Tutorial 3

Machine Learning and Big Data for Economics and Finance

List of activities

I. Complete the list of exercises in this tutorial.

II. Complete **Section 3.6 Lab: Linear Regression**. Section 3.6.6 to 3.6.7.

Exercise 1. Consider the following sample of the three random variables X_1, X_2 and Y:

Obs.	X_1	X_2	Y
1	1	2	o
2	1	3	o
3	-3	1	o
4	2	2	x
5	3	2	x
6	4	1	x
7	4	3	x

Table 1.

1. In the input space, compute the distance between each point and $\boldsymbol{x}_0 = (1, 1)$.
2. Predict Y given $X_1 = 1$ and $X_2 = 1$ using K-nearest neighbor classification for $K = 1$ and $K = 3$.

Solution.

1. Use matrix notation and label the 7 points as $\boldsymbol{x}_1$ to $\boldsymbol{x}_7$.
   Euclidean distances

$$\begin{aligned}
d(\boldsymbol{x}_1, \boldsymbol{x}_0) &= \sqrt{(1-1)^2 + (2-1)^2} &&= 1\\
d(\boldsymbol{x}_2, \boldsymbol{x}_0) &= \sqrt{(1-1)^2 + (3-1)^2} &&= 2\\
d(\boldsymbol{x}_3, \boldsymbol{x}_0) &= \sqrt{(-3-1)^2 + (1-1)^2} &&= 4\\
d(\boldsymbol{x}_4, \boldsymbol{x}_0) &= \sqrt{(2-1)^2 + (2-1)^2} &&= \sqrt{2} \approx 1.4142\\
d(\boldsymbol{x}_5, \boldsymbol{x}_0) &= \sqrt{(3-1)^2 + (2-1)^2} &&= \sqrt{5} \approx 2.2361\\
d(\boldsymbol{x}_6, \boldsymbol{x}_0) &= \sqrt{(4-1)^2 + (1-1)^2} &&= 3\\
d(\boldsymbol{x}_7, \boldsymbol{x}_0) &= \sqrt{(4-1)^2 + (3-1)^2} &&= \sqrt{13} \approx 3.6056
\end{aligned}$$

2. The closest point to $\boldsymbol{x}_0$ is $\boldsymbol{x}_1$. Thus $\mathcal{N}_0 = \{1\}$ and using 1-nearest neighbor classification

$$\Pr\{Y = \text{“o”} | \boldsymbol{X} = \boldsymbol{x}_0\} = \frac{1}{1} \sum_{i \in \mathcal{N}_0} 1(y_i = \text{“o”}) = 1$$

Using Bayes classifier, since $\Pr\{Y = \text{“o”} | \boldsymbol{X} = \boldsymbol{x}_0\} = 1 > \frac{1}{2}$, you predict Y_0 to be “o”.

The closest 3 points to $\boldsymbol{x}_0$ are $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_4$. Thus $\mathcal{N}_0 = \{1, 2, 4\}$ and using 3-nearest neighbor classification

$$\Pr\{Y = \text{“o”} | \boldsymbol{X} = \boldsymbol{x}_0\} = \frac{1}{3} \sum_{i \in \mathcal{N}_0} 1(y_i = \text{“o”}) = \frac{1}{3}(1 + 1 + 0) = \frac{2}{3}.$$

Using Bayes classifier, since $\Pr\{Y = \text{“o”} | \boldsymbol{X} = \boldsymbol{x}_0\} = \frac{2}{3} > \frac{1}{2}$, you predict Y_0 to be “o”.

Exercise 2. Load the data included in the file `MC1.csv`. The file contains a sample of size $n = 1000$ from a random variable X.

The data generating process for a new random variable Y is given by

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

where $\beta_0 = -1$, $\beta_2 = 5.1$ and $\varepsilon \sim N(0,1)$.

1. Generate a sample of size n from Y.
2. Assuming you don't know the parameters behind the data generating process, compute the least squares estimates for $\hat{\beta}_0$, $\hat{\beta}_1$ and $\hat{\sigma}$ (the standard deviation of the error term).
3. Generate 100 different samples of size n of Y and for each sample compute $\hat{\beta}_{0,m}$, $\hat{\beta}_{1,m}$ and $\hat{\sigma}_m$ where $\hat{\beta}_{0,m}$ is the estimate of β_0 in sample m for $m = 1, \ldots, 100$.
4. Using the values $\hat{\beta}_{0,m}$, $\hat{\beta}_{1,m}$ and $\hat{\sigma}_m$, compute
   a. The sample averages of each of $\hat{\beta}_{0,m}$, $\hat{\beta}_{1,m}$ and $\hat{\sigma}_m$.
   b. The sample variance of $\hat{\beta}_{0,m}$.
   c. The sample variance of $\hat{\beta}_{1,m}$.
   d. The sample covariance of $\hat{\beta}_{0,m}$ and $\hat{\beta}_{1,m}$.
5. Plot $\hat{\beta}_{0,m}$, $\hat{\beta}_{1,m}$ and $\hat{\sigma}_m$ and discuss.
6. Compare the results of the small simulation exercise with the formula $\mathrm{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}$.

Solution.

This is the first Monte Carlo exercise carried during the course.

The code giving solution to the exercise will be given. The following points are to be noted

- As this is supervised learning, all the analysis is done conditional on the inputs, i.e. fixing them. When doing a simulation, it makes sense to sample only Y and not the inputs.
- The objective of this Monte Carlo exercise is to show the behavior of least squares learning. That is, assuming we don't know how the data were generated, we run the learner over different samples to study its behavior. In particular:
  - On average, are the estimators $\hat{\beta}_0$ and $\hat{\beta}_1$ close to the truth.
  - How much do they fluctuate.
  - As a bonus, we can see if the theoretical formula $\mathrm{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ is anywhere near the result of the simulations.
- As we need to repeat the same operation $M = 100$ times, it will be useful to use a loop.

  For example

```
for (i in 1:10)
{
  print(i^2)
}
```

  This code will start by setting $i = 1$ and then printing 1 in the command line. Then it will set $i = 2$ and then print i^2 which is 4 in the command line. Then it will set $i = 3$ etc...

  Finally, it will set $i = 100$ and then print 10000 on the command line.

  Be careful to respect the syntax and put all the list of commands between braces {}.
- Notice that $\hat{\beta}_{0,m}$, $\hat{\beta}_{1,m}$ and $\hat{\sigma}_m$ are on average close to $\beta_0 = -1$, $\beta_2 = 5.1$ and $\sigma = 1$. Notice also that the actual numerical values taken by their variances of and their covariance

$$\begin{pmatrix} \mathrm{Var}(\hat{\beta}_{0,m}) & \mathrm{Cov}(\hat{\beta}_{0,m}, \hat{\beta}_{1,m}) \\ \mathrm{Cov}(\hat{\beta}_{0,m}, \hat{\beta}_{1,m}) & \mathrm{Var}(\hat{\beta}_{0,m}) \end{pmatrix}$$

  are very close to the ones in the formula

$$\begin{pmatrix} \mathrm{Var}(\hat{\beta}_0) & \mathrm{Cov}(\hat{\beta}_0, \hat{\beta}_1) \\ \mathrm{Cov}(\hat{\beta}_0, \hat{\beta}_1) & \mathrm{Var}(\hat{\beta}_0) \end{pmatrix} = \mathrm{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}$$