Compressed Sensing Recovery Algorithms

He Liu
aresmiki@163.com
Southwest Jiaotong University

Sparse optimization algorithm is the key of compressed sensing technology. At present, there are many compressed sensing (CS) recovery algorithms have been proposed. The problem of compressed sensing recovery is actually a minimization ℓ_0 problem. Solving ℓ_0 norm minimize using combinatorial search is NP-hard, two frequently used sparse recovery algorithms in compressive sensing (CS) are matching pursuit (MP) and ℓ_1 norm minimization based algorithm known as basis pursuit (BP). In order to obtain the results of approximate ℓ_0 minimization, and to avoid the NP-Hard problem, many sparse recovery algorithms based approximate ℓ_0 and ℓ_p norm based minimization problem with $p<1$ have been proposed. Here I list some of them, and provide the corresponding experimental results. The corresponding Matlab Codes of the following algorithms can be downloaded from my GitHub page, https://github.com/aresmiki.

The signal x could be estimated from noisy measurement y by solving the convex minimization problem, called second-order cone program (SOCP), as follows.

$$\begin{aligned} &\min mize && \|x\|_1 \\ &subject\,to: && \|Ax-y\|_2 \le \varepsilon \end{aligned}$$

where ε is a bound of the amount of noise in the data. For notation convenience, we rephrase the ℓ_1 minimization problem for noise free case as

$$\begin{aligned} &\min mize && \|x\|_1 \\ &subject\,to: && Ax=y \end{aligned}$$

The following recovery algorithms are considered for two optimization models, Please refer to the relevant literature for details.

1. Orthogonal Matching Pursuit (OMP) [1]

Algorithm 1: CS recovery using OMP

Input: The CS observation y, and a measurement matrix $\Omega=\Phi\Psi=\{\omega_i,\mathrm{i}=1,2,\cdots,\mathrm{m}\}$ where $\Phi \in R^{n\times m}$ and $\Psi \in R^{m\times m}$.

Initialization: Index $I=\varnothing$, residual $r=y$, sparse representaton $\theta=0\in R^m$.

Iteration:

while(stopping criterion false)

$i = \arg\max_j |< r, \omega_j >|$;

$I = I \cup \{i\}$;

$r = y - \Omega(:, I)[\Omega(:, I)]^{\dagger} y$

end while

$\theta(I) = [\Omega(:, I)]^{\dagger} y$

Output: Sparse representation θ, and the original signal $x = \Psi\theta$.

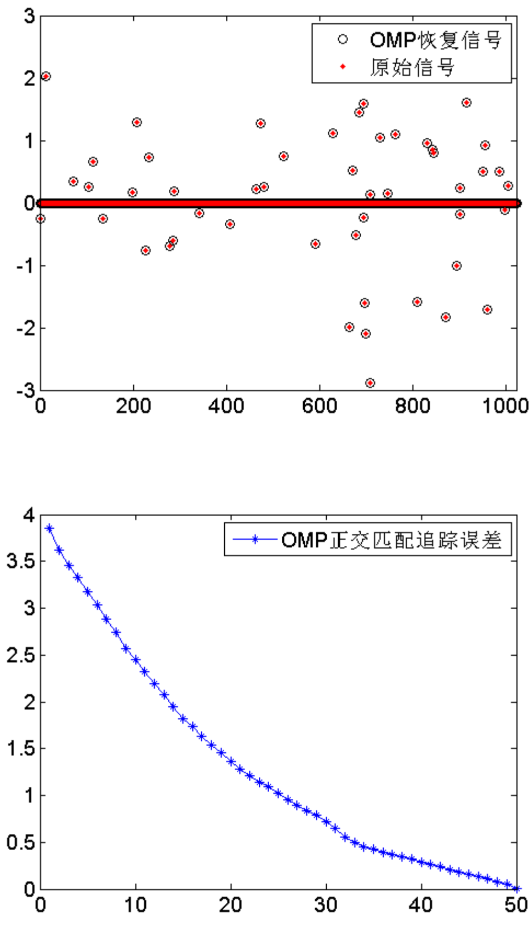


Figure.1. Recovery result by using OMP

2. Compressive Sampling Matching Pursuit (CoSaMP) [2-3]

Algorithm 2: CS recovery using CoSaMP

Input: The CS observation y, a measurement matrix $\Omega=\Phi\Psi=\{\omega_i, i=1,2,\cdots,m\}$ where $\Phi \in R^{n\times m}$ and $\Psi \in R^{m\times m}$, and the sparsity of signal x based on Ψ is s.

