Signal Processing 92 (2012) 1172–1175

Contents lists available at SciVerse ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Fast communication

Constraint removal for sparse signal recovery

Ahmet Şahin [a,b,*], Serdar Özen [a]

[a] İzmir Institute of Technology, Turkey
[b] İzmir University, Turkey

ARTICLE INFO

Article history:
Received 30 May 2011
Received in revised form
25 September 2011
Accepted 12 November 2011
Available online 19 November 2011

Keywords:
Sparse recovery
Greedy pursuit
Underdetermined system
Compressed sensing

ABSTRACT

This paper presents a new iterative algorithm called constraint removal (CR) for the recovery of a sparse signal x from an incomplete number of linear measurements y such that $y^{m\times 1}=A^{m\times n}x^{n\times 1}$ and $m<n$. It is empirically demonstrated that the CR algorithm has a recovery performance which is between basis pursuit linear programming (BP-LP) and subspace pursuit (SP) for both zero-one and Gaussian type signals.

1. Introduction

A vector is K-sparse if the number of its nonzero entries is less than or equal to the positive integer K. A vector is sparse if the majority of its entries are zero. Sparse recovery refers to the problem of reconstructing a sparse signal x from a number of linear measurements y in which the number of measurements m is smaller than the number n of the entries in x. This can be formulated as an underdetermined system of equations

$$y^{m\times 1}=A^{m\times n}x^{n\times 1} \quad (m<n). \tag{1}$$

Compressive sampling [1,2] aims to sample data in a compressed form by finding the sparsest solution to (1). The ℓ_0 norm of a vector is equal to the number of its nonzero entries and therefore to its sparsity. Due to the absence of scaling property, ℓ_0 norm is not mathematically a norm, although it became a usual practice to call it a norm. The straightforward way to find the sparsest solution to (1) is thus by minimizing the ℓ_0 norm of the solution x

$$\min\|x\|_0 \quad \text{subject to } y=Ax. \tag{2}$$

The solution to the nonconvex problem in (2) requires a combinatorial search which is NP-hard and therefore practically infeasible. It has been shown in [3–5] that if the matrix A in (1) satisfies certain properties, it is equivalent to use the ℓ_1 norm instead of the ℓ_0 norm for minimization. The ℓ_1 norm minimization problem is called *basis pursuit* [6] and is solved by linear programming methods. This is frequently referred to as BP-LP or simply LP, and is written as

$$\text{Basis Pursuit}: \quad \min\|x\|_1 \quad \text{subject to } y=Ax. \tag{3}$$

Both ℓ_0 and ℓ_1 norm minimizations are computationally complex which have been the motivation to search for alternative greedy pursuit algorithms with much lower complexities and with comparable performances. The first is the matching pursuit (MP) [7] algorithm. The MP algorithm has been followed by many derivatives in which the latest two algorithms stand out. They are the subspace pursuit (SP) [8] and the compressive sampling matching pursuit (CoSaMP) [9] algorithms. They both represent a leap in recovery performance in MP based algorithms while still preserving the low complexity profile of the MP. The

* Corresponding author at: İzmir University, Turkey.
Tel.: +90 533 621 4221; fax: +90 232 224 0909.
E-mail address: ahmet.sahin@izmir.edu.tr (A. Şahin).

0165-1684/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2011.11.014

constraint removal (CR) algorithm introduced in this presentation is the first greedy algorithm with nearly the same performance of LP. The CR algorithm has a different motivation and different initial steps than all other MP based greedy pursuits as shown in the following section.

2. Problem formulation and motivation for CR

Since the system in (1) is underdetermined, a direct solution is not possible and infinitely many solutions exist. One remedy is to force the solution to be sparse. The support of a vector x denoted by $supp(x)$ is a set which contains the indices of the nonzero elements, i.e. $supp(x)=\{i : x_i \neq 0\}$. Since both the support and the sparsity levels of the signal x are not known, the only intuitive thing to be done is to press all entries in x to zero. Doing so is more correct than false if x is indeed sparse. Pressing all entries to zero is done by vertically concatenating the underdetermined matrix A with a nxn identity matrix below and elongating the measurement vector y by padding it with n zeros downwards as shown in (5). The new system of equations become

$$y_{new}^{(m+n)\times 1} = A_{new}^{(m+n)\times n} x^{n\times 1}, \tag{4}$$

where

$$A_{new}^{(m+n)\times n} = \begin{bmatrix} A^{m\times n} \\ I^{n\times n} \end{bmatrix} \quad \text{and} \quad y_{new}^{(m+n)\times 1} = \begin{bmatrix} y^{m\times 1} \\ 0^{n\times 1} \end{bmatrix}. \tag{5}$$

