CS 3220 Spring 2010
Homework 8

Problem 1: Rank deficient least squares.

The blurring of images taken by shaky handheld cameras is a familiar problem that can be very nicely modeled as a linear transformation: each point in the ideal unblurred image, instead of being imaged as a point, is imaged as a little streak because it was moving during the exposure. A useful computational model is that the blurred image is a sum of a lot of streaks, one for each pixel in the unblurred image we would have recorded if the camera hadn't been moving. The streaks are all the same shape, and each streak is shifted to the position of the relevant pixel, and its intensity is scaled by the pixel's intensity. Because the streak is the image you'd get by photographing a point source with your shaky camera, it is called the "point spread function" or PSF.

If you know the PSF exactly (that is, you know the shape of the streak—for instance, if your camera contains an accelerometer system that can measure the camera's motion), then de-blurring the image is a linear system. For very small images this system can be explicitly formed and solved. (For practical cases, more sophisticated means are required to avoid using too much memory.)

This problem comes with a Matlab function blur_img that will take a PSF and an image (stretched out into a column vector) and blur it. If you give it a matrix, it will blur all the columns of the matrix for you. You can build the matrix transformation $\mathbf{M}_{\text{blur}}$ that computes a blurred image $\mathbf{i}_{\text{blurred}}$ from an unblurred image $\mathbf{i}_{\text{unblurred}}$:

$$\mathbf{i}_{\text{blurred}} = \mathbf{M}_{\text{blur}} \mathbf{i}_{\text{unblurred}}$$

by blurring the identity matrix (why does this work?). It also comes with a function showimgs, which shows images side by side for comparison, and a file blurring.mat that contains a 32 by 32 image of some trees and two PSFs: one that produces an out-of-focus-like blur and one that produces more of a camera-shake type of blur.

1. Compute the blurring matrices for each of the two PSFs. Plot their singular values together on one plot using semilogy. What is the condition number of each of these matrices?

2. Blur the image with each of the two PSFs, compare the results with one another, and then solve the linear systems to recover the original image from each blurred image. What is the maximum error in these reconstructions?

3. Now add a small amount of Gaussian random noise to each blurred image. Start by adding noise with standard deviation 1×10^{-4}, which you can do by adding `1e-4 * randn(size(`*vector*`))`. How does this affect your ability to recover the original image in each case? What are the maximum errors? How do you explain the difference?

4. Now let's work with a larger amount of noise: use standard deviation 5×10^{-3}. Is this much noise visible when you show the noisy blurred images on the screen? Try to recover the original image by treating the blurring matrix as rank-deficient and using SVD to get a least-squares solution. For what rank do you achieve the best result for each PSF? Which is better? How do you explain the difference?

Problem 2: Factoring low-rank matrices.

Suppose $k < n < m$ and we have a matrix $\mathbf{C}$ that we know to be the product of two unknown matrices $\mathbf{A} \in \mathbb{R}^{m \times k}$ and $\mathbf{B} \in \mathbb{R}^{k \times n}$.

1. What is rank($\mathbf{C}$)?

Now suppose we have two matrices $\mathbf{A}'$ and $\mathbf{B}'$, of the same sizes as $\mathbf{A}$ and $\mathbf{B}$, whose product is also $\mathbf{C}$.

2. What can you say about the relationships between range($\mathbf{A}$), range($\mathbf{B}$), range($\mathbf{C}$), range($\mathbf{A}^T$), range($\mathbf{B}^T$), and range($\mathbf{C}^T$)?

3. Whatever $\mathbf{A}$ is, it has to be $\mathbf{A}'\mathbf{X}$ for some $\mathbf{X}$. Why is this? If $\mathbf{A} = \mathbf{A}'\mathbf{X}$, then what is $\mathbf{B}$?

4. Suppose we have the matrix $\mathbf{C}$ and want to recover $\mathbf{A}$ and $\mathbf{B}$. Give a procedure using the SVD to find two matrices $\mathbf{A}'$ and $\mathbf{B}'$ which may not be $\mathbf{A}$ and $\mathbf{B}$ but whose product is $\mathbf{C}$.

5. The provided data file contains a matrix $\mathbf{C}$, but with some noise added. Estimate the two factors assuming $k = 2$, and make a picture by plotting the rows of $\mathbf{A}$ as a bunch of 2D points. Choose a value for $\mathbf{X}$ that makes the resulting image look like it was probably intended to look, and explain how you chose it.