EXERCISE SHEET: (MODEL-BASED) COMPRESSED SENSING

JEAN-LUC BOUCHOT

1. Day 1: Introduction, hands-on

1.1. Mathematical topics.

Exercise 1 (Regularization). Let $A \in \mathbb{R}^{m \times N}$ and $\lambda > 0$. What is the solution to

$$\operatorname{argmin}_{\mathbf{z} \in \mathbb{R}^N} \|A\mathbf{z} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{z}\|_2^2.$$

Does your result hold in complex spaces?

Exercise 2. Prove the following equivalences from class.

Let A be an $m \times N$ sensing matrix in $\mathbb{R}^N$. The following statements are equivalent

- Every s-sparse $\mathbf{x}$ is the unique s-sparse solution of $A\mathbf{z} = A\mathbf{x}$.
- $\ker(A) \cap \{\mathbf{x} \in \mathbb{R}^N : \|\mathbf{x}\|_0 \leq 2s\} = \{0\}$.
- Let $S \subset \{1, \cdots, N\}$ be any set with $|S| \leq 2s$. The submatrix A_S is injective.
- Every set of $2s$ columns of A is linearly independent.

Exercise 3. We want to understand the importance of rotations of the space on the null space property. Throughout this exercise, let $A \in \mathbb{R}^{m \times N}$ be a matrix satisfying the null space property of order s and let D_1 and D_2 be two non-singular matrices respectively in $\mathbb{R}^{m \times m}$ and $\mathbb{R}^{N \times N}$.

(1) Show that the matrix $D_1 A$ satisfies the null space property of order s.
(2) Find a counter example such that AD_2 does not satisfy the null space property. ($m = 2$, $N = 3$, $s = 1$)

Exercise 4. Let $A \in \mathbb{R}^{m \times N}$ satisfy the null-space property of order s. Prove directly without using any other theorems from the course, that $\mathbf{x} \in \ker(A)$ and $\|\mathbf{x}\|_0 \leq 2s \Rightarrow \mathbf{x} = 0$.

Exercise 5. Prove Stechking's (tail) estimate: For any $q > p$ and any $\mathbf{x} \in \mathbb{R}^N$,

$$\sigma_s(\mathbf{x})_q \leq s^{1/q - 1/p} \|\mathbf{x}\|_p.$$

1.2. Programming (Python) experiments.

Exercise 6. Program (and test!) the following functions: (adapt the inputs/ outputs to your needs...)

(1) `l2normalize`: a function that takes a matrix as an input, and returns the same matrix with its columns having ℓ_2 norm 1.
(2) `coherence`: which takes a matrix as an input and returns its coherence
(3) `random_matrix`: which takes at least three inputs: `nb_rows`, the number of rows, `nb_cols`, the number of columns, and `rdm_type`, the type of random generation used. Check the documentation for `random.random()` from `NumPy`. This function returns an `nb_rows x nb_cols` matrix whose entries are generated at random according to the distribution `rdm_type`. You may pass an extra parameter to normalize the columns of your matrix or not. `rdm_type` should at least contain *Gaussian*, *Rademacher*, and *Exponential*, but you may (should?) include more options.
(4) `subsample_DFT`: which takes two inputs, `N` the size of the signals, and `Omega` the subset of rows selected from the DFT matrix. It returns a $|\text{Omega}| \times N$ matrix constructed by extracting the rows of an $N \times N$ DFT matrix which are supported on the index set `Omega`.

Exercise 7. (1) For various random matrices, generate graphs of the their coherences with respect to m, the number of rows (fixing the number of columns N). To this end, generate 100 random matrices and compute the average, min, and max of the coherence. (*Hint:* Remember to normalize the columns of the matrix!)

Date: August 5, 2019.

(2) Using the basis pursuit procedure provided, test various settings (in terms of s, m, and N, as well as various random matrices). For 100 s-sparse random vectors, compute the average recovery rate. Let s vary and plot the percentage of recovery with respect to the sparsity s.

(3) Repeat the procedures above with random subsampled Fourier matrices, in which the support `Omega` is generated at random with a given size.

School of Mathematics and Statistics, Beijing Institute of Technology
Email address: `jlbouchot@bit.edu.cn`