Eq. (4) represents an overdetermined system with an upper and a lower part. The upper part is the initial system for which a sparse solution is sought, the lower part is an identity matrix pressing all entries in x towards zero. Therefore both parts will compromise on the solution. Multiplying both sides of (4) with A_{new}^T, the equation transforms to

$$A_{new}^T y_{new} = A_{new}^T A_{new} x. \tag{6}$$

Using the identities in (5), $A_{new}^T A_{new} = (A^T A + I)$ and $A_{new}^T y_{new} = A^T y$, Eq. (6) is equivalent to

$$A^T y = (A^T A + I)x. \tag{7}$$

Multiplying both sides with $(A^T A+I)^{-1}$ the initial solution $x = x_{init}$ is

$$x_{init} = (A^T A + I)^{-1} A^T y, \tag{8}$$

where x_{init} is the initial solution for the underdetermined equation (1), and where all entries in x are equally pressed towards zero within an overdetermined context. After finding x_{init}, an initial estimate of the support can be obtained. Assuming that the maximum expected sparsity is K, the indexes of the largest K entries in x_{init} is the first estimate of the support. This means the pressure on the entries in the support can be released by removing the corresponding K rows in Eq. (4) to obtain the new overdetermined equation

$$y_{new}^{(m+n-K)\times 1} = A_{new}^{(m+n-K)\times n} x^{n\times 1}. \tag{9}$$

Applying the same procedure described in Eqs. (6)–(8), to Eq. (9), the new estimate of the signal x is given by

$$x = (A^T A + D)^{-1} A^T y. \tag{10}$$

This time, the solution for x involves a diagonal matrix $D^{n\times n}$ in place of the identity matrix $I^{n\times n}$ in (8) due to the removed constraints on the estimated support. The diagonal matrix D is obtained by setting the elements of the identity matrix I to zero corresponding to the current support estimate. This procedure is repeated until the support does not change. The CR algorithm consists of two interleaved parts in each iteration, estimating the solution x for a given support and estimating the new support from the solution estimate x. Eqs. (4)–(9) serve only to describe the motivation behind CR. Eq. (10) is the main equation to be iterated.

3. The CR algorithm

Inverted support of a vector is obtained by logically inverting its support. Algorithm 1 describes the constraint removal algorithm. At the end of the iterations, x_{out} will be our sparse solution, and the diagonal s' of the matrix D will contain the inverted support for x_{out}. At each iteration a group of the largest K entries is selected until the group does not change. A group element may stay in the group or may be eliminated from the group at each iteration prohibiting fixed false elements. The third step involves least squares evaluation which can be solved by Gaussian elimination. Contrary to the SP and CoSaMP algorithms, CR does not require that the sparsity of the signal be known.

Algorithm 1. The constraint removal algorithm.

Input: Measurement matrix $A^{m\times n}$, measurement vector $y^{m\times 1}$ and maximum expected sparsity $K = floor(m/2)$.
Output: Signal $x_{out}^{n\times 1}$.
Initialize: $x_{init} = (A^T A + I)^{-1} A^T y$ and $x_{new} = x_{init}$.
Step-1: Set $x_{old} = x_{new}$. By setting the largest K entries in x_{new} to zeros and all others to ones, produce an inverted support vector s'.
Step-2: Produce a diagonal matrix $D = diag(s')$ from the vector s'.
Step-3: Solve $x_{new} = (A^T A + D)^{-1} A^T y$.
Step-4: If $x_{new} = x_{old}$ end iterations, set $x_{out} = x_{new}$ and terminate; else go to step 1.