Initialization: Residual $r = y$, sparse representation $\theta = 0 \in R^m$.

Iteration:

while(stopping criterion false)

$$\hat{\theta} = \Omega^T r;$$

$$p = \sup(\hat{\theta} l_{2s});$$

$$\overline{p} = p \cup \sup(\theta);$$

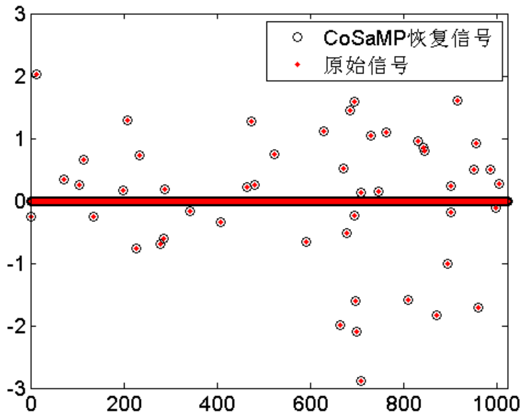
$$\overline{\theta}(\overline{p}) = \Omega^{\dagger}(:, \overline{p}) y;$$

$$\theta = \overline{\theta} l_s;$$

$$r = y - \Omega\theta;$$

end while

Output: Sparse representation θ, and the original signal $x = \Psi\theta$.



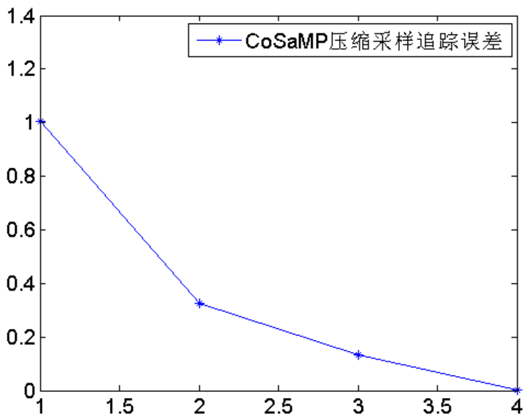


Figure.2. Recovery result by using CoSaMP

3. Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [4]

Algorithm 3: CS recovery using FISTA

Input: $L = L(f)$---A Lipschitz constant of $\nabla f(x)$.

Step 0: Take $y_1 = x_0 \in R^n, t_1 = 1$.

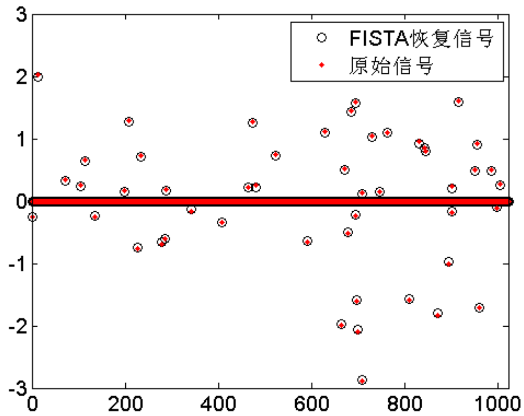
Step k **:** ($k \geq 1$) Compute

(i) $x_k = P_L(y_k)$ by solving the problem:

$$P_L(x_{k-1}) = \arg\min_x \left\{ \left\{ \frac{L}{2} \left\| x - \left(x_{k-1} - \frac{1}{L} \nabla f(x_{k-1}) \right) \right\|_2^2 + g(x) \right\} \right.$$

(ii) $t_k = \dfrac{1 + \sqrt{1 + 4t_k^2}}{2}$

(iii) $y_{k+1} = x_k + \left(\dfrac{t_k - 1}{t_{k+1}} \right) (x_k - x_{k-1})$



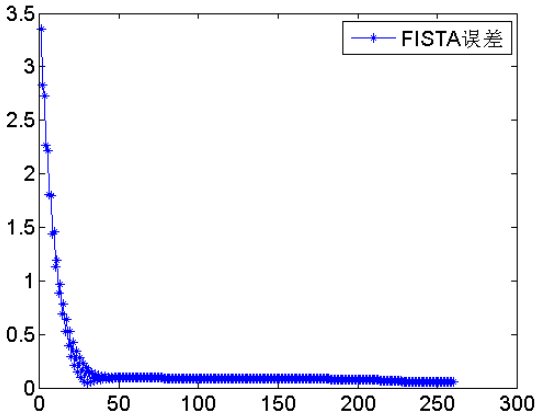


Figure.4. Recovery result by using FISTA

4. Iterative Hard Thresholding algorithms for compressive sensing

(IHT) [5-6]

Algorithm 4: CS recovery using IHT

Input: observation y , and a measurement matrix Ω .

