Homework 4

Due May 3, 2019 by 11:59pm

Instructions: Upload your answers to the questions below to Canvas. Submit the answers to the questions in a PDF file and your code in a (single) separate file. Be sure to comment your code to indicate which lines of your code correspond to which question part. There is 1 reading assignment and 4 exercises in this homework, including 1 exercise to get you started with the data competition and 1 optional exercise.

Reading Assignment

Read Ch. 10, Sec. 2 in *An Introduction to Statistical Learning* and Ch. 14, Sec. 5.1 in *The Elements of Statistical Learning*.

1 Exercise 1

In this exercise, you will implement by hand, i.e., **with pen and paper** and a five-function calculator, PCA on a simple dataset. There is no coding involved in this exercise. Please report all your derivations. Below by plotting x_1 we mean drawing by hand of x_1 in the Cartesian plane
Consider the dataset with points $x_1 = (-10, 10), x_2 = (12, -8), x_3 = (6, -14), x_4 = (-4, 4)$.

(a) Plot the data.

(b) Recall that PCA projects the data onto a subspace that is "closest" to the observations. Before calculating anything, draw a line on the plot where you think the line (i.e., shifted 1-D subspace) is that is closest to the points. Estimate the slope and intercept of the line.

(c) Center the data. For the centered data report the mean of each column (i.e., feature). Here we will assume that all variables are on the same scale, so we will not standardize them.

(d) Compute the empirical covariance matrix.

(e) Compute the eigenvalues λ_1, λ_2 of the empirical covariance matrix.

(f) Compute the eigenvectors v_1, v_2 of the empirical covariance matrix.

(g) Compute the PCA projections of the points onto a 1-dimensional space (a line). Denote the projected points by x'_1, x'_2, x'_3, x'_4.

(h) Plot x'_1, x'_2, x'_3, x'_4.

(i) Transform the projected points back to the original space using the top principal component and the mean vector from part (b). Denote these points by $\hat{x}_1$, $\hat{x}_2$, $\hat{x}_3$, $\hat{x}_4$.

(j) Plot the following quantities: The original points x_1, x_2, x_3, x_4, the principal components v_1, v_2 shifted by the mean μ of the original data, and the reconstructed points $\hat{x}_1$, $\hat{x}_2$, $\hat{x}_3$, $\hat{x}_4$. Does your plot look reasonable?

2 Exercise 2

In this problem you will generate simulated data and then perform PCA on the data. You will use *your own normalized Oja algorithm* for PCA. Note that "first two principal component score vectors" refers to the results from projecting the original data to a two-dimensional space with PCA.

(a) Generate a simulated data set with 30 observations in each of three classes (i.e. 90 observations total), and 50 features. Hint: There are a number of functions in numpy that you can use to generate data. One example is the `numpy.random.normal()` function; `numpy.random.uniform()` is another option. Be sure to add a mean shift to the observations in each class so that there are three distinct classes.

(b) Run *your own normalized Oja algorithm* on the 90 observations. This algorithm was discussed in the week 4 lecture. Plot the first two principal component score vectors. Compare your results to the ones obtained with scikit-learn's PCA algorithm. Use a different color to indicate the observations in each of the three classes. If the three classes appear separated in this plot, then you're done. If not, then return to part (a) and modify the simulation so that there is greater separation between the three classes.

3 Data Competition Project

(a) Pick two classes of your choice from the dataset.

(b) Run PCA on this subset of the data using Scikit-Learn and project the data down to two dimensions.

(c) You will be training an ℓ_2-regularized logistic regression classifier. Find the value of the regularization parameter λ using Scikit-Learn with 5-fold cross-validation.

(d) Train a classifier using ℓ_2-regularized logistic regression on the training set using **your own fast gradient algorithm**.

(e) Repeat steps (b)-(d), trying all powers of two up to the number of features in the dataset.

(f) Plot, with different colors, the *misclassification error* on the training set and on the test set vs. the dimension of the projection. Note that to obtain the performance on the test set you will need to submit to Kaggle, and you can only submit three times per day. If you only have enough time for three submissions, submit your predictions for dimensions 2, 16, and 128.

4 Optional Exercise

In this exercise, you will compute the gradients of several loss functions. Notation: for an example-label pair (x, y), the prediction is $x^T\beta$.

- $\ell(y, x^T\beta) = (y - x^T\beta)^4$
- $\ell(y, x^T\beta) = yx^T\beta - \exp(x^T\beta)$
- $\ell(y, x^T\beta) = \big(\max(0,\ 1 - yx^T\beta)\big)^2$