1 Notation

Let us define some important notations used in this note:

- The $\ell_0$ norm is defined as $\|x\|_0 :=$ the number of nonzero elements in $x$. For example, $\|[1 \ 0 \ 2\ -2.3 \ 0]^T\|_0 = 2$. Please note that the $\ell_0$ norm is not a real “norm”, because it does not satisfy the definition of norms;
- $1$ denotes the vector with 1 in all elements.

2 Introduction

To motivate the sparse learning, let us start from the linear equation case:

$$Ax = b$$

where $A \in \mathbb{R}^{n \times p}$ and $b \in \mathbb{R}^p$. In sparse learning, we are particularly interested in the situation $p > n$ (under deterministic system), which means that the number of variables is larger than the number of linear equalities. In general, the solution to (1) is not unique. “$p > n$” is a very common situation. $n$ is usually proportional to the money we should pay. $p$ is proportional to the complexity of the problem.

Let me take myself as example. I just move to Rochester and am looking for a house. I want to study the key factors which affect the house price. Specifically, I want to know the weight for each factor such that when see a new house I can tell its price is too high or low. Therefore, I have to do some investigations. I need visit a few houses and obtain the values for all factors of each house, say, the size of house, the number of floors, the number of neighbors, the distance to downtown, the score of the local elementary school, the score of the local middle school, the score of the local high school, the number of doors, the number of windows, so on and so forth. I assume a linear model “price = size $\times x_1$ + # floors $\times x_2$ + · · · + # windows $\times x_p$. Let $a_i$ and $b_i$ be the factor vector and the price for the $i$th house. We are essentially solves a linear system

$$Ax = b.$$  

Unfortunately the number of factors is large, say 1000, while the number of houses I want to investigate is very few (say, 100) because it is costly to investigate a house. In general, I cannot
recover the weight for each factor. It looks hopeless. However, if we know that the factors which affect the price is very few, that is, the true solution is very sparse, then we may have some hope. We can pursue a sparse solution to the linear equation in the following:

\[(P0) \min_x \|x\|_0 \quad s.t. \quad Ax = b.\] (2)

Unfortunately, this problem is NP-hard. To make it solvable, people use $\ell_1$ norm to approximate $\ell_0$ norm and solve the following problem:

\[(P1) \min_x \|x\|_1 \quad s.t. \quad Ax = b.\] (3)

This is a convex problem which can solved in polynomial time generally.

Here, you may ask why $\ell_1$ norm is a good approximation to $\ell_0$ norm? Simply speaking, $\ell_1$ norm is the “tightest convex” approximation to $\ell_0$ in the region $\|x\|_\infty \leq 1$.

3 Algorithm

The optimization problem in (3) is a convex but nonsmooth problem. It can be formulated into a linear programming problem which can be solved very efficiently by using mature solver or calling some functions in R or Matlab.

Introduce two variables $x_+$ and $x_-$ to denote the positive and negative part of $x$ respectively. In particular, we have

\[x = x_+ - x_- \quad \|x\|_1 = 1^T(x_+ + x_-) \quad x_+ \geq 0 \quad x_- \geq 0.\]

Now we can rewrite (3)

\[\min_{x_+, x_-} 1^T(x_+ + x_-) \quad s.t. \quad A(x_+ - x_-) = b \quad x_+ \geq 0 \quad x_- \geq 0.\] (4)

4 Theoretical Guarantees

Assumption 1. Restricted Isometry Constant (or Restricted Isometry Property) Given a matrix $A \in \mathbb{R}^{n \times p}$, the “$s$”-restricted isometry constant $\delta_s$ is defined to be the quantity satisfying

\[(1 - \delta_s)\|h\|^2 \leq \frac{1}{n}\|Ah\|^2 \leq (1 + \delta_s)\|h\|^2 \quad \forall \|h\|_0 \leq s.\] (5)

An equivalent condition to (5) is

\[\frac{1}{n}\sigma_{\max}(A_F^T A_F) \leq 1 + \delta_s \quad \frac{1}{n}\sigma_{\min}(A_F^T A_F) \geq 1 - \delta_s \quad \forall \ F \subset \{1, 2, \cdots, p\} \text{ and } |F| \leq s,\] (6)