Initialization: $x^{(0)} = 0$

Step k **:** ($k \geq 1$) Compute

(i) $x^{(1)} = \Omega^T \left[y - \Omega x^{(0)} \right]$

(ii) $x^{(2)} = x^{(0)} + ux^{(1)}$

(iii) set all but the largest K elements to zero, where K is the sparsity of the estimate

(iv) $x^{(0)} = x^{(2)}$

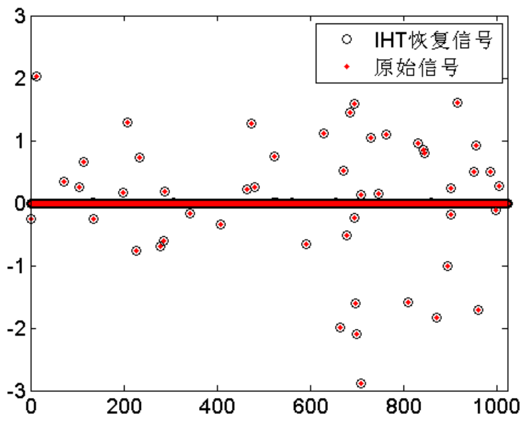


Figure.4. Recovery result by using IHT

5. Iteratively Reweighted Least Square (IRLS) [7]

Algorithm 5: CS recovery using IRLS

Input: The CS observation y , and a measurement matrix $\Omega = \Phi\Psi = \{\omega_i, \mathrm{i}=1,2,\cdots,\mathrm{m}\}$ where $\Phi \in R^{n \times m}$ and $\Psi \in R^{m \times m}$.

Initialization: sparse representation $\theta = \Omega^{\mathrm{T}} y$, $p = 1, \varepsilon = 1$.

Iteration:

while (stopping criterion false)

for i from 1 to end

$$w_i = (\theta_i^2 + \varepsilon)^{p/2-1};$$

end

$Q = diag(1./w)$;

$\theta = Q\Omega^T inv(\Omega Q \Omega^T) y$;

end whlie

Output: Sparse representation θ, and original signal $x = \Psi\theta$.

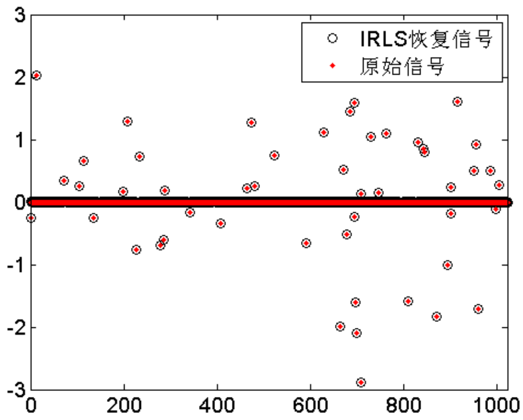


Figure.5. Recovery result by using IRLS

6. Iterative Shrinkage-Thresholding Algorithm (ISTA) [8]

Algorithm 6: CS recovery using ISTA

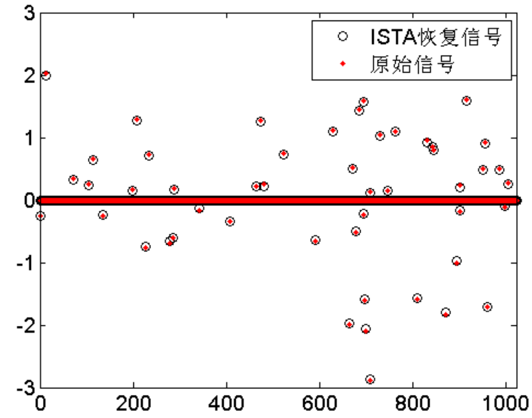
Input: $L = L(f)$---A Lipschitz constant of $\nabla f(x)$.

Step 0: Take $x_0 \in R^n$.

