ERROR CORRECTION VIA SMOOTHED L0-NORM RECOVERY

*Saman Ashkiani, Massoud Babaie-Zadeh**

Department of Electrical Engineering, Sharif University of technology, Tehran, Iran
sa.ashkiani@gmail.com
mbzadeh@sharif.edu

Christian Jutten

GIPSA-LAB and Institut Universitaire de France
Grenoble, France
christian.jutten@inpg.fr

ABSTRACT

Channel coding has been considered as a classical approach to overcome corruptions occurring in some elements of input signal which may lead to loss of some information. Proper redundancies are added to the input signal to improve the capability of detecting or even correcting the corrupted signal. A similar scenario may happen dealing with real-field numbers rather than finite-fields. This paper considers a way to reconstruct an exact version of a corrupted signal by using an encoded signal with proper number of redundancies. The proposed algorithm uses Graduated Non-Convexity method beside using a smoothed function instead of ℓ^0-norm to correct all the corrupted elements. Simulations show that our proposed algorithm substantially improves the probability of exact recovery in comparison to previous algorithms.

Index Terms— Real-field coding, Compressed Sensing, Sparse Signal Processing.

1. INTRODUCTION

In transmission of a digital signal through a communication channel, the classic approach to overcome probable noise contaminations is encoding the main signal at the transmitter and decoding at the receiver. In this paper we consider *real-field coding* [1] (in contrast to finite-field coding), in which the signal to be encoded can have any real value, i.e. the input signal $\mathbf{s}^* \in \mathbb{R}^n$. This input signal is encoded linearly by a $m \times n$ matrix, $\mathbf{A}$, where throughout this article we assume $m > n$. Then the transmitter sends $\mathbf{As}^*$ through the channel and the receiver receives:

$$\mathbf{x} = \mathbf{As}^* + \mathbf{e}^* \tag{1}$$

where $\mathbf{e}^* \in \mathbb{R}^m$ is an error vector and $\mathbf{x} \in \mathbb{R}^m$ is the corrupted received signal. Now the main question is that: can we reconstruct a clean version of the input signal by knowing $\mathbf{x}$ and $\mathbf{A}$? The answer is no in general cases. But if only a small fraction of $\mathbf{As}^*$ is corrupted (i.e., $\mathbf{e}^*$ is a sparse vector) the input signal can be recovered exactly by using a proper recovery scheme.

*This work has been partially funded by Iran Telecom Research Center (ITRC).

As $\mathbf{A}$ is a tall matrix (i.e., $m > n$), inspired by the classical method which is used in coding theory, one can use the left annihilator of $\mathbf{A}$ (i.e., the matrix $\mathbf{B} \in \mathbb{R}^{(m-n)\times m}$ such that $\mathbf{BA} = \mathbf{0}$ which is also known as the parity check matrix in coding theory) to find an estimation of $\mathbf{e}^*$ as follows:

$$\min_{\mathbf{e}} \|\mathbf{e}\|_0 \quad \text{s.t.} \quad \mathbf{Be} = \mathbf{Bx} \triangleq \mathbf{r}. \tag{2}$$

This minimization can be solved by utilizing algorithms which are proposed to find the sparsest solution of the underdetermined system of linear equations in (2) (e.g., *Basis Pursuit (BP)* [2] and *Smoothed ℓ^0 (SL0)* [3]). Then by using the pseudo-inverse of $\mathbf{A}$, the input signal can be recovered. Nonetheless, this method may have some disadvantages. Firstly, the recovery is done in two steps: Finding an estimation of $\mathbf{e}^*$ and then solving an overdetermined system of linear equations by using the preceded estimation. The problem is that, any error occurring in the process of finding $\mathbf{e}^*$ would propagate to the next step. This problem would be intensified by using less strict algorithms for solving (2). Furthermore, there may exist situations where the constraint in (2) grows in size, hence it makes the minimization difficult. Generally speaking, we would prefer eliminating constraints in optimization problems, if possible.

