Adaptive Outlier Pursuit in 1-Bit Compressive Sensing and Matrix Completion

Yi Yang
Joint work with Ming Yan and Stanley Osher

Department of Mathematics, UCLA

Outline

Introduction

- In many real world applications such as signal processing and image processing, there are all kinds of errors in the measurements during data acquisition and transmission. Some errors will damage the data seriously and make the obtained data containing no information about the true signal. Therefore, using this damaged data for signal reconstruction is useless and may worsen the performance of reconstruction methods. Methods robust to the outliers are strongly needed.
- We introduce a new variable Λ to detect the "correct" data location. $\Lambda = 0$ if the data is corrupted and 1 otherwise. Λ will be updated in each iteration according to the current result. This outlier detection technique is defined as adaptive outlier pursuit (AOP).

Outline

Compressive Sensing

- A technique for finding sparse solutions to underdetermined linear systems.

$$\min_{x \in \mathbf{R}^N} |x|_1, \; s.t. \; \Phi x = b \tag{1}$$

Here x is the original long but sparse signal. $\Phi \in \mathbf{R}^{M \times N}$ is the transformation operator with $M < N$, and the "short" signal b can save a lot of transmission time.

1-Bit Compressive Sensing[1]

1-bit compressive sensing was introduced and studied in 2008 by Boufounos and Baraniuk. The framework is as follows: Measurements of a signal $x \in \mathbf{R}^N$ are computed via

$$y = A(x) := \mathrm{sign}(\Phi x), \tag{2}$$

where the measurement operator $A(\cdot)$ is a mapping from $\mathbf{R}^N$ to the Boolean cube $\mathcal{B}^M := \{-1, 1\}^M$, and the purpose is to recover signal $x \in \sum_K^* := \{x \in S^{N-1} : \|x\|_0 \leq K\}$ where S^{N-1} is the unit hyper-sphere of dimension N.

[1] P. Boufounos and R. Baraniuk, 1-bit compressive sensing, 2008

Why 1-Bit?

- 1-bit measurements are inexpensive and fast.
- robust to amplification of the signal and other errors, as long as they preserve the signs of the measurements.
- focus on bits rather than measurements.

Some Results for 1-Bit CS Framework[2]

- A lower bound is provided on the best achievable performance of this 1-bit CS framework, and if the elements of Φ are drawn randomly from i.i.d. Gaussian distribution or its rows are drawn uniformly from the unit sphere, then the solution will have bounded error on the order of the optimal lower bound.
- A condition on the mapping A, binary ϵ-stable embedding (BϵSE), that enables stable reconstruction is given to characterize the reconstruction performance even when some of the measurement signs have changed (e.g., due to noise in the measurements).

[2] L. Jacques et.al., Robust 1-Bit Compressive Sensing via Binary Stable Embeddings of Sparse Vectors, 2011

Previous Methods: BIHT[3]

Binary iterative hard thresholding (BIHT or BIHT-ℓ_2) is the algorithm for solving

$$\begin{aligned} \min_{x} \quad & \sum_{i=1}^{M} \phi(y_i, (\Phi x)_i) \\ \text{s.t.} \quad & \|x\|_2 = 1, \quad \|x\|_0 \leq K, \end{aligned} \tag{3}$$

where ϕ is the one-sided ℓ_1 (or ℓ_2) objective:

$$\phi(x, y) = \begin{cases} 0, & \text{if } x \cdot y > 0, \\ |x \cdot y| \text{ (or } |x \cdot y|^2/2), & \text{otherwise.} \end{cases} \tag{4}$$

[3] L. Jacques et.al., Robust 1-Bit Compressive Sensing via Binary Stable Embeddings of Sparse Vectors, 2011

BIHT

Algorithm 1 BIHT

Input: Φ, $y \in \{-1, 1\}^M$, $K > 0$, $\alpha > 0$, $\mathrm{Miter} > 0$.
Initialization: $x^0 = \Phi^T y / \|\Phi^T y\|$, $k = 0$, tol $=$ inf.
while $k \leq$ Miter and $0 <$ tol **do**
  $\beta^{k+1} = x^k + \alpha \Phi^T (y - \mathrm{sign}(\Phi x^k))$.
  $x^{k+1} = \eta_K(\beta^{k+1})$.
  tol $= \|y - A(x^{k+1})\|_0$.
  $k = k + 1$.
end while
return $x^k / \|x^k\|$.

