Math 76 Homework 3: Due August 13.

August 6, 2018

Tools needed or useful for homework

MATLAB CVX: `http://web.cvxr.com/cvx/doc/quickstart.html#an-optimal-trade-off-curve` and `http://web.cvxr.com/cvx/doc/CVX.pdf`. There are a lot of ways to use CVX. One of the goals in this assignment is to see the effect of different regularization parameters λ, so make sure you choose the CVX options that allow you to vary this. You will also want to vary the number of iterations and error tolerance.

Keys for successful homework completion

1. You do **not** need to program your own optimization method from scratch. If you are interested in what is going on "under the hood" you can read section III in *Scientific Computing With Case Studies* by O'Leary.

2. You are allowed to work with others on code development, but each student must submit his/her own homework and do his/her own analysis of the results.

Problem 1

Consider the function on the interval $[-1, 1]$.

$$f(x) = \begin{cases} \frac{1}{2}(-1-x) & x \leq 0 \\ \frac{1}{2}(1-x) & x > 0. \end{cases}$$

1. Determine the Fourier coefficients $\hat{f}_k$ for $f(x)$ using the definition:

$$\hat{f}_k = \frac{1}{2}\int_{-1}^{1} f(x)e^{-ik\pi x}dx,$$

   where i is the imaginary unit defined by $i^2 = -1$. (Hint: you can solve this integral by breaking it up over two continuous intervals and then using integration by parts.)

2. What is the *decay rate* of the Fourier coefficients $\hat{f}_k$? Show this in a graph by plotting k vs. $|\hat{f}_k|$ for $k = -16, \cdots, 16$. Explain how this affects the convergence of the partial sum approximation $S_N f(x)$ given below.

3. Approximate the solution to $f(x)$ by calculating (using MATLAB)

$$S_N f(x) = \sum_{k=-N/2}^{N/2} \hat{f}_k e^{ik\pi x}$$

   on grid points $x_j = -1 + \frac{2j}{N}$, $j = 1, \cdots, N$. (Note that due to periodicity, $f(-1) = f(1)$ and $S_N f(-1) = S_N f(1)$.)

(a) Graph the solution for $N = 16$. Describe what you observe and explain your results.

(b) On one graph, plot x vs. $\log|f(x) - S_N f(x)|$ for $N = 16, 32, 64$. Explain your results.

(c) Repeat (a) and (b) for

$$S_N^{\sigma} f(x) = \sum_{k=-N/2}^{N/2} \sigma_k \hat{f}_k e^{ik\pi x}$$

where $\sigma_k = \frac{1}{2}(1 + \cos(\frac{2k\pi}{N}))$ is the *raised cosine filter*. HINT: Make sure that you are programming your filter correctly by graphing k vs. σ_k.

(d) **BONUS:** Try other filters such as the exponential filter of various orders, Hamming window, and the sharpened raised cosine filter and compare your results.

4. Now consider using the inverse method approach. Assume you are given the N vector of Fourier coefficients (that you calculated above) $\hat{\mathbf{f}}$, where the first element corresponds to $\hat{f}_{-N/2+1}$ and the last corresponds to $\hat{f}_{N/2}$. Using a trapezoidal sum to approximate an integral (you can see a calculus book for an explanation), we have

$$\hat{f}_k = \frac{1}{2}\int_{-1}^{1} f(x)e^{-ik\pi x}dx \approx \frac{1}{N}\sum_{j=1}^{N} f(x_j)e^{-ik\pi x_j},$$

which leads to the system

$$F\mathbf{f} \approx \hat{\mathbf{f}},$$

where $\mathbf{f}$ is the solution f on grid points x_j, $j = 1, \cdots, N$, and F is the discrete Fourier transform matrix with entries

$$F(\tilde{k}, j) = \frac{1}{N}e^{-ik\pi x_j}.$$

Here $\tilde{k} = k + N/2$ for $k = -N/2 + 1, \cdots, N/2$, so that F is defined as an $N \times N$ matrix.

Now solve the inverse problem for $N = 16, 32$, and 64 using:

(a) truncated SVD,

(b) ℓ_1 regularization. where you can write the solution to the problem as

$$\mathbf{f}^* = argmin_f(||F\mathbf{f} - \hat{f}||_2^2 + \lambda||L\mathbf{f}||_1),$$

The solution $\mathbf{f}^*$ is the approximation of f at grid points x_l, $l = 1, \cdots, N$ on $(-1, 1]$. Choose L appropriately for your what f is. Note that $f(x)$ is *not sparse* but $TV(f)$ and $\frac{df}{dx}$ are. Explain how to choose the operator L.

(c) You must also consider at least one of the following scenarios:

- The given data are noisy. In this case add complex Gaussian noise of mean zero to each calculated Fourier coefficient, so we instead have $\hat{g}_k = \hat{f}_k + \epsilon_k$, and the system we solve is $\mathbf{Ff} = \hat{\mathbf{g}}$.
- The Fourier data are blurry. In this case, calculate $\hat{\mathbf{f}}$ as above and then multiply by a blurring matrix $\mathbf{K}$ so that the right hand side of $\mathbf{Ff} = \hat{\mathbf{g}}$ is $\hat{\mathbf{g}} = \mathbf{K}\hat{\mathbf{f}}$. Note that the system $\mathbf{F}$ does *not* account for the blur (i.e. it is unknown, just like noise). You can also try $\hat{\mathbf{g}} = \mathbf{K}\hat{\mathbf{f}} + \epsilon$, where ϵ is a vector of complex Gaussian noise of mean zero.
- The system is very underdetermined. In this case we have fewer measurements of $\hat{\mathbf{f}}$ then needed, so we need the regularization to choose the best solution. One way to program this is to generate a *row selector matrix* $\mathbf{M}$ that zeros out $N - m$ rows (where N is the number of columns and m is the number of rows). See how little data you can get away with. Note that $\mathbf{M}$ does not need to have all of its zero rows at the bottom. You are more interested in doing some kind of sparse (random) sampling throughout the input data domain.

(d) Repeat 3(a) and 3(b) for each experiment. Clearly explain your results and compare the outcomes.

Problem 2

Let $\mathbf{x} \in \mathbb{R}^n$ be an unknown random vector and $\mathbf{y} \in \mathbb{R}^m$ be defined as

$$\mathbf{y} = \mathbf{E}\mathbf{A}\mathbf{x},$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a known transform operator and $\mathbf{E} \in \mathbb{R}^{m \times m}$ is multiplicative exponentially distributed random noise independent of $\mathbf{X}$, where the probability density function is given by

$$P_{\mathbf{E}}(e) = \lambda exp(-\lambda e), \qquad \lambda > 0.$$

Write the MAP estimate of $\mathbf{x}$ given an entropy prior.