CS/ECE/ME 532
Homework 4

1. Tikhonov regularization. Sometimes we have competing objectives. For example, we want to find a $\boldsymbol{w}$ that minimizes $\|\boldsymbol{y} - X\boldsymbol{w}\|_2$ (least-squares), but we also want the weights $\boldsymbol{w}$ to be small. One way to achieve a compromise is to solve the following problem:

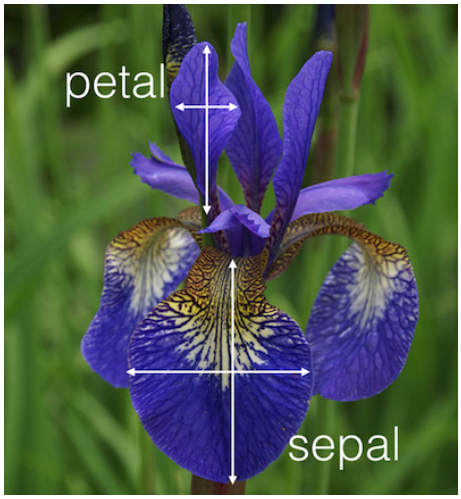
$$\text{minimize} \quad \|\boldsymbol{y} - X\boldsymbol{w}\|_2^2 + \lambda \|\boldsymbol{w}\|_2^2 \tag{1}$$

where $\lambda > 0$ is a parameter we choose that determines the relative weight we want to assign to each objective. This is called *Tikhonov regularization* (also known as L_2 regularization).

a) Solve the optimization problem (1) by finding an expression for the minimizer $\hat{\boldsymbol{w}}$.
Hint: one approach is to reformulate (1) as a modified least-squares problem with different "X" and "$\boldsymbol{y}$" matrices. Another approach is to use the vector derivative method we saw in class.

b) Suppose that $X \in \mathbb{R}^{n \times p}$, with $n < p$. Is there a unique least squares solution? Is there a unique solution to (1) ? Explain your answers.

2. In 1936 Ronald Fisher published a famous paper on classification titled "The use of multiple measurements in taxonomic problems." In the paper, Fisher study the problem of classifying iris flowers based on measurements of the sepal and petal widths and lengths, depicted in the image below.



Fisher's dataset is available in Matlab (`fisheriris.mat`) and is widely available on the web (e.g., Wikipedia). The dataset consists of 50 examples of three types of iris flowers. The sepal and petal measurements can be used to classify the examples into the three types of flowers.

a) Formulate the classification task as a least squares problem. Least squares will produce real-valued predictions, not discrete labels or categories. What might you do to address this issue?

b) Write a Matlab or Python program to "train" a classifier using LS based on 40 labeled examples of each of the three flower types, and then test the performance of your classifier using the remaining 10 examples from each type. Repeat this with many different randomly chosen subsets of training and test. What is the average test error (number of mistakes divided by 30)?

c) Experiment with even smaller sized training sets. Clearly we need at least one training example from each type of flower. Make a plot of average test error as a function of training set size.

d) Now design a classifier using only the first three measurements (sepal length, sepal width, and petal length). What is the average test error in this case?

e) Use a 3d scatter plot to visualize the measurements in (d). Can you find a 2-dimensional subspace that the data approximately lie in? You can do this by rotating the plot and looking for plane that approximately contains the data points.

f) Use this subspace to find a 2-dimensional classification rule. What is the average test error in this case?

3. In this problem you will work with an analyze the dasaset `jesterdata.mat`, which is available on the moodle site. The dataset contains an $m = 100$ by $n = 7200$ dimensional matrix X. Each row of X corresponds to a joke, and each column corresponds to a user. Each of the users rated the quality of each joke on a scale of $[-10, 10]$.

a) Suppose that you work for a company that makes joke recommendations to customers. You are given a large dataset X of jokes and ratings. It contains n reviews for each of m jokes. The reviews were generated by n users who represent a diverse set of tastes. Each reviewer rated every movie on a scale of $[-10, 10]$. A new customer has rated $k = 25$ of the jokes, and the goal is to predict another joke that the customer will like based on her k ratings. Use the first $n = 20$ columns of X for this prediction problem (so that the problem is overdetermined). Her ratings are contained in the file `newuser.mat`, also on moodle, in a vector b. The jokes she didn't rate are indicated by a (false) score of -99. Compare your predictions to her complete set of ratings, contained in the vector `trueb`. Her actual favorite joke was number 29. Does it seem like your predictor is working well?

b) Repeat the prediction problem above, but this time use the entire X matrix. Note that now the problem is underdetermined. Explain how you will solve this prediction problem and apply it to the data. Does it seem like your predictor is working? How does it compare to the first method based on only 20 users?

c) Propose a method for finding one other user that seems to give the best predictions for the new user. How well does this approach perform? Now try to find the best two users to predict the new user.

d) Use the Matlab function `svd` with the 'economy size' option to compute the SVD of

$X = U\Sigma V^T$. Plot the sprectrum of X. What is the rank of X? How many dimensions seem important? What does this tell us about the jokes and users?

e) Visualize the dataset by projecting the columns and rows on to the first three principle component directions. Use the `rotate` tool in the Matlab plot to get different views of the three dimensional projections. Discuss the structure of the projections and what it might tell us about the jokes and users.

f) One easy way to compute the first principle component for large datasets like this is the so-called power method (see `http://en.wikipedia.org/wiki/Power_iteration`). Explain the power method and why it works. Write your own code to implement the power method in Matlab and use it to compute the first column of U and V in the SVD of X. Does it produce the same result as Matlab's built-in `svd` function?

g) The power method is based on an initial starting vector. Give one example of a starting vector for which the power method will fail to find the first left and right singular vectors in this problem.