Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.2016.m1428

Vol. **9**, No. 2, pp. 169-184
May 2016

One-Bit Compressed Sensing by Greedy Algorithms

Wenhui Liu, Da Gong and Zhiqiang Xu*

LSEC, Institute of Computational Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China.

Received 9 November 2014; Accepted 7 April 2015

Abstract. Sign truncated matching pursuit (STrMP) algorithm is presented in this paper. STrMP is a new greedy algorithm for the recovery of sparse signals from the sign measurement, which combines the principle of consistent reconstruction with orthogonal matching pursuit (OMP). The main part of STrMP is as concise as OMP and hence STrMP is simple to implement. In contrast to previous greedy algorithms for one-bit compressed sensing, STrMP only need to solve a convex and unconstrained subproblem at each iteration. Numerical experiments show that STrMP is fast and accurate for one-bit compressed sensing compared with other algorithms.

AMS subject classifications: 94A12; 65H99; 65D15.

Key words: Compressed sensing, sparse signals, greedy algorithm.

1. Introduction

Compressed sensing, or compressive sensing provides a new method of data sampling and reconstruction, which allows to recover sparse signals from much fewer measurements [5,7]. Suppose that we have an unknown sparse signal $\hat{x} \in \mathbb{R}^n$ with $\|\hat{x}\|_0 \leq s$ and $s \ll n$, where $\|\cdot\|_0$ denotes the number of nonzero components. We observe the signal as

$$b = A\hat{x},$$

where $A \in \mathbb{R}^{m\times n}$ is called measurement matrix, $b \in \mathbb{R}^m$ is the vector of measurements. Compressed sensing shows that only $m = O(s\log(n/s))$ measurements are sufficient for exact reconstruction of $\hat{x}$ under many settings for the measurement matrix A [6,7].

1.1. One-Bit compressed sensing

In compressed sensing, it is supposed that the measurements have infinite bit precision. However, in practice what we get is quantized measurements. In other words, the entries in the measurement vector b must be mapped to a discrete set of values

*Corresponding author. *Email addresses:* liuwenhui11@mails.ucas.ac.cn (W. -H. Liu), gongda@lsec.cc.ac.cn (D. Gong), xuzq@lsec.cc.ac.cn (Z. -Q. Xu)

 ©2016 Global-Science Press

https://doi.org/10.4208/nmtma.2016.m1428 Published online by Cambridge University Press

$\mathcal{A}$. There are much work about the recovery of the general signal from the quantized measurements [18]. In this paper, we focus on the case where $\mathcal{A} = \{-1, 1\}$ with the mapping being done by the sign function. So we need to recover an s-sparse signal from $y := \text{sign}(b) \in \{-1, 1\}^m$. This problem is called one-bit compressed sensing, which was first introduced by Boufounos-Baraniuk [4]. In one-bit compressed sensing, we observe original signal as:

$$y = \text{sign}(A\hat{x}),$$

where $y \in \mathbb{R}^m$ with each element is sign of the corresponding element of $A\hat{x}$. That means we lost all magnitude information of $A\hat{x}$. Following [8], $x^\sharp \in \mathbb{R}^n$ is called *a solution for one-bit compressed sensing corresponding to* A *and* $\hat{x}$ if it satisfies

(i) consistence, i.e. $\text{sign}(Ax^\sharp) = \text{sign}(A\hat{x})$,

(ii) sparsity, i.e. $\|x^\sharp\|_0 \leq \|\hat{x}\|_0$.

A simple observation is that $\text{sign}(Ac\hat{x}) = \text{sign}(A\hat{x})$ and $\|c\hat{x}\|_0 \leq \|\hat{x}\|_0$ where $c > 0$ is a scale. Thus the best one-bit compressed sensing can do is to recover $\hat{x}$ up to a positive scale. Therefore, we usually expect to recover original signal on the unit Euclidean sphere.

1.2. Previous work

A straightforward way to obtain a solution for one-bit compressed sensing is to solve the following program:

$$\begin{aligned} &\min \ \|x\|_0 \\ &\text{s. t. } y = \text{sign}(Ax) \ \text{ and } \ \|x\|_2 = 1. \end{aligned} \tag{1.1}$$

Since (1.1) is computational intractable, similar with compressed sensing, one can replace the ℓ_0 norm by the more tractable ℓ_1 norm and obtain that (see [4, 9, 10])

$$\begin{aligned} &\min \ \|x\|_1 \\ &\text{s. t. } y = \text{sign}(Ax) \ \text{ and } \ \|x\|_2 = 1. \end{aligned} \tag{1.2}$$

Many algorithms have been proposed to solve (1.2). Particularly, in [9], Laska et. al. use the augmented Lagrangian optimization framework to design RSS algorithm by employing a restricted-step subroutine to solve a non-convex subproblem. Binary iterative hard thresholding (BIHT) and adaptive outlier pursuit (AOP) are introduced in [8] and [12], respectively. BIHT is a modification of iterative hard thresholding which is to solve compressed sensing problem (see [2]). AOP is a robust algorithm built on BIHT, and it is exactly BIHT when measurements are noise free. The numerical experiments in [12] show that AOP performs better than the previous existing algorithms in terms of the recovery performance. In [10], Plan and Vershynin replace the normalization constraint $\|x\|_2 = 1$ by $\|Ax\|_1 = c_0$ and give an analysis of the following convex

https://doi.org/10.4208/nmtma.2016.m1428 Published online by Cambridge University Press

program

$$\begin{aligned} &\min \ \|x\|_1 \\ &\text{s. t. } y = \operatorname{sign}(Ax) \text{ and } \|Ax\|_1 = c_0, \end{aligned}$$

where c_0 is a given positive constant. It is actually equivalent to

$$\begin{aligned} &\min \ \|x\|_1 \\ &\text{s. t. } y_i(Ax)_i \geq 0, \ i = 1, \ldots, m \\ &\quad y^\top Ax = c_0, \end{aligned}$$

where y_i means the ith element of y, and $(Ax)_i$ stands for the ith element of Ax.

