Least angle quantile lasso regression

Jing Huang
ESMT European School of Management and Technology

Abstract

In This paper, we use the least angle method to select variables in quantile regression. Quantile regression has huge application in understanding tail distribution and variable selection is helpful to study the regression under limited resource. Unlike algebra algorithm, least angle method, suggested by Efron et al. (2004), chooses variable by selecting minimum angle between dependent variable and regressors rather than minimize the distance, in which case least angle method is believed to have fast speed. Furthermore, we found that the current algorithm, like Li and Zhu (2008) is fast in speed but not sufficient to find the optimal variable sets, and exhaustive method is too prohibitive. Therefore, we used an approximation to transform the minimization problem into a simple OLS problem and suggest several ways, like least trimmed square and PCA (factor analysis), to improve the algorithm to selection variables at each searching step more efficiently. At end we conduct a simulation study, which shows PCA method has the best performance among all the algorithms we tested.

1 Introduction

The goal of quantile regression is to find out the conditional quantile of observation $\{y_i\}$ for given data set $\{x_i\}$. Suppose the τth quantile of y_i giving x_i is estimated as $\xi(x_i,\beta)$. Then $\{\xi(x_i,beta)\}$ is the solution for the following minizing problem:

$$\min_{\beta}\sum_{i}^{n}\rho_\tau(y_i-\xi(x_i,\beta)) \tag{1}$$

where $\beta=(\beta_1,\beta_2,\cdots,\beta_p)'$. and $\rho_\tau(\cdot)$ is the quantile function:

$$\rho_\tau(z)=\begin{cases}\tau\cdot z & \text{if } z\geq 0\\ (\tau-1)\cdot z & \text{if } z<0\end{cases} \tag{2}$$

In quantile lasso of this paper, we assume a linear form of $\xi(x_i,\beta)=x_i\cdot\beta$ and we try to solve the following problem:

$$\min_{\beta}\sum_{i=1}^{n}\rho_\tau(Y_i-X_i\beta) \qquad \text{s.t. } \sum_{j=1}^{p}|\beta_j|\leq t \tag{3}$$

The name 'Lasso' stands for *least absolute shrinkage and selection operator*, was first introduced by Tibshirani (1996). The LASSO model aims for the least squares regression:

$$\min_{\beta}\sum_{i=1}^{n}(y_i-\sum_{j}^{p}x_{ij}\beta_j)^2 \text{ s.t } \sum_{j}^{p}|\beta_j|<s \tag{4}$$

The advantage of Lasso model is it can select variable once at a time so it somehow *'ranks'* the importance of them. In his paper, the algorithm to solve the minimizing problem follows some traditional Lagrangian methods under Kuhn-Tucker conditions suggested by Lawson and Hanson (1974) or David Gray.

We define $\sum_{i=1}^{n}\rho_\tau(\cdot)$ as τ-distance in R^n space, denoted as d_τ. Generally τ-distance is not well defined because $d_\tau(x-y)\neq d_\tau(y-x)$. But if we keep one point fixed in both value and position(always appeared in the minuend), as Y in (3), d_τ can still help us in understanding the distance changes between Y and $X\beta$, which is essential in solving (3). Therefore, d_τ should a definition not only of distance but also of direction. From now on, we call $d_\tau(y-x)$ as the τ-distance *from x to y*. For example, if $n=2$, the points that have the same τ-distance to the origin is showed as the blue lines in Figure 1(left). It can be proved easily that all the points on the blue line have the same τ-distance to the origin. We call these blue lines *equidistance boundary*. The equidistance boundary for three dimensions is also showed in Figure 1(right). To understand what's the case in high dimension space, we can always decompose it into 2 dimensional and 3 dimensional space. Here we introduce the definition of the summation of space:

Definition: Assume V_1 and V_2 are two subspace of V: $V_1\subseteq V\subseteq R^n$, $V_2\subseteq V\subseteq R^n$, V_1+V_2 is defined as:

$$V_1+V_2=\{\alpha_1x_1+\alpha_2x_2|\forall x_1\in V_1,x_2\in V_2,\alpha_1,\alpha_2\in R\}$$

If V_1 and V_1 are orthogonal, we use $V_1\bigoplus V_2$ instead of V_1+V_2. If e_i as unit vector with i^{th} element equals to 1 and all other elements equal to zeros, then for R^n we can always choose $n2$ subspaces $V_1^2,V_2^2,\cdots,V_{n2}^2$:

$$V_i^2=\{\alpha_1\cdot e_{i1}+\alpha_2\cdot e_{i2}|\forall\alpha_1,\alpha_2\in R\}$$

and $n3$ subspaces $V_1^3,V_2^3,\cdots,V_{n3}^3$:

$$V_j^3=\{\alpha_1\cdot e_{j1}+\alpha_2\cdot e_{j2}+\alpha_3\cdot e_{j3}|\forall\alpha_1,\alpha_2,\alpha_3\in R\}$$

such that:

$$V_1^2\bigoplus V_2^2\bigoplus\cdots\bigoplus V_{n2}^2\bigoplus V_1^3\bigoplus V_2^3\bigoplus\cdots\bigoplus V_{n3}^3=R^n$$

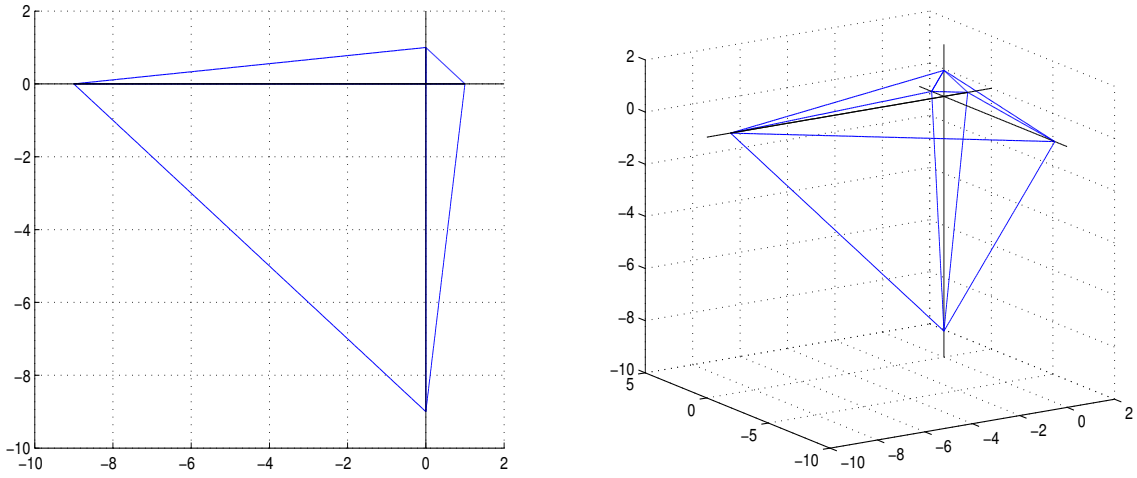


Figure 1: The equidistance boundary when $\tau = 0.9$ in 2 and 3 dimensional space

