Mathematical methods for machine learning and signal processing SS 19

Problem set 3

Veniamin Morgenshtern, Helmut Bölcskei, Erwin Riegler

Due date: 6. June 2019

Problem 1: Gram matrix

Let $\{\mathbf{a}\}_{k=1}^{N}$ be a set of vectors in $\mathbb{C}^M$. Show that the Gram matrix $\{\langle \mathbf{a}_k, \mathbf{a}_l \rangle\}_{k,l=1}^{N}$ is a Hermitian positive-semidefinite matrix, and that it is positive-definite whenever $\{\mathbf{a}\}_{k=1}^{N}$ forms a linearly independent set of vectors.

Problem 2: "l_0-norm"

In the lecture, we defined $\|\mathbf{x}\|_0$ to be the number of entries in the vector $\mathbf{x} \in \mathbb{C}^N$. Show that $\|\cdot\|_0$ is not a norm for $\mathbb{C}^N$, despite the fact that it is often referred to as "l_0-norm".

Problem 3: Fat matrix inversion

Consider a matrix $\mathbf{A} \in \mathbb{R}^{S \times N}$ with $S < N$. Consider a vector $\mathbf{y} \in \mathbb{R}^S$.

1. How many solutions does the equation $\mathbf{A}\mathbf{x} = \mathbf{y}$ have?
2. Consider a subset $\mathcal{S} \subset \{1, \ldots, N\}$ with S elements. Now we are looking for the solutions of the equation $\mathbf{A}\mathbf{x} = \mathbf{y}$ that satisfy the additional constraint: $x_j = 0$ for $j \notin \mathcal{S}$. What are the conditions on the matrix $\mathbf{A}$ so that the equation $\mathbf{A}\mathbf{x} = \mathbf{y}$ has exactly one solution x satisfying this additional constraint?

Problem 4: Compressed sensing

Let $\mathbf{x} \in \mathbb{R}^N$ be a piecewise constant vector with only a small number s of jumps. That is,

$$\mathbf{x} = [\underbrace{\alpha_1 \alpha_1 \ldots \alpha_1}_{\text{block1}} \underbrace{\alpha_2 \alpha_2 \ldots \alpha_2}_{\text{block2}} \ldots \underbrace{\alpha_s \alpha_s \ldots \alpha_s}_{\text{block}s}]^{\mathsf{T}}.$$

Suppose that $\mathbf{x}$ is unknown to us and that we only know the measurement vector

$$\mathbf{y} = \mathbf{A}\mathbf{x} \in \mathbb{R}^M, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{M \times N}$ is a known matrix modeling a linear measurement process. In the case where $M \leq N$, (1) form an underdetermined system of equations. However, compressed sensing theory tells us that it is all the same possible to recover $\mathbf{x}$ under certain conditions.

Explain how you can recover the vector $\mathbf{x}$ from the knowledge of $\mathbf{y}$ and $\mathbf{A}$ only.

Problem 5: P0 Recovery algorithm

This is a computer exercise. The point of the exercise is understand the most basic recovery algorithm $P0$.

Consider the space $\mathbb{C}^M$. Let $\mathbf{e}_k$ denote the k-th column of the $M \times M$ identity matrix $\mathbf{I}_{M\times M}$. Let $\mathbf{f}_k$ denote the k-th column of the $M \times M$ Fourier matrix

$$\mathbf{F} = \frac{1}{\sqrt{M}} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega & \cdots & \omega^{M-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{M-1} & \cdots & \omega^{(M-1)(M-1)} \end{bmatrix}$$

where $\omega = \exp(-2\pi \mathrm{i}/M)$.

Compose a frame for $\mathbb{C}^M$ by combining the ONBs $\mathcal{E}_\mathbf{I}$ and $\mathcal{E}_\mathbf{F}$:

$$\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_{2M}] = [\mathbf{e}_1, \ldots, \mathbf{e}_M, \mathbf{f}_1, \ldots, \mathbf{f}_M].$$

1. Fix s, $1 \le s \le 2M$. Choose a subset of indices $\mathcal{S} \subset \{1, \ldots, 2M\}$ of cardinality $\left|\mathcal{S}\right| = s$ at random. Choose a set of coefficients $\{x_k\}_{k\in\mathcal{S}}$ at random. Generate the signal $\mathbf{y}$ according to

$$\mathbf{y} = \sum_{k\in\mathcal{S}} x_k \mathbf{d}_k.$$

Your goal now is to implement and test algorithms that take $\mathbf{y}$ as an input and try to recover $\mathcal{S}$ and $\{x_k\}_{k\in\mathcal{S}}$ as an output.