4. Convergence

The convergence of the CR algorithm is straightforward. It is assumed that x_a is the actual K-sparse solution so that $y = Ax_a$. Referring to the main iterated equation in step 3 of Algorithm 1 and replacing y with Ax_a, the estimated solution x_{new} becomes

$$x_{new} = (A^T A + D)^{-1} A^T A x_a, \tag{11}$$

$$x_{new} = (A^T A + D)^{-1} (A^T A + D - D) x_a, \tag{12}$$

$$x_{new} = x_a - \underbrace{(A^T A + D)^{-1} D x_a}_{error}. \tag{13}$$

Eq. (13) indicates that if D contains in its diagonal the inverted support of the actual solution x_a so that $Dx_a = 0$, the solution x_{new} converges to the actual solution x_a. Finally, it needs to be shown that the diagonal of matrix D approaches the inverted support of x_a beginning from the first iteration where $D = I$. In the first iteration of the

CR algorithm

$$x_{new}[1] = x_a - (A^TA+I)^{-1}x_a, \tag{14}$$

the number of columns used in $(A^TA+I)^{-1}$ is equal to the sparsity K of x_a. If $x_{new}[1]$ is well aligned (have a similar order of magnitudes) with x_a, in the second iteration, the vector Dx_a will be sparser than x_a. As a result, the next solution $x_{new}[2]$ will be closer to and better aligned with x_a since less columns from $(A^TA+D)^{-1}$ will be involved. This will continue in the subsequent iterations until convergence occurs where $Dx_a=0$ and $x_{new}=x_a$.

From Proposition 3.1 in [9], if A satisfies the restricted isometry condition for sparsity K and parameter δ, the bounds for the error in the first iteration can be written as

$$\frac{\|x_a\|_2}{(2+\delta)} \le \|(A^TA+I)^{-1}x_a\|_2 \le \frac{\|x_a\|_2}{(2-\delta)}. \tag{15}$$

It is therefore crucial to have a good alignment in the first and subsequent iterations which depends on how close A^TA is to identity and therefore on the restricted isometry constant. Even when $\delta=0$, the norm of the error in the first iteration is not zero but is perfectly aligned with x_a, therefore in the second iteration actual solution x_a will be recovered exactly. If A satisfies the restricted isometry condition, the reconstruction will be exact yet the value of the RIC parameter δ below which reconstruction is guaranteed needs to be determined.

5. Noise and stable recovery

The measurements may be corrupted by noise such that $y=Ax+e$, where e represents the noise vector. For a stable recovery, measurement noise e and corresponding signal deviation Δx must be comparable. This can be shown using the main iterated equation in the CR algorithm $x[i]=(A^TA+D)^{-1}A^T(y+e)$. The deviation in x is given by $\Delta x=(A^TA+D)^{-1}A^Te$. Since $(A^TA+D)^{-1}$ is always non-singular, the norm of Δx will be comparable to the norm of measurement noise e ensuring stability.

6. Empirical results and comparison

For comparison, the same presentation style used in [8] is adopted for simulations. Simulations are performed for two types of sparse signal x of length 256. The first type is zero-one sparse signal in which all nonzero entries are set to one. The second type is Gaussian signal in which all entries are selected from the standard normal distribution, with zero mean and variance equal to one so that $x \sim \mathrm{N}(0_{n\times 1}, I_{n\times n})$. For both signal types, sparsity K is varied from 1 to 55. For each sparsity level 500 realizations are conducted. Sparsity K refers to the number of nonzero entries in x and therefore $K=\|x\|_0$. The number of exact recoveries is plotted against K. The Gaussian sampling matrix A is a 128×256 matrix whose elements are drawn from the standard normal distribution. Referring to Eq. (1), $m=128$ and $n=256$. For linear programming an interior point method is used.

Fig. 1 depicts the simulation results for zero-one sparse signals. CR algorithm is compared to BP-LP and SP algorithms. CR algorithm outperforms SP algorithm and have a closer performance to BP-LP method.

