Homework 6

MA430

As always, give full justification for your answers!

1. Read as much of Chapter 4 as I've given you, and anything else I handed out (e.g., the paper I wrote with Deborah Walter).

2. Let
$$\mathbf{A} = \begin{bmatrix} 1 & \sqrt{3}/2 & 0 & -1/\sqrt{2} \\ 0 & 0 & 1 & 1/\sqrt{2} \\ 0 & 1/2 & 0 & 0 \end{bmatrix}$$
Compute the RIP constants $\delta_1, \delta_2, \delta_3$ and δ_4 for $\mathbf{A}$. Hint: δ_1 and δ_4 are easy.

3. Suppose a matrix $\mathbf{A}$ satisfies the RIP of order r with constant δ_r. Show that $\mathbf{A}$ also satisfies the RIP of order q if $q < r$, and that $\delta_q \leq \delta_r$.

4. Use Theorem 4.5.4 in the notes to predict how many data (how many samples "m") are needed if $n = 10^5$, $r = 10$, $\delta = 0.5$, and $\epsilon = 0.5$. (Of course, the Theorem is very pessimistic).

5. A couple years ago some Rose students instrumented a treadmill in the SRC in order to collect "foot strike" data. An accelerometer was attached to the treadmill base and collected periodic acceleration/vibration data, to determine when someone was on the treadmill and if so, how fast they were running.

   Let's say, for simplicity, that the data was collected at 50 samples per second when someone was on the treadmill, and collected in one second windows. The goal is to use these 50 collected data points to determine when any foot strikes occurred in this window. The sensors and other hardware had limited power and capability, so most of the computation to determine when a foot strikes occured was to take place remotely. The sensor simply transmitted the data it collects, with minimal processing. In order to reduce the amount of power/bandwidth needed for transmission, the full 50 data points were not transmitted, but rather a distilled version of the data from which the foot strike times could be computed. And in fact, ideally, we'd never collect these 50 data points, but rather a distilled version of them.

   The students decided to try a compressed sensing approach. A one second window of data is a vector $\mathbf{x} \in \mathbb{R}^{50}$. This vector ought to be approximately sparse—most of the time the vector (acceleration data) is near 0, but when the runner's foot hits the treadmill the data is nonzero for a short period, maybe a few samples.

   So here is a simplified/sanitized version of the problem. We have a sparse vector $\mathbf{x}$ that has, perhaps, 2 nonzero entries (suppose there are never more than 2 foot strikes in a second, and each strike yields only one nonzero entry in $\mathbf{x}$.). Our goal is to design a $m \times 50$ measurement matrix $\mathbf{A}$ so that we can transmit just the vector $\mathbf{d} = \mathbf{A}\mathbf{x}$ (m pieces of information) instead of $\mathbf{x}$ (50 pieces of information), and then the receiver can use $\mathbf{d}$ to reconstruct $\mathbf{x}$, under the assumption that $\mathbf{x}$ is sparse. Of course we want $m << 50$, but large enough so that $\mathbf{x}$ can be reliably recovered.

(a) Suppose we decide to just choose $m = 10$ of the 50 samples to transmite, say samples $x_1, x_6, \ldots, x_{46}$. What matrix $\mathbf{A}$ does this correspond to? Why is this a terrible scheme?

(b) Try a sensing martrix $\mathbf{A}$ with normal random entries. Specifically, try the following experiment: Make up a vector $\mathbf{x} \in \mathbb{R}^{50}$ that is 2-sparse. Matrix up a random 15×50 matrix $\mathbf{A}$ with components that are normal random variables (so we're collecting 15 samples). Synthesis data $\mathbf{d} = \mathbf{A}\mathbf{x}$ and then try to recover $\mathbf{x}$ from $\mathbf{d}$ using OMP or basis pursuit. Repeat this procedure 1000 times (each time changing $\mathbf{A}$). How often do you succeed in exactly reoovering $\mathbf{x}$ with 15 samples?

(c) Repeas the last part but make $\mathbf{A}$ a matrix with entries

$$A_{jk} = \cos(\omega_j k)$$

where ω_j is a randomly chosen number, say drawn from a standard normal distribution.

(d) Repeat with a matrix $\mathbf{A}$ that is a random $0 - 1$ matrix (like in the marble problem), with each entry having a 50/50 change of being a 1 or a 0. (You can do this in Matlab with `A=randi(2,15,50)-1;`.

(e) Does one type of sensing matrix do better? Can you find something superior to these? For a real challenge, can you find a simple deterministic choice for $\mathbf{A}$ that works well?