Candès et al. [1] introduced an equivalent method which does not have the above problems. Consequently, the recovery is achieved by solving the following unconstrained minimization and only in a single step :

$$\min_{\mathbf{s}} \|\mathbf{x} - \mathbf{As}\|_0. \tag{3}$$

However, ℓ^0-norm is not a continuous function, so finding the solution of (3) needs to do a combinatorial search which becomes intractable as signal's dimension grows. By this reason one can prefer to minimize ℓ^1-norm instead, since it results in a convex optimization problem. In [1] an algorithm is proposed (i.e., *ℓ^1-norm recovery*) in which, ℓ^0-norms in (2) and (3) are replaced by ℓ^1-norm. However, using ℓ^1-norm has some consequences. In fact, for underdetermined system of linear equations in (2), an extra condition on the sparsity of $\mathbf{e}$ is required for leading to the unique sparsest solution [4] in (3). By using ℓ^1-norm a tighter condition should hold, which

may deprive us from an exact recovery for special cases, especially cases where there are a large number of corruptions.

In this paper we propose a new algorithm for decoding a real-field encoded signal, inspired from the idea of SL0 [3], and we name it *Smoothed ℓ^0-norm recovery*, in which the non-continuous ℓ^0-norm in (3) is replaced by a proper smoothed cost function. By using our algorithm, we can avoid the mentioned problems in solving a constrained minimization in (2). Moreover, significant improvement is achieved without affecting the order of complexity in comparison with ℓ^1-norm recovery, in the sense that a larger number of corruptions can be tolerated for reconstruction. However, with a little degradation in the performance (still superior to the ℓ^1-norm), highly faster recovery time can be obtained. In addition, *contrary to SL0* [3], our method needs to solve an *unconstrained* minimization problem, so it does not have all the complexities which was forced by using a gradient projection approach in implementation of SL0. Hence it is heuristically expected that Smoothed ℓ^0-norm recovery be less likely to be trapped in local minima in comparison with SL0 applied on (2).

This paper is organized as follows. In Section 2, we review some basic materials which are required in the rest of the paper. In section 3, we propose our algorithm and state some details about its performance. In section 4, we compare Smoothed ℓ^0-norm recovery with ℓ^1-norm recovery algorithm in their ability to recover corrupted encoded signals.

2. PRELIMINARIES

2.1. Uniqueness of the recovery

In general, it is not possible to exactly recover all corrupted encoded signals. In fact, $\mathbf{e}^*$ should be sparse enough in order to enable us to perform the reconstruction by solving (3). Candès et al. [1] showed that ℓ^1-norm versions of (2) and (3) are equivalent. By a similar line of reasoning it can easily be shown that this equivalency exists in the case of ℓ^0-norm (i.e., solving (3) and (2) are equivalent, which is in fact a well-known result in coding theory). Consequently, reconstruction by utilizing (3) would lead to a unique solution as far as the underdetermined system of linear equations in (2) possesses the unique sparsest solution.

2.2. Smoothed ℓ^0-norm

In [3] it is suggested to use a smoothed function to approximate ℓ^0-norm with the nice property to be continuous and also differentiable. In this paper, Gaussian smoothing function is used. Consequently, for any $\mathbf{y} \in \mathbb{R}^m$ we replace the ℓ^0-norm of $\mathbf{y}$ by:

$$\|\mathbf{y}\|_0 = m - \lim_{\sigma \to 0} \sum_{i=1}^{m} \exp\left(-\frac{y_i^2}{2\sigma^2}\right) \tag{4}$$