Step k **:** ($k \geq 1$) Compute

(i) $x_k = P_L(y_k)$ by solving the problem:

$$P_L(x_{k-1}) = \arg\min_x \left\{ \left\{ \frac{L}{2} \left\| x - \left(x_{k-1} - \frac{1}{L} \nabla f(x_{k-1}) \right) \right\|_2^2 + g(x) \right\} \right.$$



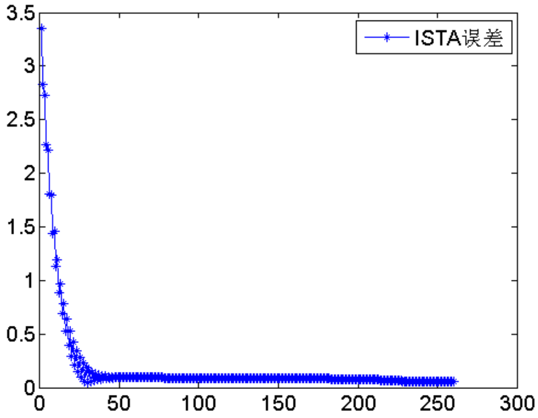


Figure.6. Recovery result by using ISTA

7. Null-Space Reweigthted Approximate l0-Pseudonorm Algorithm (NSRAL0) [9]

Algorithm 7: CS recovery using NSRAL0

Step 1

Input: Φ, x_s, σ_J, r, τ, and ε.

Step 2

Set $\xi^{(0)} = 0$, $\omega^{(0)} = e_N$, $\sigma = \max|x_s| + \tau$, and $k = 0$.

Step 3

Perform the QR decomposition $\Phi^T = QR$ and construct V_r using the last $N - M$ columns of Q.

Step 4

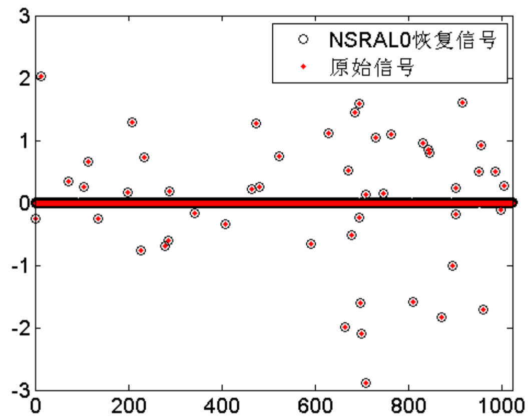
With $\omega{=}\omega^{(k)}$ and using $\xi^{(0)}$ as an initial point, apply the BFGS algorithm to solve the problem in (10), where re-weighting with parameter ε is applied using $\omega_i^{(k+1)} = \dfrac{1}{\left|x_i^{(k)}\right| + \varepsilon}$ in each iteration. Denote the solution as $\xi^{(k)}$.

Step 5

Compute $x^{(k)} = x_s + V_r \xi^{(k)}$ and update weight vector to $\omega^{(k+1)}$ using $\omega_i^{(k+1)} = \dfrac{1}{\left|x_i^{(k)}\right| + \varepsilon}$.

Step 6

If $\sigma \le \sigma_J$. stop and output $x^{(k)}$ as solution; otherwise, set $\xi^{(0)} = \xi^{(k)}, \sigma = r \cdot \sigma, k = k+1$, and repeat from Step 4.



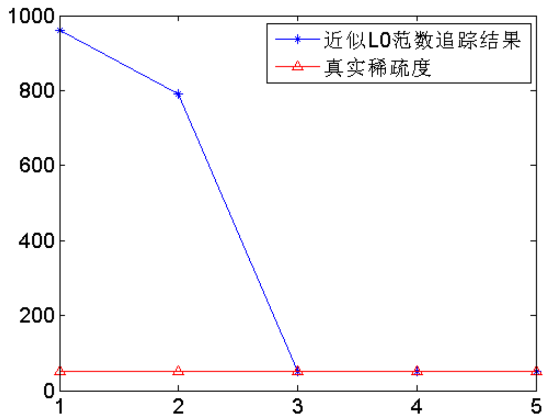


Figure.7. Recovery result by using NSRAL0

8. Reweighted L1 Minimization Algorithm (RL1) [10]

Algorithm 8: CS recovery using RL1

Step 1

Set $l = 0$ and $\omega^{(0)} = 1$ for $i = 1, 2, \cdots, m$.

