ECE-497/BME-491: Applied Biomedical Signal Processing
Laptop Day #11
Due at the end of class, February 6, 2007

Today we have the following goals

- looking at the effects of noise on inverse solutions
- looking at the use of zero order Tikhonov regularization
- looking at the use of truncated singular value decomposition (TSVD) to improve the estimate

Review of TSVD and Zero Order Tikhonov Regularization

For both of these methods we are looking at the problem

$$\underline{y} \;=\; \mathbf{Z}\underline{x}$$

where $\underline{y}$ is the measured output of a system and we want to estimate $\underline{x}$ based on this measurement. Using the singular value decomposition (SVD) , $\mathbf{Z}$ can be written as

$$\mathbf{Z} \;=\; \mathbf{U}\mathbf{S}\mathbf{V}^T$$

Here both $\mathbf{U}$ and $\mathbf{V}$ are *unitary matrices*

$$\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} \;=\; \mathbf{I}$$

and $\mathbf{S}$ is a diagonal matrix with *singular values* σ_i on the diagonals. The σ_i are arranged so they decrease as i increases. We can solve our problem by assuming

$$\begin{aligned} \underline{y} &= \mathbf{U}\underline{\beta} \\ \underline{x} &= \mathbf{V}\underline{\alpha} \end{aligned}$$

where

$$\begin{aligned} \underline{\beta} &= \mathbf{U}^T\underline{y} \\ \underline{\alpha} &= \mathbf{S}^{-1}\underline{\beta} \end{aligned}$$

This method works well as long as there is little noise in the measurement and the singular values are not very small. However, if there is significant noise or there are small singular value then we should use *truncated singular value decomposition* (TSVD). For TSVD we just truncate modes with small singular values. This is equivalent to setting the corresponding elements of $\underline{\alpha}$ to zero.

An alternative approach, *zero order Tikhonov regularization* approaches the problem by finding the value of $\underline{x}$ to solve the problem

$$\mathbf{\Pi} \;=\; ||\underline{y} - \mathbf{Z}\underline{x}||^2 + \mu||\underline{x}||^2$$

The first term indicates we want to match the data, while the second term indicates we want the magnitude of $\underline{x}$ to be small. The *regularization* parameter μ indicates the relative weight between the two terms. Usually μ is fairly small (less than 1). In class we showed the solution to the problem was given by

$$\alpha_i = \frac{\sigma_i \beta_i}{\sigma_i^2 + \mu}$$

Not that usually of the terms are removed and if $\mu = 0$ we get the same result as a regular least squares solution.

1) Go to the class website and download *laptop11.m*. The code has three arguments

- the signal to noise ratio (in dB)
- the regularization parameter μ
- the number of terms to use in the singular value decomposition solution

2) Modify the code to implement zero order Tikhonov regularization. Your code should mimic the code before it in that it prints out the estimated X value and the projected Y values. Use the same method of printing out the results.

3) Initially assume a SNR of 100 and use a value of $\mu = 0.1$. You should get fairly good estimates of X.

4) Implement truncated singular value decomposition, using the number of modes passed to the program. You should be able to truncate all of the modes except the first mode and your program should run. Your code should mimic the code before it in that it prints out the estimated X value and the projected Y values. Use the same method of printing out the results.

5) Run your code and compare the results using Tikhonov regularization and TSVD. Vary the SNR and number of modes you keep in the TSVD. Which method works better for large SNR's, which works better for small SNRs? Play around until you think you understand what is going on. Run your programs for SNR values of 100 dB, 50 dB, 20 dB, 10 dB, and 5 dB, using values of μ and N that give "good" results at each level. When you think you have good results at one SNR level, comment out the line **fid = 1** so the results will be written to a file. (Recomment when you want to play around again.) Print out your results files as you go (or rename them and then print them out). Look at the results with no regularization and the results with regularization. Some things to note:

- Look at how the measured data changes as we change the SNR. You should notice the change does not appear that large.
- When we compute $\underline{x}$ using the noisy data it gets very bad as the SNR gets large
- If we project the" bad" estimate of $\underline{x}$ forward, we do, in fact get the correct $\underline{y}$
- When we use either TSVD or Tikhonov regularization to estimate $\underline{x}$ and project the solution forward, we do not get the original $\underline{y}$ back, but we are usually fairly close.

Turn in your results files. *Note that the program uses different values of noise for each run. Do not change this, since you are not supposed to optimize for a particular realization of noise!*

6) Uncomment the code to let you run the second experiment, and repeat part 5 again.

Turn in your code.