École Nationale
Supérieure
de l'Électronique
et de ses Applications

Statistiques Multidimentionnelles et Inférentielles

Responsable du cours : Bastien Faucard
bastien.faucard@ensea.fr

E.N.S.E.A. 2023 – 2024, Semestre 8

Travaux Pratiques

Table des matières

1

TP3 : Estimation de densité

Dans ce TP, nous utiliserons le langage Python. Il y'a plusieurs manière de l'utiliser, premièrement, il faut une partie d'écriture de programmes (au format .py) qui peut seulement être faite avec n'importe quel éditeur de texte, il y'a ensuite un logiciel de compilation des programmes Python, vous pouvez utiliser celui de votre choix. Chaque graphique demandé dans ce TP sera à enregistrer au format PiNOM1NOM2Qj.png où i est le numéro de la partie en question, j est le numéro de la question au sein de la partie considérée et NOM1 et NOM2 sont les deux noms de famille des deux membres du binôme de TP. Chaque question comportant le symbole $\star$ nécessitera la création d'un graphique à enregistrer comme spécifié ci-dessus. S'il y'a plusieurs graphiques à faire pour une seule question, ils seront nommés PiNOM1NOM2Qja.png, PiNOM1NOM2Qjb.png etc...

1.1 Préparation

Ce TP fait référence au chapitre 3 du cours qui n'a pas été traité en amphi. N'hésitez pas à vous y référer.

Considérons une réalisation $x_1, \dots, x_n$ d'un échantillon $X_1, \dots, X_n$ de variables identiques et indépendantes de densité commune f. Le but de ce TP est de comparer les noyaux utilisés pour estimer la densité commune f. Il faut bien comprendre que dans la pratique f est inconnue, ici, pour comparer l'efficacité des noyaux et la taille de la fenêtre h nous allons supposer dans une première partie que f est la densité d'une gaussienne centrée réduite, dans une seconde partie nous supposerons que f est la densité d'une loi de dimension plus grande.

1. Si $K : \mathbb{R} \to \mathbb{R}_+$ est un noyau statistique et $\mu \in \mathbb{R}$ une constante, la translation de K par la constante a, $\tau_\mu K$, est-elle encore un noyau statistique ?
2. Si $K : \mathbb{R} \to \mathbb{R}_+$ est un noyau statistique et $\lambda \in \mathbb{R}*$ une constante non nulle, montrer que $d_\lambda K$ définie pour tout $x \in \mathbb{R}$ par $d_\lambda K(x) = \frac{1}{\lambda} K\left(\frac{x}{\lambda}\right)$ est encore un noyau statistique.
3. Montrer que $K = \frac{1}{2} 1_{[-1,1]}$ est un noyau statistique, on l'appel le noyau uniforme.
4. Montrer que $K(x) = (1 - |x|) 1_{[-1,1]}(x)$ est un noyau statistique, on l'appel le noyau triangle.
5. Montrer que $K(x) = \frac{3}{4}(1 - x^2) 1_{[-1,1]}(x)$ est un noyau statistique, on l'appel le noyau d'Epanechnikov.
6. Montrer que $K(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ est un noyau statistique, on l'appel le noyau gaussien.

Soit $h > 0$ une constante appelée la fenêtre. Soit K un noyau statistique. On considère la fonction $\widehat{f}_h$, définie pour tout $x \in \mathbb{R}$, par :

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

C'est l'estimation de la densité f avec la fenêtre h et le noyau K.

7. Montrer que $\widehat{f_h}$ est une densité de probabilité.

1.2 Travail en séance avec Python : partie 1

Le but de cette partie est de définir, représenter et comparer l'efficacité des quatre noyaux de la préparation pour l'estimation de la densité d'une gaussienne standard f. On suppose donc que $X_1, \ldots, X_n$ est un échantillon de taille n de variables indépendantes et identiquement distribuées selon la loi normale centré réduite de densité f. Télécharger le script TP3.py disponible sur Moodle, c'est dans ce script que vous définirez toutes les fonctions et répondrez aux questions.