Adaptive Outlier Pursuit[4]

Introducing Λ into the old problem solved by BIHT, we have the following new problem with unknown variables x and Λ:

$$\begin{aligned} \min_{x,\Lambda} \quad & \sum_{i=1}^{M} \Lambda_i \phi(y_i, (\Phi x)_i) \\ \text{s.t.} \quad & \sum_{i=1}^{M} (1-\Lambda_i) \le L, \quad \Lambda_i \in \{0,1\} \quad i = 1, 2, \cdots, M, \\ & \|x\|_2 = 1, \quad \|x\|_0 \le K. \end{aligned} \tag{5}$$

[4] M. Yan, Y. Yang and S. Osher, Robust 1-bit compressive sensing using binary matching pursuit, 2012

Two Steps

- Fix Λ and solve for x:

$$\begin{aligned} \min_{x} \quad & \sum_{i=1}^{M} \Lambda_i \phi(y_i, (\Phi x)_i) \\ \text{s.t.} \quad & \|x\|_2 = 1, \quad \|x\|_0 \leq K. \end{aligned} \tag{6}$$

- Fix x and update Λ:

$$\begin{aligned} \min_{\Lambda} \quad & \sum_{i=1}^{M} \Lambda_i \phi(y_i, (\Phi x)_i) \\ \text{s.t.} \quad & \sum_{i=1}^{M} (1 - \Lambda_i) \leq L, \quad \Lambda_i \in \{0, 1\} \quad i = 1, 2, \cdots, M. \end{aligned}$$

Given an x estimated from (6), we can update Λ in one step:

$$\Lambda_i = \begin{cases} 0, & \text{if } \phi(y_i, (\Phi x)_i) \geq \tau, \\ 1, & \text{otherwise.} \end{cases} \tag{7}$$

where τ is the L^{th} largest term of $\{\phi(y_i, (\Phi x)_i)\}_{i=1}^{M}$.

Algorithm 2 AOP

Input: Φ, $y \in \{-1,1\}^M$, $K > 0$, $L \geq 0$, $\alpha > 0$, $\mathrm{Miter} > 0$

Initialization: $x^0 = \Phi^T y / \|\Phi^T y\|$, $k = 0$, $\Lambda = \mathbf{1} \in \mathbf{R}^M$, Loc $=$ $1 : M$, tol $=$ inf, TOL $=$ inf.

while $k \leq$ Miter and $L \leq$ tol **do**

  $\beta^{k+1} = x^k + \alpha \Phi(\mathrm{Loc},:)^T (y(\mathrm{Loc}) - \mathrm{sign}(\Phi(\mathrm{Loc},:)x^k))$.

  $x^{k+1} = \eta_K(\beta^{k+1})$.

  tol $= \|y - A(x^{k+1})\|_0$.

  if tol $\leq$TOL **then**,

    Compute Λ with (7).

    Update Loc to be the location of 1-entries of Λ.

    TOL $=$ tol.

  end if

  $k = k + 1$.

end while

return $x^k / \|x^k\|$.

Algorithm 3 AOP with flips

1. **Input:** Φ, $y \in \{-1,1\}^M$, $K > 0$, $L \geq 0$, $\alpha > 0$, $\mathrm{Miter} > 0$
2. **Initialization:** $x^0 = \Phi^T y / \|\Phi^T y\|$, $k = 0$, $\Lambda = \mathbf{1} \in \mathbf{R}^M$, $y_r = y$, tol = inf, TOL = inf.

while $k \leq$ Miter and $L \leq$ tol **do**

  $\beta^{k+1} = x^k + \alpha \Phi^T (y_r - A(x^k))$.

