CS/ECE/ME 532
Homework 5: The SVD and Least Squares

1. Recall the face emotion classification problem from HW 3. Design and compare the performances of the classifiers proposed in **a** and **b**, below. In each case, divide the dataset into 8 equal sized subsets (e.g., examples $1-16$, $17-32$, etc). Use 6 sets of the data to estimate $\boldsymbol{x}$ for each choice of the *regularization parameter*, select the best value for the regularization parameter by estimating the error on one of the two remaining sets of data, and finally use the $\boldsymbol{x}$ corresponding to the best value of the regularization parameter to predict the labels of the remaining "hold-out" set. Compute the number of mistakes made on this hold-out set and divide that number by 16 (the size of the set) to estimate the error rate. Repeat this process 56 times (for the 8×7 different choices of the sets used to select the regularization parameter and estimate the error rate) and average the error rates to obtain a final estimate.

a. Truncated SVD solution. Use the pseudo-inverse $\boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T$, where $\boldsymbol{\Sigma}^{-1}$ is computed by inverting the k largest singular values and setting others to zero. Here, k is the regularization parameter and it takes values $k = 1, 2, \ldots, 9$; i.e., compute 9 different solutions, $\widehat{\boldsymbol{x}}_k$.

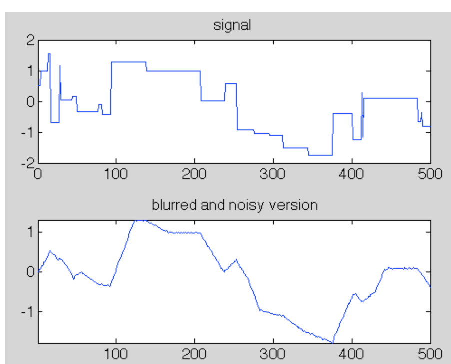
b. Regularized LS. Let $\widehat{\boldsymbol{x}}_\lambda = \arg\max_{\boldsymbol{x}} \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_2^2$, for the following values of the regularization parameter $\lambda = 0, 2^{-1}, 2^0, 2^1, 2^2, 2^3$, and 2^4. Show that $\widehat{\boldsymbol{x}}_\lambda$ can be computed using the SVD and use this fact in your code.

c. Use the original dataset to generate 3 new features for each face, as follows. Take the 3 new features to be random linear combination of the original 9 features. This can be done with the Matlab command `A*randn(9,3)` and augmenting the original matrix $\boldsymbol{A}$ with the resulting 3 columns. Will these new features be helpful for classification? Why or why not? Repeat the experiments in (a) and (b) above using the 12 features.

2. Many sensing and imaging systems produce signals that may be slightly distorted or blurred (e.g., an out-of-focus camera). In such situations, algorithms are needed to deblur the data to obtain a more accurate estimate of the true signal. The Matlab code `blurring.m` generates a random signal and a blurred and noisy version of it, similar to the example shown below. The code simulates this equation:

$$\boldsymbol{b} \;=\; \boldsymbol{A}\boldsymbol{x} \;+\; \boldsymbol{e}\,,$$

where $\boldsymbol{b}$ is the blurred and noisy signal, $\boldsymbol{A}$ is a matrix that performs the blurring operation, $\boldsymbol{x}$ is the true signal, and $\boldsymbol{e}$ is a vector of errors/noise. The goal is to estimate $\boldsymbol{x}$ using $\boldsymbol{b}$ and $\boldsymbol{A}$.



a. Implement the standard LS, truncated SVD, and regularized LS methods for this problem.

b. Experiment with different averaging functions (i.e., different values of k in the code) and with different noise levels (σ in the code). How do the blurring and noise level affect the value of the regularization parameters that produce the best estimates?