SPUR Final Paper, Summer 2012

Least Absolute Deviations Method For Sparse Signal Recovery

February 2, 2013

Author:
Jelena Markovic

Mentor:
Ruixun Zhang

Project suggested by:
Prof. Lie Wang

We consider a problem in signal processing which deals with the recovery of a high-dimensional sparse signal based on a small number of measurements. Our goal is to apply the least absolute deviations (LAD) method in an algorithm that would essentially follow the steps of the orthogonal matching pursuit (OMP) algorithm that has been used mostly in this setting. OMP can recover the signal with high probability in noiseless cases and specific noise distributions such as bounded and Gaussian. In the presence of heavy-tailed distributed noise, OMP algorithm needs a signal to be much larger than the noise in order to recover it (puts too much constraint on a signal). We consider the algorithm using LAD in all the cases above and compare the simulation results using both methods. OMP works better, i.e., recovers a higher percentage of signals in noiseless, bounded and Gaussian noise cases. On the other hand, our new LAD based method recovers a higher percentage of signals in the case when the $t(2)$-heavy tailed noise is present. This provides an alternative to the standard least-squares based methods especially in the presence of heavy tailed noises. We also provide a sufficient condition on the design matrix in order for the LAD based method to recover all signals precisely. Simulation shows the sufficient condition is satisfied with high probability for Bernoulli and Gaussian random design matrices.

1 Introduction

This work is related to the problem in the Compressed Sensing topic which is finding sparse solutions to vastly underdetermined linear systems. An underdetermined system of linear equations has more unknowns than equations and generally has an infinite number of solutions. However, if there is a unique sparse solution to the underdetermined system, then the Compressed Sensing framework allows the recovery of that solution. There are also significant connections of this project with the problem of recovering signals from highly incomplete measurements. In electrical engineering, particularly in signal processing, this problem is of great importance. The results found by David Donoho, Emmanuel Cands and Terence Tao [7] are very important for establishing the field. They showed that the number of the measurements can be small and still contain nearly all the useful information.

First we will provide background on the regression model and the least squares estimation method as in [1] and [2]. Then we will say how the least squares regression is used to develop the orthogonal matching pursuit algorithm. Finally, we will introduce the least absolute deviations method and the corresponding algorithm.

2 Regression model. Least squares estimation method.

The regression model assumes that output y linearly depends on the input variables $x_1, x_2, \dots, x_p$, $p \in \mathbb{N}$, i.e.,

$$y = \beta_1 x_1 + \cdots + \beta_p x_p.$$

This model is to be fit of n, $n \in \mathbb{N}$ points

$$y_i, x_{i1}, x_{i2}, \dots, x_{ip}, \quad i = 1, 2, \dots, n.$$

The observations y_i, where $i = 1, 2, \dots, n$ will be represented by a vector Y. The unknowns $\beta_1, \beta_2, \dots, \beta_p$ will be represented by a vector β. Let X be a matrix

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

For a given β, the vector of fitted or predicted values, $\widehat{Y}$, can be written $\widehat{Y} = X\beta$. Using the least squares estimation we will pick the coefficients $\beta = (\beta_1, \beta_2, \dots, \beta_p)^T$ to minimize the residual sum of squares

$$\begin{aligned} RSS(\beta) &= \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \\ &= \|Y - X\beta\|_2^2 \\ &= \left\| Y - \widehat{Y} \right\|_2 \\ &= (Y - X\beta)^T (Y - X\beta). \end{aligned}$$

Differentiating with respect to β we get

$$\frac{\partial RSS}{\partial \beta} = -2X^T(Y - X\beta).$$

Thus, RSS is minimal for $X^T X\beta = X^T Y$. Assuming that X has full column rank we get that $X^T X$ is nonsingular and thus the solution is unique

$$\hat{\beta} = (X^T X)^{-1} X^T Y.$$

In this case

$$\widehat{Y} = X\hat{\beta} = X(X^T X)^{-1} X^T Y.$$

Denote $P = X(X^T X)^{-1} X^T$ and columns of X as $x_1, \ldots, x_p$ (n dimensional vectors). These vectors span a subspace of $\mathbb{R}^n$. We will minimize $RSS(\beta) = \|Y - X\beta\|_2^2$ by choosing $\hat{\beta}$ so that the residual vector $Y - \widehat{Y}$ is orthogonal to this subspace. This can also be seen from the equation that minimizes RSS:

$$X^T(Y - X\beta) = 0.$$

Thus we get

$$x_i^T(Y - \widehat{Y}) = 0, \quad i = 1, 2, \ldots, p,$$

i.e.,

$$\left\langle x_i, (Y - \widehat{Y}) \right\rangle = 0, \quad i = 1, 2, \ldots, p.$$

We conclude that $Y - \widehat{Y}$ is orthogonal to all $x_1, x_2, \ldots, x_p$, i.e., space spanned by columns of the matrix X. Hence, $\widehat{Y}$ is the orthogonal projection of Y onto this subspace. The matrix P computes orthogonal projection, and hence it is also called a projection matrix.

It might happen that the columns of X are not linearly independent, so that X is not of full rank. This happens, for example, when the number of inputs p is bigger than the number of measurements n and this case is of particular interest for this project. In the next section we will present one way of reconstructing the signal β in this case if we add more restrictions to the model as shown in [3].

3 Orthogonal matching pursuit algorithm.

Assume the presence of noise in the model so that

$$Y = X\beta + \epsilon,$$

where the observation $Y \in \mathbb{R}^n$, the matrix $X \in \mathbb{R}^{n \times p}$ and the errors $\epsilon \in \mathbb{R}^n$. Let the columns of X be normalized i.e., $\|x_i\|_2 = 1$ for all $i = 1, 2, \ldots, p$. The goal is to reconstruct $\beta \in \mathbb{R}^n$ if X and Y are given, $p \gg n$, and we know that in this model the vector β is sparse. Sparsity is defined in the following way:

Definition 1 *For a given vector $\beta \in \mathbb{R}^p$, the support of β is defined to be the set $supp(\beta) = \{i : \beta_i \neq 0\}$ and β is said to be k-sparse if $|supp(\beta)| \leq k$.*

Now, we will consider the orthogonal matching pursuit algorithm for the recovery of this kind of β, high-dimensional sparse signal, based on a small number of measurements. We still have the assumption that the columns of X are normalized. For any subset $S \subset \{1, 2, \ldots, p\}$, denote by $X(S)$ a submatrix of X consisting of the columns x_i with $i \in S$. As defined in the paper [3] we will call x_i a correct variable if the corresponding $\beta_i \neq 0$ and call x_i an incorrect variable otherwise. The OMP algorithm is the following:

- Step 1: Initialize the residual $r_0 = Y$ and initialize the set of selected variables $X(c_0) = \emptyset$. Set the iteration counter $i = 1$.
- Step 2: Find the variable (column) x_{t_i} which solves the maximization problem

$$\max_{1 \leq t \leq p} |x_t^T r_{i-1}|,$$

and add the variable x_{t_i} to the set of selected variables: $c_i = c_{i-1} \cup \{t_i\}$.

- Step 3: Set $P_i = X(c_i)(X(c_i)^T X(c_i))^{-1} X(c_i)^T$ the projection onto the linear space spanned by the columns of $X(c_i)$. Now change (update) $r_i = (I - P_i)Y$.
- Step 4: If the stopping rule is achieved, stop the algorithm. Otherwise, set $i = i + 1$ and return to step 2.

This algorithm selects in the second step the column of X which is most correlated with the current residual. The column chosen in this way is not already selected because the current residual is orthogonal to all already selected columns so the scalar product of the current residual and an already selected column would be zero, hence certainly not the maximum. This column is then added to the set of selected columns. The algorithm updates the residual by projecting the observation Y onto the linear space spanned by the columns that have already been selected (Step 3) and then the algorithm iterates (Step 4). The stopping rule in Step 4 depends on the noise structure and in the noiseless case the stopping rule is naturally $r_i = 0$. The paper [3] considers three types of noise. The first is l_2 bounded noise, i.e., $\|\epsilon\| \leq b_2$ for some constant $b_2 > 0$. Another is l_∞ bounded noise where $\|X^T \epsilon\| \leq b_\infty$ for some constant $b_\infty > 0$. The third one is Gaussian noise where ϵ_i i.id. $N(0, \sigma^2)$ is considered.

One of the important restrictions that the matrix X should have in order for the sparse signal to be recovered is the *Mutual Incoherence Property* (MIP). The Mutual Incoherence is defined as:

$$\mu = \max_{1 \leq i < j \leq p} |\langle x_i, x_j \rangle|,$$

and it measures the maximum correlation between two columns. The MIP requires the mutual incoherence μ to be small i.e., the columns of the matrix X are slightly correlated (we cannot take them to be orthogonal under the condition $p \gg n$ but we can take them to be "almost orthogonal" by requiring μ to be small). It is shown in [4] that in the noiseless case the condition $\mu < \frac{1}{2k-1}$ is sufficient for recovering a k-sparse signal β i.e., under MIP the OMP algorithm presented can completely recover the vector β. In this case the algorithm will select all correct variables x_i (corresponding $\beta_i \neq 0$) and none of the incorrect ones.