1. Dans ce même script, définir quatre fonctions $K1, K2, K3, K4$ correspondant respectivement aux noyaux uniforme, triangle, d'Epanechnikov et gaussien.
2. $\star$ Représentez ces quatre noyaux sur un même graphique (utiliser une légende et des couleurs différentes). Créer une fonction pour faire cette question que vous nommerez **AllplotK** qui prendra en entrée les paramètres du graphique (le pas, xmin, xmax, les couleurs etc...) et représentera le graphique en retour.
3. Générer une réalisation de l'échantillon aléatoire X selon la loi gaussienne standard de taille n. (n est pour l'instant fixé à 100 dans le script).
4. Définir la fonction **fchapeau** qui prend comme argument une fonction K (le noyau), la fenêtre h et la réalisation de l'échantillon X et une variable x et qui retourne l'image de x par la fonction $\widehat{f_h}$.
5. $\star$ Représenter sur un même graphique la fonction f de référence ainsi que les quatres fonctions $\widehat{f_h}$ obtenues avec les noyaux $K1, K2, K3, K4$. Vous ajouterez une légende et des couleurs différentes à toutes les courbes. On fixera pour cette question $h = 2$. Vous définierez une fonction comme dans la question 2 pour faire cette question. Cette fonction sera nommée **Allplotfchapeauh2**.
6. $\star$ Refaire la question précédente avec $h = 1$. Qualitativement, est-ce que l'estimation diffère plus lorsque l'on fait varier le noyau utilisé ou la fenêtre h utilisée ? La nouvelle fonction pour cette question sera nommée **Allplotfchapeauh1**.
7. $\star$ Reprendre les deux questions précédentes pour $n = 10$ puis $n = 1000$. Pour cette question, quatre graphiques doivent être construits : le premier pour $(n, h) = (10, 2)$, le second pour $(n, h) = (10, 1)$, le suivant pour $(n, h) = (1000, 2)$ et le dernier pour $(n, h) = (1000, 1)$. Vous détaillerez votre raisonnement dans le script et commenterez les résultats obtenus. Revenir ensuite à la valeur de $n = 100$ dans le script pour la suite du TP.
8. Nous allons calculer l'erreur quadratique d'une estimation : soit

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

la somme de carrés des écarts entre l'image de t_i par l'estimation $\widehat{f_h}$ et l'image de t_i par f, où $\{t_0, t_1, t_2, \ldots, t_{500}\}$ est une discrétisation de l'intervalle $[-5, 5]$ de pas $10/500$.
Autrement dit

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

Définir une fonction **SCE** qui prend comme paramètre une fonction (le noyau considéré), la fenêtre h, la densité de référence f et qui retourne $SCE(h)$.

9. Définir une fonction **lemeilleurh** qui prend une fonction (le noyau en question) et une autre fonction f (la référence) en paramètres et retourne l'index divisé par 100 du minimum de la liste $\{SCE(k/100)\}_{1 \leq k \leq 200}$. Pour chaque noyau, la meilleur fenêtre pour l'etimation de la fonction de référence est donnée par cette fonction.
10. $\star$ Définir alors une fonction qui représente graphiquement les quatres estimations de densité pour ces quatre noyaux avec les fenêtre obtenue via la fonction **lemeilleurh**. Définir pour ce faire, la fonction **Allplotfchapeauhoptimal**

1.3 Travail en séance avec Python : partie 2

Nous allons maintenant exploiter les fonctionnalités de **scikit-learn**. La fonction **estimationdensite** présente dans le script sert à effectuer une estimation de densité par noyau gaussien (**kernel**=$'gaussian'$) avec pour fenêtre h dont la densité de référence est un mélange gaussien (de deux gausiennes de moyennes mu1, mu2 et d'écartyypes sigma1, sigma2). Cette fonction fait appel au package scikit-learn.

1. $\star$ Executer cette fonction avec $mu1 = 0$, $mu2 = 5$ et $sigma1 = sigma2 = 1$, $N = 100$ et $h = 0.75$.
2. Comparer à tout autre paramètre fixés comme dans la question précédente, l'influence de la fenêtre h. On pourra tester des valeurs de h comprises entre 0.2 et 1.5. Commenter.
3. Faite varier les paramètres des deux lois gaussiennes qui définissent le mélange gaussien. Commenter.
4. Faite varier N et commenter.
5. $\star$ On peut aussi tester d'autres noyaux par exemple en remplaçant '*gaussian*' dans le code par '*epanechnikov*'. Réaliser ce graphique en executant la fonction **estimationdensite2** avec les mêmes paramètres que ceux de la question 1.