2.3. Graduated Non-Convexity

Graduated Non-Convexity (GNC) [5] is a deterministic minimization algorithm for finding the global minimum of a non-convex, unconstrained and continuous cost function like (4). Suppose we want to minimize $F(\mathbf{s}\,;\sigma)$ where $F : \mathbb{R}^n \to \mathbb{R}$, $\mathbf{s}$ is an n-dimensional real signal and σ is a real parameter. Our goal is to minimize $F(\mathbf{s}\,;\sigma)$ for a small σ (say σ_{target}). To achieve this goal and as an initial step, σ_0 is set large enough to make $F(\mathbf{s}\,;\sigma_0)$ strictly convex. Hence we can find the minimum value of the convex cost function by a simple local minimization algorithm (e.g., *steepest descent*) regardless of the initial value for $\mathbf{s}$. The gained minimizer $\mathbf{s}_{\sigma_0}$ at this step is set as an initial value for the next step to find $\mathbf{s}_{\sigma_1}$, where $\sigma_1 < \sigma_0$. As σ is decreased, $F(\mathbf{s}\,;\sigma)$ becomes non-convex and local minima may occur. At each step i, the minimum of $F(\mathbf{s}\,;\sigma_i)$ is used as a starting point to locate the minimum for the next (smaller) σ_{i+1} , using a steepest descent approach (i.e., for each i, $\sigma_{i+1} < \sigma_i$, which ends at the final value of σ_{target}). Since the value of σ_{i+1} has only slightly decreased, we hope that the minimizer of $F(\mathbf{s}\,;\sigma_{i+1})$ is not too far from the minimizer for the previous (larger) σ_i, and hence we hope not to get trapped into a local minimum. It is also important to note that the GNC approach is a heuristic method which cannot provide a mathematical guarantee to find the global minimizer in the general case. But by decreasing σ slowly enough, we can achieve good results, as it is experimentally shown in results of Section 4.

3. SMOOTHED ℓ^0-NORM RECOVERY

3.1. The main idea

The main objective of this article is to find the sparsest solution of (3) by using a proper smoothed cost function like (4) instead of the ℓ^0-norm. We define our cost function as:

$$F(\mathbf{s}\,;\sigma) \triangleq m - \sum_{i=1}^{m} \exp\left(-\frac{(x_i - \mathbf{a}_i\mathbf{s})^2}{2\sigma^2}\right) \tag{5}$$

where $\mathbf{a}_i$ denotes the i-th row of $\mathbf{A}$. The objective minimization argument becomes:

$$\min_{\mathbf{s}} F(\mathbf{s}\,;\sigma) \tag{6}$$

for $\sigma = \sigma_{\text{target}}$ where σ_{target} is very small. The function in (6) is non-convex, hence we are facing a continuous and differentiable function with lots of local minima. By using GNC idea we try to find an approximate solution for global minimum of (6), which we expect to be a proper solution for (3), our original problem. Initially, as will be explained in the next section, the algorithm starts from a very large σ, for which the global minimum of (6) can be calculated. Then, the recovery is done using two nested iterations. The internal loop is a simple steepest descent method for minimizing $F(\mathbf{s}\,;\sigma)$ for a fixed σ. In the external loop and based on the previous statements about GNC, σ is decreased gradually in each step.