Step 2

Solve the weighted l_1 minimization problem

$$x^{(l)} = \arg\min \left\| W^{(l)}(y - \Phi x) \right\|_{l_1}.$$

Step 3

Updata the weights; let $r^{(l)} = y - \Phi x^{(l)}$ and for each $i = 1, \cdots, m$, define

$$\omega^{(l+1)} = \frac{1}{\left| r_i^{(l)} \right| + \varepsilon}.$$

Step 4

Terminate on convergence or when l attains a specified maximum number of iterations $l_{\max}$.Otherwise, increment l and go to Step 2.

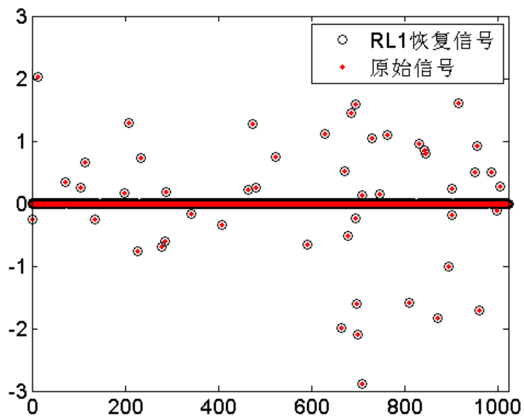


Figure.8. Recovery result by using RL1

9. Robust Smoothed l0-Pseudonorm Algorithm (RSL0) [11]

Algorithm 9: CS recovery using RSL0

Step 1

Let $\hat{s}_0$ equal to the minimum l^2 norm solution of $s = x$.

Step 2

Choose a suitable decreasing sequence for σ, $[\sigma_1, \cdots, \sigma_K]$.

Step 3

Let $\sigma = \sigma_K$.

Maximize (approximately) the function F_σ on the feasible set $\delta = \{s | \|As - x\|_2 \le \varepsilon\}$;

Initialization: $s = \hat{s}_{k-1}$

For $j = 1, \cdots, L$

Let $\Delta s = [s_1 e^{-s_1^2/2\sigma^2}, \cdots, s_m e^{-s_m^2/2\sigma^2}]^{\mathrm{T}}$.

Let $s \leftarrow s - \mu_0 \Delta s$.

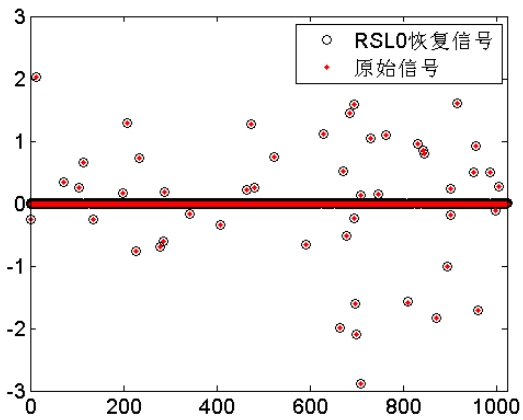
If $\|As - x\|_2 > \varepsilon$ projiect s on $As = x$ with:

$s \leftarrow s - A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}(As - x)$.

Set $\hat{s}_K = s$.

Step 4

Final answer is $\hat{s} = \hat{s}_K$.



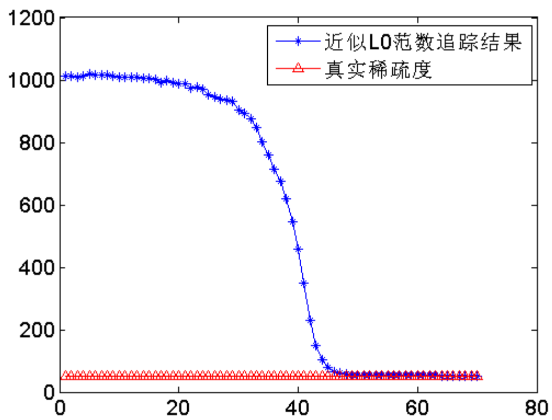


Figure.9. Recovery result by using RSL0

10. L1_SplitBregmanIteration (SBIL1) [12]

Algorithm 10: CS recovery using SBIL1

While $\left\|u^k-u^{k-1}\right\|_2>tol$

For $n=1$ to N

$u^{k+1}=\min_u H(u)+\frac{\lambda}{2}\left\|d^k-\Phi(u)-b^k\right\|_2^2$, use a wide variety of optimization techniques to solve this problem.

