

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

**ANSI***net*

Asian Network for Scientific Information  
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

## Iterative Weighted Gradient Projection for Sparse Reconstruction

Jun Deng, Guanghui Ren, Yansheng Jin and Wenjing Ning  
School of Electronics and Information Technology, Harbin Institute of Technology,  
Harbin, Heilongjiang 150001, China

**Abstract:** Finding sparse solution to undetermined linear systems is one of the fundamental challenging issues in compressive sensing problems and other signal processing applications. This study has presented a novel iterative weighted gradient projection algorithm, referred to as the IWGP, to recover sparse signal in large-scale settings. IWGP is based on a widely used weighted filter technique in signal processing which reduces undesirable influence so that gradient projection can be applied to achieve computational efficiency. Numerical experiments are carried out and the results demonstrate the proposed algorithm is significantly faster than the fastest known methods for the  $l_1$  minimization programs and further show that the computational time isn't sensitive to the sparsity level of original signal.

**Key words:** Compressive sensing, sparse reconstruction, gradient projection, convex optimization, residual

### INTRODUCTION

Sparse signal reconstruction from a set of underdetermined or ill-conditioned linear measurements is a fundamental problem in signal processing and statistics. This problem has over the last few years found application areas from compressive sensing (Donoho, 2006; Guochang *et al.*, 2010) to source coding (Goyal *et al.*, 1998), image restoration (Figueiredo and Nowak, 2003; Yu *et al.*, 2010) and source localization (Picard and Weiss, 2010).

Suppose  $A$  is a measurement matrix in  $\mathbb{R}^{M \times N}$  ( $M < N$ ) and  $n$  is a noise vector in  $\mathbb{R}^M$ , then we are given measurements  $y \in \mathbb{R}^M$  of the form:

$$y = Ax + n \quad (1)$$

If there are no restrictions on the measurement matrix  $A$  and the original signal  $x$ , then the problem of finding an estimate  $\hat{x}$  is NP-hard in general. While there are at least five major classes of computation techniques for solving sparse reconstruction problem (Tropp and Wright, 2010), namely greedy pursuit (Mallat and Zhifeng, 1993), Bayesian framework (Wipf and Rao, 2004), nonconvex optimization (Chartrand, 2007), brute force (Miller, 2002) and convex relaxation (Chen *et al.*, 1998).

Greedy pursuit, one of the most popular ways at present (Chen *et al.*, 2008), is used in approaches such as Matching Pursuit (MP) (Mallat and Zhifeng, 1993), Orthogonal Matching Pursuit (OMP) (Pati *et al.*, 1993), Stagewise Orthogonal Matching Pursuit (StOMP)

(Donoho *et al.*, 2006) and Stagewise Weak Gradient Pursuit (SWGPP) (Blumensath and Davies, 2009). Greedy pursuit iteratively refines a sparse signal representation by successively choosing one or more elements on a dictionary that have the highest correlation with current residual. Bayesian framework employs relevance vector machine for estimating the underlying signal while nonconvex optimization procedures develop a related nonconvex problem and attempt to find a stationary point. Brute force strategies search through all possible support sets to minimize the number of possibilities. Another very popular approach is to solve the convex unconstrained optimization problems of the form:

$$\min_x \|y - A\hat{x}\|_2^2 + \tau \|\hat{x}\| \quad (2)$$

Here,  $\tau$  is a regularization positive parameter which balances the sparsity of the data against the data fidelity (Tropp and Wright, 2010). Many optimization algorithms have been proposed to solve the unconstrained (but nonsmooth) formulation (2), such as Interior-Point (IP) methods (Kim *et al.*, 2007), Iterative Shrinkage/Thresholding (IST) (Daubechies *et al.*, 2004) algorithms, GPSR (Figueiredo *et al.*, 2007) and SpaRSA (Wright *et al.*, 2009).