  $x^{k+1} = \eta_K(\beta^{k+1})$,

  tol $= \|y - A(x^{k+1})\|_0$.

  if tol $\leq$TOL **then**,

    Compute Λ with (7).

    Set $(y_r)_i = y_i$ if $\Lambda_i = 1$, $(y_r)_i = -y_i$ if $\Lambda_i = 0$.

    TOL = tol.

  end if

  $k = k + 1$.

end while

return $x^k / \|x^k\|$.

Numerical Experiments

Here AOP is implemented in the following four ways:

- AOP with one-sided ℓ_1 objective (AOP);
- AOP with one-sided ℓ_1 objective and sign flips (AOP-f);
- AOP with one-sided ℓ_2 objective (AOP-ℓ_2);
- AOP with one-sided ℓ_2 objective and sign flips(AOP-ℓ_2-f).

The four algorithms, together with BIHT and BIHT-ℓ_2, are studied and compared.

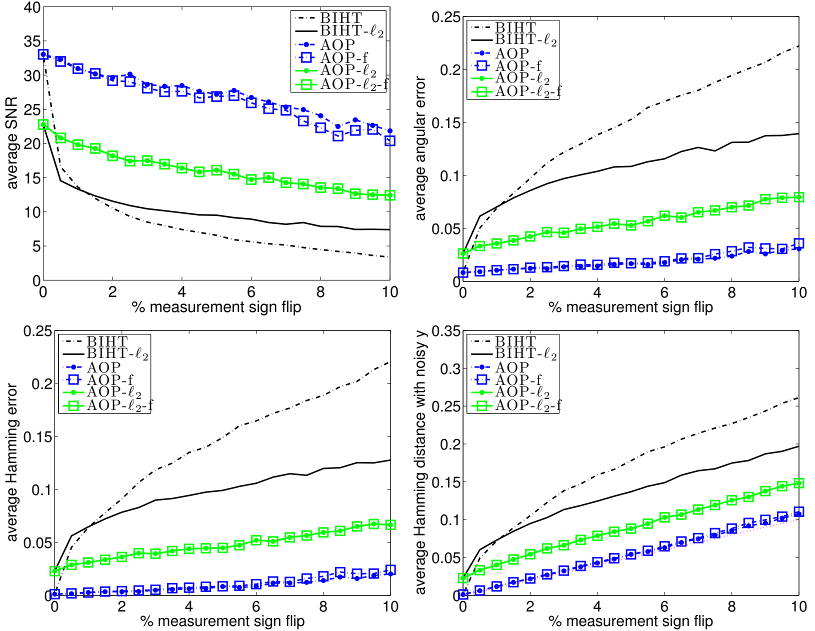
- signal-to-noise ratio (SNR): $10\log_{10}(\|x\|^2/\|x-x^*\|^2)$;
- angular error: $\arccos\langle x, x^*\rangle/\pi$;
- Hamming error: $\|A(x)-A(x^*)\|_0/M$;
- Hamming distance between $A(x)$ and y, defined as $\|A(x)-y\|_0/M$;