Apparently $2 \times n2 + 3 \times n3 = n$. By the way above, we decomposed R^n into subspaces that are mutually orthogonal and each subspace can be illustrated either in Figure 1(left) and Figure 1(right). Also, one can show the equidistance boundary for R^n can be decomposed into equidistance boundaries of subspaces illustrated in Figure 1. For example, for a R^5 space we can decompose it into two subspaces $V_1^2 = \{(\alpha_1, \alpha_1, 0, 0, 0) | \alpha_1, \alpha_2 \in R\}$ and $V_1^3 = \{(0, 0, \gamma_1, \gamma_2, \gamma_3) | \gamma_1, \gamma_2, \gamma_3 \in R\}$.

So to minimize $\sum_{i=1}^{n} \rho_\tau(Y_i - X_i\beta)$ is equivalent to find the smallest equidistance boundary. Till now, we figure out the objective function to be minimized in the R^n space. Our next step is to setup a geometric view of the restriction $\sum_{j=1}^{p} |\beta_j| \leq t$. Each n observations of regressor X_j can be taken as a vector in R^n. Their linear combination $\sum_{j=1}^{p} \beta_j \cdot X_j$ forms a hyperplane. For a certain level t, $\sum_{j=1}^{p} |\beta_j| \leq t$ restrict above combination to a sub-hyperplane. To illustrate this sub-hyperplane we could use the same decomposition discussed above accordingly. The purpose to do this is only for the convenience to show the progress of least angle methods in a 2 dimensional graph. The real algorithm does not need these decompositions.

Suppose now we have to vectors: x_1 and x_2, it is important to understand the shape of $\beta_1 \cdot x_1 + \beta_2 \cdot x_2$ under the restriction $|\beta_1| + |\beta_2| = t$. We claim that it is a straight line. To show this, we take three different points $p_1 = t \cdot x_1 + 0 \cdot x_2$, $p_2 = 0 \cdot x_1 + t \cdot x_2$ and $p_3 = (1-\alpha)t \cdot x_1 + \alpha t \cdot x_2$ where $\alpha \in (0,1)$. We need to show that $\vec{p_1 p_2} = \lambda \cdot \vec{p_1 p_3}$, then p_1,p_2 and p_3 are on the same line. Since p_3 can be any point between p_1 and p_2, by showing they lay on the same line we actually prove $\beta_1 \cdot x_1 + \beta_2 \cdot x_2$ is a straight line when $|\beta_1| + |\beta_2| = t$. The proof can be easily done by seeing:

$$\begin{aligned} \vec{p_1 p_2} &= 0 \cdot x_1 + t \cdot x_2 - (t \cdot x_1 + 0 \cdot x_2) \\ &= -t \cdot x_1 + t \cdot x_2 \\ &= (\frac{1}{\alpha})[-\alpha t \cdot x_1 + \alpha t \cdot x_2] \\ &= \lambda(p3 - p1) \\ &= \lambda \vec{p_1 p_3} \end{aligned}$$

which shows it's indeed a straight line. In rest part of this paper, we call it $t-$restriction line.

2 Least angle regression

From now, we decomposed high dimensional quantile lasso problem into a smaller dimensional space so that we could see the how least angle regression progress works. As showed in Figure 2, Y

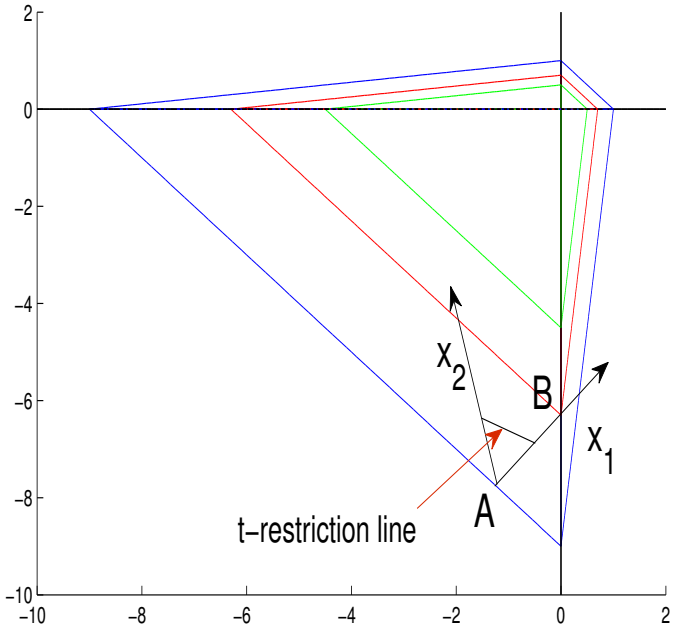


Figure 2: The equidistance boundary and t-restriction line when $\tau = 0.9$ in 2 space

is the origin point and A is the start point indicates the situation when $\sum_{j=1}^{p} |\beta_j| = 0$. As t grows, the $t-$restriction line moves towards the origin so that it can reach lower equidistance boundary. At the start point, the equidistance boundary is the blue line. As we can easily see from the figure, if A goes along the direction of x_1, the equidistance can shrink faster if it goes along x_2. The reason for this is the angle between the normal vector of the current equidistance boundary and x_1 is smaller than the angle between the normal vector of the current equidistance boundary and x_2. At the point x_1 is chosen rather x_2 by minimizing the angle between current equidistance boundary. So the method is called *least angle regression*.

