Estimation non paramétrique
Master 1 MIMSE – Université de Bordeaux

– TP 2 : Estimation de densité par la méthode du noyau –

Partie 1 : Construction d'un estimateur à noyau

Soit $X_1, \ldots, X_n$ un échantillon i.i.d. de variables alétoires réelles de densité f. Un estimateur $\hat{f}_h$ à noyau de f s'écrit sous la forme

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right), \quad x \in \mathbb{R}$$

où h est le paramètre de fenêtre et $K : \mathbb{R} \to \mathbb{R}^+$ un noyau.

1) Écrire une fonction `KG` qui prend en paramètre $u \in \mathbb{R}$, et renvoie $K_G(u)$ où K_G est le noyau gaussien centré réduit.

2) Écrire une fonction `KE` qui prend en paramètre $u \in \mathbb{R}$ et renvoie $K_E(u)$ où K_E est le noyau d'Epanechnikov.

3) Écrire une fonction `fchap_G` qui prend en paramètres un vecteur X, $x \in \mathbb{R}$, $h > 0$, et renvoie la valeur de l'estimation de Parzen-Rosenblatt au point x de l'échantillon X muni du noyau gaussien et de la fenêtre h.

4) Même question avec `fchap_E` pour le noyau d'Epanechnikov.

5) Simuler un échantillon X_0 de taille $n = 50$ selon la loi gaussienne centrée réduite.

6) Compléter le code ci-dessous pour tracer la densité f_0 de la loi $N(0, 1)$, ainsi que les données de l'échantillon X_0.

```
# Choix d'une grille de points x
nbpts = 1000
x = seq(-4,4,length.out=nbpts)

# Visualisation des donnees et de la vraie densite f0
f0 = ...
plot(x,f0,type="l",lwd=2)
points(X0,rep(0,50),col="green",lwd=2)
```

7) A l'aide de l'exemple ci-dessous, estimer la densité de l'échantillon X_0 avec les deux estimateurs à noyaux précédents `fchap_G` et `fchap_E` pour $h = 0.2$ puis pour $h = 0.05$ et enfin pour $h = 0.8$. Que constate-t-on ?

```
# densité estimée par noyau d'Epanechnikov sur la grille de points x
h=0.2
hatf=rep(0,nbpts)
cpt=0
```

```
for (y in x){
cpt=cpt+1
hatf[cpt]= fchap_E(X0,y,h)
}
# Figure : la vraie densité, son estimation et les données
plot(x,2*f0,type="n",lwd=2,lty="dotdash")
lines(x,f0,type="l",lwd=2,lty="dotdash")
lines(x,hatf,type = "l",col="red", lwd=2)
points(X0,rep(0,50),col="green",lwd=2)
```

La fonction `density` de R permet en fait d'implémenter directement un estimateur à noyau pour différents choix de K et h. Utiliser la commande `help(density)` pour regarder l'ensemble des paramètres de cette fonction. Voici un exemple d'utilisation pour un noyau Gaussien avec $h = 0.1$:

```
hatf = density(X0,bw=0.1,kernel="gaussian",n=nbpts,from=-4, to= 4)$y
```

8) Reprendre la question précédente en utilisant cette fois la fonction `density`.

Partie 2 : Validation Croisée Leave-One-Out

L'erreur quadratique intégrée est donnée par : $R(h) = J(h) + \|f\|_2^2$ où

$$J(h) = \mathbb{E}\left(\int (\hat{f}_h(x))^2 dx - 2\int \hat{f}_h(x) f(x) dx\right).$$

Ainsi, minimiser l'erreur quadratique intégrée revient à minimiser J. Cependant, comme on ne connaît pas f, on ne peut pas calculer J. On va l'estimer comme suit. Pour $i = 1 \dots n$, on construit un estimateur $\hat{f}_{h,-i}$ de la densité à partir de l'ensemble des données privé de la i-ème observation

$$\hat{f}_{h,-i}(x) = \frac{1}{(n-1)h} \sum_{j=1, j\neq i}^{n} K\left(\frac{X_j - x}{h}\right), \quad x \in \mathbb{R}.$$

On estime ensuite $J(h)$ par

$$\hat{J}(h) = \int_{\mathbb{R}} (\hat{f}_h(x))^2 dx - \frac{2}{n}\sum_{i=1}^{n} \hat{f}_{h,-i}(X_i)$$

qui est sans biais. Puis, on cherche h_{opt} qui minimise $\hat{J}(h)$.

9) Construire un vecteur `Jchap` qui contient, pour un noyau gaussien et pour l'échantillon X_0, les valeurs de $\hat{J}(h)$ pour chaque h dans `seq(0.1,1.6,length.out=40)`.

10) Déterminer h_{opt} et tracer la densité correspondante.

11) Que fait la fonction `bw.ucv` ? La valeur obtenue est-elle proche de celle issue de la validation croisée ? Tracer les deux densités correspondantes sur un même graphique.