Moreover, in compressed sensing, one develops many greedy algorithms to recover the sparse signals, such as OMP, CoSaMP, ROMP, OMMP and subspace pursuit etc (see [14–17, 19]). Motivated by these algorithms in compressed sensing, one also designs greedy algorithms for one-bit compressed sensing. Particularly, the matching sign pursuit (MSP) algorithm is presented in [3]. However, MSP suffers from time-consuming since it solves a non-convex and constrained sub-problem at each iteration. Naturally, one may be interested in designing more efficiently greedy-type algorithm for one-bit compressed sensing, which is also the starting point of this project.

1.3. Our contribution

The aim of this paper is to present the sign truncated matching pursuit (STrMP) algorithm, which is a new greedy algorithm to solve one-bit compressed sensing. In particular, motivated by [10], we replace the unit ℓ_2-norm constraint by $\|Ax\|_1 = c_0$ where c_0 is any fixed positive constant, and hence we consider the following optimization problem:

$$\begin{aligned} &\min \ \|x\|_0 \\ &\text{s. t. } y = \operatorname{sign}(Ax) \text{ and } \|Ax\|_1 = c_0. \end{aligned} \tag{1.3}$$

A key step of STrMP algorithm is to use

$$j_0 := \operatorname*{argmax}_i |A_i^\top y|$$

to choose the first index j_0, where $A_i^\top$ denotes transposition of the ith column of A. We also prove that $j_0 \in \operatorname{supp}(\hat{x})$ with high probability provided $m = O(s \log n)$ and A is a Gaussian matrix. If $j_0 \in \operatorname{supp}(\hat{x})$, we can remove the constraint $\|Ax\|_1 = c_0$ and transform (1.3) to the program in the form of

$$\begin{aligned} &\min \ \|z\|_0 \\ &\text{s. t. } y = \operatorname{sign}(Pz + q), \end{aligned}$$

where $P \in \mathbb{R}^{m \times (n-1)}$ and $q \in \mathbb{R}^m$, which is more convenient for designing greedy algorithms. We will introduce this and the algorithm in detail in Section 2. STrMP algorithm overcomes the bottleneck of MSP that a non-convex and constrained problem

https://doi.org/10.4208/nmtma.2016.m1428 Published online by Cambridge University Press

need to be solved at each iteration. In fact, STrMP just need to solve a convex and unconstrained sub-problem at each iteration. Hence, numerical experiments show that STrMP outperforms previous existing algorithms in terms of speed. Moreover, the numerical experiments also show that the recovery performance of STrMP is better than that of MSP and RSS, and is similar with that of BIHT or AOP. The last but not the least, besides the sparsity level, no parameter need to be adjusted in STrMP by the user.

1.4. Terminology and organization

In the following of this paper, we denote by A_i the ith column of matrix A, and A_{ij} the ith row and jth column component of A. We use e_i to denote the unit vector with the ith element being 1 and other elements being zero. For a vector $y \in \mathbb{R}^m$, $\mathrm{diag}(y) \in \mathbb{R}^{m\times m}$ denotes the diagonal matrix whose diagonal elements are corresponding elements of y. The symbol $[n]$ denotes the index set $\{1, 2, \ldots, n\}$. For a subset $T \subset [n]$, $|T|$ denotes the number of elements in T. For a vector $x \in \mathbb{R}^n$ and a subset $T \subset [n]$, we use $x_T \in \mathbb{R}^{|T|}$ to denote the vector containing the entries of x indexed by T. We use $x|_T \in \mathbb{R}^n$ to denote the vector whose entries indexed by T are corresponding entries to x and the entries indexed by T^c are zero. For a matrix $A \in \mathbb{R}^{m\times n}$, $A_T \in \mathbb{R}^{m\times|T|}$ denotes a matrix which contains the columns of A indexed by T. We define the *sign truncated* function as $(\cdot)_- := \min\{\cdot, 0\}$.

The rest of this paper is organized as follows. In Section 2, we derive the STrMP algorithm, and also present the theorem which shows that one can choose the first index successfully with high probability provided $m = O(s\log(n-s))$ and A is a Gaussian matrix. The STrMP-ℓ_1 algorithm, which is an adjustment of the STrMP algorithm, is introduced in Section 3. The numerical results, comparing with other algorithms, are illustrated in Section 4. We conclude our results in Section 5.

2. STrMP algorithm

We derive the algorithm STrMP in this section. The algorithm is built on the following theorem, which shows that one can find a $x^\sharp$ which is a solution for one-bit compressed sensing by solving a program without the normalized constraint $\|Ax\|_1 = c_0$.

Theorem 2.1. *Suppose that $j_0 \in \mathrm{supp}(\hat{x})$ and $y^\top A_{j_0} \neq 0$ where $y = \mathrm{sign}(A\hat{x})$. Suppose that $z^\sharp \in \mathbb{R}^{n-1}$ is a solution to*

$$\begin{aligned} &\min \ \|z\|_0 \\ &s.\ t.\ y = \mathrm{sign}(Pz + q) \end{aligned} \tag{2.1}$$

where

$$P = \left(I - \frac{A_{j_0}y^\top}{y^\top A_{j_0}}\right) A_{[n]\setminus\{j_0\}}, \quad q = \frac{c_0}{y^\top A_{j_0}} A_{j_0}.$$

https://doi.org/10.4208/nmtma.2016.m1428 Published online by Cambridge University Press

Suppose that $x^\sharp$ is defined by

$$x^\sharp_{[n]\setminus\{j_0\}} := z^\sharp, \quad x^\sharp_{j_0} := \frac{c_0 - y^\top A_{[n]\setminus\{j_0\}} z^\sharp}{y^\top A_{j_0}}. \tag{2.2}$$

Then $\|Ax^\sharp\|_1 = c_0$ and $x^\sharp \in \mathbb{R}^n$ is a solution for one-bit compressed sensing corresponding to A and $\hat{x}$.

Remark 2.1. According to Theorem 2.1, the $x^\sharp$ defined by (2.2) is s-sparse and consistent. Then Theorem 2 in [8] shows that if A is a Gaussian matrix, then

$$\|x^\sharp/\|x^\sharp\|_2 - \hat{x}/\|\hat{x}\|_2\|_2 = O\left(\frac{s}{m}\log\frac{mn}{s}\right),$$

holds with high probability.

