CS4786/5786: Machine Learning for Data Science, Spring 2015
2/24/2015: Cheatsheet for Compressed Sensing

---

1 Recap of Random Projections

Basic idea of random projection:

- Fill the $K \times d$ matrix W as follows. For each entry (i, j), flip a fair coin; if heads, enter $+1/\sqrt{K}$, whereas if tails, enter $-1/\sqrt{K}$.
- Projections are obtained as follows: For each $t \in \{1, \ldots, n\}$,

$$\mathbf{y}_t = W\mathbf{x}_t$$

Question: In general can we recover $\mathbf{x}_t$'s based on $\mathbf{y}_t$'s?
Answer: In general, NO. When $n > d$ we have a system of underdetermined linear equations.

2 Sparse Recovery

ℓ_0 (norm) of a vector $\mathbf{x} \in \mathbb{R}^d$ measures "sparsity" of vector $\mathbf{x}$ and is given by

$$\|\mathbf{x}\|_0 = \# \text{ non-zero entries of } \mathbf{x}$$

Examples:

$$\text{Eg. 1: } \mathbf{x} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \|\mathbf{x}\|_0 = 1, \qquad \text{Eg. 2: } \mathbf{x} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -1 \\ 0 \end{bmatrix} \Rightarrow \|\mathbf{x}\|_0 = 2$$

Question: If every row of the data matrix was sparse, then can we recover $\mathbf{x}_t$'s from $\mathbf{y}_t$'s?

Definition 1 (Restricted Isometry Property)**.** *Projection matrix W of size $K \times d$ satisfies (ϵ, s)-RIP property, if for all pairs of $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$ with $\|\mathbf{x}\|_0, \|\mathbf{x}'\|_0 \leq s$,*

$$(1 - \epsilon)\left\|\mathbf{y} - \mathbf{y}'\right\|_2 \leq \left\|\mathbf{x} - \mathbf{x}'\right\|_2 \leq (1 + \epsilon)\left\|\mathbf{y} - \mathbf{y}'\right\|_2$$

where $\mathbf{y} = W\mathbf{x}$ and $\mathbf{y}' = W\mathbf{x}'$ the K dimensional projection

The above property basically says that for sparse $\mathbf{x}$'s, projection with matrix W preserves distances to within factor $1 \pm \epsilon$

If we pick W to be the matrix from the random projections method with $K \approx \frac{s \log d}{\epsilon^2}$, then this matrix satisfies (ϵ, s)-RIP property with high probability.

2.1 Recovering $\mathbf{x}_t$'s

Assume that all $\mathbf{x}_t$'s are at least s sparse, ie. $\|\mathbf{x}_t\|_0 \leq s$.

Method:

$$\tilde{\mathbf{x}}_t = \underset{\mathbf{x}:\mathbf{y}_t = W\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_0 \tag{1}$$

If W is an $(s,\epsilon)-$RIP matrix then $\tilde{\mathbf{x}}_t = \mathbf{x}_t$ for all t's.

We have, $\tilde{\mathbf{y}}_t = W\tilde{\mathbf{x}}_t = \mathbf{y}_t$ Hence $\|\tilde{\mathbf{y}}_t - \mathbf{y}_t\|_2 = 0$. Now note that since $\tilde{\mathbf{x}}_t$ is the solution to optimization problem in Eq. 1, it is at least s sparse (since $\mathbf{x}_t$ itself is s sparse and can be used as a solution if no other solution is sparse enough). However, by RIP condition, since both $\tilde{\mathbf{x}}_t$ and $\mathbf{x}_t$ are s sparse,

$$\|\mathbf{x}_t - \tilde{\mathbf{x}}_t\|_2 \leq (1+\epsilon)\|\mathbf{y}_t - \tilde{\mathbf{y}}_t\|_2 = 0$$

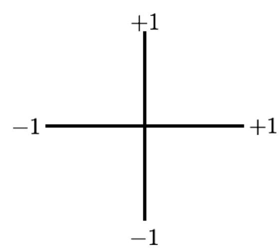
Hence for any t, $\tilde{\mathbf{x}}_t = \mathbf{x}_t$ and so we have perfect recovery.