One question might be raised here: Can x_1 and x_2 be chosen at the same time? In this situation, both x_1 and x_2 have the same angle with the normal vector of the current equidistance boundary. Therefore, the $t-$restriction line should be parallel with the current equidistance boundary. But on the other hand, if the regressors are from a stochastic progress, it can be showed that the probability that these two lines are parallel has zeros measurement. So in the algorithm, we just pre-exclude this possibility and distribute all β to, for example, x_1 in the figure.

As t grows, one element of $Y - X\beta$ reaches 0. 0 is the benchmark in the quantile function (2). In our figure, it means the normal vector of the current equidistance boundary will be changed. This moment is showed in Figure 2 as the $t-$restriction line moves to point B. Now we have two options, the first approach is to keep 0 in the norm vector, by which mean we need to add another covariate, like x_2 in the figure to maintain the 0. The second approach is to ignore this 0, taking that optimal path has already passed this vertex and comes to another hyperplane. Then the norm vector is needed

to be recalculated, which lead us to case as the initial step.

3 Algorithm

First we start from $\sum|\beta| = 0$. As discuss above, we choose Y as the origin of the n dimensional space, then the coordinate of the start point is $-Y$. the next step is to find out the normal vector of the equidistance boundary that $-Y$ stands. To do this, we can find a hyperplane that contains $-Y$. Notice that any hyperplane is spanned by the by some non-parallel lines each of which can be determined by two points in that hyperplane. Therefore, we could consider the points constitute the plane, for simplicity, we choose points from the axis. Like $\alpha \cdot e_i$, where e_i is a unit vector where only i^{th} element is 1 all other elements are zeros. For a n dimensional space, we could find n such points which are both on the axis and sharing the same τ-distance as the start point. With these n points we could construct $n-1$ different lines and then a $n-1$ dimensional hyperplane and the normal vector of this hyperplane can be easily calculated by the outer product of those directions of the lines.

Example 1: Initially $-Y = [2,-5,-4,1]'$ and $\tau = 0.9$. It's easy to calculate $\sum\rho_\tau(Y) = 8.4$. Consider four points in the R^4 space: $P_1 = (\frac{8.4}{0.9},0,0,0)$, $P_2 = (0,-\frac{8.4}{0.1},0,0)$, $P_3 = (0,0,-\frac{8.4}{0.1},0)$ and $P_4 = (0,0,0,\frac{8.4}{0.9})$. Let $f(\cdot) = \sum\rho_\tau(\cdot)$, we shall have $f(P_1) = f(P_2) = f(P_3) = f(P_4) = f(Y) = 8.4$. Since $f(\cdot)$ is a linear function, one can conclude that any point on the hyperplane spanned by $\{P_1,P_2,P_3,P_4\}$ has the same tau-distance as Y. To find out the norm vector of this plane, we can find three 3 lines on that plane, which can be chosen as $\vec{P_1P_2}$, $\vec{P_2P_3}$,$\vec{P_3P4}$. Their directions are $(-\frac{8.4}{0.9},-\frac{8.4}{0.1},0,0)$, $(0,\frac{8.4}{0.1},-\frac{8.4}{0.1},0)$ and $(0,0,\frac{8.4}{0.1},\frac{8.4}{0.9})$. The norm vector of this plane can be calculated as the outer product of $\vec{P_1P_2}$, $\vec{P_2P_3}$,$\vec{P_3P4}$.

$$\vec{P_1P_2} \wedge \vec{P_2P_3} \wedge \vec{P_3P4} = \begin{vmatrix} i & j & h & k \\ -\frac{8.4}{0.9} & -\frac{8.4}{0.1} & 0 & 0 \\ 0 & \frac{8.4}{0.1} & -\frac{8.4}{0.1} & 0 \\ 0 & 0 & \frac{8.4}{0.1} & \frac{8.4}{0.9} \end{vmatrix} \tag{5}$$

After standardization, this norm vector is $v = [-0.0781,0.7028,0.7028,-0.0781]'$. Therefore, use the same procedure above, with simple mathematics, we can conclude: suppose currently $-(Y - X\beta_k)$ is $P = [p_1,p_2,...,p_n]'$, the norm of the hyperplane contains S is $V = [v_1,v_2,\cdots,v_n]'$, where

$$v_i = \begin{cases} \frac{\tau-1}{\alpha} & \text{if } p_i > 0 \\ 0 & \text{if } p_i = 0 \\ \frac{\tau}{\alpha} & \text{if } p_i < 0 \end{cases}$$

α is positive and makes $\sum v_i^2 = 1$. v is the direction which makes $-(Y - X\beta_k)$ decrease at fastest speed.

For simplicity we standardize every regressor X_i to $||X_i|| = 1$, so that when we calculate the correlation between them, which automatically becomes inner product which is just the cosine value of the angle. Suppose we are at step k. At this moment, there are m_k observations satisfy $y_i - x_i\beta_k = 0$. We use the elbow definition from Li and Zhu (2008):

$$\varepsilon_k = \{i : y_i - x_i\beta_k = 0\}$$

ε_k means the elbow at step k and $|\varepsilon_k|$ is the number of elements in the ε_k. In our assumption, currently $|\varepsilon_k| = m_k$. As discussed above in the first step, we faced two options at the moment, the first:

Option 1: we need to chose $m_k + 1$ covariate from X to maintain these m_k zeros in the elbow. To do this, we first calculate the norm vector. Denote $P_k = Y - X\beta_k$ which is a n vector with m_k zeros. The space decomposition idea can be applied here to write P_{n-m_k} as the projection onto the subspace where P_i is not zero. Then the hyperplane and the norm

can be found out through algorithm as discussed in **Example 1**. Suppose the norm vector is $V^k = [v_1^k, v_2^k, \cdots, v_n^k]'$ with $v_j^k = 0$, $j \in \varepsilon_k$. To maintain the elbow, we need to find a linear combination of $k+1$ covariates from X, that is to say, we need to find $\beta_{i_1}^k, \beta_{i_2}^k, \cdots, \beta_{i_{m_k+1}}^k$, such that:

$$\beta_{i_1}^k \cdot x_{j,i_1} + \beta_{i_2}^k \cdot x_{j,i_2} + \cdots + \beta_{i_{m_k+1}}^k \cdot x_{j,i_{m_k+1}} = 0 \tag{6}$$

