Master 1 Mathématiques Appliquées et Statistique **2021-2022**
parcours MSS et CMI

TP3 - Statistique Non Paramétrique

Estimation non paramétrique d'une régression

Le rapport est à rendre avant le mercredi prochain (20 Avril).

Exercise 1 **L'estimateur régressogramme**

(a) Construire l'estimateur régressogramme (à classes de même taille) d'une fonction de régression m définie sur $[0, 1]$ en complétant le code ci-dessous :

```
> # construire l'estimateur régressograme comme une fonction de N, de l'échantillon (X_i,
> # ou N est le nombre de classes / intervalles / bins
> # et x est un point dans [0,1]
> estimateur_reg <- function(x, N, echX, echY){
+   #assertation
+   stopifnot(length(echX)==length(echY))
+   n <- length(echX)
+   # découper [0, 1] en N classes de même taille
+   breakpoints <- seq(0, 1, length.out = N + 1)
+   # calculer la valeur de l'éstimateur régressogramme en point x: <<m_chap_x>>
+   for (i in 1:N){
+     # réperer la bonne classe <<i>> ou se trouve x
+     if(x >= breakpoints[i] & x < breakpoints[i+1]){
+       # calculer la moyenne des Y_i dont X_i tombent dans la classe <<j>>
+
+
+       is_echX_in_classj <- #à complèter
+       echY_in_classj <- echY[is_echX_in_classj]
+       m_chap_x <- #à complèter
+
+
+       break
+     }
+   }
+   return(m_chap_x)
+ }
```

(b) **Simulation : l'évaluation d'une méthode sur les donées idéales.**

Utiliser le code ci-dessous pour

- simuler un $n = 1000$ - échantillon $(X_1, Y_1), \ldots, (X_n, Y_n)$ i.i.d. de (X, Y) dont la vraie relation est donné par $Y = m(X) + \varepsilon$. On prends $X_i \overset{i.i.d.}{\sim} \mathcal{U}(0,1)$, $m(x) = \sin(2\pi x^2)^2$, et $(\varepsilon_1, \ldots, \varepsilon_n)$ *i.i.d de* $\varepsilon \sim \mathcal{N}(0, 0.5^2)$.

Le but de la simulation est de vérifier si l'etimateur proposé peut trouver la vraie relation m depuis les donées idéales (les hypothéses sont verifiés). Sioui, l'etimateur est alors valable pour les donées réelles.

- appliquer l'estimateur sur les données simulées avec $N = 10, 30, 50$,

- tracer la fonction éstimée sur $[0, 1]$ (superposer les données simulées et la vraie relation m à estimer).

Quel N est le meilleur ?

```
> # le modèle de regression est vrai pour les données simulées
> set.seed(1)
> echX_ <- runif(1000)
> echEpsilon <- rnorm(1000, 0, 0.5)
> echY_ <- sin(2*pi*echX_^2)^2 + echEpsilon
> # appliquer et tracer l'estimateur sur les données simulées
> x_list <- seq(0,0.999,length.out = 1000)
> estimation_reg <- sapply(x_list, function(x_){
+                                                return(estimateur_reg(x_,50,echX_,echY_
+                                                }
+                            )
> plot(echX_, echY_, cex = 0.5)
> lines(x_list,sin(2*pi*x_list^2)^2, type="s", col = "red", lwd=2)
> lines(x_list,estimation_reg, type="s", col = "blue", lwd=2)
```

Exercise 2 **L'estimateur à noyau**

(a) Construire un estimateur à noyau (la densité de X est inconnue) avec le noyau Gaussien en complètant le code ci-dessous.

Rapel de la définition :

Si $\mathbf{K} : \mathbb{R} \to \mathbb{R}$ est un noyau statistique d'ordre 1, ***l'estimateur de Nadaraya-Watson*** vaut

$$\widehat{m}_{NW}(x) = \begin{cases} \dfrac{\sum_{i=1}^{n} \mathbf{K}\left(\frac{X_i - x}{h}\right) Y_i}{\sum_{k=1}^{n} \mathbf{K}\left(\frac{X_k - x}{h}\right)} & \text{si } \sum_{k=1}^{n} \mathbf{K}\left(\frac{X_k - x}{h}\right) \neq 0 \\ 0 & \text{sinon.} \end{cases} \tag{0.1}$$

```
> #construire l'estimateur à noyau Gaussien comme une fonction de h, de l'échantillon (X_
> estimateur_noyau_Gaussien <- function(x, h, echX, echY){
+   #assertation
```

```
+   stopifnot(length(echX)==length(echY))
+   # calculer la valeur de l'éstimateur à noyau en point x: <<m_chap_x>>
+
+   #à complèter
+
+   return(m_chap_x)
+ }
```

(b) Appliquer l'estimateur à noyau Gaussien à l'échantillon de l'exercise 1 avec 3 valeurs de h à votre choix qui donnent respectivement sous-lissage, sur-lissage, et ni l'un ni l'autre dans leurs estimations. On notera cette dernière h^* dans la suite.