This study has concentrated on GPSR methods. There are two main problems associated with the application of GPSR to large scale data sets. On the one hand, the computational time is easily affected by original signal sparsity and number of measurements. On the other

hand, there is a great deal of interference of other uselessly searching directions in GPSR so that the computational time per reconstruction is long if without continuation schemes. Inspired by greedy pursuit methods, we introduce a variant of GPSR together with iterative weighted vector estimate and we only update the searching directions of  $\tilde{x}^{(n)}$  based on the weighted vector. We refer to the resulting approach as IWGP (Iterative Weighted Gradient Projection). We show that IWGP retains all the merits of the GPSR methods and greatly decreases the computational time. We use numerical experiments to show IWGP can handle larger scale compressive sensing problems more efficiently than GPSR and IST. We also compare the sensitivity of IWGP with those of GPSR and IST in terms of change of number of nonzero components and number of measurements.

**GRADIENT PROJECTION FOR SPARSE RECONSTRUCTION**

The algorithms discussed and developed in this study are all part of GPSR algorithms. Given a vector  $x$  and a measurement matrix  $A$ , the approach of GPSR is to express Eq. 2 as a Bound-Constrained Quadratic Program (BCQP) (Wu *et al.*, 2010) via splitting the variable  $x$  into positive and negative parts (Figueiredo *et al.*, 2007) and can be rewritten Eq. 2 as:

$$\min_z \|y - A(\tilde{x}_p - \tilde{x}_n)\|_2^2 + \tau \mathbf{1}^T (\tilde{x}_p - \tilde{x}_n) \tag{3}$$

s.t.  $\tilde{x}_p \geq 0, \tilde{x}_n \geq 0$

where,  $\tilde{x} = \tilde{x}_p - \tilde{x}_n$ ,  $\tilde{x}_p = (\tilde{x})_+$  and  $\tilde{x}_n = (-\tilde{x})_+$ .

Equation 3 can be more properly rewritten in a standard BCQP form as:

$$\min_z G(\tilde{z}) = \mathbf{b}^T \tilde{z} + \frac{1}{2} \tilde{z}^T B \tilde{z} \tag{4}$$

s.t.  $\tilde{z} \geq 0$

Where:

$$\tilde{z} = \begin{bmatrix} \tilde{x}_p \\ \tilde{x}_n \end{bmatrix}, \mathbf{b} = \tau \mathbf{1} + \begin{bmatrix} -A^T y \\ A^T y \end{bmatrix}$$

And:

$$B = \begin{bmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{bmatrix}$$

The gradient of the objective function  $G(\tilde{z})$  is defined as:

$$\nabla G(\tilde{z}) = \mathbf{b} + B\tilde{z} \tag{5}$$

GPSR chooses the negative gradient  $-\Delta G(\tilde{z})$  to search for each iterate  $\tilde{z}^{(k)}$ :

$$\tilde{z}^{(k+1)} = \tilde{z}^{(k)} - \alpha^{(k)} \nabla G(\tilde{z}^{(k)}) \tag{6}$$

where,  $\alpha^{(k)}$  is the step size. This can be solved by two strategies in GPSR, called GRSR-Basic algorithm and GPSR-BB algorithm respectively. In GPSR-Basic, the step size  $\alpha^{(k)}$  is defined initially as:

$$\alpha_0 = \frac{(\mathbf{g}^{(k)})^T \mathbf{g}^{(k)}}{(\mathbf{g}^{(k)})^T B \mathbf{g}^{(k)}} \tag{7}$$

Where:

$$\mathbf{g}_i^{(k)} = \begin{cases} (\nabla G(\tilde{z}^{(k)}))_i, & \text{if } \tilde{z}^{(k)} > 0 \text{ or } ((\nabla G(\tilde{z}^{(k)}))_i) < 0 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Then it is achieved the proper step size until a sufficient decrease is attained in  $G$  by running a backtracking line search. Alternatively, in order to avoid the fussy inner searching loop, the Barzilai and Borwein equation (Figueiredo *et al.*, 2007) is applied to approximate to the Hessian of  $G$  at  $\tilde{z}^{(k)}$  in GPSR-BB. So the step size is calculated simply as:

$$\alpha^{(k)} = \frac{(\theta^{(k)})^T \theta^{(k)}}{(\theta^{(k)})^T B \theta^{(k)}} \tag{9}$$