and

$$|\beta_{i_1}^k| + |\beta_{i_2}^k| + \cdots + |\beta_{i_{m_k+1}}^k| = 1 \tag{7}$$

where $x_{j,i}$ means the j^{th} observation from i^{th} covariate. Denote X_i as the i^{th} covariate of X, then $X_i = [x_{1i}, x_{2i}, \cdots, x_{ni}]'$. And denote $X^k = \beta_{i_1}^k \cdot X_{i_1} + \beta_{i_2}^k \cdot X_{i_2} + \cdots + \beta_{i_{m_k+1}}^k \cdot X_{i_{m_k+1}}$. Since V^k is the direction that $P = Y - X\beta_k$ decreases at fastest speed in term of tau-distance, we need to find such X^k that the inner production of V^k and X^k is biggest. In other words, such choice of covariate X solves the problem

$$\mathscr{A}_k = [i_1, i_2, \cdots, i_{m_k+1}] = argmax < V^k, X^k > \tag{8}$$

$\mathscr{A}_k$ is called the current active set.

The algorithm of Option 1 will increase the element of $\mathscr{A}_k$ by one. This does not necessarily mean add one variable to $\mathscr{A}_k$, which is the idea in Li and Zhu (2008), but instead, remove h ($h \leq |\mathscr{A}_k| = m_k$) elements of index out of it and add $h+1$ elements in. After this procedure, $\mathscr{A}_{k+1}$ will have $m_k + 1$ elements.

Option 2: Remove one element from the elbow ε_k and then follow the process in **Option 1**: calculate norm vector, find out m_k covariates satisfy (6) and(8) under the new elbow. This algorithm requires us to try all the $|\varepsilon_k|$ elements in ε_k: each time remove one element and do (6) and(8) to find out V_i^k and X_i^k ($i = 1, \cdots, |\varepsilon_k|$), record the out of $d_i =< V_i^k, X_i^k >$ and move to the next element. After all elements in ε_k have been tried, find out the choice of I that makes biggest d_I. Then denote $\tilde{V}^k = V_I^k$ and $\tilde{X}^k = X_I^k$.

After process of **option 1** and **option 2**, we can compare the two inner product $< V_k, X_k >$ and $< \tilde{V^k}, \tilde{X^k} >$, find the biggest one between these two and write the new pair as $< V_k, X_k >$. (if $< V_k, X_k >$ is chosen as the biggest pair, then $< V_k, X_k >$ does not change.) After the direction is chosen, we shall let P_k goes along this direction. Suppose before step k, there are already some coefficients are not zeros. Denote $\hat{\beta} = [\hat{\beta}_1, \hat{\beta}_2, \cdots, \hat{\beta}_p]'$ is $p \times 1$ vector. $\hat{\mathscr{A}_k} = \{i | \hat{\beta}_i \neq 0 \text{ before step } k\}$. We call $\hat{\mathscr{A}_k}$ the total active set in comparing current active set $\mathscr{A}_k$ at step k. After every step, the elements in current active set $\mathscr{A}_{k+1}$ will be added into the total active set $\hat{\mathscr{A}_k}$ and the new total active set will be denoted as $\hat{\mathscr{A}_{k+1}}$ accordingly. Since P_k goes along the direction X_k, we can write this path as $P_k + alpha \cdot X^k$, where α is a positive number. As $P_k + \alpha \cdot X^k$ is walking along this path, β also goes along the path $\beta = \hat{\beta} + \alpha\beta^k$. Therefore, α increase from 0 until either of the following two events happen: **1):** one element of $P_k + alpha \cdot X^k$ reaches 0, or **2):** the constraint of β in (3) is bounded. If **1)** happens, replace k with $k+1$ and the above algorithm will be repeated. If **1)** happens, the whole algorithm will stop, which means the calculation is completed.

The main difference between our algorithm and the algorithm of Li and Zhu (2008) is: ours allows to chose variables from wider range. At each step, our algorithm consider all possibilities while their algorithm only considers adding one variable at each time while keeping those ones that have been chosen already in past steps. The advantage and disadvantage are obvious: our algorithm is more efficient in minimizing the tau-distance since we consider all possibilities which of course contain also their choices while their model requires much less calculation cost. e.g. If we need to choose 15 variables out of 30, our algorithm will try C_{30}^{15} possibilities and theirs only need to calculate $30 - (15 - 1) = 16$ distance. The calculation of C_{30}^{15} outcomes is almost impossible even nowadays, not to mention choosing variables under bigger p. So a method to avoid trying all possibilities is very much necessary. in the following section, we will discuss ways to decrease calculation cost.

4 Efficiency improvement

The first step we shall take to decrease calculation time in to find a way to simplify the calculation in (6) and (8). (6) is a linear equation systems with m_k equations m_{k+1} unknowns: $\beta^k_{i_1}, \beta^k_{i_2}, \cdots, \beta^k_{i_{m_k+1}}$. These unknowns can be solved out with the additional equation (7). Rewrite (6) as:

$$\begin{cases} \beta^k_{i_1} x_{j_1 i_1} + \cdots + \beta^k_{i_{m_k+1}} x_{j_1 i_{m_k+1}} = 0 \\ \beta^k_{i_1} x_{j_2 i_1} + \cdots + \beta^k_{i_{m_k+1}} x_{j_2 i_{m_k+1}} = 0 \\ \vdots \\ \beta^k_{i_1} x_{j_{m_k} i_1} + \cdots + \beta^k_{i_{m_k+1}} x_{j_{m_k} i_{m_k+1}} = 0 \end{cases} \tag{9}$$

consider a plane made by directions (or lines):

$$\begin{matrix} [x_{j_1 i_1}, x_{j_1 i_2}, \cdots, x_{j_1 i_{m_k+1}}] \\ [x_{j_2 i_1}, x_{j_2 i_2}, \cdots, x_{j_2 i_{m_k+1}}] \\ \vdots \\ [x_{j_{m_k} i_1}, x_{j_{m_k} i_2}, \cdots, x_{j_{m_k} i_{m_k+1}}] \end{matrix} \tag{10}$$

according to (9), $\beta^k = [\beta^k_{i_1}, \beta^k_{i_2}, \cdots, \beta^k_{i_{m_k+1}}]'$ is orthogonal to that plane spanned by above directions. Furthermore, we can right $< V^k, X^k >$ in (8) as:

$$\sum_{j=1}^{n} v^k_j \cdot (\sum_{h=1}^{m_k+1} \beta^k_{i_h} X_{j,i_h}) = \sum_{h=1}^{m_k+1} \beta^k_{ih} \cdot (\sum_{j=1}^{n} v^k_j x_{j,i_h}) \tag{11}$$