(c) Retrouver votre estimation avec celui donné par R en utilisant les fonctions `ksmooth` et `locpoly` pour $h = h^*$, à l'aide du code ci-dessous.

```
> h_star <-
+
+ # le package utlise une autre déf du bandwidth, en fonction de quartiles de la Gaussian
+ # qui est égale à 2.67*h (avec h de notre définition)
+ # plus d'explication:
+ # https://stats.stackexchange.com/questions/396324/what-does-bandwidth-in-kernel-regres
+ estimation_ksmooth <- ksmooth(echX_, echY_, kernel = "normal", bandwidth = 2.67*h_star,
> plot(echX_, echY_, cex = 0.5)
> lines(x_list, estimation_noyau_Gaussien,col="green", lwd=3)
> lines(estimation_ksmooth,col="blue",type='s', lwd=1)
```

La méthode d'estimation de Nadaraya-Watson est un cas particulier de la technique des polynômes locaux qui est réalisée dans R avec la fonction locpoly. L'estimateur de Nadaraya-Watson=l'estimateur est lié au degré zéro et s'obtient ici :

```
> library(KernSmooth)
> estimation_locpoly <- locpoly(echX_, echY_,degree=0,bandwidth=h_star,gridsize=1000,rang
> plot(echX_, echY_, cex = 0.5)
> lines(x_list, estimation_noyau_Gaussien,col="green", lwd=3)
> lines(estimation_locpoly,col="blue",type='s', lwd=1)
```

(d) **Observer l'effet local du lissage.** Ajouter l'observation $(0.5, -5)$ dans l'échantillon, et puis appliquer l'estimateur à noyau sur l'échantillon augmenté avec $h = h^*$. Quelle est la difference entre cette dernière estimation et la précédante ?

Exercise 3 **Sélection de la fenêtre h : la validation croisée (Leave One Out)**

On va maintenant chercher une approche qui permet de choisir l'hypermaramètre h en autonomie (dans la pratique), et puis la programmer.

On d'abords cherche une merique qui évalue la performance de l'éstimateur à noyau $\hat{m}$ appliqué sur un jeu de données $(X_1, Y_1), ..., (X_n, Y_n)$. Un choix courant en pratique est la somme des carrés des résidus (SCR) :

$$\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{m}_h(X_i)\right)^2.$$

Donc, la valeur de h qui minimise la SCR est optimale. Mais en effet le minimiseur est toujours $h = 0$ pour la formule ci-dessur, car $\hat{m}_h$ est calculé avec Y_i déjà (plus d'explication, cf. Section 4.3.2 https ://bookdown.org/egarpor/NP-UC3M/kre-i-bwd.html#kre-i-bwd-cv), donc on considère au lieu son variant

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{m}_{h,-i}(X_i)\right)^2, \tag{0.2}$$

où $\widehat{m}_{h,-i}(x)$ est l'estimateur de Nadaraya-Watson sur les n couples privés du couple (X_i, Y_i).

Comme la validation croisée utilisée dans le cadre de l'estimation de densité, on cherche à lier $\hat{m}_{h,-i}(x)$ à $\hat{m}_h(x)$ pour faciliter le code.

Pour ceci, on note $\omega_j(x)$ le poids $\dfrac{\mathbf{K}\left(\dfrac{X_j - x}{h}\right)}{\sum_{k=1}^{n}\mathbf{K}\left(\dfrac{X_k - x}{h}\right)}$, donc $\hat{m}_h(x) = \sum_{j=1}^{n}\omega_j(x)Y_j$.

D'autre part, on note $\omega_j^{-i}(x)$ le poids $\dfrac{\mathbf{K}\left(\dfrac{X_j - x}{h}\right)}{\sum_{k=1,k\neq i}^{n}\mathbf{K}\left(\dfrac{X_k - x}{h}\right)}$, donc $\hat{m}_{h,-i}(x) = \sum_{j=1,j\neq i}^{n}\omega_j^{-i}(x)Y_j$.

D'après le calcul direct, on peut trouver

$$\omega_j^{-i}(x) = \frac{\omega_j(x)}{1 - \omega_i(x)}. \tag{0.3}$$

Injecter Equation 0.3 dans $\hat{m}_{h,-i}(x)$, on a trouvé ainsi le lien désiré

$$\hat{m}_{h,-i}(x) = \frac{\hat{m}_h(x) - \omega_i(x)Y_i}{1 - \omega_i(x)}$$

La SCR (0.2) est devenue

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \hat{m}_h(X_i)}{1 - \omega_i(X_i)}\right)^2.$$

(a) Créer une fonction de R qui prend la fenêtre h et l'échantillon $(X_1, Y_1)..., (X_n, Y_n)$ comme les entrées et sort $\widehat{R}(h)$, en complètant le code ci-dessous.

```
> R_chap <- function(h, echX, echY){
+   #calculer le vecteur numérateur: (Y1-m_chap_h(X1), Y2-m_chap_h(X2),..., Yn-m_chap_h(X
+
+   vect_numérateur <- #à complèter
+
+   #calculer le vecteur dénominateur: (1-w1(X1), 1-w2(X2),..., 1-wn(Xn))
+   wi_Xi <- function(Xi){
+     int_K <- (echX - Xi)/h
+     val <- dnorm(0)/sum(dnorm(int_K))
+     return(val)
+   }
+   vect_dénominateur <- 1-sapply(echX, wi_Xi)
```

```
+
+   return( mean( vect_numérateur^2 / vect_dénominateur^2 ) )
+ }
```

(b) Tracer la courbe de $\hat{R}(h)$ en fonction de h, avec l'échantillon donné dans les exercises précedents, puis repérer son minimum.

```
> h_list <- seq(0.01, 0.1, 0.001)
>
> ## à complèter
>
```