Based on Theorem 2.1, if one knows an index $j_0 \in \mathrm{supp}(\hat{x})$ in advance, one can construct a solution for one-bit compressed sensing corresponding to A and $\hat{x}$ by solving a program in the form of (2.1). STrMP algorithm uses

$$j_0 = \underset{i\in[n]}{\mathrm{argmax}} |A_i^\top y|$$

to choose the index j_0, and we also prove that $j_0 \in \mathrm{supp}(\hat{x})$ with high probability provided $m = O(s\log n)$ and A is a Gaussian matrix:

Theorem 2.2. *Let $\hat{x}$ be an s-sparse vector in $\mathbb{R}^n$. Let $A \in \mathbb{R}^{m\times n}$ be a random matrix with independent standard normal entries. Assume for some $\epsilon > 0$*

$$m \geq \frac{\pi}{2} s\left(\epsilon + \sqrt{\epsilon^2 + 2\log(n-s)}\right)^2.$$

Then

$$\underset{i\in[n]}{\mathrm{argmax}} |(A^\top \mathrm{sign}(A\hat{x}))_i| \in \mathrm{supp}(\hat{x})$$

holds with probability at least $1 - 2e\cdot\exp(-c\epsilon^2)$, where c is an absolute constant.

Remark 2.2. We also make numerical experiments to test the success probability of the choice of the first index. We set $n = 1000,\ s = 15$ and change m within the range $[30,\ 200]$. We repeat the experiment 100 times for each m and calculate the success rate. The numerical results show that $m \approx 150$ measurements are enough for correctly selecting the first index with the success rate being 1.

We now focus on (2.1). A simple observation is that we can rewrite (2.1) as

$$\begin{aligned} &\min\ \|z\|_0 \\ &\text{s. t. } Cz + d \geq 0, \end{aligned} \tag{2.3}$$

https://doi.org/10.4208/nmtma.2016.m1428 Published online by Cambridge University Press

where $C = \mathrm{diag}(y)P$ and $d = \mathrm{diag}(y)q$. We use greedy algorithms to find an approximate solution to (2.3). To state the algorithm, we recall that the *sign truncated function* $(\cdot)_-$ which is defined by $(\cdot)_- := \min\{0, \cdot\}$. The algorithm begins with an initial support set $\Lambda^0 = \emptyset$ and an estimate $z^0 = 0$. At the kth iteration, the algorithm computes the product of $C^\top$ and the sign truncated vector $(Cz^k + d)_-$ and form a proxy, where the truncated function $(\cdot)_-$ is applied component-wise to $Cz^k + d$. And use

$$j^k = \underset{i}{\mathrm{argmax}} |C_i^\top (Cz^k + d)_-|$$

to choose a new index where $C_i^\top$ denotes transposition of C_i. Set $\Lambda^{k+1} := \Lambda^k \cup \{j^k\}$. Next we solve a convex problem to enforce the consistence:

$$z^{k+1} := \underset{z}{\mathrm{argmin}} \|(Cz + d)_-\|_2^2 \ \text{ s. t. } \mathrm{supp}(z) \subset \Lambda^{k+1}, \tag{2.4}$$

which is essential unconstrained in a lower dimensional space. We use two-point step size gradient method [1] to solve it and we introduce the algorithm in detail in Appendix B. We conclude above fact and formulate our algorithm in Algorithm 1.

Algorithm 2.1 The Sign Truncated Matching Pursuit (STrMP) Algorithm

Inputs: $A \in \mathbb{R}^{m\times n},\ y \in \{-1, 1\}^m,\ c_0 > 0,\ s \in \mathbb{Z}^+$, residual tolerance $\varepsilon > 0$

Initialization: $\Lambda^0 = \emptyset \subset [n]$, $z^0 = 0$, $k = 0$

Compute the first index $j_0 = \underset{i\in[n]}{\mathrm{argmax}} |(A^\top y)_i|$

Set

$$C = \mathrm{diag}(y)\left(I - \frac{A_{j_0} y^\top}{y^\top A_{j_0}}\right) A_{[n]\setminus\{j_0\}},$$

$$d = \frac{c_0}{y^\top A_{j_0}} \mathrm{diag}(y) A_{j_0}.$$

while $\|(Cz^k + d)_-\|_2^2 \geq \varepsilon$ and $k \leq s - 1$

Match: $h^k = C^\top (Cz^k + d)_-$

Identify: $j^k = \underset{i}{\mathrm{argmax}} |h_i^k|$

Update: $\Lambda^{k+1} = \Lambda^k \cup \{j^k\}$

Solve:

$$z^{k+1} = \underset{z\in\mathbb{R}^{n-1}}{\mathrm{argmin}} \|(Cz + d)_-\|_2^2 \ \text{ s. t. } z|_{(\Lambda^{k+1})^c} = 0$$

Increase iteration count: $k = k + 1$

end while

Set $x^k_{[n]\setminus\{j_0\}} = z^k, \quad x^k_{j_0} = \frac{c_0 - y^\top A_{[n]\setminus\{j_0\}} z^k}{y^\top A_{j_0}}$

Return: $\frac{x^k}{\|x^k\|_2}$

https://doi.org/10.4208/nmtma.2016.m1428 Published online by Cambridge University Press

Remark 2.3. The code in Algorithm 2.1 describes a version of the STrMP algorithm. Similar to OMP, there are many adjustments of STrMP. Particularly, in the identify step, one can select many atoms per iteration instead of one atom, which is also helpful for improving the performance.

Remark 2.4. A simple observation is that the output result of STrMP algorithm is independent of c_0 provided $c_0 \gg \varepsilon$. In fact, for different positive c_0, the solution z^{k+1} in the update step is the same up to a positive scale. And hence, in the identify step, STrMP algorithm chooses the same index j^k even c_0 is different.