Four cases

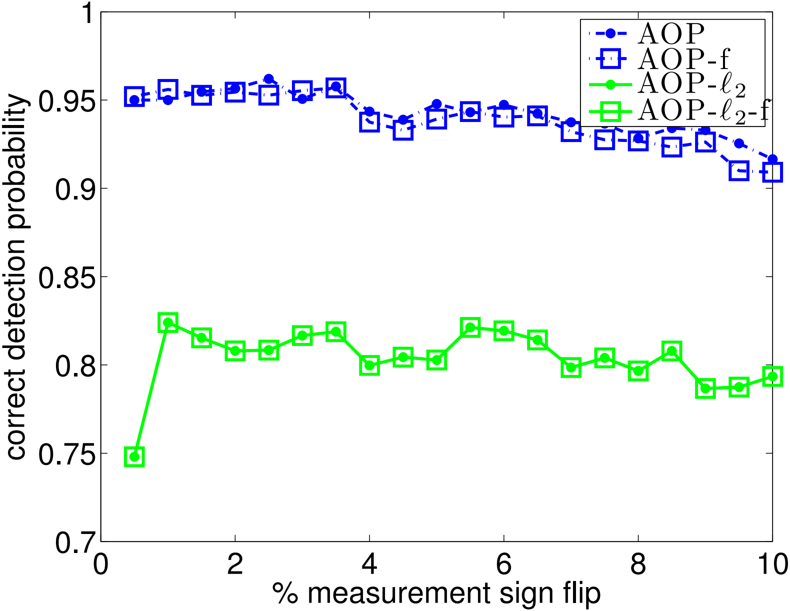
Here we consider the performance of these algorithms in the following four different cases.

- Different noise levels.
- Different M/N ratios.
- High noise levels.
- L unknown.

