text{cheat}} + \beta\vec{c}_{\text{MP}}. \quad (16)$$

Thus, with $\beta = 1$, the initial coefficients are the Moore-Penrose coefficients, which is the starting point for SAGI as described.

Figure 1 illustrates the average over 20 random tests of the cost values obtained for SAGI when one synthesis vector is turned on using the Gini index in the upper plot and the $\ell^{0.1}$-score in the lower plot. Using the Gini index, the average final cost is equal to the cheat cost. This indicates that the correct coefficients have been identified and the sparsest solution has been obtained (with respect to the Gini index). The average initial cost increases steadily with $\beta$ as the initial coefficients consist of a higher fraction of the Moore-Penrose coefficients. In the lower part of the figure the average final cost is once again equivalent to the cheat cost,

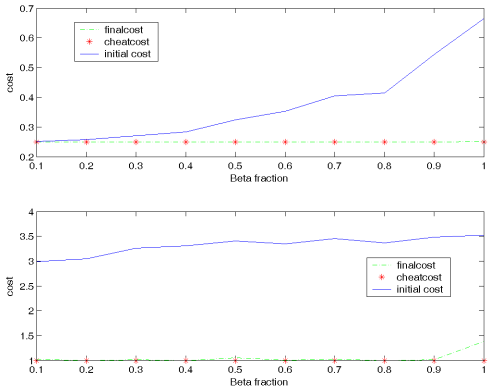

Fig. 1: This figure shows the average final cost, initial cost and cheat cost obtained when SAGI is performed on a signal which consists of one synthesis atom. The costs in the upper graph were calculated using the Gini index and the $\ell^{.01}$-score was used in the lower plot.

save for the final value of $\beta$ where the Moore-Penrose coefficients are the initial coefficients. This means that SAGI fails to find the sparsest solution when using the $\ell^{0.1}$-score as a measure of sparsity. In both plots a significant reduction in cost from the average initial cost is shown. The sparsity of the solution also varies as a function of $\beta$ yet in all cases sparsification has been achieved.

In Figure 2, the results from the tests with two synthesis atoms turned on are shown. In this case, when the Gini index is used as a measure of sparsity (upper plot), the SAGI coefficients are always more sparse than both the initial and cheat coefficients. This is not the case when the $\ell^{0.1}$-score is used (lower plot). While SAGI does increase the sparsity of the solution from the initial coefficients, on average it does not find a solution as sparse as the two 1's two 0's coefficients used to generate the signals.

## 5 Conclusions

We have formulated a greedy sparsification algorithm that iteratively alters the magnitude of the largest coefficient not yet considered while compensating so as to guarantee perfect reconstruction. At each step, the modification is made so as to maximize the sparsity of the coefficients. Initial tests on 2D random data show that the algorithm is perhaps better suited to the Gini index measure of sparsity, as opposed to the $\ell^{0.1}$-score measure. Further tests and analysis including comparing the method to other techniques are needed to better understand the algorithm.

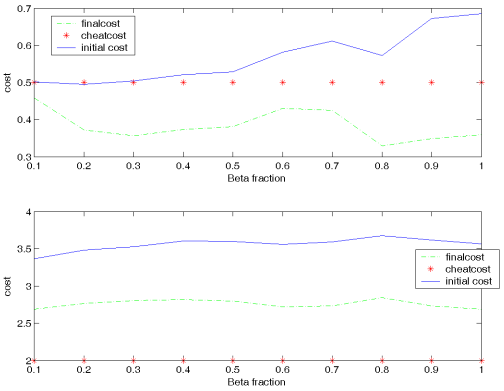

Fig. 2: This figure shows the average final cost, initial cost and cheat cost obtained when SAGI is performed on a signal which consists of two synthesis atoms. The costs in the upper graph were calculated using the Gini index and the $\ell^{.01}$-score was used in the lower plot.

## References

[1] O. Christensen, *An introduction to Frames and Reisz Bases*. Boston: Birkhäuser, 2003.

[2] D. L. Donoho, "Sparse components analysis and optimal atomic decompositions. constructive approximation," *Constructive Approximation*, vol. 17, pp. 353–382, 2001.

[3] J. Karvanen and A. Cichoki, "Measuring sparseness of noisy signals," in *ICA03*, 2003.

[4] S. G. Mallat and Z. Zhang, "Matching pursuit with time frequency dictionaries," *IEEE Transcations on Signal Processing*, vol. 41, pp. 3397 – 3415, 1993.

[5] S. Chen, D. Donoho, and M. Saunders, "Atomic decompostition by basis pursuit," *SIAM J. Comput.*, vol. 20, pp. 33 – 61, 1998.

[6] I. Daubechies, "Time-frequency localization operators: a geometric phase space approach," *IEEE Transcations on Information Theory*, vol. 34, pp. 605 – 612, 1998.

[7] C. Gini, "Measurement of inequality of incomes," *Economic Journal*, vol. 31, pp. 124–126, 1921.