- Input: Coding matrix $\mathbf{A}$, corrupted entries $\mathbf{x}$ and σ_{target}
- output: $\mathbf{s}_{\text{min}}$
- Initialization:
  1. $\mathbf{s}^{(0)} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{x}$
  2. $\mathbf{e}^{(0)} = \mathbf{x} - \mathbf{A}\mathbf{s}^{(0)}$
  3. Set $\sigma_0 > k\|\mathbf{x} - \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{x}\|_\infty$, where k is a controlling parameter.
  4. Constant parameters κ and μ_0.
  5. $i = 0$
- **while** $\sigma_i > \sigma_{\text{target}}$ **do**
  1. if $i > 0$: $\sigma_i \leftarrow \kappa\sigma_{i-1}$
  2. minimize $F(\mathbf{s}\,;\sigma_i)$ (steepest descent method)
     - $\mathbf{s} \leftarrow \mathbf{s}^{(i)}$
     - **For** $l = 1, \ldots, L$
       (a) $\mathbf{z} \leftarrow \left[e_1 \exp(-\frac{e_1^2}{2\sigma_i^2}, \ldots, e_m \exp(-\frac{e_m^2}{2\sigma_i^2})\right]$
       (b) $\mathbf{s} \leftarrow \mathbf{s} - \mu_0\mathbf{A}^T\mathbf{z}$
       (c) $\mathbf{e} \leftarrow \mathbf{x} - \mathbf{A}\mathbf{s}$
  3. $i \leftarrow i + 1$
  4. $\mathbf{s}^{(i)} \leftarrow \mathbf{s}$
- **return**: $\mathbf{s}_{\text{min}} \leftarrow \mathbf{s}^{(i)}$

Fig. 1. Smoothed ℓ^0-norm recovery algorithm.

We repeat previous actions and at each step the minimizer $\mathbf{s}$ is set as an initial value for the next step until the σ_{target} is reached. The final algorithm is detailed in pseudocode form in Fig. 1.

3.2. Initialization

In order to use GNC effectively, we have to choose a sufficiently large σ_0, i.e. theoretically infinity, and an appropriate $\mathbf{s}^{(0)}$, where the superscript denote the iteration number, so we can use steepest descent to find an approximate global minimum of (6). By setting $\nabla F(\mathbf{s}) = \mathbf{0}$ we have:

$$\mathbf{A}^T\left[e_1\left(\exp\left(-\frac{e_1^2}{2\sigma^2}\right)\right), \ldots, e_m\left(\exp\left(-\frac{e_m^2}{2\sigma^2}\right)\right)\right]^T = \mathbf{0}. \tag{7}$$

Now, by having $\sigma \to \infty$:

$$\mathbf{A}^T\mathbf{e} = \mathbf{0} \Longrightarrow \mathbf{A}^T(\mathbf{x} - \mathbf{A}\mathbf{s}) = \mathbf{0} \Longrightarrow \mathbf{s} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{x}. \tag{8}$$

So, initially we choose $\mathbf{s}^{(0)} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{x}$. Now, we want to choose a practical σ_0 for our simulations which acts like infinity. Regarding to the fact that we want our cost function to be convex near $\mathbf{s}^{(0)}$, it is necessary that $\nabla F^2(\mathbf{s}^{(0)})$ be positive definite. It is easy to show that this leads to $\sigma > |e_i^{(0)}|$, $\forall i = 1, \ldots, m$, where $e_i^{(0)}$ is the i-th component of $\mathbf{e}^{(0)} \triangleq \mathbf{x} - \mathbf{A}\mathbf{s}^{(0)}$. Moreover, by choosing 3 to 5 times of this threshold, we can also be assured that $\exp(-e_i^{(0)}/(2\sigma_0^2)) \approx 1$ and acts as if $\sigma \to \infty$.

3.3. Further remarks about the algorithm

In this part some important remarks about the algorithm of Fig. 1 are stated for further clarifications.

1. In theory, we should have $\sigma_{\text{target}} \to 0$, but for our algorithmic implementation we should choose a practical small σ_{target} as a stopping criterion for our iterative algorithm. The smaller σ_{target} we choose, the better quality we will have in recovery.

2. As it was stated before, it is very important to decrease σ in a proper manner in order not to be trapped in a wrong local minimum. The decrease of σ toward σ_{target} is controlled by a decrease factor $0 < \kappa < 1$. However κ can control the balance between the quality of the recovery on one hand and computational load on the other hand. A too fast decrease in σ can also result in a wrong local minimum and lack of convergence.

3. Steepest descent method is chosen as a minimization method in our algorithm. However, the internal loop can be repeated for a fixed and small number of times (L). In other words by decreasing σ gradually, it is not necessary to wait until final convergence of the steepest descent, we only need to be in the vicinity of the global minimum. This choice have a significant effect on the speed of the proposed algorithm.