Another condition used in [4] is called the Exact Recovery Condition (ERC). Let $T = \{i : \beta_i \neq 0\}$ be the support of β and let $X(T)$ be the set of columns of X corresponding to the support T. Define

$$M = \max_{x \in X \setminus X(T)} \{\|(X(T)^T X(T))^{-1} X(T)^T x\|_1\}.$$

The condition $M < 1$ is called the Exact Recovery Condition. It is shown in [4] that the ERC is a sufficient condition for the exact recovery of the support of the signal β in the noiseless case. The value M is not computable as it depends on the unknown support T.

Generating random matrices with MIP puts some limits on n, k and p. From [5] we see that in order for the MIP to hold for a random matrix where all x_{ij} are i.id., roughly the sparsity k should satisfy

$$k < \frac{1}{4}\sqrt{\frac{n}{\log p}}.$$

For example, if we take $k = 2$, we get $\log p < \frac{n}{(4k)^2} = \frac{n}{64}$, which implies $n < p < e^{\frac{n}{64}}$. The smallest n that satisfies this property is $n = 381$ in which case p has to satisfy $381 < p < 385$. We need $p \gg n$ thus n needs to be bigger than this lower bound. For $n = 1,000$ we get $1,000 < p < 6,107,328$ which makes p much bigger than n. Because the MIP condition puts too much constraint on the dimensions of the data matrix X, we see from the following theorem from [6] that OMP can recover a k-sparse signal when the number of measurements n is nearly proportional to k but with some probability.

Theorem 2 *(OMP with Gaussian Measurements): Fix $\delta \in (0, 0.36)$, and choose $n \geq Ck \ln(\frac{p}{\delta})$. Suppose that β is an arbitrary k-sparse signal in $\mathbb{R}^p$. Draw n measurement vectors $X_1, X_2, \ldots, X_n$ independently from the standard Gaussian distribution on $\mathbb{R}^p$. Given the data $\{\langle \beta, X_i \rangle : i = 1, 2, \ldots, n\}$, OMP can reconstruct the signal with probability exceeding $1 - 2\delta$. The constant satisfies $C \leq 20$. For large values of k, it can be reduced to $C \approx 4$.*

In [6] it is also shown from the simulation results that this theoretical bound is qualitatively correct even though it is slightly pessimistic.

4 Least absolute deviations method.

For this method we will use the same model as used to describe the least squares estimation. Here, we will try to minimize RSA, residual sum of absolute values, i.e.,

$$RSA = \sum_{i=1}^{n} |y_i - \sum_{j=1}^{p} x_{ij}\beta_j| = \|Y - X\beta\|_1.$$

Differentiating RSA with respect to β we get

$$\frac{\partial RSA}{\partial \beta} = -X^T \mathrm{sgn}(Y - X\beta),$$

for all real vectors β for which the function is differentiable. The sign function for a random variable is defined as

$$\mathrm{sgn}(x) = \begin{cases} -1 & : x < 0 \\ 0 & : x = 0 \\ 1 & : x > 0. \end{cases}$$

Also, we are using $\text{sgn}(r) = (\text{sgn}(r_1), \text{sgn}(r_2), \dots, \text{sgn}(r_n))$ for $r = (r_1, r_2, \dots, r_n)$ a real vector. In order to minimize RSA we need to find β so that these partial derivatives are as close as possible to zero. Thus, we will change the OMP algorithm in the following way:

- Step 1: Initialize the residual $r_0 = Y$ and initialize the set of selected variables $X(c_0) = \emptyset$. Set the iteration counter $i = 1$.
- Step 2: Find the variable (column) x_{t_i} not already in c_{i-1} which solves the maximization problem

$$\max_{1 \le t \le p \text{ and } t \notin c_{i-1}} \left| x_t^T \text{sgn}(r_{i-1}) \right|,$$

and add the variable x_{t_i} to the set of selected variables: $c_i = c_{i-1} \cup \{t_i\}$.
- Step 3: Find the new residual using LAD method: calculate the new residual $r_i = Y - X(c_i)\beta_{c_i}$ by minimizing the corresponding $RSA = \|Y - X(c_i)\beta_{c_i}\|_1$.
- Step 4: If $i \ge k$ stop the algorithm. Otherwise, set $i = i + 1$ and return to Step 2.

In Step 2, we find a column which is most correlated with the sign of the current residual, but not already selected. Also, in Step 3, we do not use projections (this is specific for the least squares estimation), but we use the package *quantile regression* in R, which computes coefficients and residuals in this case.

The stopping times for the cases with noise in OMP algorithm used in [3] do not use the sparsity level as given. They recover the signal without having k as input. The stopping times used there depend on the norm of the residual and noise type, but they are constructed only for the specific noise distributions mentioned in Section 3. The case of $t(2)$ distributed noise is not considered in [3] because OMP does not work very well with this noise type (heavy-tailed), i.e., it could recover the signal only if its the non-zero components are very large, so that the noise would be much smaller compared with the signal. This restriction on the signal is too big, so that is mainly why we consider alternative methods to deal with this type of noise. We take that the sparsity of β is given as input (k is known), so we have exactly k iterations in the algorithm.

Now, we want to find the conditions on the matrix X for which LAD would recover the signal. From the model we have

$$Y = x_1\beta_1 + x_2\beta_2 + \cdots + x_k\beta_k.$$

In this case the set of true variables is $X_T = (x_1, \dots, x_k)$ and the set of incorrect variables is denoted as $X_F = (x_{k+1}, \dots, x_p)$. In order for the algorithm to select the correct variable in the first step we need to have

$$\|X_T^T \text{sgn}(Y)\|_\infty > \|X_F^T \text{sgn}(Y)\|_\infty. \tag{1}$$

If we put $\beta^* = (\beta_1, \dots, \beta_k)$ then we can write $Y = X_T\beta^*$, and the above inequality becomes

$$\|X_T^T \text{sgn}(X_T\beta^*)\|_\infty > \|X_F^T \text{sgn}(X_T\beta^*)\|_\infty.$$

By using Hölder's inequality we get:

$$\|\beta^*\|_1 \|X_T^T \text{sgn}(X_T\beta^*)\|_\infty \ge \langle \beta^*, X_T^T \text{sgn}(X_T\beta^*) \rangle.$$

From $\langle \beta^*, X_T^T \mathrm{sgn}(X_T\beta^*)\rangle = (\beta^*)^T X_T^T \mathrm{sgn}(X_T\beta^*) = \|X_T\beta^*\|_1$ and using the inequality above we get

$$\|X_T^T \mathrm{sgn}(X_T\beta^*)\|_\infty \geq \frac{\|X_T\beta^*\|_1}{\|\beta^*\|_1},$$

so the sufficient condition for the recovery of a correct variable in the first step becomes

$$\frac{\|X_T\beta^*\|_1}{\|\beta^*\|_1} > \|X_F^T \mathrm{sgn}(X_T\beta^*)\|_\infty. \tag{2}$$

Because the condition in (2) should hold for all $\beta^* \in \mathbb{R}^k$ we get that (1) holds for Y equals any linear combination of the true variables $x_1, \dots, x_k$. Thus, we can write

$$\|X_T^T \mathrm{sgn}(r)\|_\infty > \|X_F^T \mathrm{sgn}(r)\|_\infty,$$

where r is any of the residuals (a residual is some linear combination of the true variables). From here, we get that in all iterations the column which is most correlated with the sign of the current residual is a true variable, i.e., one of the first k columns. We need to make sure it is not an already selected true variable. This can be accomplished if we change the definition of the sign function so that sgn(0) can be any number between -1 and 1. The derivative of $\|Y - X(c_i)\beta_{c_i}\|_1$ is equal to $-X(c_i)^T \mathrm{sgn}(Y - X(c_i)\beta_{c_i})$ in all points $\beta_{c_i} \in \mathbb{R}^i$ in which the function is differentiable. The new definition of the sign function allows us to have the value of $-X(c_i)^T \mathrm{sgn}(Y - X(c_i)\widehat{\beta_{c_i}})$ at the point $\widehat{\beta_{c_i}} = \min_{\beta_{c_i}} \arg(\|Y - X(c_i)\beta_{c_i}\|_1)$ to be exactly 0. This means that the sign vector (using the new definition of the sign function) of each new residual $Y - X(c_i)\beta_{c_i}$ is orthogonal to all already selected columns. Thus we get that none of the column can be selected twice. It means that the condition in (2) is sufficient for the exact recovery of all true variables in LAD algorithm in the noiseless case.

Let us see what the inequality in (2) becomes in the following cases of $k = 1$ and $k = 2$.

• For $k = 1$ we have that $Y = x_1\beta_1$, and in order to select the corect variable in the first (and only) step we need to have

$$\frac{\|x_1\beta_1\|_1}{|\beta_1|} = \|x_1\|_1 > |\langle \mathrm{sgn}(Y), x_t\rangle| = |\langle \mathrm{sgn}(x_1), x_t\rangle|, \quad \text{for all } 2 \leq t \leq p.$$

If we assume that all columns of the matrix X have the same l_1 norm and that all entries of the matrix X are non-zero, we get exact recovery if x_t, for all $t \geq 2$, have the sign vector different than the sign vector of x_1 and different than the sign vector of $-x_1$.