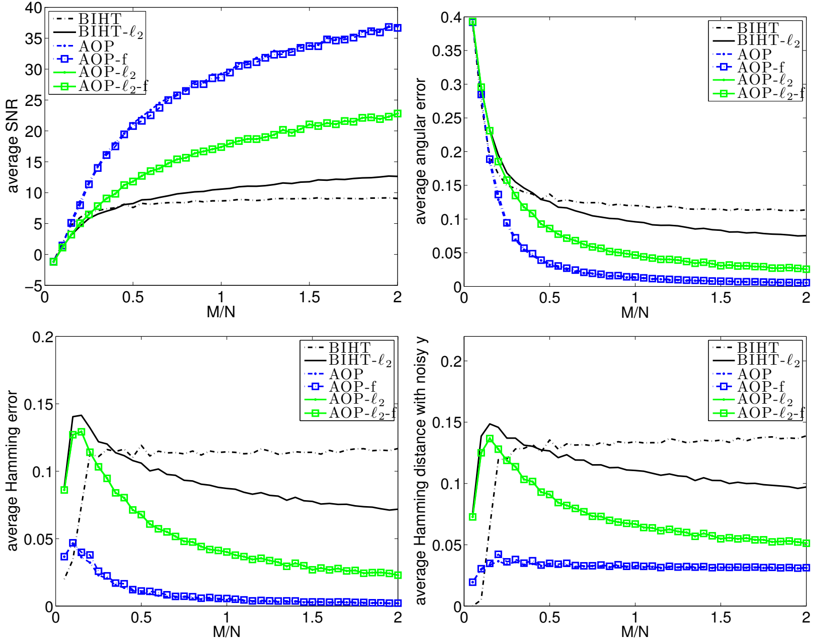
Different noise levels



Accuracy of "clean" data detection



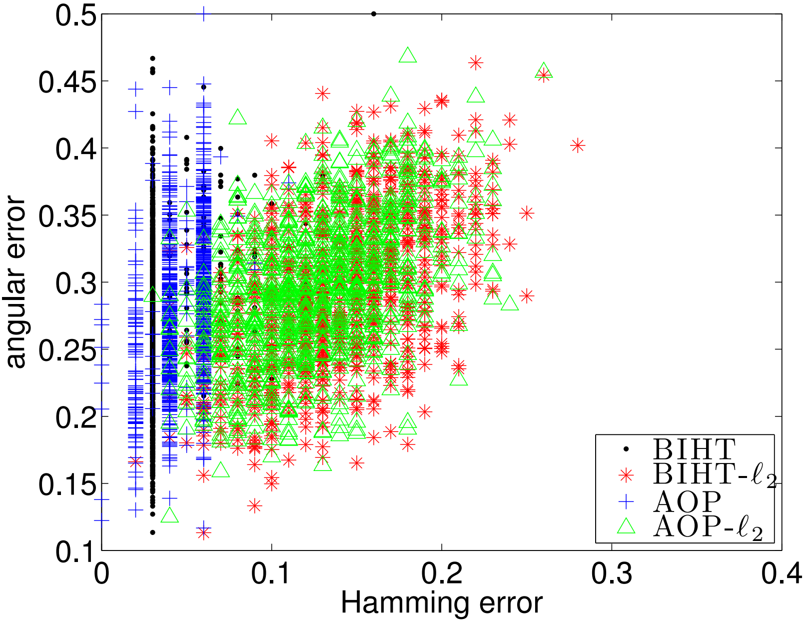
Different sample rate M/N



Hamming error vs angular error

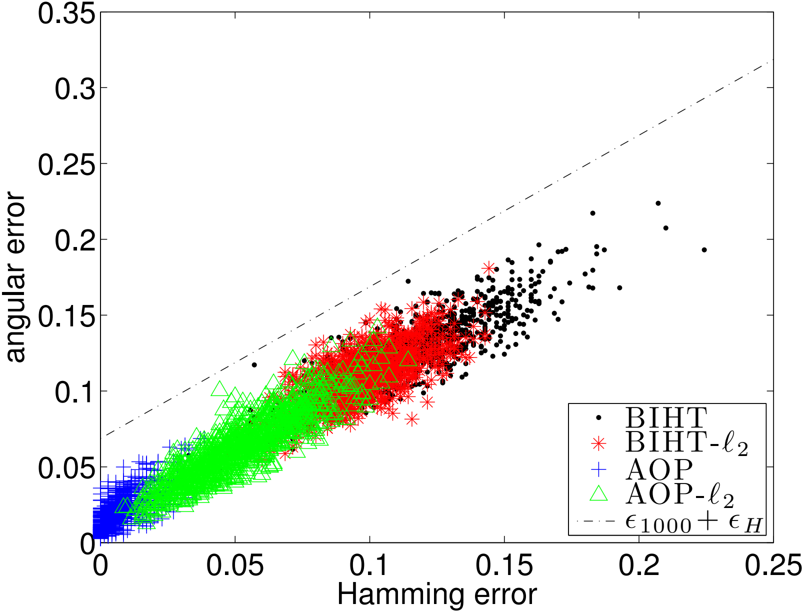
- We plot the Hamming error between $A(x)$ and $A(x^*)$ vs angular error for four algorithms with different M/N. We can see that almost all the blue $(+)$ points stay in the lower left part of the graph for $M = 0.7N$ and $M = 1.5N$, which proves that AOP gives more consistent results compared with other three algorithms.