4. In the steepest descent method the iteration form is $\mathbf{s} \leftarrow \mathbf{s} - \mu\nabla F(\mathbf{s}\,;\sigma)$. In our algorithm μ should be decreasing in a harmony with σ, i.e. for smaller values of σ, $F(\mathbf{s}\,;\sigma)$ would change more and hence smaller step sizes should be chosen. It is shown in [3] that $\mu_\sigma = \mu_0\sigma^2$ would be a proper step size as a function of σ. Note that in Fig. 1 we have just written μ_0, and σ^2 disappears: this is due to the definition $\mathbf{z} \triangleq \sigma^2\nabla F(\mathbf{s}\,;\sigma)$ which leads to cancellation of σ^2 in the updating term.

4. SIMULATION RESULTS

This section is dedicated to numerical results obtained from our recovery algorithm in comparison with ℓ^1-norm recovery which was previously proposed in [1]. Suppose $\mathbf{x}$ and $\mathbf{A}$ in (1) are given. The main objective of this part will be finding the breakpoint in the fraction of sparsity of the noise vector (i.e., ρ where $\|\mathbf{e}^*\|_0 = \rho m$) in which each algorithm fails to recover the input signal (i.e., fails to decode properly). We define *exact recovery* as a recovery under which more than 40dB in *reconstruction SNR* (defined as $10\log_{10}\left(\|\mathbf{s}^*\|_2^2/(\|\mathbf{s}^* - \mathbf{s}\|_2^2)\right)$) is achieved. The simulation results are based on finding the average rate of exact recovery which is calculated over 200 trials. Entries of the coding matrix $\mathbf{A}$ and the input signal $\mathbf{s}^*$ are independent random values with a Gaussian distribution of zero mean and unit variance. Each column of $\mathbf{A}$ is divided by $\sqrt{m}$ to be normalized. For generating a sparse error vector $\mathbf{e}^*$, at first a random support with specified fraction of m (i.e. the parameter ρ defines the sparsity level as well as the corruption rate) is chosen at each trial. Then elements within

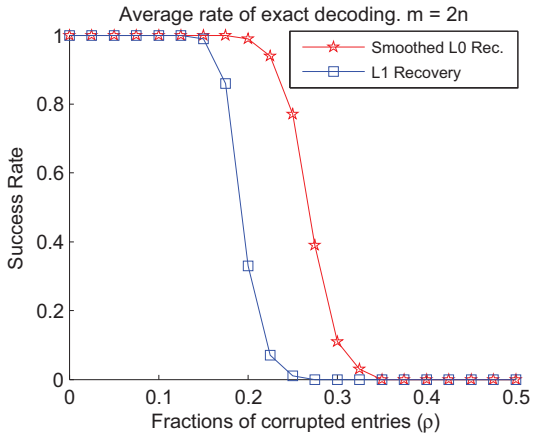

Fig. 2. Average rate of exact decoding for the case of $m = 2n$ and $n = 100$. Smoothed ℓ^0-norm recovery's parameters are set as: $\sigma_{\text{target}} = 0.0001$, $\kappa = 0.82$, $L = 3$ and $\mu_0 = 2$.

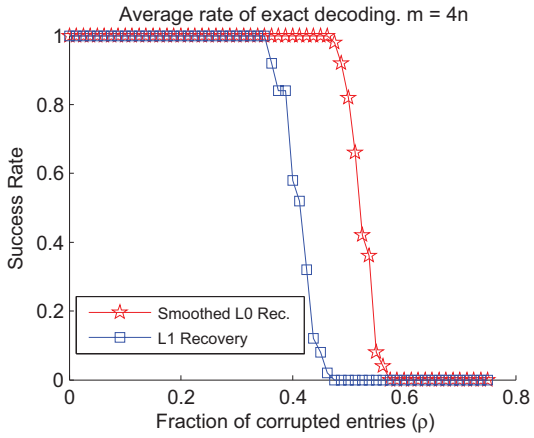

Fig. 3. Average rate of exact decoding for the case of $m = 4n$ and $n = 100$. Smoothed ℓ^0-norm recovery's parameters are set as: $\sigma_{\text{target}} = 0.0001$, $\kappa = 0.87$, $L = 3$ and $\mu_0 = 2$.