$d^{k+1}=\min_d |d|+\frac{\lambda}{2}\left\|d-\Phi\left(u^{k+1}\right)-b^k\right\|_2^2$, compute the optimal value of d using shrinkage operators:

$$d^{k+1}=shrink\left(\Phi(u)+b^k,1/\lambda\right),\ \ shrink(x,\gamma)=\frac{x}{|x|}\max\left(|x|-\gamma,0\right)$$

End

$b^{k+1}=b^k+\left(\Phi\left(u^{k+1}\right)-d^{k+1}\right)$

End

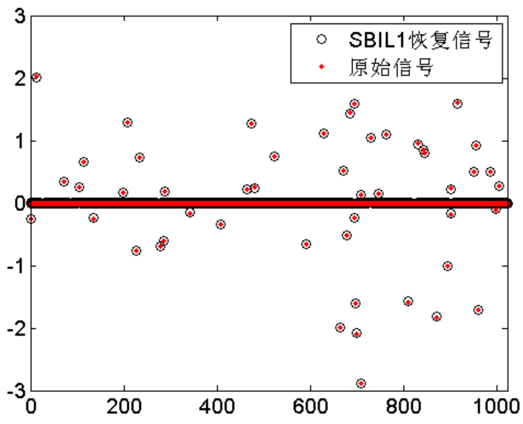


Figure.10. Recovery result by using SBIL1

11. Smoothed l0-Pseudonorm Algorithm (SL0) [13]

Algorithm 11: CS recovery using SL0

Step 1

Set $\hat{s}_0=A^{\dagger}x$. Choose a suitable decreasing sequence for $\sigma:[\sigma_1\cdots\sigma_J]$.

Step 2

For $j=1,\cdots,J$:

Let $\sigma=\sigma_i$.

Maximize $F_\sigma(s)$ subject to $As=x$, using L iteration of steepest ascent:

Initialization: $s=\hat{s}_{j-1}$.

For $l = 1, 2, \cdots, L$

Let $s \leftarrow s + (\mu\sigma^2)\nabla F_\sigma(s)$.

Project s back onto the feasible set $\{s \mid As = x\}$:

$s \leftarrow s - A^{\dagger}(As - x)$.

Set $\hat{s}_j = s$.

Step 3

Final answer is $\hat{s} = \hat{s}_J$.

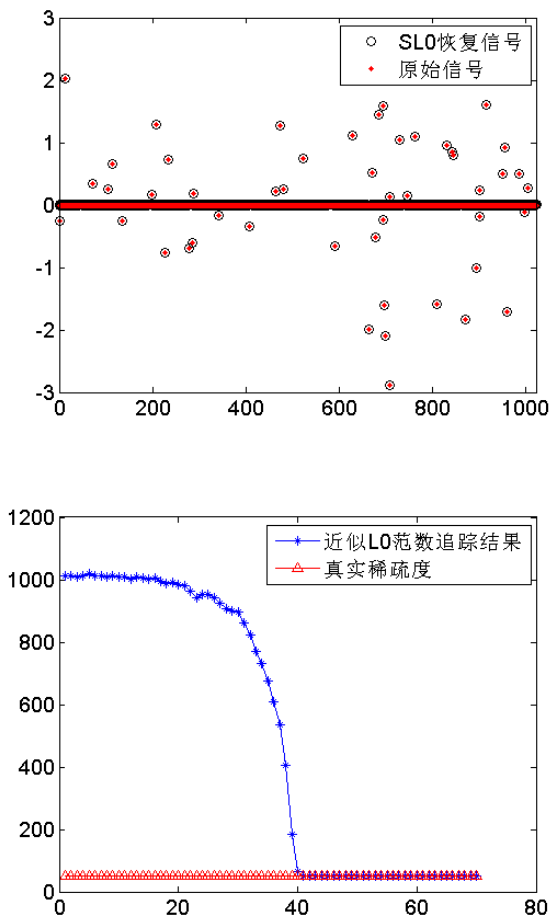


Figure.11. Recovery result by using SL0

12. Minimization of Approximate Lp Pseudonorm Using a Quasi-Newton Algorithm (UALP) [14]

Algorithm 12: CS recovery using UALP

Step 1

Input: $p, \varepsilon_1, \varepsilon_J$, and J.

set $\xi^{(1)} = 0$.