Denote $Z^k = [\sum_{j=1}^{m_k} v^k_j x_{j,i_1}, \sum_{j=1}^{m_k} v^k_j x_{j,i_2}, \cdots, \sum_{j=1}^{m_k} v^k_1 x_{j,i_{m_k+1}}]$. Then (11) can be written as the inner product: $< \beta^k, Z^k >$, which has the geometric meaning as the projection of Z^k on β^k times the length of β^k. Given that β^k is orthogonal to (10), if $||\beta^k||^2 = 1$, $< \beta^k, Z^k >$ is the distance between Z^k and the plane (10). This reminds us of the Ordinary Least Square(OLS): Consider

$$\bar{Y} = \bar{X}\theta + \varepsilon \tag{12}$$

Suppose the OLS estimator of θ is $\hat{\theta}$, let error term is $u = \bar{Y} - \bar{X}\hat{\theta}$. Then $u' \cdot u$ measures the distance between $\bar{Y}$ and space of $\bar{X}$. In our case $\bar{Y} = Z^k$ and $\bar{X}$ is a $(m_k + 1) \times m_k$ matrix constituted by vectors in (10), this guarantees that the norm vector of $\bar{X}$ has unique directions. In other words: for any two directions, say γ_1 and γ_2, which are both orthogonal to $\bar{X}$: $\gamma_1' \cdot \bar{X} = \gamma_2' \cdot \bar{X} = 0$, γ_1 and γ_2 must be linear depended: there exists a non-zero α. such that $\gamma_1 = \alpha\gamma_2$. Since both the error term u and β^k are orthogonal to $\bar{X}$ (or Z^k) : $u' \cdot \bar{X} = 0$. So we can conclude u and β^k has the same direction. Together with (7), we can see

$$\beta^k = \frac{u}{\sum |u|}. \tag{13}$$

u is the error term from (12). The object in (11) becomes:

$$\begin{aligned} (Z^k)' \cdot \beta^k &= \frac{\bar{Y}' u}{\sum |u|} \\ &= \frac{\bar{Y}'(\bar{Y} - \bar{X}(\bar{X}'\bar{X})^{-1}\bar{X}'\bar{Y})}{\sum |u|} \\ &= \frac{u'u}{\sum |u|} \end{aligned} \tag{14}$$

Now we have simplified the problem: let $\bar{Y}$ be $p \times 1$ vector defined like Z^k but use all covariates, $\bar{X}$ is a $p \times m_k$ matrix like (10) but all the covariates. We need to find $m_k + 1$ observations from $\bar{Y}$ and $\bar{X}$ to form new $\tilde{Y}$ and $\tilde{X}$. Under such choice: $\tilde{u}'\tilde{u}/\sum |\tilde{u}|$ (14) is maximized, where $\tilde{u}$ is the error term from regression $\tilde{Y} = \tilde{X}\theta$. Unlike $u = \bar{Y} - \bar{X}(\bar{X}'\bar{X})^{-1}\bar{X}'\bar{Y}$, there is no general form for $\sum |u|$, so we use $u'u$

as an approximation of (14). This is not very an accurate approximation, but we can use this idea to narrow down the choice at least a little bit. e.g. choosing 15 variables out of 30 is too prohibitive, but choosing 15 variables out of 20 require much less calculating power. So we can preliminarily choose N sets that make first N biggest $u' \cdot u$, then try every possibility in within these smaller choices of sets. In the following section, we will discuss several ways to find the smaller set containing the potential best choices.

4.1 Least trimmed square

Consider the linear mode in (12), the *least trimmed squares(LTS)* estimator (Rousseeuw and Leroy (2005)) is defined as:

$$\hat{\theta}_{LTS} = arg\min_{\theta} \sum_{i=1}^{h} \{(\bar{Y}_i - \bar{X}_i\theta)^2\}_{1:n} \tag{15}$$

where $\{(\bar{Y}_i - \bar{X}_i\theta)^2\}_{1:n}$ is the i^{th}-order statistic from listing all $\{(\bar{Y}_i - \bar{X}_i\theta)^2\}$. This method chooses the h "best fitting" observations and is originally used to detect outliers. Agulló (2001) have discussed several algorithm for computing the LTS estimator based on swap observations between those h selected ones and those $n-h$ non-selected ones. He argues: Let $M = \bar{X}'\bar{X}$ in (12), if we add one observation $[y,x]$ to and run the regression again, the sum of squared residual will increase by:

$$u^{2+} = \frac{(y - x\hat{\theta})^2}{1 + xM^{-1}x'} \tag{16}$$

If we remove on observation $[\tilde{y},\tilde{x}]$ and run the regression again, the sum of squared residual will decreased by:

$$u^{2-} = \frac{(\tilde{y} - \tilde{x}\hat{\theta})^2}{1 - \tilde{x}M^{-1}\tilde{x}'} \tag{17}$$

To apply LTS algorithm to our case, we need only change the *min* sign in (15) with *max*. We choose the *Minimum-maximum exchange algorithm (MMEA)* in Agulló (2001)'s paper.In his paper he mentioned the exchange algorithm does not guarantee the exact LTS estimator, several tries with different starting set should be made. So at each time, we start with 5 different initial set. If at current stage we can choosing $m_k + 1$ variables from p candidates, a sketch of the algorithm can be described as:

1. Run the regession (12) with all the observation. Rearrange the residuals $\{u_1, u_2, \cdots, u_p\}$ in descending order, choose the first $m_k + 1$ observations to form the initial set, label it as "$M_k + 1$". The rest unselected observations form the set "$P - M_k - 1$".
2. Remove an observation $[y,x]$ from $M_k + 1$ that makes smallest decrease in $\sum u_i^2$ according to (17). Replace it with an observation $[\tilde{y},\tilde{x}]$ from Q that makes the biggest increase in $\sum u_i^2$ according to (16). Switch them then update $M_k + 1$ and $P - M_k - 1$.
3. Repeat step 2 until such exchange does not increase the total sum of squared residual.
4. Try with another 4 different starting points.

