Math 76 Homework 1: Due July 16.

July 1, 2018

Tools needed or useful for homework

1. **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.
2. **Challenge Files**: These are available by following the link `www.siam.org/books/fa03`. They correspond to the challenges in *Deblurring Images: Matrices, Spectra, and Filtering*, by Hansen et. al.
3. **Relevant software:** There is also some helpful code and information at `http://www.cs.umd.edu/~oleary/SCCS/cs_deblur/index.html`. Many of the codes there will be useful in understanding how to write efficient code.
4. **MATLAB imaging toolbox**: You can download different images and blur functions. The documentation will show you how to convert images to index files and back again.

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.
3. Do not use very large data sets, as the cvx program is not well suited for this. Eventually if you want to use a larger image, you are free to find a better software package. CVX will be fine for most one-dimensional problems.
4. You will have a lot of flexibility in how you choose the problem, so do not expect your results to match anyone else's. Have fun and try different scenarios to fully appreciate the consequences of changing the data.

Practice Problems

Do Challenges 1 - 4 in *Deblurring Images: Matrices, Spectra, and Filtering.* This should not be turned in with your homework, but it will help you to see how parameters are chosen for different problems.

Problem set up

Let f be a piecewise continuous function on an interval $[a, b]$. You can also choose f to be a sparse signal. (For simplicity you may want to choose $[-1, 1]$ as this will help you to re-use your code for the next assignment.)

- For example choose f to be piecewise constant, piecewise linear, piecewise polynomial, or a piecewise trigonometric polynomial.
- Since f is *piecewise continuous*, there should be a few points of discontinuity in the function. For example, you might choose
$$f(x) = \begin{cases} -\frac{x+1}{2} & x \leq 0 \\ -\frac{x-1}{2} & x > 0. \end{cases}$$
For this $f(x)$ there is one discontinuity. *Use your imagination.*
- Suppose the function is blurred, that is $g(x) = (K * f)(x)$, where K is a blurring kernel, and that our measuring device gives us the noisy blurred data $\tilde{g} = \mathbf{g} + \eta$ at a discrete set of grid points, $x_j = a + (b-a)(j-1)/(m-1)$, $j = 1, \cdots, m$. Here η is random Gaussian noise.
- We now have the discrete inverse problem to solve:
$$P\mathbf{f} = \mathbf{g},$$
where P is the corresponding blurring matrix (that is, the blurring kernel is converted into a linear system, see page 83 in O'Leary's book for an explanation). You should choose a blurring matrix such as one provided in *Deblurring Images: Matrices, Spectra, and Filtering* and different levels of noise.
- You will need to choose m, the amount of data, that ensures convergence. You should do examples with $n = \alpha m$, where $\alpha < 1$ for an overdetermined system and $\alpha > 1$ for an underdetermined system.
- You should use the following regularizations:
  1. truncated SVD,
  2. ℓ_2 regularization (this is Tikhonov regularization when $L = I$),
  3. ℓ_1 regularization.

  In all cases, you can write the solution to the problem as
$$\mathbf{f}^* = argmin_f(||P\mathbf{f} - \tilde{g}||_2^2 + \lambda||L\mathbf{f}||_p^p),$$
where we choose either $p = 1$ or $p = 2$. The solution $\mathbf{f}^*$ is the approximation of f at grid points x_l, $l = 1, \cdots, n$ on $[a, b]$. (Do you understand why?)
- Choose L appropriately for your what f is. Explain.
- You can also consider the following scenarios:
  - The system is ill conditioned: The blurring matrix is $n \times n$ but has at least one singular value that is close to zero. Even when there is no noise ($\eta = \mathbf{0}$) it may be difficult to recover the underlying function f.
  - The system has a lot of noise: Even when the matrix is not ill conditioned, if there is a lot of noise it can be very difficult to recover the underlying function f.
  - The system is very underdetermined: In this case we have fewer measurements of $\mathbf{g}$ 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 sampling throughout the input data domain.

- Analyze your results. You should prepare a well written report, with a clear explanation of how you set the problem up, what paramters you chose, and how you varied your experiments. You should use pictures to illustrate your comments. Your write up *must* do the following:

  1. General comparison of different types of regularization for these problems. When does one regularization appear to be superior to the others?
  2. Discuss how robust each method is with respect to the regularization parameter λ. That is, do the results vary wildly for different choices of regularization parameters?
  3. Discuss how much data are needed in order to achieve meaningful results.
  4. Discuss how each method responds to added noise and/or blur.
  5. Discuss the efficiency of the method. Note that this is a bit out of your control because you are using a canned software package. But you should be able to remark if the algorithm took seconds, minutes, or hours to compute.

Shameless plug: These problems are regularly solved in imaging and signal processing. The methods are becoming part of the hardware in MRI and other types of medical imaging devices, and are becoming more common in areas such as astronomy and seismology. There are several extensions to this homework that you can consider for your term project.