• If sparsity equals 2, we have $Y = x_1\beta_1 + x_2\beta_2$, where $\beta_1 \neq 0$ and $\beta_2 \neq 0$. In order for LAD algorithm in the first step to select a correct variable (x_1 or x_2) the sufficient condition is

$$\frac{\|x_1\beta_1 + x_2\beta_2\|_1}{|\beta_1| + |\beta_2|} > |\langle x_t, \mathrm{sgn}(x_1\beta_1 + x_2\beta_2)\rangle|, \quad \text{for all } t \geq 3.$$

Now, we need to put a condition on a matrix X, so that the inequality in (2) would be true for all $\beta^* \in \mathbb{R}^k$. For OMP algorithm we have seen that it is enough to have the Mutual Incoherence Propery on the data matrix, but for LAD it is harder to simplify the above inequality. Thus, the idea is to bound LHS and RHS of (2) with a lower and upper bound respectively (finding the infimum of the LHS and the supremum of the RHS over all real vectors β^* of length k is what we will consider), and then see in what cases we can compare these two bounds.

Let us now define the following quantities.

Definition 3 *For a $n \times k$ matrix A, we say that its infimum-norm equals*

$$\eta(A) = \underset{v \in \mathbb{R}^k \setminus \{0\}}{inf} \frac{\|Av\|_1}{\|v\|_1}.$$

Note that the quantity defined above is not a norm on $\mathbb{R}^{n \times k}$ (vector space of all matrices with n rows and k columns).

Definition 4 *For a $n \times l$ matrix B, we will call the restricted matrix norm the following quantity*

$$\xi(B)_A = \underset{v \in \mathbb{R}^k \setminus \{0\} s.t. Av \neq 0}{sup} \frac{\|B^T sgn(Av)\|_\infty}{\|sgn(Av)\|_\infty},$$

depending on a $n \times k$ matrix A.

The matrix operator $\xi(B)_A$ is defined similarly to the $\|\cdot\|_{\infty,\infty}$ norm of a matrix B^T, but the supremum in $\xi(B)_A$ is not taken over the whole space $\mathbb{R}^n$. In Definition 4 the supremum is taken over the sign space of the subspace of $\mathbb{R}^n$. That subspace depends on a matrix A; it is spanned by the columns of A. Let us denote that subspace as S_A, and also define

$$\text{sgn}(S_A) = \{s : s \in \mathbb{R}^n \text{ and } (\exists v \in S_A)(s = \text{sgn}(v))\},$$

for a $n \times k$ matrix A. Then, Definition 4 becomes

$$\xi(B)_A = \sup_{s \in \text{sgn}(S_A) \setminus \{0\}} \frac{\|B^T s\|_\infty}{\|s\|_\infty}.$$

Using these two definitions we get the following bounds

$$\frac{\|X_T \beta^*\|_1}{\|\beta^*\|_1} > \eta(X_T), \quad \text{for all } \beta^* \in \mathbb{R}^k,$$

and

$$\|X_F^T \text{sgn}(X_T \beta^*)\|_\infty < \xi(X_F)_{X_T}, \text{ for all } \beta^* \in \mathbb{R}^k.$$

If we put the condition

$$\eta(X_T) > \xi(X_F)_{X_T} \tag{3}$$

on the data matrix X we get the recovery of a true variable in the first step. We see that this condition is similar to the Exact Recovery Condition (mentioned in Section 3, introduced in [4]) in the sense that it is not computable because it depends on the set of true variables. In the following section we will see how likely a random data matrix X is to satisfy this inequality.

5 LAD Recovery Condition for Bernoulli and Gaussian data matrices.

Let us call the conditon in (3) LAD Recovery Condition. We want to see how strong the condition is, and that is why we will compute these bounds for a standard normal data matrix and a Bernoulli data matrix. The setup is the following. In each step we generate a $n \times p$ matrix X and compute $\eta(X_T)$ and $\xi(X_F)_{X_T}$ for a fixed value of k. The data matrix X is a random matrix so we can take the first k columns to be the correct variables. Thus, we get the matrix X_T to be $n \times k$ submatrix of X, i.e., X_T contains exactly first k columns of X. Then X_F is $n \times (p-k)$ submatrix of X and contains the last $(p-k)$ columns of X. Now we compute $\eta(X_T)$ as the minimum of $\frac{\|X_T v\|_1}{\|v\|_1}$ over 100 vectors $v \in \mathbb{R}^k$. Each of the coordinates v_i, $1 \leq i \leq k$, of the vector v is chosen from the uniform $[-1, 1]$ distribution. Similary, $\xi(X_F)_{X_T}$ is computed as the maximum of $\|X_F^T \text{sgn}(X_T v)\|_\infty$ over 100 vectors $v \in \mathbb{R}^k$ whose coordinates are chosen as previously. Alse we compute the average values of $\eta(X_T)$ and $\xi(X_F)_{X_T}$ for 100 data matrices X all of the same size $n \times p$. Moreover, we measure how many out of 100 data matrices satisfy the inequality $\eta(X_T) > \xi(X_F)_{X_T}$.

For $p = 256$, $k = 4$ and a Bernoulli matrix X we get that for $n \geq 150$ the inequality is satisfied for over 90 matrices. Also, for $n \geq 150$ the inequality is satisfied for over 90 matrices in the case of a standard normal matrix X. These results are presented in Table 1. Also, the results for $p = 1024$ are presented in Table 2 and Table 3 for both Gaussian and Bernoulli data matrix X.

From these results we see that LAD Recovery Condition is likely to happen for n large enough. For example for $p = 256$ we get that the condition is satisfied with a probability over 0.9 for $n \geq 150$. So, for these values of n and p we get from these experimental results that the recovery would be exact. For example, if we use theoretical bound for MIP condition, which is a computable condition for OMP, we see that we need $16k^2 < \frac{n}{\ln(256)}$ (elaborated in Section 3, from [5]). From this we get $16k^2 < \frac{n}{\ln(256)} < \frac{256}{\ln(256)}$, i.e., we conclude that we can recover only the sparsity $k = 1$ using theoretical estimations (in that case n is almost the same value as p). MIP is a simplified condition for OMP algorithm and easily computable, but we see that these theoretical bounds are too pesimistic and we usually need numerical results. For LAD method we do not have a single computable condition for the data matrix X (the condition that does not depend on the set of correct variables), but the inequality in (3) we got for LAD recovery is very likely to happen in specific data matrices, such as Bernoulli and Gaussian. This propery makes the condition very useful for applications.

n	< 60	80	100	120	140	160	180	200	220	240
Bernoulli	0	2	15	60	88	93	99	100	100	100
Gaussian	0	0	5	44	76	96	99	98	100	100

Table 1: The table showing how many out of 100 measurement matrices X satisfty the inequality $\eta(X_T) > \xi(X_F)_{X_T}$ for different values of n (X is of dimension $n \times 256$, $k = 4$). Here we present the results when X is a standard normal matrix and a Bernoulli matrix.

In Figures 1, 2 and 3 we present the mean values of both $\eta(X_T)$ and $\xi(X_F)_{X_T}$ for the following cases $(p = 256, k = 4)$, $(p = 1024, k = 5)$, and $(p = 1024, k = 10)$ for different

n	<120	140	160	180	200	220	240	260	>280
Bernoulli	0	6	42	73	96	97	100	100	100
Gaussian	0	3	22	49	81	96	97	99	100

Table 2: The table showing how many out of 100 measurement matrices X satisfty the inequality $\eta(X_T) > \xi(X_F)_{X_T}$ for different values of n (X is of dimension $n \times 1024$, $k = 5$). Here we present the results when X is a standard normal matrix and a Bernoulli matrix.

n	<240	260	280	300	320	340	360	380	400	420	440	>500
Bernoulli	0	1	10	25	44	65	87	92	94	98	99	100
Gaussian	0	1	7	18	39	66	80	86	98	95	100	100

Table 3: The table showing how many out of 100 measurement matrices X satisfty the inequality $\eta(X_T) > \xi(X_F)_{X_T}$ for different values of n (X is of dimension $n \times 1024$, $k = 10$). Here we present the results when X is a standard normal matrix and a Bernoulli matrix.

values of n, ranging from 10 to p. In Figures 4, 5 and 6 the mean values of the same quantities are shown for a standard Gaussian data matrix. From the figures, we can see how likely the sufficient condition we get is to be satisfied. For a fixed dimension p, as the number of observations n increases, $\eta(X_T)$ increases faster than $\xi(X_F)_{X_T}$. Therefore, for Bernoulli and Gaussian design matrices, the sufficient condition we get is satisfied with high probability.

6 Comparing two algorithms.

This section compares the two algorithms based on the simulation results. We considered both the noiseless case and the case when the noise is present. The important conclusion is that the algorithm that uses LAD works better than OMP algorithm when the heavy-tailed noise is present in the model. In other cases, both methods behave similarly, although OMP recovers the signal with higher probability.

Let us describe the experimental setup. The same setup is used in [6] for the simulation results of OMP in the noiseless case. In each trial we use the k-sparse signal β with first k components (out of p) to be equal to one. Also, we generate an $n \times p$ measurement matrix X as a standard Gaussian random matrix. Then, we execute both OMP and LAD algorithms with $Y = X\beta$ in the noiseless case. Finally, we check whether the set of selected columns is equal to the set of correct columns (variables). If they are identical then the algorithm has succeeded with probability one. For each triple (k, n, p) we perform 100 independent trials.