2. Implement the $P0$-recovery algorithm discussed in class. Recall that this algorithm searches through all possible subsets $\mathcal{S} \subset \{1, \ldots, 2M\}$ of cardinality s, starting from the smallest $s = 1$ and increasing s step-by-step. For each $\mathcal{S}$ the algorithm searches for a solution $\{x_k\}_{k\in\mathcal{S}}$ of the equation

$$\mathbf{y} = \sum_{k\in\mathcal{S}} x_k \mathbf{d}_k.$$

If the solution exists, the algorithm stops and gives $\mathcal{S}$ and $\{x_k\}_{k\in\mathcal{S}}$ as an output.

3. Experiment with different values of M and s. Do you observe perfect recovery? How does the speed of the algorithm depend on M and on s?

Problem 6: Coherence in sines and spikes

Consider the space $\mathbb{C}^M$. Let $\mathbf{e}_k$ denote the k-th column of the $M \times M$ identity matrix $\mathbf{I}_{M\times M}$. Let $\mathbf{f}_k$ denote the k-th column of the $M \times M$ DFT matrix

$$\mathbf{F} = \frac{1}{\sqrt{M}} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega & \cdots & \omega^{M-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{M-1} & \cdots & \omega^{(M-1)(M-1)} \end{bmatrix}$$

where $\omega = \exp(-2\pi \mathrm{i}/M)$.

Compose a frame for $\mathbb{C}^M$ by combining the ONBs $\mathcal{E}_\mathbf{I}$ and $\mathcal{E}_\mathbf{F}$:

$$\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_{2M}] = [\mathbf{e}_1, \ldots, \mathbf{e}_M, \mathbf{f}_1, \ldots, \mathbf{f}_M].$$

1. Compute the coherence $\mu(\mathbf{D})$.
2. What is the largest sparsity s such that successful recovery via P0 or Basic Pursuit is guaranteed for every vector $\mathbf{x}$ with $\|\mathbf{x}\|_0 \leq s$ from the measurements $\mathbf{y} = \mathbf{D}\mathbf{x}$?

Problem 7: Coherence in super-resolution

Let $\mathbf{F}_{\text{lo}}$ denote low frequency part of the DFT matrix, i.e. the $M \times N$ matrix consisting of the $M = 2f_c + 1$ rows

$$[1 \ \omega^k \ \cdots \ \omega^{k(N-1)}]$$

where $\omega = \exp(-2\pi \mathrm{i}/N)$ for $k = -f_c, \ldots, f_c$.

1. Compute the coherence $\mu(\mathbf{F}_{\text{lo}})$.
2. What is the largest sparsity s such that successful recovery via P0 or Basic Pursuit is guaranteed for every vector $\mathbf{x}$ with $\|\mathbf{x}\|_0 \leq s$ from the measurements $\mathbf{y} = \mathbf{F}_{\text{lo}}\mathbf{x}$?

Problem 8: Super-resolution experiment

This is a computer exercise. Python is the preferred language. Let $N = 128$ be the length of the discrete signal $\mathbf{x}$, and $f_c = 16$ be the cut-off frequency.

1. Generate the $N \times N$ DFT matrix $\mathbf{F}$. Generate a sparse signal $\mathbf{x}$, start with sparsity level $s = 2$, start with nonnegative signal.
2. Compute and plot $\mathbf{F}\mathbf{x}$. Compute and plot $\mathbf{F}^{\mathsf{H}}\mathbf{F}\mathbf{x}$. Is it true that $\mathbf{F}^{\mathsf{H}}\mathbf{F}\mathbf{x} = \mathbf{x}$?
3. Generate $\mathbf{F}_{\text{lo}}$, the matrix of size $2f_c \times N$, corresponding to the low frequencies in $\mathbf{F}$.
4. Compute $\mathbf{y} = \mathbf{F}_{\text{lo}}^{\mathsf{H}}\mathbf{F}_{\text{lo}}\mathbf{x}$. Does the result look like a low-pass version of $\mathbf{x}$?
5. Install cvxpy version higher than 1.0.0. Implement the basis pursuit (l_1 minimization) recovery algorithm in cvxpy.
6. Does cvxpy recover the signal $\mathbf{x}$ perfectly from $\mathbf{y}$?
7. Experiment with higher s. Report your findings.
8. Experiment with signal that has positive and negative elements. Report your findings.