According to section 3, at each step, there are two options to be applied. Therefore, the computation cost for this LTS (MMEA) at the step where there are m_k element in the elbow is the sum of the computation cost from these two options. For the first option, we need to choose $m_k + 1$ variables from totally p candidates. Each exchange requires $m_k + 1 + (p - m_k - 1) = p$ calculations, times in each calculation, we need to calculate M^{-1}. If we apply Cholesky factorization, we need to solve a m_k linear equations system. So the total calculation cost is $p \cdot m_k$ for option1. For option 2, we have to try to remove one index at each time from the elbow, and then choose m_k variables, so the calculation cost is for option 2 is $m_k \cot p \cdot (m_k - 1)$. Hence the totally calculation cost of each step for this method is $p \cdot m_k + m_k \cot p \cdot (m_k - 1) + O(m_k^2 \cdot p) = O(p^3)$.

4.2 Adverse least trimmed square

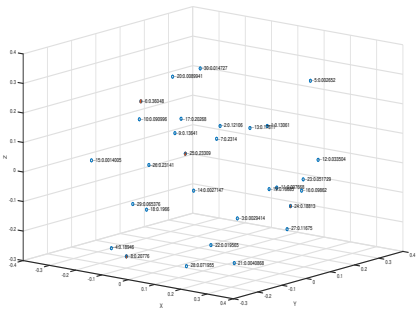
Since we are looking for the subset of observations that maximizing the sum of squared residual, another approach would naturally be finding the subset that minimizing the sum of squared residual and then exclude this subset from choosing candidates. Therefore, we can indeed use (15) to narrow down the potential observations that fit our profile. Since we are actually looking for (14) instead of $u'u$, we need to loose the constraint a little bit. E.g. we want to choose 5 observations out of 30 candidates, we can set $h = 20$ in (15). After calculation, we find these 20 and exclude them from the total 30 observations, which left us with 10 observations. Then we try all possibilities of C_{10}^5 choices to find out the best 5 that maximizing (14).

The algorithm of this approach is very similar to the one in subsection 4.1. The computation cost is also similar expect M now has the rank of $p - m_k - N$, N is the additional number of variable we use to make sure we have include right selection even we use un-precise approximation (in above example $N = 5$). So the computation for preliminary choosing $O(m_k \cdot p \cdot (p - m_k - N)) = O(p^3)$. After that, process of $C_{m_k+N+1}^{m_k+1} = O(((m_k + N + 1)/(m_k + 1))^{m_k+1})$ trials for option 1 and $m^k \cdot C_{m_k+N}^{m_k} = O(((m_k + N)/m_k)^{m_k})$ trials for option 2 need to be calculated. In total, the calculation cost is: $O(p^3) + O(((m_k + N + 1)/(m_k + 1))^{m_k+1}) + O(((m_k + N)/m_k)^{m_k})$.

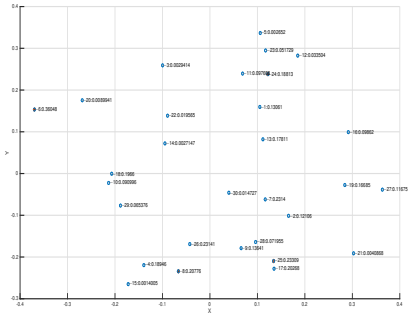
4.3 PCA (factor analysis)

In many cases, the choice of variable exhibit clustering properties. E.g. when we run the whole regression (12), calculate $u = \bar{Y} - \bar{X}\hat{\theta}$, the 2^{nd}, 5^{th} and 7^{th} element of u might be small, but when we choose 2^{nd}, 5^{th} and 7^{th} observations to make the new $\tilde{X}$ and $\tilde{Y}$, then run the OLS regression, the sum of square residual will become big. Ideally, to form a large sum of square residual $u'u$, it does not only require that at each observation level, the distance between $\tilde{Y}_i$ and $\tilde{X}_i$ needs to be relatively big. The distance among all $\tilde{X}_i$ better to be big too, in which case they can "better" span a plane. The idea is heuristic: decompose $\bar{X}\bar{X}'$ (not $\bar{X}'\bar{X}$) to get the principle components or factors. Then find the observations that constitute the most of the $var(\bar{X}')$. One example of each analysis is showed in Figure 3:

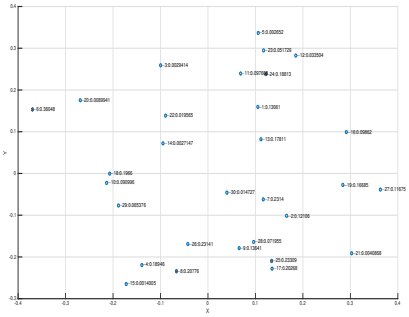
Figure 3 is a factor analysis of $\bar{X}\bar{X}'$ where $\bar{X}$ is a 30×3 covariate matrix. Our goal is to find out $3 + 1 = 4$ observations that make (14) biggest. First, we run the regression with whole observations, and record the 30 regression residuals (as showed in each sub-fig). $\bar{X}\bar{X}'$ has a rank of 3, so it has only 3 non-zero eigenvalue and 3 corresponding eigenvectors. These 3 eigenvalues contain 30 3-dimensional points who are displayed in Figure 3. Furthermore, we tried all the C_3^40 combinations and found out index $\{6, 8, 24, 25\}$ together make the largest (14), they are market with red dots in the figure. We can see these red points come from different quadrants. This gives us an intuition how to choice observations: find one index from one quadrant. Note that there are 8 quadrants in three dimensional world, so we need at first find 4 quadrants from totally 8 candidates. We don't need to consider all points, e.g. point 5 comes with residual 0.002652 should be definitely excluded from considering. So we can set a threshold like we keep only the first half of the observation with largest regression residuals. If we applied this threshold to our example in the figure, we shall delete all observations with residuals smaller than 0.1077 and keep only half (15) observations. After this filtering, we see $\{4, 8, 18\}$ are in $1st$ quadrant $(neg(-), neg(-), neg(-))$, $\{19, 27\}$ are in the $2nd$ quadrant $(pos(+), neg(-), neg(-))$, $\{2, 7, 9, 17, 25\}$ are in the $6th$ quadrant $(pos(+), neg(-), pos(+))$, $\{6\}$ are in the $7th$ quadrant $(pos(+), pos(+), neg(-))$. So if we choose $\{1st, 2nd, 6th, 7th\}$ quadrants from all C_8^4 combinations, given that we choose one observation from each quadrant, we have to try $3 \times 2 \times 5 \times 1 = 30$ combinations. This is the computation cost if we choose $\{1st, 2nd, 6th, 7th\}$ quadrants, we need to try other combination of quadrants too. In the end, there are totally 501 com-