Where:

$$\theta^{(k)} = (\tilde{z}^{(k)} - \alpha^{(k)} \nabla G(\tilde{z}^{(k)}))_+ - \tilde{z}^{(k)}$$

Note that GPSR-Basic ensures objective function decrease per iteration and GPSR-BB makes the iteration easier. They are easy to implement and do not appear to require application-specific tuning. In reality, GPSR algorithms tend not to work very well for large-scale settings because they spend much computational time. Instead, it is necessary to apply specialized techniques, such as warm-start and continuation. A more detailed discussion of GPSR algorithms can be found by Figueiredo *et al.* (2007).

**ITERATIVE WEIGHTED GRADIENT PROJECTION**

In this study, we propose a novel approach, called IWGP which combines a weighted vector with GPSR to

achieve a sparse approximation solution to Eq. 1. More precisely, we mean that it is more efficient to compute the gradient  $\Delta G(\tilde{z})$  and to solve Eq. 2 by GPSR. In this section, we discuss how the algorithm works efficiently as well as the different ways to choose threshold, describing the IWGP framework formally.

In fact the gradient  $\Delta G(\tilde{z})$  contain more information than its form suggests, by further analyzing its particular Eq. 5. For a given  $\tilde{z} = [\tilde{x}_p^T \tilde{x}_n^T]^T$ , we have:

$$\begin{aligned} \nabla G(\tilde{z}) &= \mathbf{b} + \mathbf{B}\tilde{z} \\ &= \mathbf{b} + \mathbf{B} \begin{bmatrix} \tilde{x}_p \\ \tilde{x}_n \end{bmatrix} \\ &= \tau \mathbf{1} + \begin{bmatrix} -\mathbf{A}^T(\mathbf{y} - \mathbf{A}(\tilde{x}_p - \tilde{x}_n)) \\ \mathbf{A}^T(\mathbf{y} - \mathbf{A}(\tilde{x}_p - \tilde{x}_n)) \end{bmatrix} \end{aligned} \quad (9)$$

We define a residual vector (Liejun, 2011):

$$\mathbf{r} = \mathbf{y} - \mathbf{A}\tilde{\mathbf{x}} \quad (10)$$

So Eq. 9 can be rewritten as:

$$\nabla G(\tilde{z}) = \begin{bmatrix} \nabla G(\tilde{x}_p) \\ \nabla G(\tilde{x}_n) \end{bmatrix} \quad (11)$$

Where:

$$\nabla G(\tilde{x}_p) = \tau \mathbf{1} - \mathbf{A}^T \mathbf{r}, \nabla G(\tilde{x}_n) = \tau \mathbf{1} + \mathbf{A}^T \mathbf{r} \quad (12)$$

It may be worth noting that the gradient of the objective function in Eq. 11 can be constitution of a constant vector  $\tau \mathbf{1}$  and a vector  $\mathbf{A}^T \mathbf{r}$  which means that the gradient  $\Delta G(\tilde{z})$  includes the information of residual correlation between the current residual  $\mathbf{r}$  and measurement matrix  $\mathbf{A}$ .

Now we may concentrate on showing that the proposed algorithm makes full use of the residual correlation to build up a weighted vector to filter out unwanted gradient value. We define a vector  $\mathbf{J}$  in  $\mathbb{R}^N$  by:

$$\mathbf{J} = \text{hard}(|\mathbf{c}|, \text{Th}) \quad (13)$$

where,  $\mathbf{c} = \mathbf{A}^T \mathbf{r}$ ,  $\text{Th}$  is a threshold parameter,  $\text{hard}(x, \text{th}) = 2 \times \text{sign}(|x| - \text{th}) + 1$  is the well-known hard-threshold function. We merge the newly identified vector with the previous weighted vector estimate, thereby updating the estimate:

$$W_i = \begin{cases} 1, & \text{if } J_i = 1 \text{ or } W_i = 1 \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

Since the original sparse signal  $\mathbf{x}$  is a signal populated primarily with zeros, we can be only interested in the gradient directions of the locations of the nonzeros in  $\tilde{z}$ .