The first plot, Figure 7, describes the results of the simulations for $p = 256$. It shows what fraction (of 100 trials) was recovered correctly as a function of n for both methods and using different sparsity levels $k = 4, 12, 20, 28, 36$. As expected, when the number of non-zero components increases, more measurements are necessary for signal recovery. From this graph we see that OMP recovers a higher fraction of signals than LAD for all five sparsity levels we tried. Also, we see that they "behave" similarly in the sense that both of the methods recover the signals completely if the sparsity level is small enough ($k = 4, 12$) and n large enough. Figure 8 shows the simulation results of both methods also in the

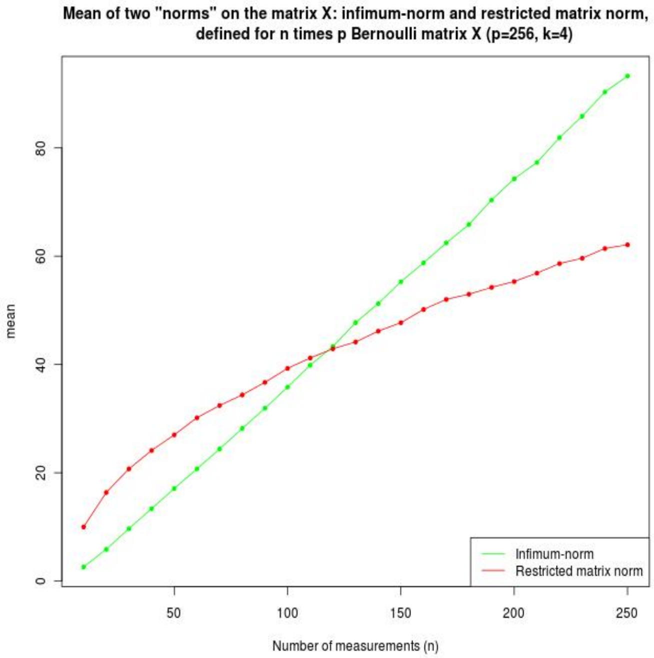

Figure 1: The average values of the quantities $\eta(X_T)$ (infimum-norm) and $\xi(X_F)_{X_T}$ (restricted matrix norm), computed for 100 Bernoulli $n \times p$ matrices X, as a function of n. In this case $p = 256$ and $k = 4$.

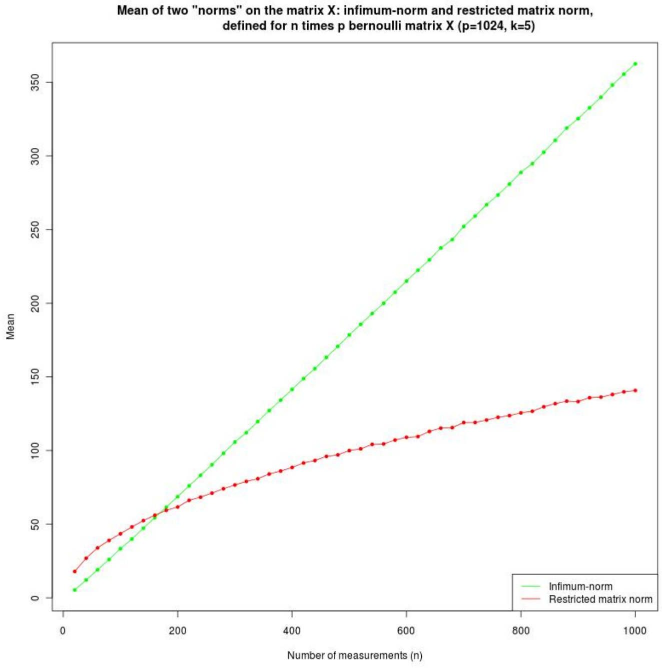

Figure 2: The average values of the quantities $\eta(X_T)$ and $\xi(X_F)_{X_T}$, computed for 100 Bernoulli $n \times p$ matrices X, as a function of n. In this case $p = 1024$ and $k = 5$.

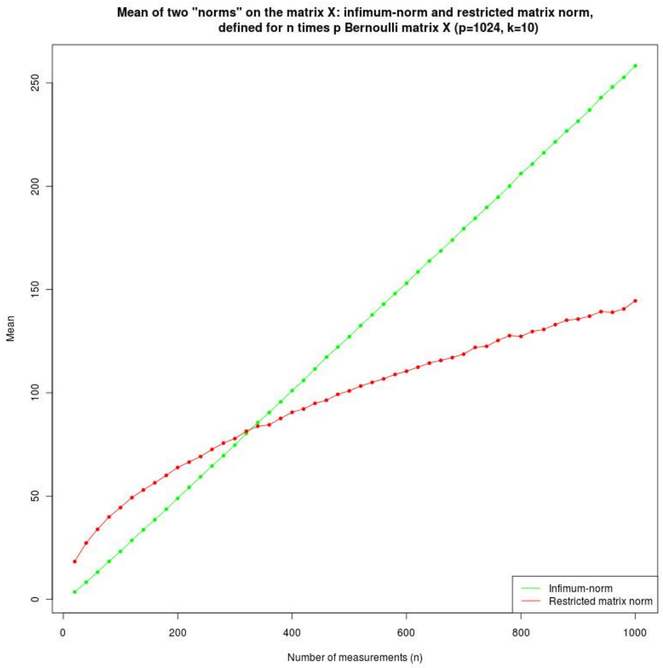

Figure 3: The average values of the quantities $\eta(X_T)$ and $\xi(X_F)_{X_T}$, computed for 100 Bernoulli $n \times p$ matrices X, as a function of n. In this case $p = 1024$ and $k = 10$.

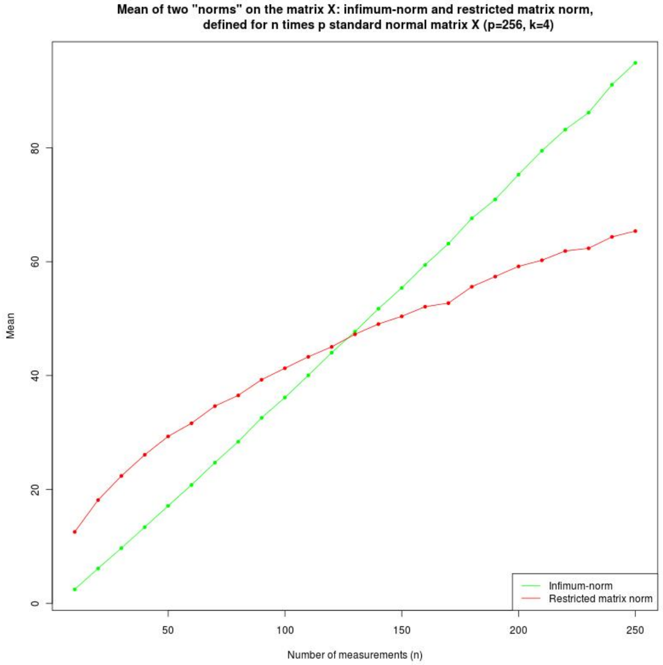

Figure 4: The average values of the quantities $\eta(X_T)$ and $\xi(X_F)_{X_T}$, computed for 100 standard Gaussian $n \times p$ matrices X, as a function of n. In this case $p = 256$ and $k = 4$.

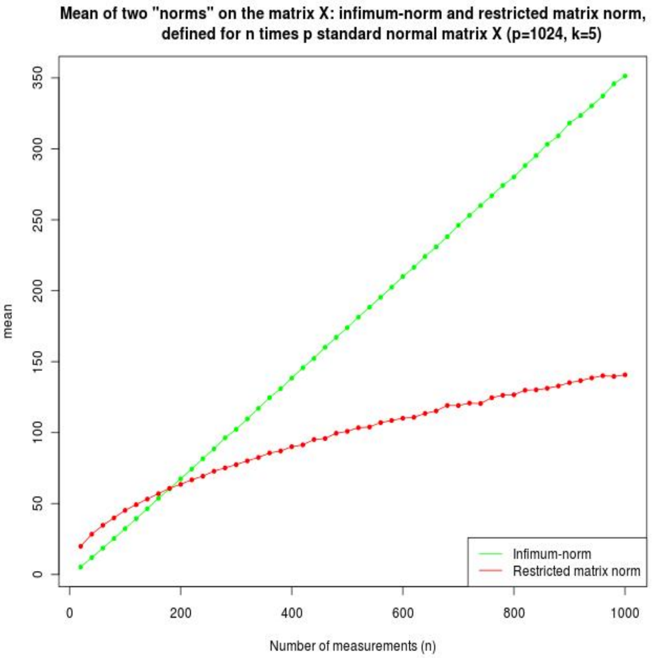

Figure 5: The average values of the quantities $\eta(X_T)$ and $\xi(X_F)_{X_T}$, computed for 100 standard Gaussian $n \times p$ matrices X, as a function of n. In this case $p = 1024$ and $k = 5$.