the previous support are chosen randomly and independently with a Gaussian distribution with zero mean and unit variance (i.e., we expect elements of $\mathbf{e}^*$ and $\mathbf{As}^*$ to have the same order of energy). Now by having $\mathbf{x} = \mathbf{As}^* + \mathbf{e}^*$, the recovered version $\mathbf{s}$ and $\mathbf{s}^*$ are compared. Simulations are run for two different scenarios of $m = 2n$ and $m = 4n$. Average rate of exact recovery (under previously stated definition) is plotted for both scenarios in Fig. 2 and Fig. 3, respectively. ℓ^1-norm recovery is done by using ℓ_1-MAGIC[1] package. As it is seen in these figures, in both cases, our proposed method provides better recovery rate, even with higher fraction of corruptions. In the case of $m = 2n$, ℓ^1-norm recovery succeeds as long as corruption fraction does not exceed about $\rho = 0.15$ while Smoothed ℓ^0-norm recovery can survive until about $\rho = 0.20$. In the case of $m = 4n$, breakdown for ℓ^1-norm recovery occurs at $\rho = 0.35$ while for Smoothed ℓ^0-norm recovery it happens at about $\rho = 0.5$. It is also important to note that utilizing Smoothed ℓ^0-norm recovery does not impose more complexity in its implementation. In fact, in our algorithm (similar to SL0), we have parameters κ an σ_{target} that determine a trade-off between quality and speed (for example closer κ to 1 results in better quality, but lower speed). Figures 2 and 3 have been plotted for a set of parameters of Smoothed ℓ^0-recovery for which the simulation times of both algorithms are approximately equal (0.019s for $m = 2n$ and 0.054s for $m = 4n$). Alternatively, by choosing other parameters with our algorithm, we could gain a better speed at similar qualities (not plotted here due to limited space).

[1]For ℓ_1-MAGIC, we have used MATLAB code available at http://www.acm.caltech.edu/l1magic/

5. CONCLUSION

In this article, we proposed an algorithm to recover the exact input signal from a corrupted version of the input signal after being encoded by a tall coding matrix. The algorithm aims at solving (3) by using a proper GNC algorithm and by replacing the ℓ^0-norm by a smoothed function. The overall performance of the algorithm is significantly better than ℓ^1-norm recovery and tolerates larger amount of corruptions. As it was mentioned, there is no general statement which guarantees the GNC approach to converge. However, Mohimani et al. [6] has recently proposed a convergence analysis for SL0. So it may be possible to derive a similar convergence analysis for this algorithm, too.

6. REFERENCES

[1] E. Candès, M. Rudelson, T. Tao, and R. Vershynin, "Error correction via linear programming," *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2005.

[2] S.S. Chen, D.L. Donoho, and M.A. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, pp. 129–159, 2001.

[3] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A Fast Approach for Overcomplete Sparse Decomposition Based on Smoothed ℓ^0 Norm," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 289–301, 2009.

[4] I.F. Gorodnitsky and B.D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, 1997.

[5] A. Blake and A. Zisserman, *Visual reconstruction*, MIT Press, 1987.

[6] H. Mohimani, M. Babaie-Zadeh, I. Gorodnitsky, and C. Jutten, "Sparse Recovery using Smoothed ℓ^0 (SL0): Convergence Analysis," *Arxiv preprint arXiv:1001.5073*, 2010.