Specifically, we define a new gradient vector:

$$\begin{aligned} \nabla G(\tilde{z})_w &= \begin{bmatrix} \nabla G(\tilde{x}_p) \cdot \mathbf{W}^T \\ \nabla G(\tilde{x}_n) \cdot \mathbf{W}^T \end{bmatrix}, \\ &= \begin{bmatrix} (\nabla G(\tilde{x}_p))_w \\ (\nabla G(\tilde{x}_n))_w \end{bmatrix} \end{aligned} \quad (15)$$

Where:

$$(\nabla G(\tilde{x}_p))_w = \nabla G(\tilde{x}_p) \cdot \mathbf{W}^T, (\nabla G(\tilde{x}_n))_w = \nabla G(\tilde{x}_n) \cdot \mathbf{W}^T \quad (16)$$

With the direction to move given by Eq. 15, we can simply compute the step size as Eq. 9 and update the next estimate  $\tilde{\mathbf{x}}$  by Eq. 6.

**Threshold selection:** To recover the original signal  $\mathbf{x}$ , we hope to know which columns of  $\mathbf{A}$  participate in the measurement vector  $\mathbf{y}$ . The idea behind the algorithm is to identify columns in a thresholding fashion. At each iteration, we firstly wish to determine a weighted vector which represents the column of  $\mathbf{A}$  that is most strongly correlated with the remaining part of  $\mathbf{y}$ . Then we start to search from a new approximation along the negative gradient filtered by the weighted vector. After proper iterations, we wish that the algorithm will have identified the correct vector of columns.

These considerations motivate a few number of possible threshold selection. One simple selection is:

$$\text{Th}_{\text{stomp}} = \frac{t \|\mathbf{r}^{(k-1)}\|_2}{\sqrt{M}} \quad (17)$$

where,  $t$  typically takes values in the range  $2 \leq t \leq 3$ . The selection strategy is inspired by Gaussian behavior of residual Multiple-Access Interference (MAI) (Donoho *et al.*, 2006).

A similar strategy is called stagewise weak element selection (Blumensath and Davies, 2009). This selection takes advantage of a threshold based on the maximum of  $|\mathbf{r}_i|$ . Based on this approach, the threshold is calculated as:

$$\text{Th}_{\text{sw}} = \beta \|\mathbf{A}^T \mathbf{r}^{(k-1)}\|_{\infty} \quad (18)$$

where,  $\beta \in (0, 1)$  is a weakness parameter.

Both these strategies are well suited to the class of problems addressed in this study. In the experiments, unless otherwise noted, we choose (18), with  $\beta = 0.75$  as the threshold.

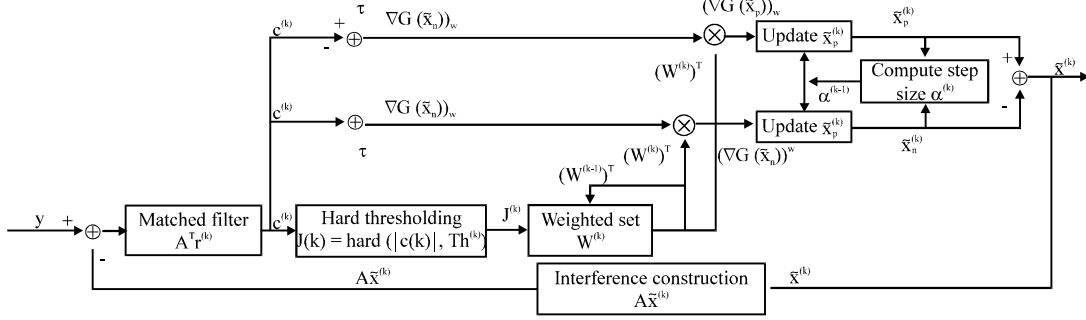


Fig. 1: Schematic Representation of the IWGP algorithm

**The IWGP framework:** IWGP operates in  $k$  iterations, building up a sequence of estimations  $\tilde{x}^{(0)}, \tilde{x}^{(1)}, \tilde{x}^{(2)}, \dots$  by searching along a sequence of negative weighted gradient vectors  $-\nabla G_w^{(1)}, -\nabla G_w^{(2)}, \dots$ . Moreover, the algorithm also maintains a sequence of estimates  $W^{(0)}, W^{(1)}, \dots$ , of weighted coefficients. Figure 1 gives a diagrammatic representation.