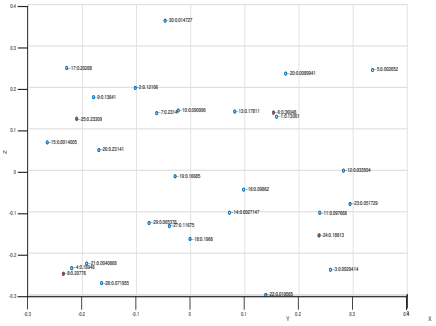
(a) 3d perspective



(b) Perspective from axis X and Y



(c) Perspective from axis X and Z



(d) Perspective from axis Y and Z

Figure 3: An example of observation selection through PCA(factor analysis) method. The numbers affiliated with spots are regression residuals. Red spots indicate the optimal observations calculated through exhaustive method(try all possibilities).

binations, this is significantly smaller than $C_{30}^{4} = 27405$. The difference will increase when we need to choose a large number of observations. E.g, when we need to choose 12 observations out of 30, since lots of quadrants contain only one observation or no observation at all, above PCA algorithm produces only 455 combinations. On the other hand, if we want to try all possible combinations, we have to calculate $C_{30}^{12} = 86493225$ times.

The computation cost PCA method is can not be consistently concluded. It depends on how the eigenvectors are distributed in the entire space. E.g. If we are choosing 4 observations out of 15 as above. So there are 8 quadrants to consider. It's like randomly throwing 15 balls into 8 boxes, we then choose 4 boxes from them and calculate the combination among them. Although it is impossible to say exactly what is the combination at each step, we can work out an upper bound. Since for any integer k, $(\frac{k}{2})^2 > (\frac{k}{2}-1)(\frac{k}{2}+1)$. Follow this intuition, the combination with max number of possibilities should "equally" dividing the candidates into quadrants so that each quadrant should have the same number of candidate. Suppose we are choosing m_k+1 observations from totally $[\gamma \cdot p]$ (γ is the cutting quantile, if we keep only half observations according to regression residual, $\gamma = 0.5$) candidates the upper bound can be calculated as:

$$C_{[\gamma \cdot p]}^{m_k+1} [\frac{[\gamma \cdot p]}{2^{m_k}}]^{m_k+1} \tag{18}$$

$[x]$ means the smallest integer that is bigger than x. Apply (18) to our example we shall have the upper bound equal to 1140 bigger than 501, which is what we get from the exact computation. At later stage of selection, m_k become larger (2^{m_k} can not exceed $[\gamma \cdot p]$, if it does, we have to increase

γ), $[\frac{p/2}{2^{m_k}}]$ will approach one, so (18) goes to $C_{[\gamma\cdot p]}^{m_k+1}$. In the example above where $m_k = 11$ and $\gamma p = 15$, $C_{[\gamma\cdot p]}^{m_k+1} = C_{15}^{12} = 455$, which matches our empirical result.

Denote (18) as $q(m_k, p)$, as discuss in the introduction of other algorithms, the computation cost at each step of *PCA* method requires $q(m_k, p)\cdot(m_k+1) + m_k \cdot q(m_k-1, p)\cdot m_k$.

4.4 Delete one observation at a time ("Kick-out")

At last, we provide another intuitive way to detect the best subset. As discussed above, the biggest difficulty of choosing observation is the clustering property the observations have and calculating all the combinations require heavy computation cost. To reduce computation load, the most straight way is to ignore this clustering property and analysis each observation independently. Following this idea, we suggest the algorithm below:

Process is described as following:

1 Run the regression (12) with all observations and record the regression residuals u.

2 Find the observation with smallest residual and delete it from the candidate group.

3 Repeat step 1 and 2 until idea number of observations are still left in the candidate group.

In practice, like in subsection 4.2, we can stop the algorithm when the number of observation in the candidate group is still large than the actual number we want, and then try the all the possibilities within this bigger set.

The computation cost also contains two parts as in subsection 4.2: the preliminary selecting and final trying out all remaining possibilities. The first part involve $p-(m_k+1+N)$ steps of OLS regression for option 1 and $m_k\cdot(p-(m_k+N))$ steps of OLS for option 2. Each OLS consume m_k+1 computation cost for option 1 and m_k for option 2. So the total cost for first part is $(p-(m_k+1+N))\cdot(m_k+1)+m_k^2\cdot(p-(m_k+N)) = O(p^3)$. The computation cost from the second part is the same as subsection 4.2: $p\cdot m_k + m_k\cot p\cdot(m_k-1)+O(m_k^2\cdot p) = O(p^3)$. Therefore, the total computation cost for this method is $(p-(m_k+1+N))\cdot(m_k+1)+m_k^2\cdot(p-(m_k+N))+p\cdot m_k+m_k\cot p\cdot(m_k-1)+O(m_k^2\cdot p) = O(p^3)$.

4.5 Mix-strategy

Since we used the approximation for (14) by ignoring the denominator. LTS, Adv.LTS or PCA might be inaccurate to find the variables. Since exhaustive method (try all possible combinations) is not feasible when at the step where we need to choose a large number of variables. it is still achievable when we choose small number set. Hence, the mix-strategy of using multiple methods is worthy to be tried: At the early stage, when we use a small number of variables, we use the exhaustive method, when the number of candidates become larger, we use the method discussed above. In the simulation, we choose the threshold of switching method when we have the number of variables to be chosen bigger than 3.

