GEORGIA INSTITUTE OF TECHNOLOGY

SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

ECE 6254 Spring 2018

Problem Set #5

Assigned: 4 Apr 18
Due Date: 18 Apr 18

Suggested reading:

- *Elements of Statistical Learning* (by Hastie, Tibshirani, and Friedman): Section 3.2 (pages 44–56) discusses least squares regression; Section 3.4 (pages 61–79) covers ridge regression and the Lasso; Sections 12.3.6–12.3.7 (pages 434–437) provide an introduction to kernel methods in regression.
- *Learning from Data* (by Abu-Mostafa, Magdon-Ismail, Lin): Section 3.2 (pages 82–88) contains an alternative introduction to linear regression.

Homework is due in class at the beginning of the class period on the due date.

For **distance learning students**, the homework should be scanned and uploaded to the t-square assignments page **one week after the regular due date**.

PROBLEM 5.1:

In this problem I'd like you to use the following code to generate a dataset to evaluate various approaches to regression in the presence of outliers.

```
import numpy as np
np.random.seed(2018)
n = 100
xtrain = np.random.rand(n)
ytrain = 0.25 + 0.5*xtrain + np.sqrt(0.1)*np.random.randn(n)
idx = np.random.randint(0,100,10)
ytrain[idx] = ytrain[idx] + np.random.randn(10)
```

The code above generates training data by selecting random values for the x_i's, then computing $f(x) = \frac{1}{4} + \frac{1}{2}x$ and adding a small amount of Gaussian noise to each observation. It then follows by creating some "outliers" in the y_i's by picking 10 random entries and adding a much larger amount of noise to just those elements.

In the problems below, you should find a linear fit to this data. In all of the methods below, there will be one or more parameters to set. You can do this manually using whatever approach you like. (Do not go crazy optimizing these, just tune the parameters until your estimate looks reasonable.)

(a) To begin, find a linear fit using the code for ridge regression that you produced in the Boston House Prices problem from the last problem set. Report the value of λ that you selected, and report the slope and intercept of your linear fit. Submit your code.

(b) Next, I would like you to find a linear fit using the Huber loss. This can be done via

```
from sklearn import linear_model
reg = linear_model.HuberRegressor(epsilon = 1.35, alpha=0.001)
reg.fit(xtrain.reshape(-1,1),ytrain)
```

You have two parameters to choose here: ϵ (which controls the shape of the loss function and needs to be greater than 1.0) and α (the regularization parameter). Report the values of ϵ and α you selected, and report the slope and intercept of your linear fit (see `reg.intercept_` and `reg.coef_`). Y Submit your code.

(c) Finally, I would like you to find a linear fit using support vector regression. This can be done via

```
from sklearn.svm import SVR
reg = SVR(C=1.0, epsilon=0.1, kernel='linear')
reg.fit(xtrain.reshape(-1,1),ytrain)
```

You have two parameters to choose here: C and ϵ. Report the values of C and ϵ you selected, and report the slope and intercept of your linear fit. Submit your code.

PROBLEM 5.2:
In this problem we consider the scenario described in Lecture 13, where x is drawn uniformly on $[-1, 1]$ and $y = \sin(\pi x)$ and we are again given $n = 2$ training samples. Here we will consider an alternative approach to fitting a line to the data based on Tikhonov regularization. Specifically, we let

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \qquad \boldsymbol{A} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} b \\ a \end{bmatrix}.$$

We will then consider Tikhonov regularized least squares estimators of the form

$$\widehat{\boldsymbol{\theta}} = (\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{A}^T\boldsymbol{y}. \tag{1}$$

(a) How should we set $\boldsymbol{\Gamma}$ to reduce this estimator to fitting a constant function (i.e., finding an $h(x)$ of the form $h(x) = b$)? (Hint: For the purposes of this problem, it is sufficient to set $\boldsymbol{\Gamma}$ in a way that just makes $a \approx 0$. To make $a = 0$ exactly requires setting $\boldsymbol{\Gamma}$ in a way that makes the matrix $\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{\Gamma}^T\boldsymbol{\Gamma}$ singular — but note that this does not mean that the regularized least-squares optimization problem cannot be solved, you must just use a different formula than the one given in (1).)

(b) How should we set $\boldsymbol{\Gamma}$ to reduce this estimator to fitting a line of the form $h(x) = ax + b$ that passes through the observed data points (x_1, y_1) and (x_2, y_2)?

(c) (Optional) Play around and see if you can find a (diagonal) matrix $\boldsymbol{\Gamma}$ that results in a smaller risk than either of the two approaches we discussed in class. You will need to do this numerically using Python or MATLAB. Report the $\boldsymbol{\Gamma}$ that gives you the best results. (You can restrict your search to diagonal $\boldsymbol{\Gamma}$ to simplify this.)

PROBLEM 5.3:

As described in class, K-means clustering with the Euclidean distance inherently assumes that each pair of clusters is linearly separable. This of course may not be the case in practice. In this problem you will derive a strategy for dealing with this limitation that we did not discuss in class. Specifically, you will show that like so many other algorithms we have discussed in class, K-means can be "kernelized."

(a) Let z_{ij} be an indicator that is equal to 1 if $\boldsymbol{x}_i$ is currently assigned to the j^{th} cluster and 0 otherwise. Show that the rule for updating the j^{th} cluster center $\boldsymbol{m}_j$ given this cluster assignment can be expressed as

$$\boldsymbol{m}_j = \sum_{i=1}^{n} \alpha_{ij} \boldsymbol{x}_i.$$

Specifically, express the α_{ij} in terms of the z_{ij}.

(b) Given two data points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, show that $\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2$ can be computed using only (linear combinations of) inner products.

(c) Given the results of the previous parts, show how to compute $\|\boldsymbol{x}_i - \boldsymbol{m}_j\|_2^2$ using only (linear combinations of) inner products between the data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.

(d) Describe how to use the results from the previous parts to "kernelize" the K-means clustering algorithm described in class.

PROBLEM 5.4:

The final exam is coming up in just a few weeks. The exam will be comprehensive in covering material from the beginning of the course up through to the last day of class. In this problem I would like you to think about what might make a good exam question. Write out your question and then answer it. If you write a good one, you just might see it on the final exam!

For the purposes of this problem, write your question to be of the "short answer" variety. Since it can be difficult to craft open-ended questions, your question should be in the form of a multiple choice question with 2-5 possible answers.

Please submit this question on its own sheet of paper, as it will be split out from the rest of the assignment.