3. STrMP-ℓ_1 algorithm

In the update step of the STrMP algorithm, we use $\min_z \|(Cz+d)_-\|_2$ to enforce the consistence. Obviously, If $z^\sharp$ is supposed to be the solution, we can expect $(Cz^\sharp + d)_-$ is exactly sparse. Inspired by compressed sensing, we can replace ℓ_2-norm by ℓ_1-norm. Thus one may consider to solve following subproblem in the update step

$$z^{k+1} := \operatorname*{argmin}_z \|(Cz+d)_-\|_1 \ \text{ s. t. } \operatorname{supp}(z) \subset \Lambda^{k+1}, \tag{3.1}$$

since ℓ_1-norm is more effective to characterize the sparsity than ℓ_2-norm. We apply quasi-Newton method to solve the subproblem (3.1). Correspondingly, in the match step, we compute

$$h^k = C^\top \operatorname{sign}((Cz^k + d)_-),$$

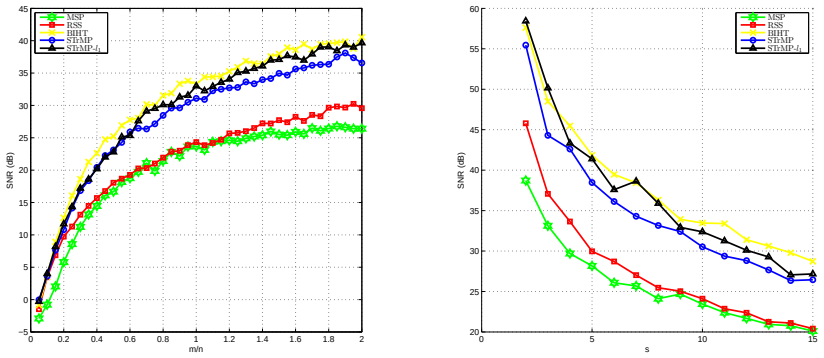
which is exactly a subgradient of $\|(Cz+d)_-\|_1$ at z^k (see [8]), and in identify step we still set $j^k = \operatorname*{argmax}_i |h_i^k|$. Here, to state conveniently, we set $\operatorname{sign}(0) = 0$. In the following of this paper, we denote this algorithm by STrMP-ℓ_1.

4. Numerical experiment

In this section, we make numerical experiments to compare the performance of STrMP with that of other existing methods as mentioned before, such as BIHT, MSP and RSS with showing that both STrMP and STrMP-ℓ_1 are fast and accurate.

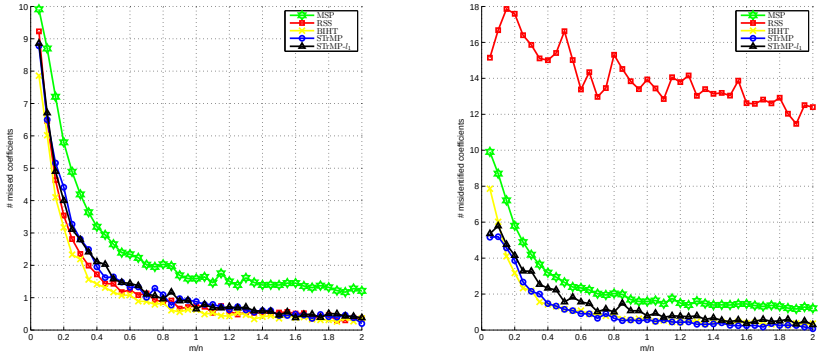
In our experiments, the measurement matrix $A \in \mathbb{R}^{m\times n}$ is generated by Gaussian random matrix. The original signal $\hat{x} \in \mathbb{R}^n$ is s-sparse, and its non-zero coefficients are drawn from standard normal distribution, and it is normalized to have unit ℓ_2 norm. In all following experiments, we set $n = 1000$. And in all following figures, the blue line with circles denotes STrMP, the black line with upward triangle denotes STrMP-ℓ_1, the yellow line with cross denotes BIHT, the red line with squares denotes RSS and the green line with hexagram denotes MSP. Again, we do not test AOP algorithm here since it is exactly BIHT when measurements are noise free.

https://doi.org/10.4208/nmtma.2016.m1428 Published online by Cambridge University Press



(a) SNR (fixed $s = 10$) (b) SNR (fixed $m = 1000$)

Figure 1: Accuracy experiments: Averaged SNR for (a) fixed $s = 10$ and $n = 1000$ while m is changed with m/n between 0.05 and 2, and (b) fixed $n = 1000$ and $m = 1000$ while s is between 1 and 15. STrMP, STrMP-ℓ_1 and BIHT have similar performance, RSS and MSP perform similarly. And STrMP, STrMP-l_1 and BIHT are better than others.



(a) number of missed coefficients (b) number of misidentified coefficients

Figure 2: Accuracy experiments: Averaged number of missed coefficients for (a) fixed $s = 10$ and $n = 1000$ with m/n is between 0.05 and 2. Averaged number of misidentified coefficients for (b) $s = 10$ and $n = 1000$ with m/n is between 0.05 and 2. STrMP, STrMP-ℓ_1, BIHT and RSS have similar performance in number of missed coefficients, STrMP, STrMP-ℓ_1, BIHT and MSP have similar performance in number of misidentified coefficients. However, the result of RSS in number of misidentified coefficients shows poor performance.

4.1. Accuracy test

In this experiment, we test the reconstruction accuracy in three different ways: the average signal-to-noise ratio (SNR), the average number of missed coefficients and the average number of misidentified coefficients, which are defined as follows

https://doi.org/10.4208/nmtma.2016.m1428 Published online by Cambridge University Press

- SNR:
$$10\log_{10}\left(\frac{\|x^\sharp\|^2}{\|x^\sharp-\hat{x}\|^2}\right)$$

- Number of missed coefficients:
$$|\{i\in[n]:\hat{x}_i\neq 0 \text{ and } x_i^\sharp=0\}|$$

- Number of misidentified coefficients:
$$|\{i\in[n]:\hat{x}_i=0 \text{ and } x_i^\sharp\neq 0\}|.$$

Here SNR is used to measure overall reconstruction performance of the recovery algorithms. Since the original signal $\hat{x}$ is sparse, it is also important to identify the support of original signal. To measure this kind of performance, it is helpful to calculate the number of missed coefficients and of misidentified coefficients.