$M/N = 0.1$

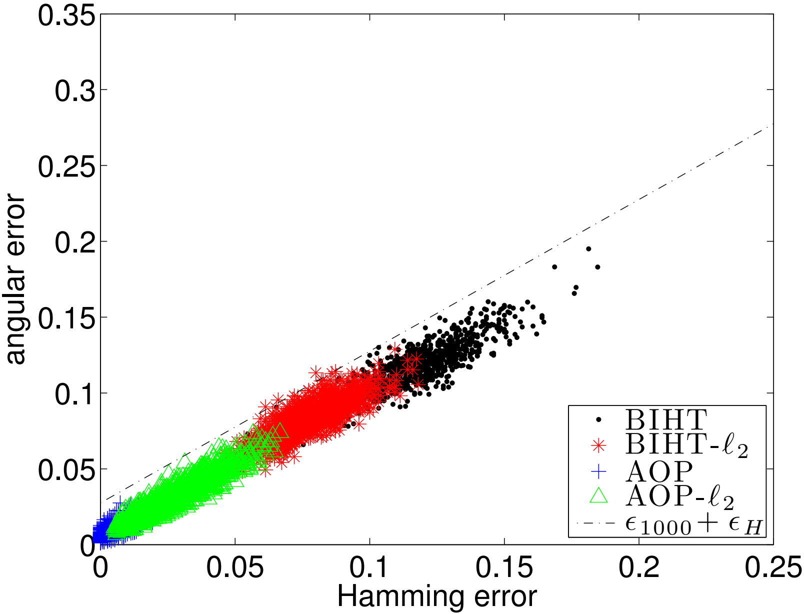


angular error
Hamming error
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0
0.1
0.2
0.3
0.4
BIHT
BIHT-ℓ_2
AOP
AOP-ℓ_2

$M/N = 0.7$

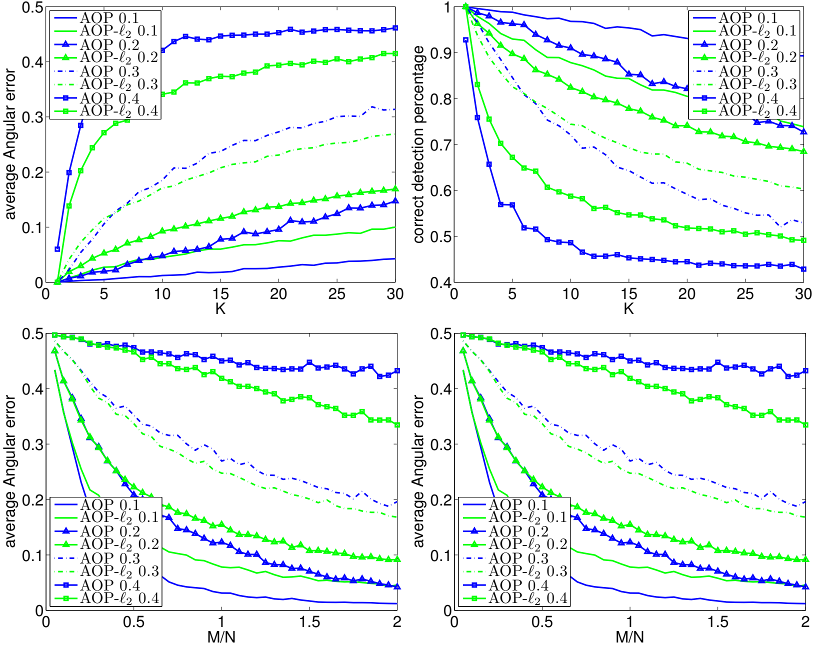

0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
angular error
0
0.05
0.1
0.15
0.2
0.25
Hamming error
BIHT
BIHT-ℓ_2
AOP
AOP-ℓ_2
$\epsilon_{1000} + \epsilon_H$

$M/N = 1.5$



0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
angular error
0
0.05
0.1
0.15
0.2
0.25
Hamming error
BIHT
BIHT-ℓ_2
AOP
AOP-ℓ_2
$\epsilon_{1000} + \epsilon_H$

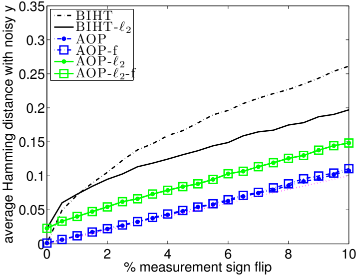
High noise levels



L is not given

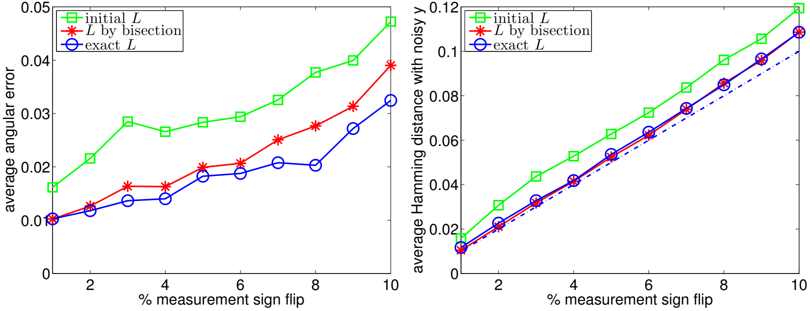
When no preknowledge about L is given, what shall we do? Follow these steps!