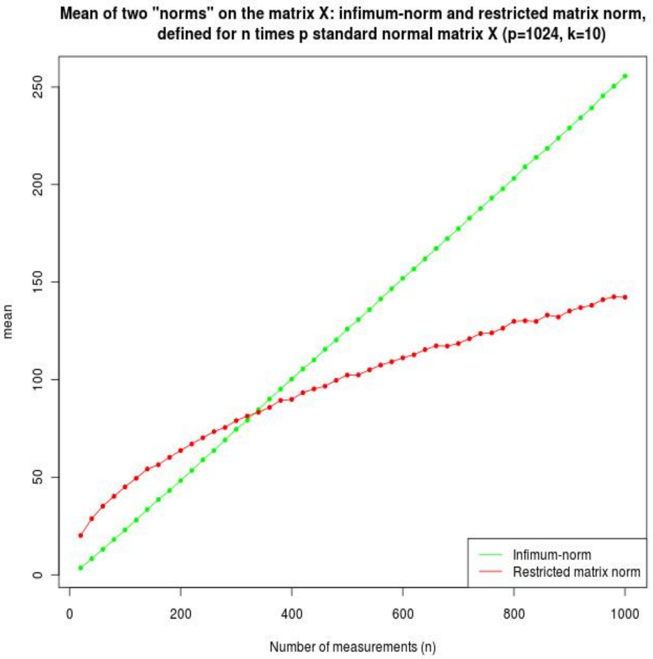

Figure 6: The average values of the quantities $\eta(X_T)$ and $\xi(X_F)_{X_T}$, computed for 100 standard Gaussian $n \times p$ matrices X, as a function of n. In this case $p = 1024$ and $k = 10$.

noiseless case but with $p = 1024$. Here we used three different sparsity levels $k = 5, 10, 15$. This graph shows the results are similar to the case $p = 256$. In the case $p = 1024$, a signal β has only k non-zero components, thus it is more sparse realtive to its length than the signal in the case where $p = 256$.

We can compare the results for LAD algorithm obtained here with the numerical results from Section 5. For $(p = 256, k = 4)$ we see from Figure 7 that for $n \geq 60$ the signal is recovered in over 90 out of 100 cases. For the same value of p and k we see from Table 1 that LAD Recovery Condition for a standard Gaussian matrix X holds in 90 out of 100 cases for $n \geq 150$. From Figure 7 we see that for $n \geq 150$ we have the exact recovery in all trials, so the theoretical bound is correct although does not match the bounds from the experimental results completely. The other results we get by comparing results from this section with the results from Section 5 are presented in Table 4. In this table we compare the lower bounds on n so that for all n greater then those bounds we have the exact recovery in all 100 trails (using LAD algorithm in the noiseless case) and we have LAD Recovery Condition holding in all 100 standard Gaussian matrices X respectively.

(p, k)	LAD algorithm	LAD Recovery Condition
$(p = 256, k = 4)$	$n \geq 120$ (from Figure 7)	$n \geq 210$ (from Table 1)
$(p = 1024, k = 5)$	$n \geq 140$ (from Figure 8)	$n \geq 280$ (from Table 2)
$(p = 1024, k = 10)$	$n \geq 280$ (from Figure 8)	$n \geq 520$ (from Table 3)

Table 4: Comparing experimental results for LAD algorithm in the noiseless case (presented in this section) with the numerical results for LAD Recovery Condition for a standard Gaussian data matrix (from the Section 5). For example, for $(p = 256, k = 4)$ we get that for $n \geq 120$, LAD algorithm would exactly recover the signal in all 100 trials (from Figure 7). Also for $(p = 256, k = 4)$ we get that for $n \geq 210$ LAD Recovery Condition holds in all 100 standard Gaussian matrices X (from Table 1). In the same way we read the results for $(p = 1024, k = 5)$ and $(p = 1024, k = 10)$.

In the next plot, Figure 9, we have the noise introduced into the model i.e., $Y = X\beta + \varepsilon$. We will use X and β generated as above (in the noiseless case) with the noise from the standard Gaussian distribution. In the simulations with noise we will use the algorithm as in [6] which supposes that the sparsity number k is given as input (elaborated in Section 4). Thus, we perform exactly k steps in both algorithms and then compare whether the selected columns are all correct ones. Here we see that in this case OMP also works better than LAD i.e., recovers the higher percentage of input signals. Also, we have the simulations results for $p = 1024$ presented in Figure 10 which are similar to the results for $p = 256$.

Next, in Figure 11, we change noise to have t-distribution of 2 degrees of freedom. $t(2)$ distribution resembles the bell shape of a normally distributed variable with mean 0 and variance 1, except that it has a heavier tail. As the number of degrees of freedom increases, the t-distribution approaches the standard normal distribution. In this case we see from the simulation results that LAD recovers a higher fraction of signals than OMP when the sparsity is small enough ($k = 4$ and $k = 12$). For $k = 20$ the methods recover approximately the same fraction of signals and for even greater k we get that OMP is better. We see that for larger values of k the data matrix X is less likely to satisfy the conditions for LAD than the conditon for OMP. In Figure 12 we take $p = 1024$ and from these results we see that LAD reaches the fraction recovery equal to 1.00, while OMP never reaches 1.00 even for small sparsity levels. In the Figure 13 for $p = 256$ again, we use $t(2)$ distributed noise multiplied by 2, so that the noise is even larger. We see that in this case LAD recovers higer fraction of signals compared to OMP in all cases we tried.

Least absolute deviations regression is more robust than the least squares regression. That was the main intuitian for expecting LAD to work better than OMP in the cases where larger errors are more likely to occur. From the simulation results we see that is exactly what happens: for $t(2)$ noise we see that the algorithm that uses LAD regression works better than the one that uses least squares regression.

7 Conclusion.

The theoretical and numerical work in this project demonstrates that LAD is an effective way for signal recovery, especially in the presence of heavy-tailed noise. Our result offers an alternative to the standard OMP algorithm in the case of large noise. We proposed a sufficient condition on design matrix X under which the LAD-based algorithm is guaranteed to select the correct variable in noiseless setting. This condition is verified in simulation with Bernoulli and Gaussian design matrices. Possiblly the following statement could be proved rigorously: For a Bernoulli and a Gaussian measurement matrix X LAD Recovery Condition holds with high probability.

Higher recovery of signals can be achieved using backward steps in the greedy algorithm. In [8] the Adaptive Forward-Backward Algorithm is used by minimizing L_2 norm of the errors. One of the suggestions for future work would be to use this greedy algorithm with L_1 norm with the kinds of noises used in this paper.

Acknowledgment

The author wishes to thank Ruixun Zhang for colaboration on this project, Prof. Lie Wang for proposing this project and his insight on the theoretical part. Special thanks to Prof. Pavel Etingof for organizing the Summer Program in Undergraduate Research SPUR at MIT.

