TP3 : Density estimation

In this tutorial we will use the Python language. There are several ways to use it, first of all, you need a program writing part (in .py format) which can only be done with any text editor, then there is a program to compile Python programs, you can use the one of your choice. Each graph requested in this TP will be saved in the format Pi-NAME1NAME2Qj.png where i is the number of the part in question, j is the number of the question within the part considered and NAME1 and NAME2 are the two last names of the two members of the TP pair. Each question with the symbol ? will require the creation of a graph to be recorded as specified above. If there are several graphs to be made for a single question, they will be named PiNAME1NAME2Qja.png, PiNAME1NAME2Qjb.png and so on...

1 Preparation

This tutorial refers to chapter 3 of the course that was not covered in class. Feel free to refer to it.

Consider a realization $x_1, \ldots, x_n$ of a sample $X_1, \ldots, X_n$ of identical and independent variables of common density f. The goal of this tutorial is to compare the kernels used to estimate the common density f. It is important to understand that in practice f is unknown, here, to compare the efficiency of the kernels and the size of the window h we will assume in a first part that f is the density of a reduced centered Gaussian, in a second part we will assume that f is the density of a higher dimensional law.

1. If $K : \mathbb{R} \to \mathbb{R}_+$ is a statistical kernel and $\mu \in \mathbb{R}$ a constant, is the translation of K by the constant a, $\tau_\mu K$, still a statistical kernel ?
2. If $K : \mathbb{R} \to \mathbb{R}_+$ is a statistical kernel and $\lambda \in \mathbb{R}*$ is a nonzero constant, show that $d_\lambda K$ defined for all $x \in \mathbb{R}$ by $d_\lambda K(x) = \frac{1}{\lambda} K\left(\frac{x}{\lambda}\right)$ is still a statistical kernel.
3. Show that $K = \frac{1}{2} 1_{[-1,1]}$ is a statistical kernel, we call it the uniform kernel.
4. Show that $K(x) = (1 - |x|) 1_{[-1,1]}(x)$ is a statistical kernel, it is called the triangle kernel.
5. Show that $K(x) = \frac{3}{4}(1 - x^2) 1_{[-1,1]}(x)$ is a statistical kernel, it is called the Epanechnikov kernel.
6. Show that $K(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ is a statistical kernel, it is called the Gaussian kernel.

Let $h > 0$ be a constant called the window. Let K be a statistical kernel. Consider the function $\widehat{f_h}$, defined for all $x \in \mathbb{R}$, by :

$$\widehat{f_h}(x) = \frac{1}{n} \sum_{k=1}^{n} d_h \tau_{X_i} K(x)$$

This is the density estimate f with window h and kernel K.

7. Show that $\widehat{f_h}$ is a probability density.

2 Session work with Python : part 1

The aim of this part is to define, represent and compare the efficiency of the four kernels of the preparation for the estimation of the density of a standard Gaussian f. We suppose that $X_1, \ldots, X_n$ is a sample of size n of independent and identically distributed variables according to the normal distribution of density f. Download the TP3.py script available on Moodle, it is in this script that you will define all the functions and answer the questions.

1. In the same script, define four functions $K1, K2, K3, K4$ corresponding respectively to the uniform, triangle, Epanechnikov and Gaussian kernels.

2. $\star$ Represent these four kernels on the same graph (use a legend and different colors). Create a function to do this question that you will name **AllplotK** which will take as input the parameters of the graph (the step, xmin, xmax, the colors etc...) and will represent the graph in return.

3. Generate a realization of the random sample X according to the standard Gaussian distribution of size n. (n is currently set to 100 in the script).

4. Define the function **fhat** which takes as arguments a function K (the kernel), the window h and the sample realization X and a variable x and returns the image of x by the function $\widehat{f}_h$.

5. $\star$ Show on the same graph the reference function f and the four functions $\widehat{f}_h$ obtained with the kernels $K1, K2, K3, K4$. You will add a legend and different colors to all the curves. We will fix for this question $h = 2$. You will define a function like in question 2 to do this question. This function will be named **Allplotfhath2**.

6. $\star$ Repeat the previous question with $h = 1$. Qualitatively, does the estimate differ more when varying the kernel used or the window h used? The new function for this question will be named **Allplotfchapeauh1**.

7. $\star$ Repeat the two previous questions for $n = 10$ then $n = 1000$. For this question, four graphs must be constructed: the first one for $(n, h) = (10, 2)$, the second one for $(n, h) = (10, 1)$, the next one for $(n, h) = (1000, 2)$ and the last one for $(n, h) = (1000, 1)$. You will detail your reasoning in the script and comment on the results obtained. Return to the value of $n = 100$ in the script for the rest of the exercise.

8. We will compute the squared error of an estimate: let be

$$SCE(h) = \sum_{i=0}^{500} (\widehat{f}_h(t_i) - f(t_i))^2$$

the sum of squares of the differences between the image of t_i by the estimate $\widehat{f}_h$ and the image of t_i by f, where $\{t_0, t_1, t_2, \ldots, t_{500}\}$ is a discretization of the interval $[-5, 5]$ of step $10/500$.
In other words

$$-5 = t_0 < t_1 = -5 + 10/500 < t_2 = -5 + 20/500 < \cdots < t_{500} = -5 + 5000/500 = 5$$

Define a function **SCE** that takes a function (the considered kernel), the window h, the reference density f as parameters and returns $SCE(h)$.

10. Define a function **thebesth** which takes a function (the kernel in question) and another function f (the reference) as parameters and returns the index divided by 100 of the minimum of the list $\{SCE(k/100)\}_{1\leq k\leq 200}$. For each kernel, the best window for the estimation of the reference function is given by this function.

11. Define then a function which represents graphically the four density estimates for these four kernels with the windows obtained via the function **thebesth**. To do this, define the function **Allplotfhathoptimal**.

3 Session work with Python : part 2

We are now going to exploit the functionalities of **scikit-learn**. The function **densite estimation** present in the script is used to carry out a density estimation by Gaussian kernel (**kernel**=$'gaussian'$) with for window h whose reference density is a Gaussian mixture (of two Gaussians of means mu1, mu2 and of deviations sigma1, sigma2). This function uses the scikit-learn package.

1. $\star$ Execute this function with $mu1 = 0$, $mu2 = 5$ and $sigma1 = sigma2 = 1$, $N = 100$ and $h = 0.75$.

2. Compare the influence of the window h to any other parameter fixed as in the previous question. We can test values of h between 0.2 and 1.5. Comment on this.

3. Vary the parameters of the two Gaussian distributions that define the Gaussian mixture. Comment.

4. Vary N and comment.

5. $\star$ One can also test other kernels for example by replacing '*gaussian*' in the code by '*epanechnikov*'. Realize this graph by executing the function **estimationdensite2** with the same parameters as in question 1.