The numerical results are depicted in Fig. 1 and Fig. 2. In these experiments, as mentioned before, we set $n=1000$. In Fig. 1(a), Fig. 2(a) and Fig. 2(b), we fix $s=10$ and change m/n within the range $[0.05,2]$ with the step 0.05. And hence 40 different m/n are considered. For each m/n, we perform 100 trials and record the average value. In Fig. 1(b), we fix $n=1000$ and $m=1000$ while s is between 1 and 15. We also repeat the experiment 100 times for each s and plot the mean value. The plots in Fig. 1 and Fig. 2 show that STrMP, STrMP-ℓ_1 and BIHT perform similarly. The performance of STrMP-ℓ_1 and BIHT is slightly better than STrMP in SNR. They are much better than MSP and RSS. Fig. 2(b) shows that RSS exhibits poorer performance for the number of misidentified coefficients which are also observed and analyzed in [9]. We may intuitively explain this phenomenon as objective function of the model RSS solved is ℓ_1 norm (not ℓ_0 norm), so the reconstruction result is usually not sparse. However, there is a projection process in BIHT algorithm, which guarantee the reconstruction signal is sparse. STrMP, STrMP-ℓ_1 and MSP use ℓ_0 norm model which can also produce sparse solution. Thus STrMP, STrMP-ℓ_1, MSP and BIHT have less mistakes in misidentified coefficients than RSS.

4.2. Consistency test

In this subsection, we test whether $x^\sharp$ is consistent. To do that, we measure the Hamming error which is defined as follows:

- Hamming error:
$$\|\mathrm{sign}(Ax^\sharp)-y\|_0/m.$$

For each m/n, we also perform 100 trials and record the average values. The plots in Fig. 3 show that STrMP, STrMP-ℓ_1 and BIHT work very well and they have better performance than that of MSP and RSS.

https://doi.org/10.4208/nmtma.2016.m1428 Published online by Cambridge University Press

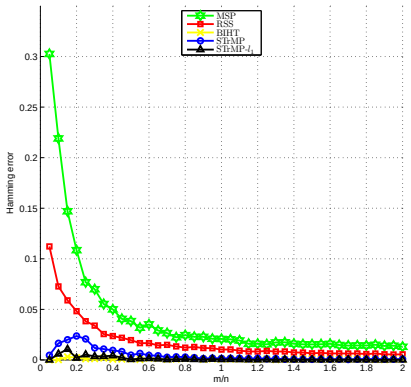
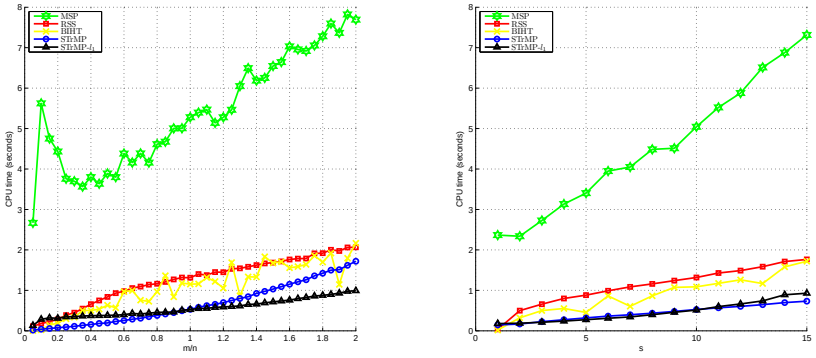


Figure 3: Consistency experiments: Averaged Hamming error between $\mathrm{sign}(Ax^\sharp)$ and y for $s = 10$ with m/n is between 0.05 and 2. As the figure depicts that STrMP and STrMP-ℓ_1 have the similar performance as BIHT in terms of consistency test, and they are better than others.



(a) CPU time (fixed $s = 10$) (b) CPU time (fixed $m = 1000$)

Figure 4: Speed experiments: Averaged CPU time for (a) $s = 10$ with m/n is between 0.05 and 2, (b) $m = 1000$ with s is between 1 and 15. These demonstrate that STrMP and STrMP-ℓ_1 take least time comparing with other algorithms.

4.3. Speed test

At last, we test the speed of the algorithms by recording the average computational time. Fig. 4 depicts that STrMP and STrMP-ℓ_1 outperform MSP significantly, and they are also faster than BIHT and RSS. The CPU time of MSP increases quickly as the measurements and the sparsity increase. The reason is that, in the MSP algorithm, a nonconvex and constrained subproblem need to be solved at each iteration. However, time consumed by STrMP grows very slowly as measurements and sparsity increase, which is particularly striking for STrMP and STrMP-ℓ_1. This is because of the sub-

https://doi.org/10.4208/nmtma.2016.m1428 Published online by Cambridge University Press

problems of the STrMP algorithms are convex and unconstrained.

5. Conclusion

In this paper, we have introduced a fast and accurate greedy algorithm for the one-bit compressive sensing. The subproblem in STrMP is convex and unconstrained. And hence, STrMP algorithm is faster than previously existing one-bit compressed sensing algorithms. The choice of the first index plays an important role in STrMP. We prove that the first index belongs to the support of the original signal with high probability provided $m = O(s \log n)$. The numerical experiments show that the recovery performance of STrMP and STrMP-ℓ_1 algorithm is similar with that of BIHT, and they are much better than that of RSS and MSP. One can note that the main part of STrMP is as concise as OMP. So, it will be very interesting to investigate the convergence property of STrMP using the technology developed in the study of OMP [13–15].

Appendix A. Proofs of Theorems 2.1 and 2.2

Proof. [Proof of Theorem 2.1] To this end, we only need to prove that $x^\sharp$ satisfies

(i) consistence, i.e. $\operatorname{sign}(Ax^\sharp) = y$,

(ii) sparsity, i.e. $\|x^\sharp\|_0 \leq \|\hat{x}\|_0$,

(iii) normalization, i.e. $\|Ax^\sharp\|_1 = c_0$.

