Sparsity and Optimization
Week 2 Exercise problems

The following are light exercises intended to give you practice using the notions covered in last week's lecture and explore some related ideas. Feel free to try as few or many as you find helpful!

(1) **For fun.** A robot stands on the edge of a cliff. He is programmed to repeatedly take one step forward (toward the cliff) with probability 1/3 and one step backward (away from the cliff) with probability 2/3. Assuming that the robot can continue this process forever and that there is infinite room for him to walk away from the cliff, find the probability he will fall off the cliff.

(2) **For fun.** Consider the following game. There are two pieces of paper face down, each with a number written on it, which you can assume were generated randomly such that the probability the numbers are the same is zero. You may look at one of the numbers, and your goal is then to guess whether the other number is higher or lower than the number you just saw. If you are correct, you win. Give a strategy so that you win with probability 1/2. Give a strategy where you win with probability strictly greater than 1/2 (yes, this is possible!).

(3) Suppose $\mathcal{A} : \mathbb{R}^{n\times n} \to \mathbb{R}^m$ is a linear operator that has no rank-$2r$ (or less) matrices in its null space (other than the zero matrix). Prove that if X is rank-r that the solution to:

$$\hat{X} = \operatorname{argmin} \operatorname{rank}(Z) \quad \text{subject to} \quad \mathcal{A}(Z) = \mathcal{A}(X)$$

is the matrix $\hat{X} = X$.

(4) There is a useful variant of compressed sensing called *one-bit* compressed sensing, where instead of obtaining the measurements $y = Ax$, we obtain only the binary data $y = \operatorname{sign}(Ax)$, where $\operatorname{sign}(w) = -1$ when $w < 0$ and 1 otherwise (and does so entry-wise on vectors). This problem appears in many applications, including some classification problems which we will see more of this week. Explain why it is impossible to reconstruct the norm of x from binary measurements of the form $y = \operatorname{sign}(Ax)$.

(5) Suppose you know that $x \in \mathbb{R}^2$ with $\|x\|_2 = 1$ and $\|x\|_0 = 1$. How many measurements of the form $y_i = \operatorname{sign}(\langle a_i, x\rangle)$ are necessary to identify x? What would the measurements look like?

(6) Suppose you know that $x \in \mathbb{R}^2$ with $\|x\|_0 = 1$ but the norm of x is unknown. Suppose you are now free to not only design measurement vectors a_i but also thresholds ("dithers") τ_i and take measurements of the form $y_i = \operatorname{sign}(\langle a_i, x\rangle - \tau_i)$. How might you design a_i and τ_i if you want to estimate x within an accuracy of ε? How many measurements does your method require? Are they non-adaptive or adaptive (do

your choices of a_i and τ_i depend on previously taken measurements or not)?

(7) **Computation.** Play with some code for low-rank matrix recovery, that we talked about last week. Download the code at `http://www.math.ucla.edu/~deanna/stoGreedy.zip` . Open `demo_LowRankMatrixRecovery.m` in the LowRankMatrixRecovery folder to view the code, and run this script to see results for several levels of rank and measurements m. For example, how many measurements are required for 100% reconstruction of rank-5 matrices using StoIHT? Using IHT?

(8) **Computation.** Consider the 1-bit compressed sensing problem mentione in (2) above. The BIHT algorithm, which appears in (13) and (14) of the paper `https://arxiv.org/pdf/1104.3160.pdf` , is a greedy method to recover x from these measurements. Write a short script that tests this method on sparse signals and their one-bit measurements. (Note: the step size τ can usually be set at $1/(\|A\|)$, but you should try several choices and see how they affect convergence!)

(9) See Lemma 5 of the paper referenced in the problem above. This shows that the update step of BIHT is in the direction of a subgradient of the objective function. Can you imagine a modification of BIHT to use stochastic updates, that on average move in the direction of this subgradient? Test your approach!