Meantime a precise framework of the algorithm is defined as follows.

#### Algorithm IWGP:

**Step 1:** Start with initial ‘solution’  $\tilde{x}^{(0)} = 0$ , initial residual  $r^{(0)} = y$  and the weighted vector  $W^{(0)} = 0$ . The iteration counter  $k$  starts at  $k = 1$

**Step 2:** Apply matched filtering to the current residual, achieving the vector of residual correlations:

$$c^{(k)} = A^T r^{(k-1)} \quad (19)$$

**Step 3:** Compute the threshold  $Th^{(k)}$  from Eq. 18, the identified vector  $J^{(k)}$  as in Eq. 13 and then yield the weighted vector  $W^{(k)}$  as in Eq. 14.

**Step 4:** Compute the weighted gradient  $(\nabla G(\tilde{x}_p^{(k)}))_w, (\nabla G(\tilde{x}_n^{(k)}))_w$  as in Eq. 16

**Step 5:** Calculate the new estimate and the new residual:

$$\begin{aligned} \tilde{x}_p^{(k)} &= \tilde{x}_p^{(k-1)} - \alpha^{(k-1)} (\nabla G(\tilde{x}_p^{(k-1)}))_w, \\ \tilde{x}_n^{(k)} &= \tilde{x}_n^{(k-1)} - \alpha^{(k-1)} (\nabla G(\tilde{x}_n^{(k-1)}))_w \end{aligned}$$

$$\tilde{x}^{(k)} = \tilde{x}_p^{(k)} - \tilde{x}_n^{(k)}$$

$$r^{(k)} = y - A^T \tilde{x}^{(k)}$$

**Step 6:** Compute the step size  $\alpha^{(k-1)}$  from Eq. 9

**Step 7:** Check the stopping criterion which is chosen as suggested by Figueiredo *et al.* (2007)  $\| \min(z, \Delta G(z)) \| \leq \text{tolP}$  and if it is satisfied, set

$\tilde{x} = \tilde{x}^{(k)}$  as the final output of the iteration; otherwise, set  $k = k + 1$  and return to step 2

## EXPERIMENTS

This section illustrates experimentally that IWGP is a powerful algorithm for several types of sparse reconstruction problems of the form Eq. 2. All the experiments were carried out on a personal computer with an Intel Pentium (R) Dual E2200 processor and 1.99GB of memory.

Let us describe the experimental setup. In each trial, we start with a  $d$ -sparse signal  $x$  and its measurements  $y$  according to the model in Eq. 1. In all of the experiments below, we have used an  $M \times N$  Gaussian matrix as our measurement matrix  $A$ , with all entries independently distributed Normal  $(0, 1)$  and then orthonormalizing the rows. We used parameter of stopping criterion  $\text{tolP} = 10^{-5}$ .

In our first experiment, we consider a length  $N = 4096$  signal  $x$  that contains  $d = 100$  randomly placed  $\pm 1$  spikes. To simulate measurement noise, zero-mean Gaussian white noise with variance  $10^{-4}$  is added to each of the  $M$  measurements that define the observation  $y$ . In the example  $M = 1024$ ,  $\tau = 0.005 \|A^T y\|_\infty$  and the reconstructions are implemented by GPSR and IWGP.

Figure 2 shows that the original signal and the reconstruction results with GPSR-BB and IWGP, respectively. Because of noisy reconstruction, GPSR-BB cannot recover the underlying sparse signal exactly, nor can IWGP. However, the IWGP reconstruction exhibits a bit lower Mean Squared Error (MSE) with respect to the original signal.