We first consider

$$\begin{aligned}\operatorname{sign}(Ax^\sharp) &= \operatorname{sign}(A_{[n]\setminus\{j_0\}} z^\sharp + x^\sharp_{j_0} A_{j_0}) \\ &= \operatorname{sign}\left(\left(I - \frac{A_{j_0} y^\top}{y^\top A_{j_0}}\right) A_{[n]\setminus\{j_0\}} z^\sharp + \frac{c_0}{y^\top A_{j_0}} A_{j_0}\right) \\ &= \operatorname{sign}(Pz^\sharp + q) = y,\end{aligned}$$

which implies (i). We turn to (iii). Note that

$$\begin{aligned}\|Ax^\sharp\|_1 = y^\top Ax^\sharp &= y^\top A_{[n]\setminus\{j_0\}} z^\sharp + y^\top A_{j_0} x^\sharp_{j_0} \\ &= y^\top A_{[n]\setminus\{j_0\}} z^\sharp + y^\top A_{j_0} \frac{c_0 - y^\top A_{[n]\setminus\{j_0\}} z^\sharp}{y^\top A_{j_0}} = c_0.\end{aligned}$$

We arrive at (iii). We next consider (ii). To this end, we need to show that

$$\|x^\sharp\|_0 \leq \|\hat{x}\|_0.$$

https://doi.org/10.4208/nmtma.2016.m1428 Published online by Cambridge University Press

Without loss of generality, we can assume that $\|A\hat{x}\|_1 = y^\top A\hat{x} = c_0$ (otherwise, we can multiply $\hat{x}$ by a positive constant), which implies that

$$\hat{x}_{j_0} = \frac{c_0 - y^\top A_{[n]\setminus\{j_0\}}\hat{x}_{[n]\setminus\{j_0\}}}{y^\top A_{j_0}}. \tag{A.1}$$

Then a simple calculation shows that

$$\begin{aligned} y &= \operatorname{sign}(A\hat{x}) = \operatorname{sign}(A_{[n]\setminus\{j_0\}}\hat{x}_{[n]\setminus\{j_0\}} + A_{j_0}\hat{x}_{j_0}) \\ &= \operatorname{sign}(P\hat{x}_{[n]\setminus\{j_0\}} + q). \end{aligned}$$

Here, in the last equality, we use (A.1). And hence $\hat{x}_{[n]\setminus\{j_0\}}$ satisfies the constraint condition (2.1) which implies that

$$\|z^\sharp\|_0 \le \|\hat{x}_{[n]\setminus\{j_0\}}\|_0 = \|\hat{x}\|_0 - 1.$$

So, the definition of $x^\sharp$ implies that

$$\|x^\sharp\|_0 \le \|z^\sharp\|_0 + 1 \le \|\hat{x}\|_0.$$

We arrive at the conclusion. □

To prove Theorem 2.2, we first introduce Hoeffding-type inequality (see [11, Proposition 5.10]).

Lemma A.1 (Hoeffding-type inequality)**.** *Let $\zeta_1, \ldots, \zeta_n$ be independent centered sub-gaussian random variables, and $K = \max_i \|\zeta_i\|_{\psi_2}$. Then for every $a = (a_1, \ldots, a_n) \in \mathbb{R}^n$ and every $\epsilon \ge 0$, we have*

$$P\left\{\left|\sum_{i=1}^n a_i\zeta_i\right| \ge \epsilon\right\} \le e \cdot \exp\left(-\frac{c_1\epsilon^2}{K^2\|a\|_2^2}\right),$$

where $c_1 > 0$ is an absolute constant.

Here, for a sub-gaussian random variable ξ,

$$\|\xi\|_{\psi_2} := \sup_{p\ge 1}(\mathbb{E}|\xi|^p)^{1/p}$$

(see also [11, Definition 5.7]). For a standard normal variable ξ, $\|\xi\|_{\psi_2}$ is bounded by $\sqrt{2}$. We are now turning to the proof of our theorem.

Proof. [Proof of Theorem 2.2] Without loss of generality, we can suppose that $\|\hat{x}\|_2 = 1$ and $\hat{x}_\jmath > 0$ where $\jmath := \operatorname{argmax}_{i\in[n]} |\hat{x}_i|$. We claim that, for any $\epsilon > 0$

$$\begin{aligned} &\mathbb{P}\Big\{\max_{k\notin\operatorname{supp}(\hat{x})} |\langle A_k, y\rangle| < \sqrt{m\epsilon^2 + 2m\log(n-s)}\Big\} \\ &\qquad \ge 1 - \exp\left(-\frac{\epsilon^2}{2}\right), \end{aligned} \tag{A.2}$$

https://doi.org/10.4208/nmtma.2016.m1428 Published online by Cambridge University Press

and

$$\begin{aligned}\mathbb{P}\left\{|\langle A_\jmath, y\rangle| \geq 2\cdot m\frac{\hat{x}_\jmath}{\sqrt{2\pi}} - \epsilon\sqrt{m}\right\}\\ \geq 1 - e\cdot\exp\left(-\frac{c_1\epsilon^2}{8}\right).\end{aligned} \tag{A.3}$$

Combining (A.2) and (A.3), to prove the conclusion we just need to show that

$$2\cdot m\frac{\hat{x}_\jmath}{\sqrt{2\pi}} - \epsilon\sqrt{m} \geq \sqrt{m\epsilon^2 + 2m\log(n-s)}, \tag{A.4}$$

which is equivalent to

$$\sqrt{m}\cdot\frac{2\hat{x}_\jmath}{\sqrt{2\pi}} \geq \sqrt{\epsilon^2 + 2\log(n-s)} + \epsilon. \tag{A.5}$$

Indeed, note that $\hat{x}_\jmath = \|\hat{x}\|_\infty \geq \frac{1}{\sqrt{s}}$ with $\hat{x}$ being s-sparse, we have

$$\sqrt{m}\cdot\frac{2\hat{x}_\jmath}{\sqrt{2\pi}} \geq \sqrt{\frac{2}{\pi}}\cdot\sqrt{\frac{m}{s}} \geq \sqrt{\epsilon^2 + 2\log(n-s)} + \epsilon,$$

which implies (A.5) and hence (A.4) holds. Here, in the last inequality, we use

$$m \geq \frac{\pi}{2}s\Big(\epsilon + \sqrt{\epsilon^2 + 2\log(n-s)}\Big)^2. \tag{A.6}$$

Hence, if the condition (A.6) is satisfied, we have

$$\operatorname*{argmax}_{i\in[n]} |(A^\top \operatorname{sign}(A\hat{x}))_i| \in \operatorname{supp}(\hat{x})$$

holds, with probability at least $1 - 2e\cdot\exp(-c\epsilon^2)$, which implies the conclusion, where $c := \min\{1, c_1/8\}$.