where $F$ is an index set and $A_F$ is a submatrix of $A$ consisting of $A$’s columns with index in $F$. This condition approximately measures the difference (dependence) among multiple feature vectors in $A$. In the extreme case, if all feature pairs $A_i$ and $A_j$ are orthogonal to each other and are normalized to $\sqrt{n}$, then we have that all $A_F^T A_F$’s are diagonal matrices, which implies that $\delta_s = 0$. In the contrary, we have $\delta_s$ is much larger than 1. Intuitively, if the dependence between feature vectors are strong, it is difficult to recovery the true coefficients $x$. Therefore, we expect a weak dependence (or small $\delta_s$). The following theorem tells us that how small $\delta_s$ should be.
Theorem 1. Given any $A$ and $b$. Assume there exists a sparse vector $x^*$ solves

$$Ax^* = b$$

and its sparsity level is bounded by $s$, that is,

$$\|x^*\|_0 \leq s.$$

If the $3s$-Restricted isometry constant defined for the matrix $A \in \mathbb{R}^{n \times p}$ satisfies

$$\delta_{2s} < \sqrt{2} - 1,$$

then the solution to (3) is $x^*$, which also solves (2).

This theorem basically tells us that when features are different enough, we solve use (3) to solve (2) and recover the true value of $x^*$. This condition $\delta_{2s} < \sqrt{2} - 1$ looks sort of reasonable. However, how realistic is it? The following definition tells us how easy (or difficult) to meet it.

Theorem 2. Let $A \in \mathbb{R}^{n \times p}$ be a Gaussian random matrix, that is, all elements of $A$ follow i.i.d. Gaussian distribution. There exists a constant $C$ such that if

$$n \geq C s \log p,$$

then the $2s$-RIC defined for $A$ satisfies

$$\delta_{2s} < \sqrt{2} - 1.$$

with probability

$$1 - \exp\{-n/4\}.$$

This results suggests that if the number of samples is more than the order $O(s \log p)$, the sufficient condition to recover $x^*$ is satisfied with high probability.

5 Noise Case

Sometimes, the observation vector $b$ is disturbed by noise:

$$b = Ax^* + \epsilon$$

where $\epsilon$ is the noise vector. Then the problem is how to recover the sparse signal $x^*$? Three famous formulations: LASSO

$$\text{(LASSO)} \quad \min_x \frac{1}{2} \|Ax - b\|^2 + \lambda \|x\|_1,$$

Dantzig Selector

$$\text{(Dantzig Selector)} \quad \min_x \|x\|_1 \quad \text{s.t.} \quad \|A^T(Ax - b)\|_\infty \leq \lambda,$$
and basis pursuit de-noising (BPDN)

\[
\begin{align*}
\text{(BPDN)} \quad & \min_x \|x\|_1 \\
& \text{s.t. } \|Ax - b\|_2 \leq \varepsilon.
\end{align*}
\]

All of these formulations are convex. LASSO and BPDN can be formulated into quadratic programming (QP), while Dantzig Selector can be formulated into a linear programming (LP). Both QP and LP problems can be solved by using solvers in Matlab or R. However, it may not be the most efficient way using general LP or QP solvers, due to their special structures. We will introduce several optimization algorithms which are more efficient than using mature solvers.

In the noiseless case, we know that a RIC condition \(\delta_{2s} < \sqrt{2} - 1\) can guarantee recover any \(s\)-sparse signal \(x^\ast\) perfectly. In the noise case, do we still have any theoretical guarantees?

**Theorem 3.** Assume that \(x^\ast\) has \(s\) nonzeros, \(\delta_{2s} < \sqrt{2} - 1\) for matrix \(A\) and elements in the noise vector \(\epsilon\) follow from i.i.d. Gaussian distribution \(\mathcal{N}(0, \sigma^2)\). Then we have that for the estimator provided by LASSO or Dantzig Selector (DS) if choose \(\lambda = O(||A^T\epsilon||_\infty)\), then

\[
\|x_{\text{LASSO, DS}} - x^\ast\| \leq O\left(\frac{\sqrt{s} \lambda}{n}\right),
\]

and that for the estimator provided by BPDN, if choose \(\varepsilon = O(||\epsilon||^2)\), then

\[
\|x_{\text{BPDN}} - x^\ast\| \leq O\left(\frac{\sqrt{\varepsilon}}{\sqrt{n}}\right).
\]

holds.

From random matrix theory, if \(A \in \mathbb{R}^{n \times p}\) is a Gaussian random matrix and \(n \geq O(s \log p)\), then with high probability we have

\[
\|x_{\text{LASSO, DS}} - x^\ast\| \leq O\left(\sqrt{\frac{s \log p}{n}} \sigma\right)
\]

and

\[
\|x_{\text{BPDN}} - x^\ast\| \leq O(\sigma).
\]

We can see a nice property for LASSO and DS: if the ratio \(\frac{s \log p}{n}\) converges to zero, the estimate error would vanish. Unfortunately, BPDN does not have this nice property.