5 Simulation study

We generate the regressor X of size $n = 50$ and $p = 30$ from random normal distribution. The error term ε is also from random normal. $\beta_1, \beta_2, \cdots, \beta_{30}$ are generate from uniformly distribution, if $\beta_i <= 5$, force $\beta_i = 0$.The Y is generated as:

$$Y_i = X_i\cdot\beta + \varepsilon_i$$

To check the performance under different constraints of β, we set different t in (3). At each simulation, let $T = \sum_i^p |\beta_i|$, we let t be a portion of: $t = \alpha \cdot T$. In the results we show, α is chosen to equal to 0.2, 0.5 and 0.8. In each scenario with different α, we generate X^j, Y^j $(j = 1, \cdots, 20)$ and β 20 times. After each generating, we use the different algorithms to solve the problem (3). When $\sum_{i=1}^p |\beta_i|$ in bounded (reaches t), we record the process time and $d_j = \sum_{i=1}^n \rho_\tau(Y_i^j - X_i^j \beta)$. Therefore, in each scenario, we should have 20 process time and d recorded. Within these 20 generating, the distances d_j are not comparable, a standardization process is needed. In each j^th generating, denote d_j^i as the distance from i^{th} method, we then define "Score" for each method as:

$$S_j^i = \frac{d_j^i}{\sum_{i=1}^n d_j^i} \tag{19}$$

In each scenario, after 20 process time are recorded and 20 scores are calculated, we compute the average time and average score (with variance) and display them in Table 1. When α is small, we expect not so many variables will be selected at the time the constraint is bounded, so we didn't calculate the "Mix" algorithm which is aiming in reducing computation load in the large set selecting cases. "Exhaustive" method is too time-consuming at later the stage of selecting, so we didn't consider it when α is relatively big.

model	$\alpha = 0.2$		$\alpha = 0.5$		$\alpha = 0.8$	
	Score (Var.)	Time	Score (Var.)	Time	Score (Var.)	Time
"Li-Zhu"	1.0103(.39e-3)	0.2515s	1.1442(.0065)	0.2708s	1.1777(0.0181)	0.6172s
"Kick-out"	0.9980(.02e-3)	2.2256s	0.9702(.0001)	248.6s	-	-
"Exhaustive"	0.9978(.02e-3)	65.0302s	-	-	-	-
"LTS"	0.9982(.02e-3)	2.2900s	0.9725(.0002)	225.3s	0.9665(.0008)	749.6s
"Adv.LTS"	0.9981(.01e-3)	1.879s	0.9973(.0003)	31.7876	0.9943(.0007)	498.7
"PCA"	0.9975(.03e-3)	3.5840s	0.9712(.0001)	383.4s	0.9501(.0016)	1297s
"Mix (LTS)"	-	-	0.9731(.0002)	36.22s	0.9697(.0006)	241.3s
"Mix (Adv.LTS)"	-	-	0.9988(.0006)	41.81s	0.9906(.0017)	481.3s
"Mix(PCA)"	-	-	0.9726(.0002)	360.2s	0.9511(.0006)	1418s

Table 1: Result of different method with simulated data, the red highlight indicate the best choice within each column

We see that when α is small (therefore, t is small), the results are quite close, but Li and Zhu's method is dominant with much less calculation time. When the constraint is loose (big α and t), although is still winning with time, Li and Zhu's method is less optimal in term of minimizing the distance. The simulation result shows "PCA" seems to produce the best way to minimizing $\sum_i^n \rho_\tau(Y_i - X_i\beta)$ with $L1$ norm constraint.

6 Discussion and future work

In this paper, we provide a feasible algorithm to solve constraint problem (3) with improved performance in term of minimizing the tau-distance: $\sum \rho_\tau(\cdot)$. Like a lot of other maximization problem, the algorithm we suggested does not guarantee a globe optimal. There might be another solution which can further decrease the $\sum_i^n \rho_\tau(Y_i - X_i\beta)$ with the same $\sum_{i=1}^p |\beta_i|$. But due to highly non-differentiable property of both object function and constraint function, we believe this solution is difficult to find: Even we can try all the possibilities at each step and use the best, we can only make sure that the path between two steps is optimal, but we can't say the combined path of multi-step is still optimal.

E.g. the distance between AB is smaller than the distance between AC, so starting from place A, the optimal path is AB compare with AC. However, B is not our final destination, but D. It is highly possible that $AB+BD$ is bigger than $AC+CD$. The worse problem, in our case, is: we even don't know the existence of D, even B nor C. We just know we are starting from A, B and C are gotten from calculation. With other calculation procedures, uncountable points like B or C can be derived, like in an ocean. So, we can only say our method has some improvement comparing with the existent one, but it is still not sufficiently the exact solution of problem (3).

Also in this paper, we didn't talk about the stopping criteria which is considered as an important part for variable selecting. The reason for this is that we think these criterion have already been fully discussed in Li and Zhu (2008). The main purpose of this paper is to provide a geometrical perspective in solving such problem and to offer some alternative algorithms with improved ability to minimize the objective function under $L1$ constraint.

In the future, it is worthy to consider the minimization problem under $L2$ norm constrain: $\sum_{i=1}^{p}(\beta)^2 \leq t$. As discussed in section 4, by using $L2$ norm, the maximization problem in each step will be directly become the calculating sum of squared residual without approximation. Another idea can be drawn from seeing the performance of PAC (factor analysis) in these kinds of question. Since working good in our case, PAC may be considered as an alternative algorithm in LTS and outlier detecting.

References

AGULLÓ, J. (2001): "New algorithms for computing the least trimmed squares regression estimator," *Computational Statistics & Data Analysis*, 36, 425–439.

EFRON, B., T. HASTIE, I. JOHNSTONE, AND R. TIBSHIRANI (2004): "Least angle regression," *The Annals of statistics*, 32, 407–499.

LAWSON, C. L. AND R. J. HANSON (1974): *Solving least squares problems*, vol. 161, SIAM.

LI, Y. AND J. ZHU (2008): "L 1-norm quantile regression," *Journal of Computational and Graphical Statistics*, 17, 163–185.

ROUSSEEUW, P. J. AND A. M. LEROY (2005): *Robust regression and outlier detection*, vol. 589, John Wiley & Sons.

TIBSHIRANI, R. (1996): "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288.