In the second experiment, we compare the speed of IWGP with that of other recently proposed algorithms, namely, variants of GPSR and IST (Daubechies *et al.*, 2004). All the parameters are similar in the first experiment except that we perform 100 independent trials. Table 1 reports the average CPU times needed by the

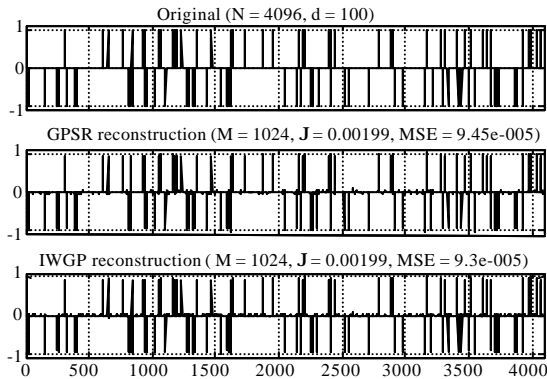


Fig. 2: From top to bottom: original signal, reconstruction via GPSR-BB and IWGP

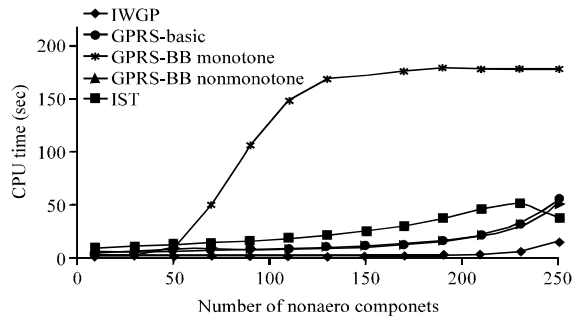


Fig. 3: CPU time and number of nonzero components

Table 1: Average CPU times

Algorithm	CPU time (sec)	Iterations	MSE
GPSR-Basic	8.0944	239	9.95e-5
GPSR-BB monotone	6.8333	401	9.99e-5
GPSR-BB nonmonotone	6.0778	401	9.02e-3
IST	17.0340	681	1.16e-3
IWGP	1.4305	84	9.83e-5

several algorithms. The finding in this table presents IWGP is significantly faster than the others. More in detail, IWGP is approximate 12 times faster than IST and about 5 times faster than GPSR.

The plot in Fig. 3 presents how the CPU time of IWGP changes when the number of nonzero components increases. For comparison, also shown are the results obtained with GPSR and IST. And the noise's variances in this example are fixed the same and the results are averaged over 100 runs. In general the CPU time of the algorithms is in line with the increase of the number of nonzero components. However, IWGP keeps this growth very mild. In other words, IWGP is insensitive to the change in the sparsity level.

Then let us see how much CPU time of the algorithms is required to recover a fixed sparsity level of signal with

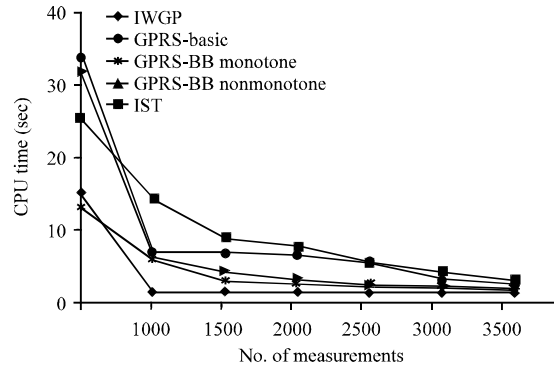


Fig. 4: CPU time and number of measurements

change in the number of measurements. Results in Fig. 4 display that CPU time of IWGP (as well as GPSR and IST) is affected by increase in the number of measurements. Not surprisingly, IWGP costs the least CPU time solving sparse reconstruction problem although the others also decrease gradually as the number of measurements grows.

### CONCLUSION

We have presented a novel gradient projection algorithm to more effectively solve the solution to sparse signal reconstruction problems arising in compressive sensing and other inverse problems. We have employed a weighted technique to filter out unwanted gradient components and the resulting algorithm is well suited to handling large-scale settings. In the numerical experiments, we have tested IWGP by utilizing a large-scale example. The novel algorithm is significantly faster compared to state-of-the-art algorithms. Moreover, the decrease of sparsity level can hardly interfere the computational time of the algorithm derived in this paper. The numerical results further show that IWGP is superior to GPSR algorithms and IST.