- "guess" an initial L.



- "train" this L value: bisection method.

Cont.


initial L
L by bisection
exact L
average angular error
% measurement sign flip
0.05
0.04
0.03
0.02
0.01
0
2
4
6
8
10
initial L
L by bisection
exact L
average Hamming distance with noisy y
% measurement sign flip
0.12
0.1
0.08
0.06
0.04
0.02
0
2
4
6
8
10

Outline

Introduction

- Matrix completion (MC) is a popular problem in many areas of science and engineering, where we need to reconstruct a low-rank matrix from incomplete samples of its entries.
-

$$\min_{X \in \mathbf{R}^{m \times n}} \ \operatorname{rank}(X), \ \text{s.t. } X_{i,j} = M_{i,j}, \ \forall (i,j) \in \Omega \tag{8}$$

$$\Longrightarrow \min_{X \in \mathbf{R}^{m \times n}} \ \|X\|_*, \ \text{s.t. } X_{i,j} = M_{i,j}, \ \forall (i,j) \in \Omega. \tag{9}$$

- Most of the MC algorithms can not perform stably when the given data is corrupted by outliers.
- We propose a robust method for recovering the low-rank matrix with adaptive outlier pursuit (AOP) when part of the measurements are damaged by outliers.

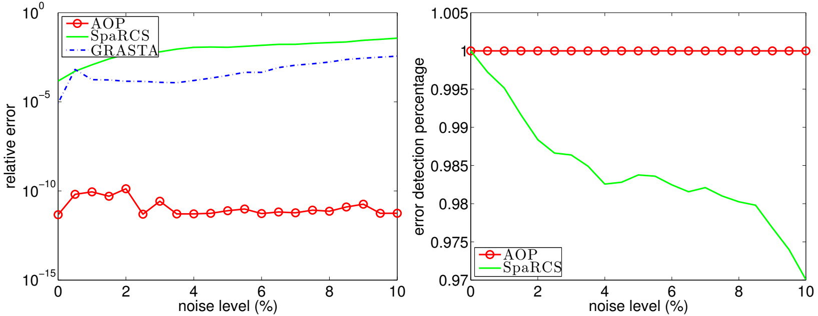
Numerical experiments

Here we fix the size of the matrix, and compare the performance of our algorithm (AOP) with two other algorithms (SpaRCS and GRASTA) in the following cases.

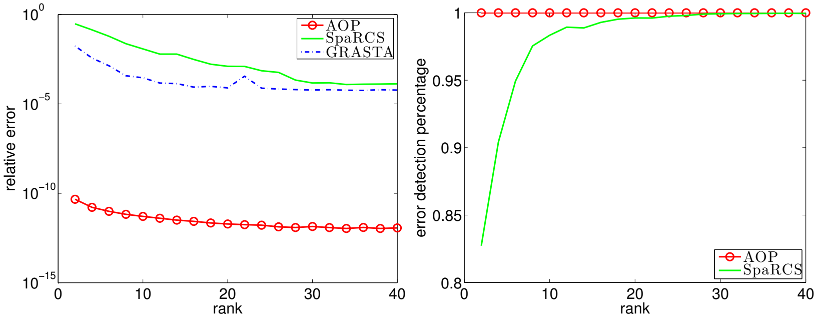
- Different noise levels.
- Different ranks.

Different noise levels

The noise level is defined as the ratio of the number of outliers to the number of given entries.



Different ranks



Outline

Conclusion and Future Work

- General form of robust 1-Bit Compressive Sensing
- Impulse Noise Removal