To this end, we still need to prove (A.2) and (A.3). We first consider (A.2). According to $y = \operatorname{sign}(A\hat{x})$, we obtain that the entries of y are $i.i.d.$ Bernoulli random variables, and they are independent of entries of A_k for those $k \notin \operatorname{supp}(\hat{x})$. Note that $\langle A_k, y\rangle \sim \mathcal{N}(0, m)$ provided $k \notin \operatorname{supp}(\hat{x})$. The Gaussian concentration inequality implies

$$P\{|\langle A_k, y\rangle| < t\} \geq 1 - \exp\left(-\frac{t^2}{2m}\right),$$

where $t > 0$. Since $\hat{x}$ is s-sparse, there are $(n-s)$ entries do not belong to $\operatorname{supp}(\hat{x})$. By using the union bound, we have

$$\mathbb{P}\Big\{\max_{k\notin\operatorname{supp}(\hat{x})} |\langle A_k, y\rangle| < t\Big\} \geq 1 - (n-s)\exp\left(-\frac{t^2}{2m}\right).$$

https://doi.org/10.4208/nmtma.2016.m1428 Published online by Cambridge University Press

Taking $t = \sqrt{m\epsilon^2 + 2m\log(n-s)}$, we obtain that

$$\mathbb{P}\{\max_{k\notin \mathrm{supp}(\hat{x})} |\langle A_k, y\rangle| < \sqrt{m\epsilon^2 + 2m\log(n-s)}\} \geq 1 - \exp\left(-\frac{\epsilon^2}{2}\right).$$

We next turn to (A.3). Without loss of generality we can assume that $\hat{x}_j > 0$. Note that $A_{ij}y_i,\ i \in [m]$ are $i.i.d.$ random variables since y_i only depends on the ith row of A. Then for every $\epsilon \geq 0$

$$\mathbb{P}\{|A_{ij}y_i| \geq \epsilon\} = \mathbb{P}\{|A_{ij}| \geq \epsilon\} \leq \exp\left(-\frac{\epsilon^2}{2}\right),$$

which implies that $A_{ij}y_i,\ i \in [m]$ are $i.i.d.$ sub-gaussian random variables. We next calculate the expectation of $A_{ij}y_i$. Let $\xi := A_{ij}$ and $\eta := \sum_{k\neq j} A_{ik}\hat{x}_k$. Then $\xi \sim \mathcal{N}(0,1)$ and $\eta \sim \mathcal{N}(0, 1-\hat{x}_j^2)$. Also, note that ξ and η are independent and

$$\begin{aligned}\mathbb{E}(A_{ij}y_i) &= \mathbb{E}\left(A_{ij}\cdot \mathrm{sign}\left(A_{ij}\hat{x}_j + \sum_{k\neq j} A_{ik}\hat{x}_k\right)\right)\\ &= \mathbb{E}(\xi\cdot \mathrm{sign}(\xi\hat{x}_j + \eta)).\end{aligned}$$

Now we have $\mathbb{E}(A_{ij}y_i) = 2\hat{x}_j/\sqrt{2\pi}$ from (A.7).

$$\begin{aligned}\mathbb{E}(A_{ij}y_i) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \xi\mathrm{sign}(\xi\hat{x}_j+\eta)\frac{1}{\sqrt{2\pi}}e^{-\frac{\xi^2}{2}}\frac{1}{\sqrt{2\pi(1-\hat{x}_j^2)}}e^{-\frac{\eta^2}{2(1-\hat{x}_j^2)}}\,\mathrm{d}\xi\,\mathrm{d}\eta\\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{-\frac{\eta}{\hat{x}_j}}(-\xi)\frac{1}{\sqrt{2\pi}}e^{-\frac{\xi^2}{2}}\frac{1}{\sqrt{2\pi(1-\hat{x}_j^2)}}e^{-\frac{\eta^2}{2(1-\hat{x}_j^2)}}\,\mathrm{d}\xi\,\mathrm{d}\eta\\ &\quad + \int_{-\infty}^{\infty}\int_{-\frac{\eta}{\hat{x}_j}}^{\infty}\xi\frac{1}{\sqrt{2\pi}}e^{-\frac{\xi^2}{2}}\frac{1}{\sqrt{2\pi(1-\hat{x}_j^2)}}e^{-\frac{\eta^2}{2(1-\hat{x}_j^2)}}\,\mathrm{d}\xi\,\mathrm{d}\eta\\ &= 2\frac{\hat{x}_j}{\sqrt{2\pi}}. \qquad \text{(A.7)}\end{aligned}$$

By using Lemma 1, we can get

$$\mathbb{P}\left\{\left|\sum_{i=1}^{m}\left(A_{ij}y_i - \mathbb{E}(A_{ij}y_i)\right)\right| \geq \epsilon\sqrt{m}\right\} \leq e\cdot\exp(-c_1\epsilon^2/8),$$

where c_1 is an absolute constant. Then

$$\mathbb{P}\left\{\left|\sum_{i=1}^{m}A_{ij}y_i\right| \geq m\cdot\mathbb{E}(A_{ij}y_i) - \epsilon\sqrt{m}\right\} \geq 1 - e\cdot\exp(-c_1\epsilon^2/8).$$

□

https://doi.org/10.4208/nmtma.2016.m1428 Published online by Cambridge University Press

Remark A.1. Suppose that $\hat{x}_j = \pm\frac{1}{\sqrt{s}}$ holds for all $j \in \mathrm{supp}(\hat{x})$. The proof of Theorem 2.2 implies that, under the condition of Theorem 2.2, the s indices corresponding to the largest magnitude entries in the vector $|A^\top y|$ are the support of $\hat{x}$ with high probability.

