Programming Exercises Week 3

Machine Learning/Advanced Machine Learning
IT University of Copenhagen

Fall 2019

Theoretical Exercises 3.1: from the Book

As stated on the learnit-website, solve the following exercises from the book:

(5.10.5) You don't need to derive a formulae!
Look at Fig. 5.7. of the book. How would the new version change the appearance of the ellipsoids?

(5.10.6) You don't need to derive a formulae!
Present your solution graphically and describe.

(6.14.4) You don't need to derive a formulae!
Give the steps to a solution, i.e. how to optimize.

Programming Exercise 3.2: Regression

The file **bodyfat.txt** contains a data set with measurements of body fat percentage, age, weight, height, and ten body circumference measurements for 252 men. The columns are: 1) Density, 2) %Fat, 3) Age, 4) Weight, 5) Height, 6) Neck, 7) Chest, 8) Abdom, 9) Hip, 10) Thigh, 11) Knee, 12) Ankle, 13) Biceps, 14) F-arm, 15) Wrist. By using the mathematical tools learnt in the lecture, in the following make a multivariate linear model to predict the body fat percentage from the remaining observations.

(a) Estimate a 1D regression model from the data to predict the factor `fat` (column 2), using the variable `Abdomen` (column 8). (This conforms to the first programming exercise.) Report the estimated parameters.

(b) Divide the data set to independent training (90%) and test sets (10%). Use the variable `Abdomen` (eight column) (M1) and the columns 3 to 15 (M2) to predict `fat`. Report the estimated parameters and the RMSE on training and test set.

(c) Choose at least two criteria to compare the models and discuss the differences. For example: (R)MSE, number of parameters, ...

Programming Exercise 3.3: PCA

(a) Compute a PCA for two dimension of the bodyfat data. Use columns 2 and 8, such that $\boldsymbol{X} \in \mathbb{R}^{252\times 2}$, $N = 252$, $D = 2$. Reminder of how to compute the PCA:

(1) Compute mean $\boldsymbol{m} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n$

(2) Subtract mean from each sample, put them in new data matrix

$$(\boldsymbol{x}_n - \boldsymbol{m}) =: \widetilde{\boldsymbol{x}} \in \mathbb{R}^D \Longrightarrow \widetilde{\boldsymbol{X}} \in \mathbb{R}^{N \times D}$$

(3) Compute covariance matrix

$$\boldsymbol{C} = \frac{1}{N} \widetilde{\boldsymbol{X}}^{\mathrm{T}} \widetilde{\boldsymbol{X}} \in \mathbb{R}^{D \times D}$$

(4) Compute eigenvectors and eigenvalues of the covariance matrix. (Hint: `numpy.linalg.eig`.)
(5) Sort the eigenvectors in descending order with respect to their eigenvalues. The matrix containing the eigenvectors is $\boldsymbol{W} \in \mathbb{R}^{D \times D}$.
(6) Select the first $K \leq D$ eigenvectors $\boldsymbol{W} \in \mathbb{R}^{D \times K}$ for projection in tasks (b),(c) and reconstruction in task (d). The Eq. (6.9) from the book is for one vector input $\boldsymbol{x}$ and output $\boldsymbol{z}$:

$$\mathbb{R}^K \ni \boldsymbol{z} = \boldsymbol{W}^{\mathrm{T}} \underbrace{(\boldsymbol{x} - \boldsymbol{m})}_{\widetilde{\boldsymbol{x}}} \in \mathbb{R}^D \qquad \text{(revisit Book Eq. (6.9))}$$

Extend to multiple points

$$\boldsymbol{Z}^{\mathrm{T}} = \boldsymbol{W}^{\mathrm{T}} \widetilde{\boldsymbol{X}}^{\mathrm{T}} \qquad \text{(PE5.1)}$$

(b) Project the 2D original data $\boldsymbol{X}$ to $\boldsymbol{Z} \in \mathbb{R}^{252 \times K}$, $K = 2$.

(c) Project the 2D original data $\boldsymbol{X}$ to 1D $\boldsymbol{Z} \in \mathbb{R}^{252 \times K}$, $K = 1$.
Hint: For plotting the 1D data in 2D space use $\boldsymbol{x} = \boldsymbol{Z} \in \mathbb{R}^{252 \times 1}$ and $\boldsymbol{y} = \boldsymbol{0} \in \mathbb{R}^{252 \times 1}$

(d) Approximate the original data by $K = 1$.
Hint: Remember to add the mean back and consider that the original data dimension must be reconstructed.

(e) Plot the results for each step, i.e.: original data, projected data for $K = 2$, projected data for $K = 1$, and approximated original data.