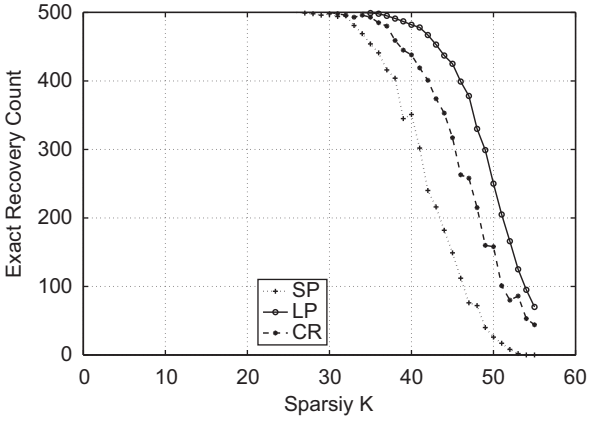

Fig. 1. Simulation results for zero-one signals whose entries can only be 1 and 0.

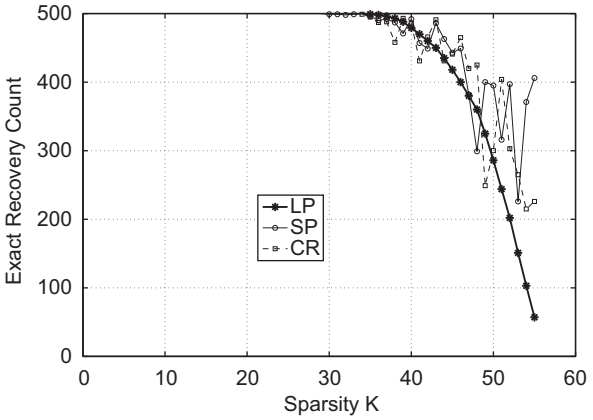

Fig. 2. Simulation results for Gaussian random signals.

Fig. 2 depicts the simulation for Gaussian sparse signals. CR algorithm is compared to BP-LP and SP methods. Both CR and SP having similar performances exhibit somewhat oscillatory behavior as the sparsity level K approaches 55. Plots are taken from actual simulation data without interpolation to demonstrate oscillations.

7. Complexity

The proposed CR algorithm has a complexity of $O(n^3)$, apparently equal to that of the LP due to the matrix inversion in step-3 of Algorithm 1. However in simulations the CR algorithm converges in an average of 10 iterations and using Gaussian elimination instead of matrix inversion in step-3 decreases the runtime further. Besides, the Gaussian elimination procedure terminates earlier than expected for sparse signals which cause additional decrease in runtime. Although CR is not as fast as greedy pursuits SP and CoSaMP, its speed is more closely aligned with them than the ℓ_1 minimization by BP-LP.

8. Conclusion

CR performs better than SP for zero-one type sparse signals with around equal performance for Gaussian type signals. Despite its apparent high complexity $O(n^3)$, its simulation runtimes were far below than expected and below that of BP-LP perhaps due to much fewer number of iterations and early termination of the Gaussian elimination for sparse signals. In addition CR does not require prior sparsity knowledge.

References

[1] D. Donoho, Compressed sensing, IEEE Transactions on Information Theory 52 (4) (2006) 1289–1306.
[2] E. Candès, Compressive sampling, Proceedings of International Congress of Mathematics, vol. 3, Madrid, Spain, 2006, pp. 1433–1452.
[3] D.L. Donoho, For most large underdetermined systems of linear equations, the minimal l_1 solution is also the sparsest solution, Communications on Pure and Applied Mathematics 59 (7) (2006) 907–934.
[4] E. Candès, T. Tao, Decoding by linear programming, IEEE Transactions on Information Theory 51 (2005) 4203–4215.
[5] E.J. Candès, The restricted isometry property and its implications for compressed sensing, Compte Rendus de l' Academie des Sciences, Paris, Serie I 346 (2008) 589–592.
[6] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, SIAM Journal on Scientific Computing 20 (1988) 33–61.
[7] S. Mallat, Z. Zhang, Matching pursuits with time–frequency dictionaries, IEEE Transactions on Signal Processing 41 (12) (1993) 3397–3415.
[8] W. Dai, O. Milenkovic, Subspace pursuit for compressive sensing: closing the gap between performance and complexity. CoRR, abs/0803.0811, 2008.
[9] D. Needell, J.A. Tropp, CoSaMP: iterative signal recovery from incomplete and inaccurate samples, Applied and Computational Harmonic Analysis 26 (3) (2009) 301–321.