MATH-UA 252/MA-UY 3204 - Fall 2022 - Homework #7

Problem 1. Let $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$, and $b \in \mathbb{R}^m$, where $m < m$. In general, underdetermined linears systems admit either no solutions, or infinitely many solutions. For instance, if our goal is to find some x satisfying $Ax = b$, and we try to solve the least squares problem:

$$\text{minimize} \quad \|Ax - b\|_2^2 \tag{1}$$

we run into trouble with the pseudoinverse $A^\dagger = (A^\top A)^{-1} A^\top$, since $A^\top A$ is rank deficient (why?). The standard way to get around this problem is to *regularize* the problem. As you saw in the midterm, if you instead solve the regularized least squares problem:

$$\text{minimize} \quad \|Ax - b\|_2^2 + \sigma \|x\|_2^2 \tag{2}$$

where $\sigma > 0$.

Download the Nile time series data set `nile.csv`. The first column is the year and the second column is the river flow. Load this data set into numpy and eliminate some values (e.g., assume 10 entries in the middle of the time series are missing). We will try to use a discrete cosine transform (DCT) basis to recover the missing values. Let n be the number of points in the time series, and use the following code:

```
X = scipy.fft.dct(n, norm='ortho')
```

to create an orthogonal DCT matrix. Note that the columns of X will correspond to cosines of progressively higher frequency (you can plot a few to see). Let f be the vector of flow rates. Then $\alpha = X^{-1} f = X^\top f$ gives the coefficients of f in the DCT basis.

Let I be an index vector giving the *non*-missing values of the time series. Then $X_I \in \mathbb{R}^{|I| \times n}$ restricts the columns of the DCT matrix to those known values. The linear system $X_I \alpha = f_I$ is an underdetermined linear system. Try to solve this problem using least squares and regularized least squares. Plot the resulting flow rate versus the year. Does this do a good job of filling in the missing values? Does ℓ_2 regularization help? Explain your findings.

Problem 2. The *basis pursuit* problem is the following minimization problem:

$$\begin{aligned} \text{minimize} \quad & \|x\|_1 \\ \text{subject to} \quad & Ax = b, \end{aligned} \tag{3}$$

where $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$, and $\|x\|_1 = \sum_{i=1}^n |x_i|$. Typically, the linear system $Ax = b$ is *underdetermined*, so that $m < n$. Note that minimizing $\|x\|_1$ promotes *sparsity* in x—there will tend to be more zero entries. Our goal will be to solve:

$$\begin{aligned} \text{minimize} \quad & \|\alpha\|_1 \\ \text{subject to} \quad & X_I \alpha = f_I, \end{aligned} \tag{4}$$

in order to compute a sparse solution to the underdetermined linear system.

1. Prove that the basis pursuit problem (3) is equivalent to the following standard form LP:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^n u_i + \sum_{i=1}^n v_i \\ \text{subject to} \quad & \begin{bmatrix} \Phi_I & -\Phi_I \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = f_I, \\ & u \geq 0 \\ & v \geq 0 \end{aligned} \tag{5}$$

2. Solve this LP using `scipy.optimize.linprog`.

3. Compare your results with what you got in Problem 1.