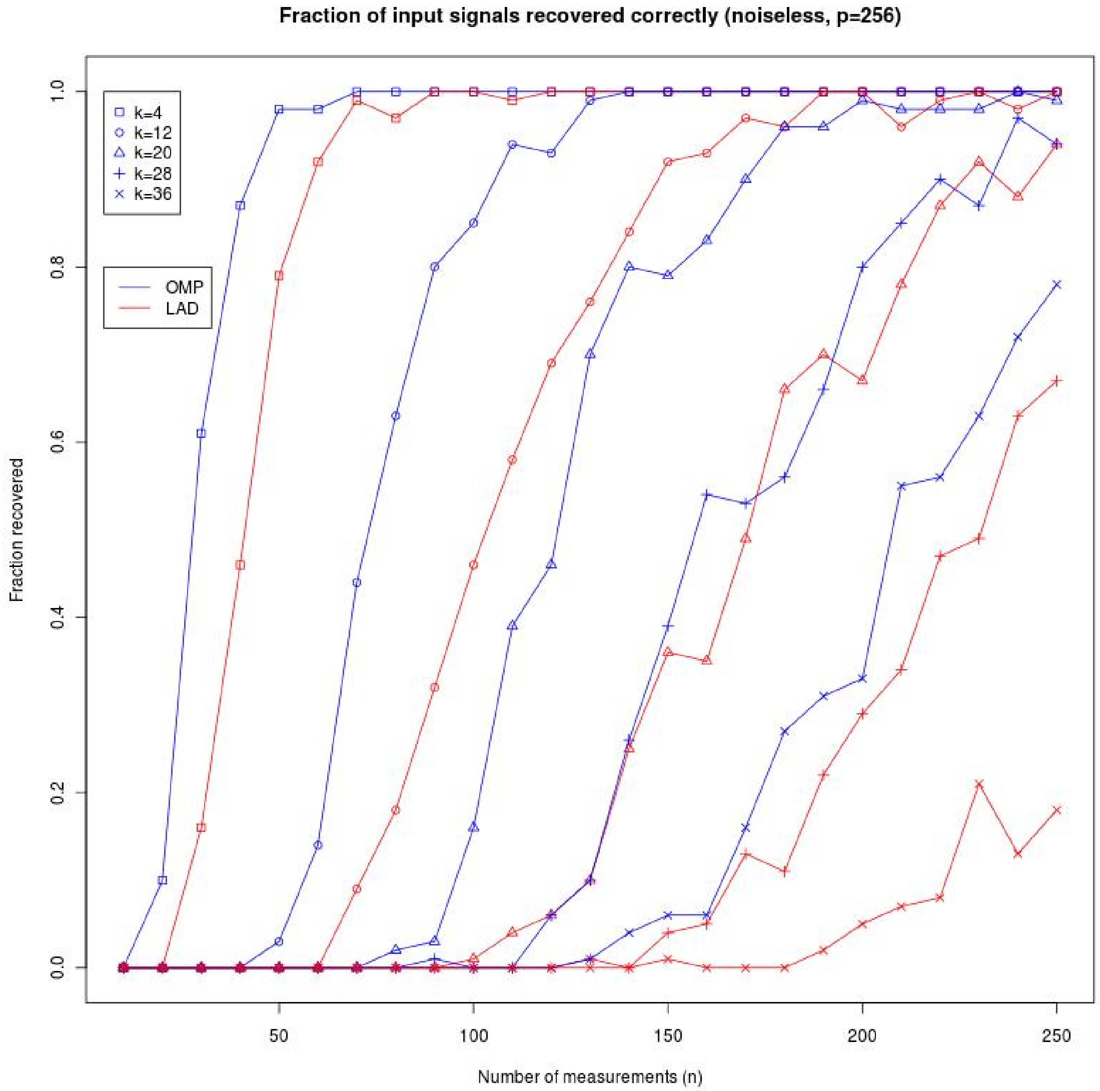

Figure 7: The fraction of 100 input signals recovered as a function of a number of measurements n for different sparsity levels k in dimension $p = 256$ in the noiseless case. Here we take five different sparsity levels $k = 4, 12, 20, 28, 36$ for both algorithms OMP and LAD. n is increased by 10.

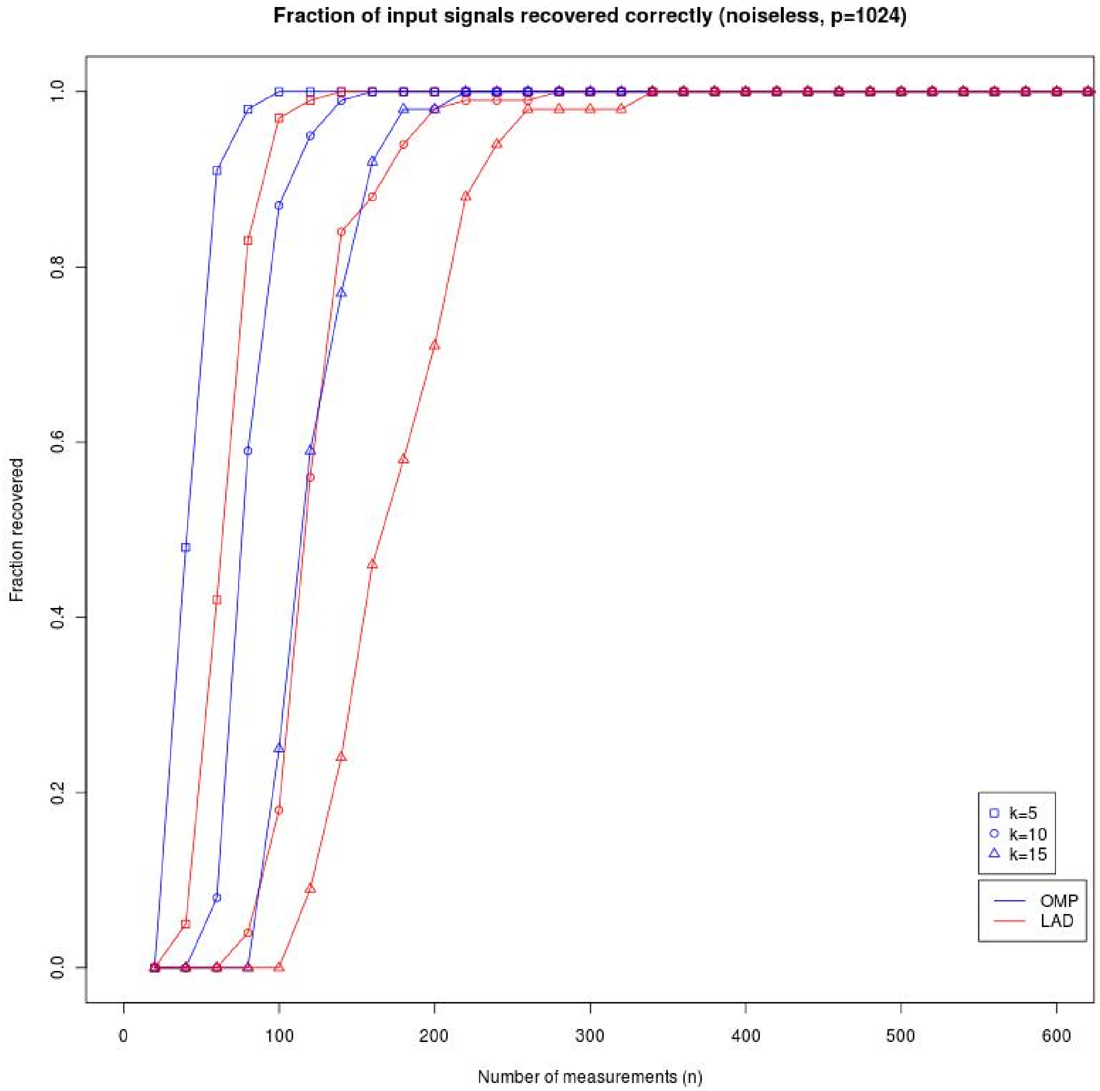

Figure 8: The fraction of 100 input signals recovered as a function of a number of measurements n for three different sparsity levels k (5, 10, 15) in dimension $p = 1024$ in the noiseless case. For $n > 600$ both methods for all three sparsity levels recovered all of 100 input signal, i.e., the fraction recovered in these cases is 1.00. n is increaed by 20.

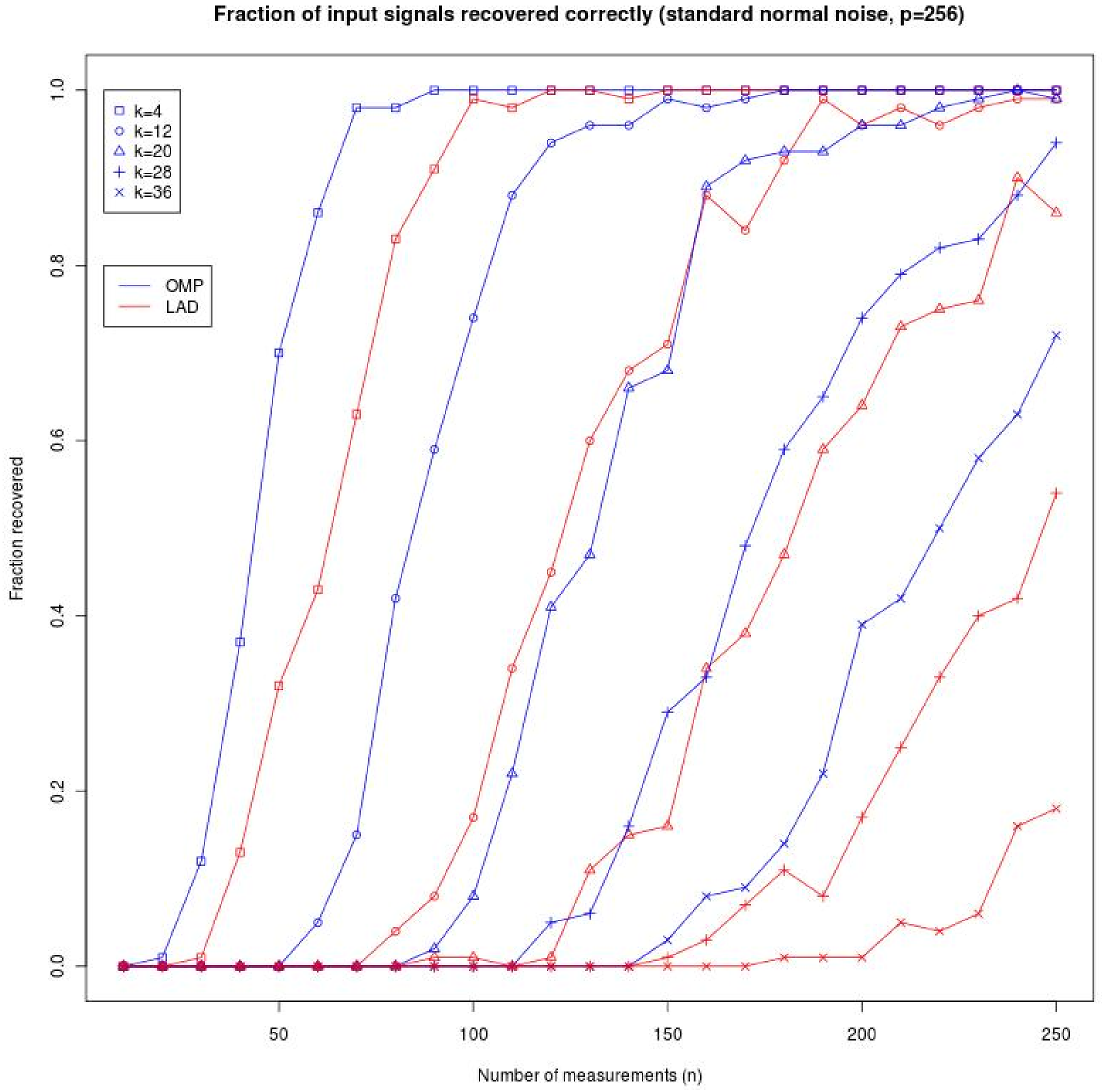

Figure 9: The fraction of 100 input signals recovered as a function of a number of measurements n for different sparsity levels k in dimension $p = 256$ in the presence of standard Gaussian noise.