### REFERENCES

Blumensath, T. and M.E. Davies., 2009. Stagewise weak gradient pursuit. *IEEE Trans. Signal Process.*, 57: 4333-4346.

Chartrand, R., 2007. Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Process. Lett.*, 14: 707-710.

Chen, S., D. Donohoh and M. Saunders, 1998. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20: 33-61.

Chen, J., J. Fan, X. Cao and Y. Sun, 2008. GRFR: Greedy rumor forwarding routing for wireless sensor/actor networks. *Inform. Technol. J.*, 7: 661-666.

- Daubechies, I., M. Frieze and C. Mol, 2004. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure. Applied Math.*, 57: 1413-1457.1002/CPA.20042.
- Donoho, D.L., 2006. Compressed sensing. *IEEE Trans. Inform. Theory*, 52: 1289-1306.
- Donoho, D.L., Y. Tsaig, I. Drori and J.L. Starck, 2006. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. Stanford Technical Report, 2006, pp: 1-39. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.115.5221&rep=rep1&type=pdf>.
- Figueiredo, M. and R. Nowak, 2003. An EM algorithm for wavelet-based image restoration. *IEEE Trans. Image Process.*, 12: 906-916.
- Figueiredo, M.A.T., R.D. Nowak and S.J. Wright, 2007. Gradient projection for sparse reconstruction application to compressed sensing and other inverse problem. *IEEE J. Sel. Top. Sign. Proces.*, 1: 586-597.
- Goyal, V.K., M. Vetterli and N. Thao, 1998. Quantized overcomplete expansions in RN: Analysis, synthesis and algorithms. *IEEE Trans. Inform. Theory*, 44: 16-31.
- Guochang, W., Z. Yadong and Y. Xiaohui, 2010. Sampling theory: From shannon sampling theorem to compressing sampling. *Inform. Technol. J.*, 9: 1231-1235.
- Kim, S., K. Koh, M. Lustig, S. Boyd and D. Gorinevsky, 2007. An interior-point method for large-scale  $l_1$ -regularized least squares. *IEEE J. Sel. Top. Sign. Proces.*, 1: 606-617.
- Liejun, W., 2011. An improved water-filling power allocation method in MIMO OFDM systems. *Inform. Technol. J.*, 10: 639-647.
- Mallat, S.G. and Z. Zhifeng, 1993. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.*, 41: 3397-3415.
- Miller, A., 2002. *Subset Selection in Regression*. 2nd Edn., Chapman and Hall, London, UK., 256.
- Pati, Y.C., R. Rezaifar and P.S. Krishnaprasad, 1993. Orthogonal matching pursuit: Ication to wavelet decomposition. *Conf. Rec. Asilomar Conf. Signals Syst. Comput.*, 1: 40-44.
- Picard, J.S. and A.J. Weiss, 2010. Bounds on the Number of Identifiable Outliers in Source Localization by Linear Programming. *IEEE Trans. Image Process.*, 58: 2884-2895.
- Tropp, J. and S. Wright, 2010. Computational Methods for Sparse Solutions of Linear Inverse Problems. *Proc. IEEE*, 98: 948-958.
- Wipf, D. and B. Rao, 2004. Sparse Bayesian learning for basis selection. *IEEE Trans. Signal Process.*, 52: 2153-2164.
- Wright, S., R. Nowak and M. Figueiredo, 2009. Sparse reconstruction by separable approximation. *IEEE Trans. Signal Process.*, 57: 2479-2493.
- Wu, H., L. He, L. Qin, T. Feng and R. Shi, 2010. Solving interval quadratic program with box set constraints in engineering by a projection neural network. *Inform. Technol. J.*, 9: 1615-1621.
- Yu, X., Y. Gao, X. Yang, C. Shi and X. Yang, 2010. Image restoration method based on least-squares and regularization and fourth-order partial differential equations. *Inform. Technol. J.*, 9: 962-967.