HW1

High-dimensional data analysis

Due: 20/5/2021

Question 1 (40 points)

Dataset

1. Download the dataset named **Chemical Composition of Ceramic Samples** from UCI. Save it as a .csv file.

2. Remove the first 2 columns. You should have 88 rows of data and 17 columns.

3. Pick column number 4 to be the Y column. Set the remaining 16 columns to be the dataset X (You can try to repeat these tests with another column as y, just for fun..).

4. Add a first column of ones to the data. This column will be associated with the first coefficient w_0. Your X data should now be of size 88×17, and Y of size 88×1.

5. To make the columns more correlated (for this example), create 6 new features that are sums of existing columns and concatenate them to the dataset. For example new-feature1 = column 2 + column 11, etc.

6. At the end of this step, your X data should now be of size 88×23.

Stability of w

1. Randomly create 3 subsets of datasets X^A, X^B, X^C. Each should hold 60 rows of data.

2. Compute the associated regression coefficients $\mathbf{w^A}, \mathbf{w^B}, \mathbf{w^C}$.

3. Plot them in a single figure, each set of coefficients in a different color. In your plot, do not plot the first coefficient w_1^A, w_1^B, w_1^C (it is much larger than the others).

4. Explain the plotted results.

Ridge regression

1. Add the ridge penalty to the computations of your regression coefficients.

2. Re-plot the coefficients $\mathbf{w^A}, \mathbf{w^B}, \mathbf{w^C}$ for three different values of λ. Choose λ's in a way that the difference is seen.

3. Find a small value for λ in which the coefficients are stable across all three subset. In particular $\|w_i^A - w_i^B\| \leq 5$, $\|w_i^A - w_i^C\| \leq 5$ and $\|w_i^B - w_i^C\| \leq 5$, for $i = 2 \dots 23$

4. Explain the contribution of the ridge regression here. Are there any drawbacks of using it?

Lasso regression

1. Split $\{X, Y\}$ into a train and test set with a ratio of 80%-20%.

2. Run the Lasso regression (use: sklearn.linear_model.Lasso) with 10 different values of λ (λ is called *alpha* (α) in the sklearn function).

3. For each such run:

   - Save the Lasso regression coefficients.
   - Predict for the test points (use the function *predict*) and calculate the MSE by $\text{MSE}_\alpha = \sum(y_i - \hat{y})^2$.

4. Plot a graph of the regression's MSE (mean square error) vs. α.

5. Propose an appropriate value for α based on this graph. How many features are non-zero for this value?

Question 2 (20 points)

1. Show that if S,T are positive definite matrices, then S+T is also positive definite.

2. Given a positive definite matrix A and a vector x, $x \neq 0$, show that the angle between x and Ax is smaller that 90°.
   Hint: Use the energy property together with the equality $\frac{u \cdot v}{\|u\| \|v\|} = cos(\theta)$, where u and v are two vectors and θ is the angle between them.

Question 3 (40 points)

1. Understanding PCA:

   (a) Write a function named ”myPca” that implements PCA step by step. It receives a dataset

   - Input - a dataset and a number d.
   - Output - the means and stds of the dataset’s columns, the normalized dataset, the principal componentes (eigenvectors, W), the eigenvalues (Lambdas) and the data projected onto the first d PCs.
   - In the function, normalize the data, compute the co-variance matrix and its spectral decomposition.

   (b) Download the wine dataset from (it may be easier to save it in a .csv format), and split it into data and label (label is in the 1st column).

   (c) Choose 3 columns, using your ”myPca” function and additional code, and create the ”reconstruction plot” (see Figure 1 (left)), as well as the two-dimensional PC plot (see Figure 1 (right)).

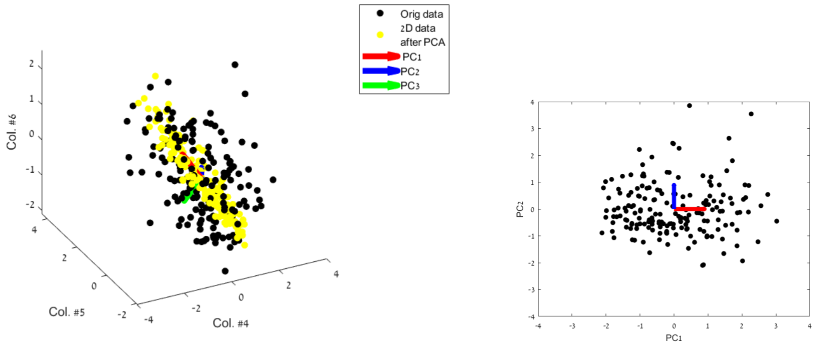

Figure 1: PCA: Left - Original points in black, PC directions and projection of the data onto the first 2 PCs (in yellow). Right - Projection of the data onto the first 2 PCs.

2. **Dry bean classification using PCA and KNN:**

Data preparation

- Download the Dry Bean dataset from the UCI repository.
- It may be useful to change the labels into numeric values, 1-7.
- Since the data is organized by the labels, we first will want to permute the rows. Use an appropriate function to scramble the order of the database.
- Split the data into a 'data' matrix and a labels' vector.
- Split the data=labels into a train+test set (10,000 samples) and a validation set (3611 samples).

Classification using PCA+KNN - building the model

On the train-test samples, we would like to find the number of PCs, d, that results with the highest KNN classification accuracy.

- For that, for each value of d ($d = 1 : 16$), run a 5-fold cross validation (train = 8000 samples, test = 2000 samples) PCA+KNN for classification.
- In particular, in each fold:
  - Apply PCA to the train data and project it to the top d dimensions.
  - Project the test data onto this space.
  - Use KNN (with k=3) in the d-dimensional space to predict the class of the test points.
  - Calculate the classification accuracy for each fold.
- Calculate the average accuracy for each value of d.
- Save the dimension d that results with the highest average accuracy.

Build a model from the train+test data and predict for the validation data

- Project the train+test data onto the first d PCs, where d is the dimension that was found to be best for the train-test splits. If $d > 3$, then plot the projection of the data onto the first 3 PCs.
- Project the validation set onto the d top PCs and use KNN to predict the point's class labels.
- Calculate the accuracy.
- Plot the reduced train data in 3D and color each point by it label.
- In the same figure, plot the projected validation data in black points.
- Plot the Eigenvalues of the constructed PCA.
- Calculate the percentage of explained variance and the reconstruction error.

- Use a 'bar' plot to visualize the coefficients of PC1, PC2, PC3 and PC4 (Each PC is a D-dimensional vector, D=16).
- What are the most dominating features that contribute to PC3 and to PC4?