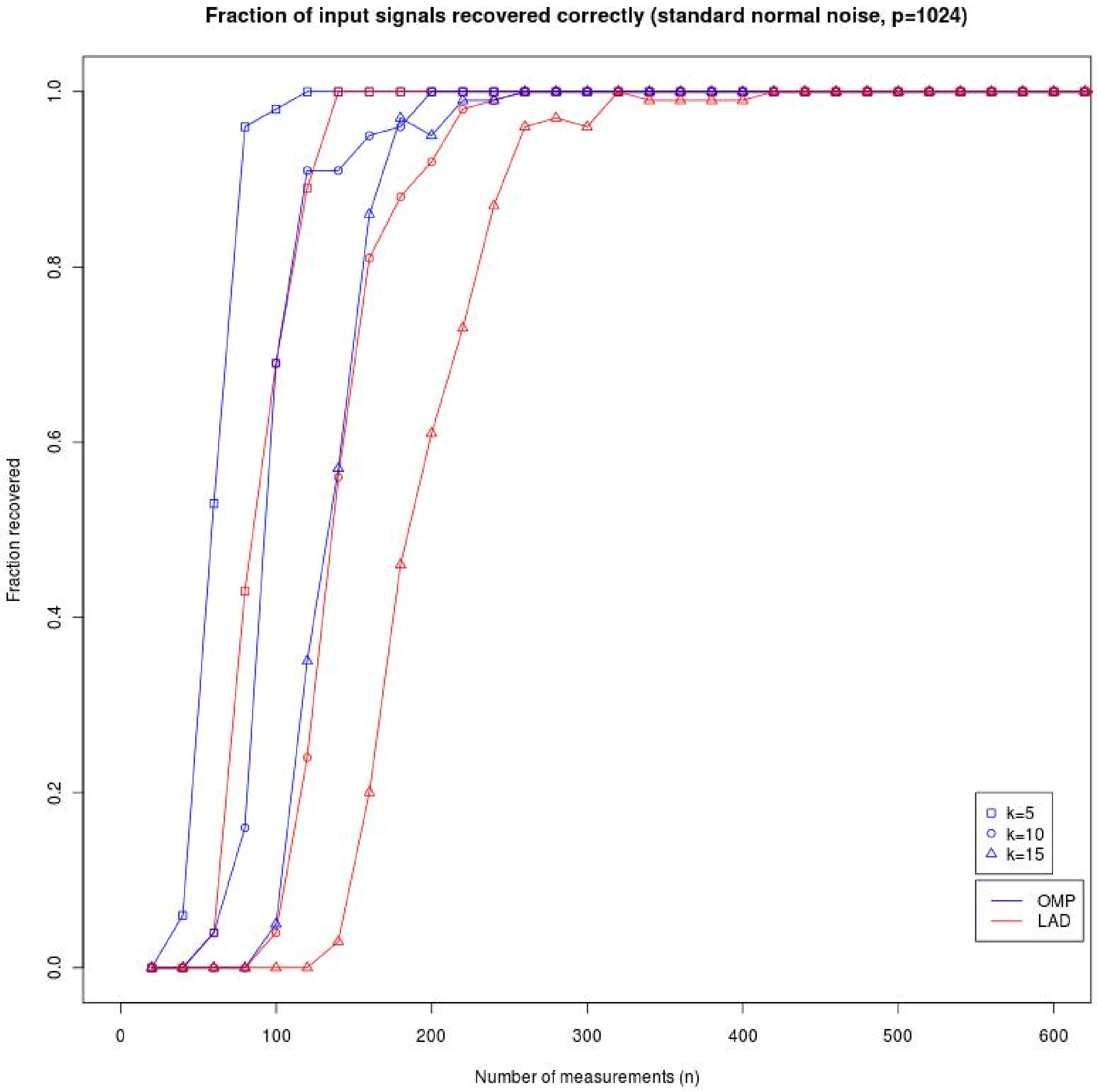

Figure 10: The fraction of 100 input signals recovered as a function of a number of measurements n for different sparsity levels k in dimension $p = 256$ in the presence of a standard Gaussian noise. For $n > 600$ both methods for all three sparsity levels recovered all of 100 input signal, i.e., the fraction recovered in these cases is 1.00.

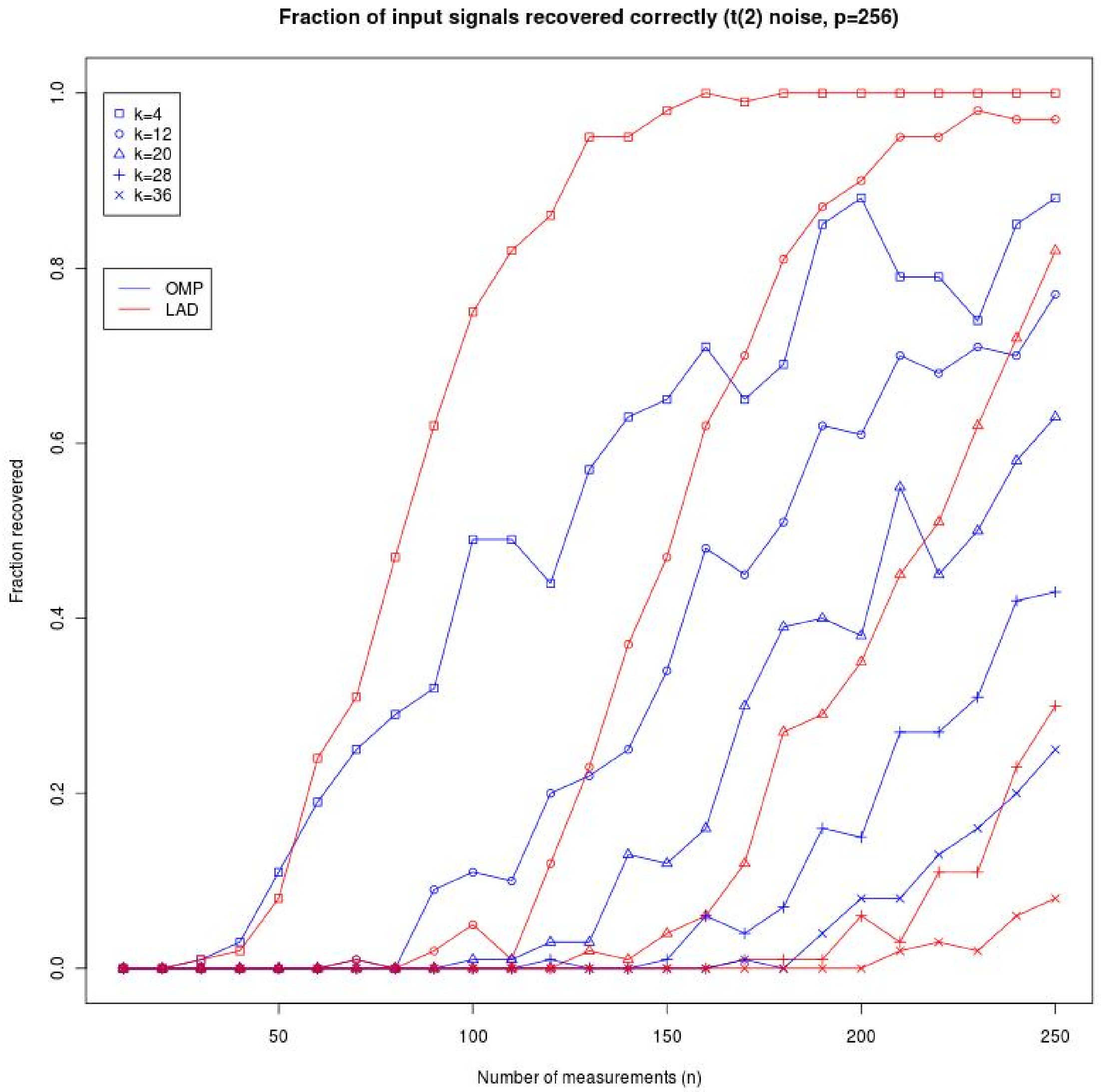

Figure 11: The fraction of 100 input signals recovered as a function of a number of measurements n for different sparsity levels k in dimension $p = 256$ in the presence of $t(2)$ distributed noise.