Appendix B. Two-point step size gradient method

The Step 7 of STrMP algorithms is to use the two-point step size gradient method [1]. We show it in Algorithm B.1. The algorithm is usually to solve unconstrained optimization problems in the form of

$$\min_{x\in\mathbb{R}^n} f(x), \tag{B.1}$$

where $f(x)$ is differentiable. For more details see [1].

Algorithm B.1 Two-Point Step Size Gradient Method

Inputs: $0 \le \varepsilon \ll 1$
Initialization: $x_1 \in \mathbb{R}^n$, $k := 1$
while $\|\nabla f(x_k)\|_2 > \varepsilon$ **do**
$d_k = -\nabla f(x_k)$
if $k = 1$ **then**
using exact line search or other methods to find first step α_1.
else
$\alpha_k = s_{k-1}^\top y_{k-1}/\|y_{k-1}\|_2^2$, where

$$s_{k-1} = x_k - x_{k-1},\ y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1}).$$

end if
$x_{k+1} := x_k + \alpha_k d_k$, $k := k + 1$
end while
Return: x_k

Acknowledgments Research of Zhiqiang Xu was supported by NSFC grant (11171336, 11422113, 11021101, 11331012) and by National Basic Research Program of China (973 Program 2015CB856000).

References

[1] J. Barzilai, and J. M. Borwein, *Two-Point Step Size Gradient Methods*, IMA Journal of Numerical Analysis, vol. 8, no. 1 (1988), pp. 141–148.
[2] T. Blumensath, and M. E. Davies, *Iterative Hard Thresholding for Compressed Sensing*, Applied and Computational Harmonic Analysis, vol. 27, no. 3 (2009), pp. 265–274.
[3] P. T. Boufounos, *Greedy Sparse Signal Reconstruction from Sign Measurements*, Asilomar Conference on Signals, Systems and Computers, 2009, pp. 1305–1309.

https://doi.org/10.4208/nmtma.2016.m1428 Published online by Cambridge University Press

[4] P. T. Boufounos, and R. G. Baraniuk, *1-Bit Compressive Sensing*, Proceedings of the 42nd Annual Conference on Information Sciences and Systems (CISS), 2008, pp. 16–21.
[5] E. J. Candes, and J. K. Romberg, and T. Tao, *Stable Signal Recovery from Incomplete and Inaccurate Measurements*, Communications on Pure and Applied Mathematics, vol. 59, no. 8 (2006), pp. 1207–1223.
[6] E. J. Candes, and T. Tao, *Decoding by Linear Programming*, IEEE Transactions on Information Theory, vol. 51, no. 12 (2005), pp. 4203–4215.
[7] D. L. Donoho, *Compressed Sensing*, IEEE Transactions on Information Theory, vol. 52, no. 4 (2006), pp. 1289–1306.
[8] L. Jacques, J. N. Laska, and P. T. Boufounos, and Richard G. Baraniuk, *Robust 1-Bit Compressive Sensing Via Binary Stable Embeddings of Sparse Vectors*, IEEE Transactions on Information Theory, vol. 59, no. 4 (2013), pp. 2082–2102.
[9] J. N. Laska, Z. Wen, W.-T. Yin, and R. G. Baraniuk, *Trust, But Verify: Fast and Accurate Signal Recovery from 1-Bit Compressive Measurements*, IEEE Transactions on Signal Processing, vol. 59, no. 11 (2011), pp. 5289–5301.
[10] Y. Plan, and R. Vershynin, *One-Bit Compressed Sensing by Linear Programming*, Communications on Pure and Applied Mathematics, vol. 66, no. 8 (2013), pp. 1275–1297.
[11] R. Vershynin, *Introduction to The Non-Asymptotic Analysis of Random Matrices*, Compressed Sensing: Theory and Applications, in Yonina C. Eldar, and Gitta Kutyniok editors, Cambridge University Press, 2012, pp. 210–268.
[12] M. Yan, Y. Yang, and S.J. Osher, *Robust 1-Bit Compressive Sensing Using Adaptive Outlier Pursuit*, IEEE Transactions on Signal Processing, vol. 60, no. 7 (2012), pp. 3868–3875.
[13] T. Zhang, *Sparse Recovery with Orthogonal Mathching Pursuit under RIP*, IEEE Transactions on Information Theory, vol. 57, no. 9 (2011), pp. 6215–6221.
[14] J. A. Tropp, *Greed Is Good: Algorithmic Results for Sparse Approximation*, IEEE Transactions on Information Theory, vol. 50, no. 10 (2004), pp. 2231–2242.
[15] D. Needell, and J. A. Tropp, *CoSaMP: Iterative Signal Recovery from Incomplete and Inaccurate Samples*, Applied and Computational Harmonic Analysis, vol. 26, no. 3 (2009), pp. 301–321.
[16] D. Needell, and R. Vershynin, *Uniform Uncertainty Principle and Signal Recovery via Regularized Orthogonal Matching Pursuit*, Foundations of Computational Mathematics, vol. 9, no. 3 (2009), pp. 317–334.
[17] W. Dai, and O. Milenkovic, *Subspace Pursuit for Compressive Sensing Signal Reconstruction*, IEEE Transactions on Information Theory, vol. 55, no. 5 (2009), pp. 2230–2249.
[18] Y. Wang, and Z.-Q. Xu, *The performance of PCM quantization under tight frames representations*, SIAM J. MATH. ANAL., vol. 44, no. 4 (2012), pp. 2802–2823.
[19] Z.-Q. Xu, *The Performance of Orthogonal Multi-Matching Pursuit under the Restricted Isometry Property*, J. Comp. Math., 33 (2015), pp. 495–516.

https://doi.org/10.4208/nmtma.2016.m1428 Published online by Cambridge University Press