Solving Eq. 1 is time consuming. Can we do better?

3 ℓ_1 Norm, Sparse Recovery, compressed sensing

It turns out that we can come up with efficient algorithm for sparse recovery. Before we see that this algorithm is, let us see pictorially what the $\|\cdot\|_0$ looks like. In 2 dimensions for instance, if we plot all points $\mathbf{x}$ such that $\|\mathbf{x}\|_0 \leq 1$ and which are such that on any coordinate i, $|\mathbf{x}[i]| \leq 1$, it looks like the figure below.

$$B_0(1) = \left\{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_0 \leq 1, \forall i \leq d, |\mathbf{x}[i]| \leq 1\right\}$$

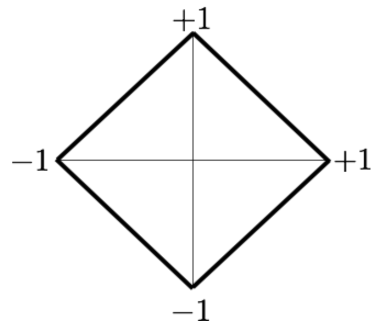


Now this is a non-convex body and hence hard to optimize over. The idea in sparse recovery using what is termed ℓ_1 norm is to replace the $\|\mathbf{x}\|_0$ by the term

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{d} |\mathbf{x}[i]|$$

that is the sum of absolute values of the coordinates of $\mathbf{x}$. The ℓ_1 ball in 2d looks like:

$$B_1(1) = \left\{ \mathbf{x} \in \mathbb{R}^d : \sum_{i=1}^{d} |\mathbf{x}[i]| \leq 1 \right\}$$

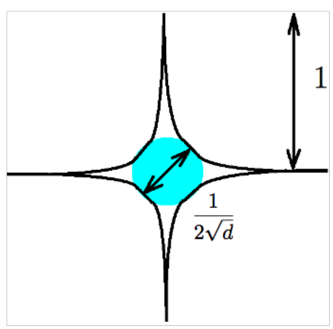


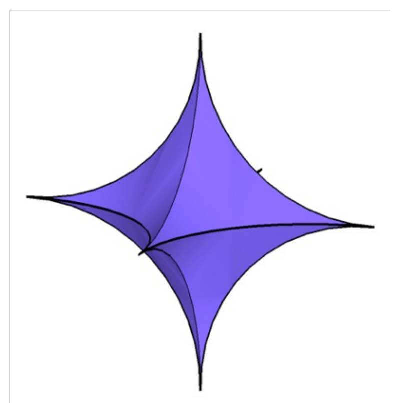
The recovery algorithm is given by

$$\hat{\mathbf{x}}_t = \underset{\mathbf{x}:\mathbf{y}_t = W\mathbf{x}}{\operatorname{argmin}} \ \|\mathbf{x}\|_1 \tag{2}$$

Why does ℓ_1 ball support sparse solutions?

The answer to this is more pronounced when one looks at the ℓ_1 ball in high dimensions. The following figure shows the 2 dimensional and 3 dimensional depiction of the ℓ_1 ball in $\mathbb{R}^d$ when d is large.





The above Figure borrowed from Roman Vershynin's notes. As dimension increases most of the volume is located at the center with "tentacles" protruding out. These tentacles are the reason for sparsity.

If W is an $(s, \epsilon)-$RIP matrix for small enough ϵ, then $\hat{\mathbf{x}}_t = \mathbf{x}_t$ for all t's.

So in effect, the compressed sensing algorithm picks a random $k \times d$ matrix W just as in random projections example. Obtains $\mathbf{y}_t = W\mathbf{x}_t$ just as in random projections. When it comes time to recountruct $\hat{\mathbf{x}}_t$ we solve the optimization problem in Eq. 2