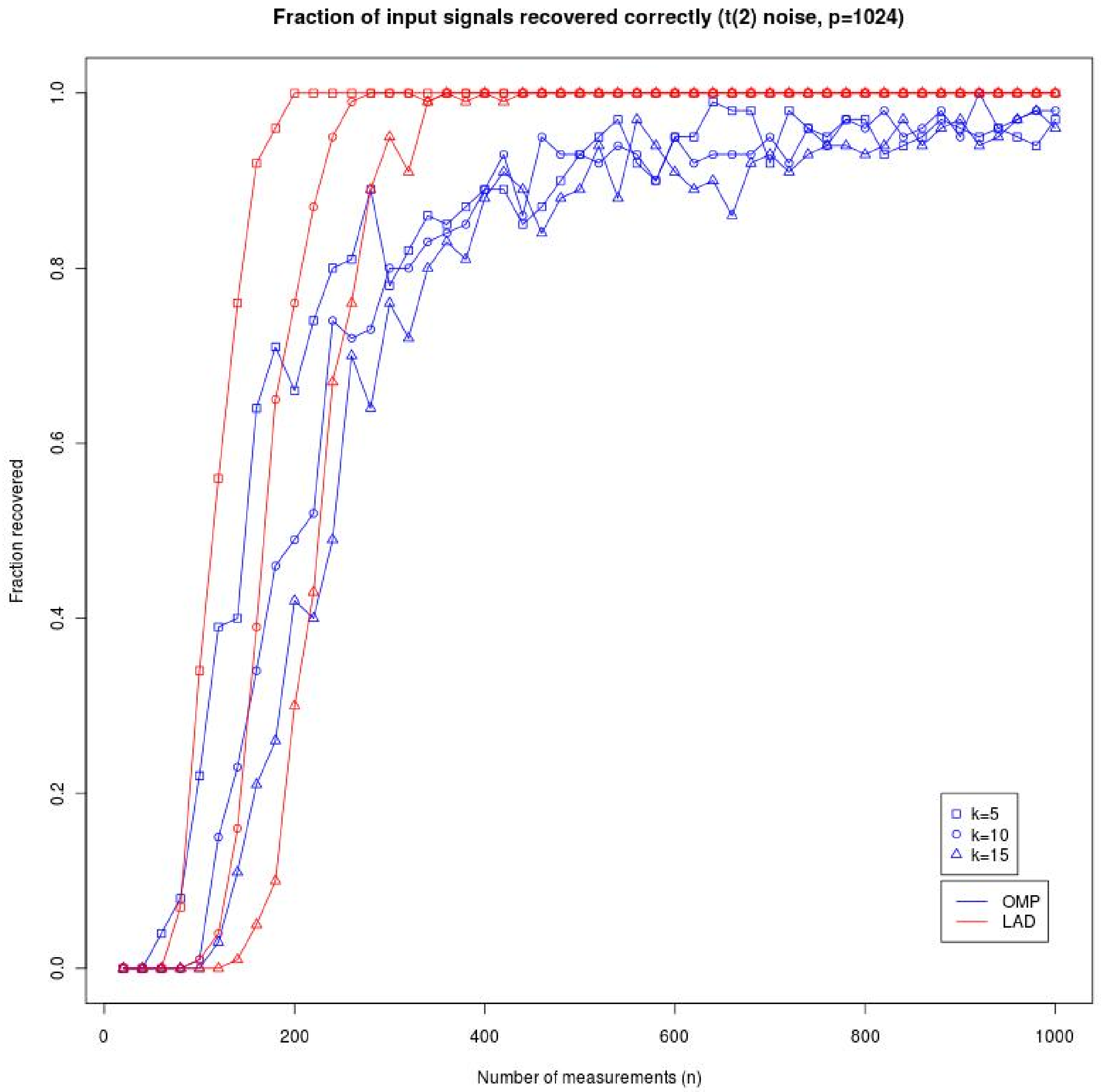

Figure 12: The fraction of 100 input signals recovered as a function of a number of measurements n for different sparsity levels k in dimension $p = 1024$ in the presence of $t(2)$ distributed noise.

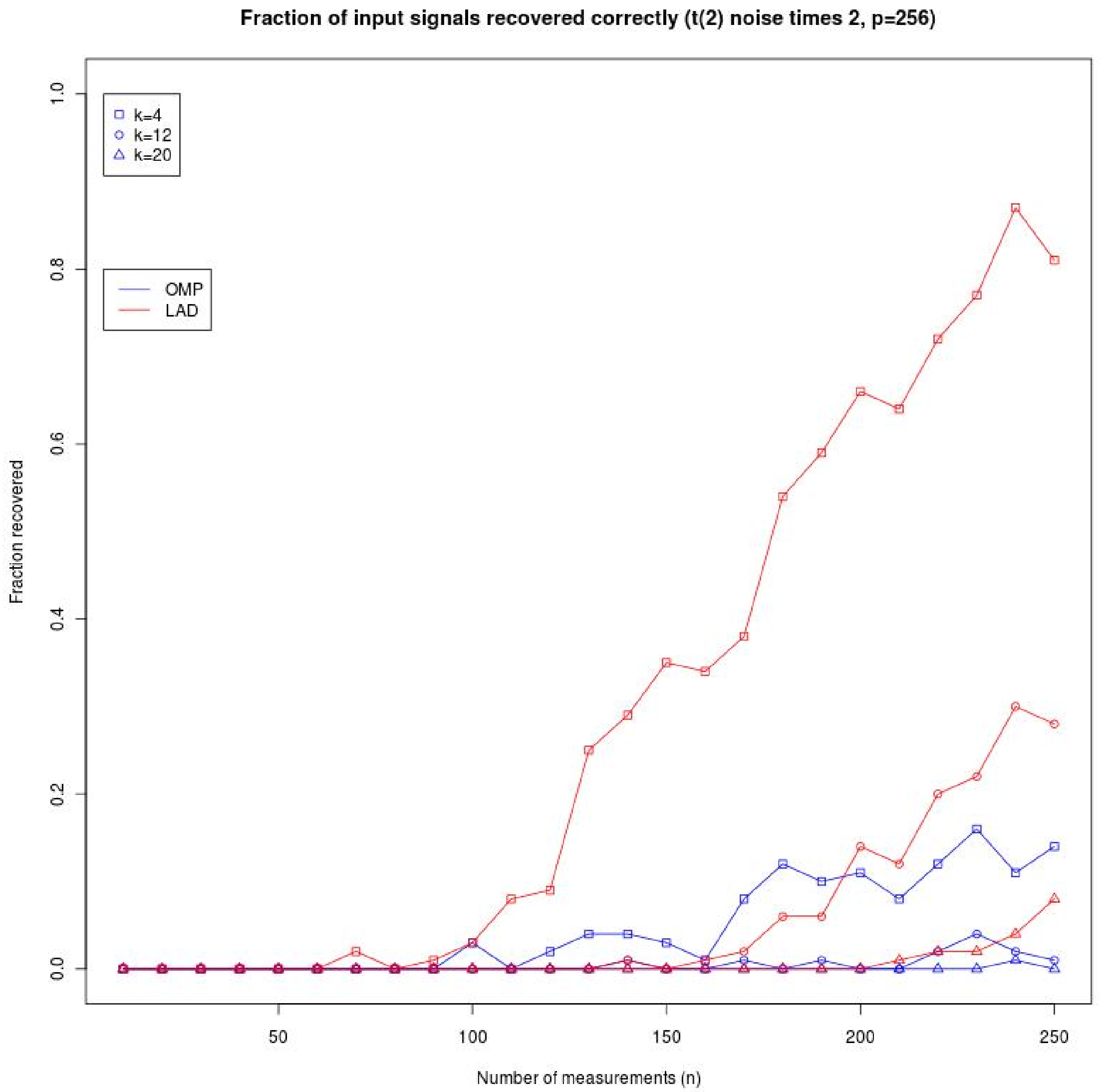

Figure 13: The fraction of 100 input signals recovered as a function of a number of measurements n for different sparsity levels k in dimension $p = 256$ where the noise is $t(2)$ distributed noise multiplied by 2. Here $k = 4, 12, 20$. For greater values of k both algorithms have the fraction recovered equal to 0.00.

References

[1] J. A. Rice. *Mathematical Statistics and Data Analysis.* Belmont, California: Duxbury Press, 1995.

[2] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning - Data Mining, Inference and Prediction.* Stanford, California: Springer Series in Statistics, 2008.

[3] T. Cai, and L. Wang. *Orthogonal Matching Pursuit for Sparse Signal Recovery With Noise.* IEEE Trans. on Inf. Theory, Vol. 57, No. 7, 2011.

[4] J. A. Tropp. *Greed is good: Algorithmic results for sparse approximation.* IEEE Trans. Inf. Theory, Vol. 50, pp 2231-2242, 2004.

[5] T. Cai, and T. Jiang. *Limiting Laws of Coherence of Random Matrices with Applications to Testing Covariance Structure and Construction of Compressed Sensing Matrices.* Annals of Statistics, Vol. 39, No 3, 2010.

[6] J. A. Tropp, and A. C. Gilbert. *Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit.* IEEE Trans. Inf. Theory, Vol. 53, No. 12, 2007.

[7] E. J. Candes and T. Tao *Decoding by Linear porgramming* IEEE Trans. Inf Theory, vol. 51, no. 12, pp 4203-4215, 2005.

[8] T. Zhang *Adaptive Forward-Backward Greedy Algorithm for Learning Sparse Representations* IEEE Trans. Inf Theory, vol. 57, issue 7, pp 4689-4708, 2011