Step 2

Compute ε_i for $i = 2, 3, \cdots, J-1$ using $\varepsilon_i = e^{-\beta i}$ where $\beta = \log(\varepsilon_1 / \varepsilon_J) / (J-1)$.

Step 3

Use $x_s = Q_1 R^{-\mathrm{T}} y, V_r = Q_2$ to compute x_s and V_r.

Step 4

Repeat for $k = 1, \cdots, J$

Set $\varepsilon = \varepsilon_k$ and use ξ^k as an initial point. Apply the **BFGS** algorithm to solve the problem in

$$\min_{\xi} imize \quad F_{p,\varepsilon}(\xi) = \sum_{i=1}^{N} [(x_{si} + v_i^{\mathrm{T}} \xi)^2 + \varepsilon^2]^{p/2}.$$

Denote the solution as ξ^{k+1}.

Step 5

Set $\xi^* = \xi^{(J+1)}, x = x_s + V_r \xi^*$, and terminate.

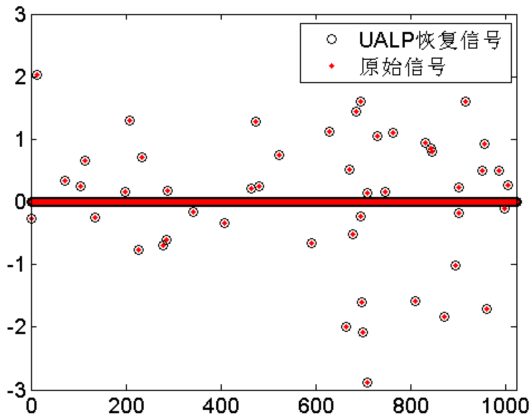


Figure.12. Recovery result by using URLP

Reference

[1] Joel A. Tropp and Anna C. Gilbert Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit，IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 53, NO. 12.

[2] Needell D，Tropp J A CoSaMP：Iterative signal recovery from incomplete and inaccurate samples[J]．Applied and Computation Harmonic Analysis，2009，26：301-321.

[3] D.Needell, J.A. Tropp．CoSaMP: Iterative signal recoveryfrom incomplete and inaccurate

samples[J]. Communications of theACM，2010，53(12)：93-100.
[4] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” SIAM J. Imaging Sciences, vol. 2, no. 1, pp. 183-202, 2009.
[5] Blumensath T, Davies M E. Iterative hard thresholding for compressed sensing[J]. Applied & Computational Harmonic Analysis, 2009, 27(3):265-274.
[6] Blumensath T, Davies M E. Iterative Thresholding for Sparse Approximations[J]. Journal of Fourier Analysis and Applications, 2008, 14(5):629-654.
[7] Chartrand and W. Yin, “Iteratively Reweighted Algorithms for Compressed Sensing,” 2008.
[8] I. Daubechies, M. Defrise, and C. D. Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,” Comm. Pure Appl. Math., vol. 57, pp. 1413-1457, 2004.
[9] J. K. Pant, W.-S. Lu, and A. Antoniou,“Reconstruction of sparse signals by minimizing a re-weighted approximate l0-norm in the null space of the measurement matrix,” IEEE Inter. Midwest Symp. on Circuits-Syst, pp. 430–433, 2010.
[10] Candès E J, Wakin M B, Boyd S P. Enhancing sparsity by reweighted L1 minimization.[J]. Journal of Fourier Analysis & Applications, 2007, 14(5):877-905.
[11] H. Mohimani, M. Babie-Zadeh, and C. Jutten,“A fast approach for overcomplete sparse decomposition based on smoothed l0-norm,” IEEE Trans. Signal Process., vol. 57, no. 1, pp. 289-301, Jan. 2009.
[12] Yin W, Osher S, Goldfarb D, et al.Bregman Iterative Algorithms for L1 Minimization with Applications to Compressed Sensing[J]. Siam Journal on Imaging Sciences, 2008, 1(1):143-168.
[13] H. Mohimani, M. Babie-Zadeh, and C. Jutten,“A fast approach for overcomplete sparse decomposition based on smoothed l0-norm,” IEEE Trans. Signal Process., vol. 57, no. 1, pp. 289-301, Jan. 2009.
[14] Pant J K, Lu W S, Antoniou A.Unconstrained regularized Lp -norm based algorithm for the reconstruction of sparse signals[C] IEEE International Symposium on Circuits and Systems. IEEE, 2011:1740-1743.