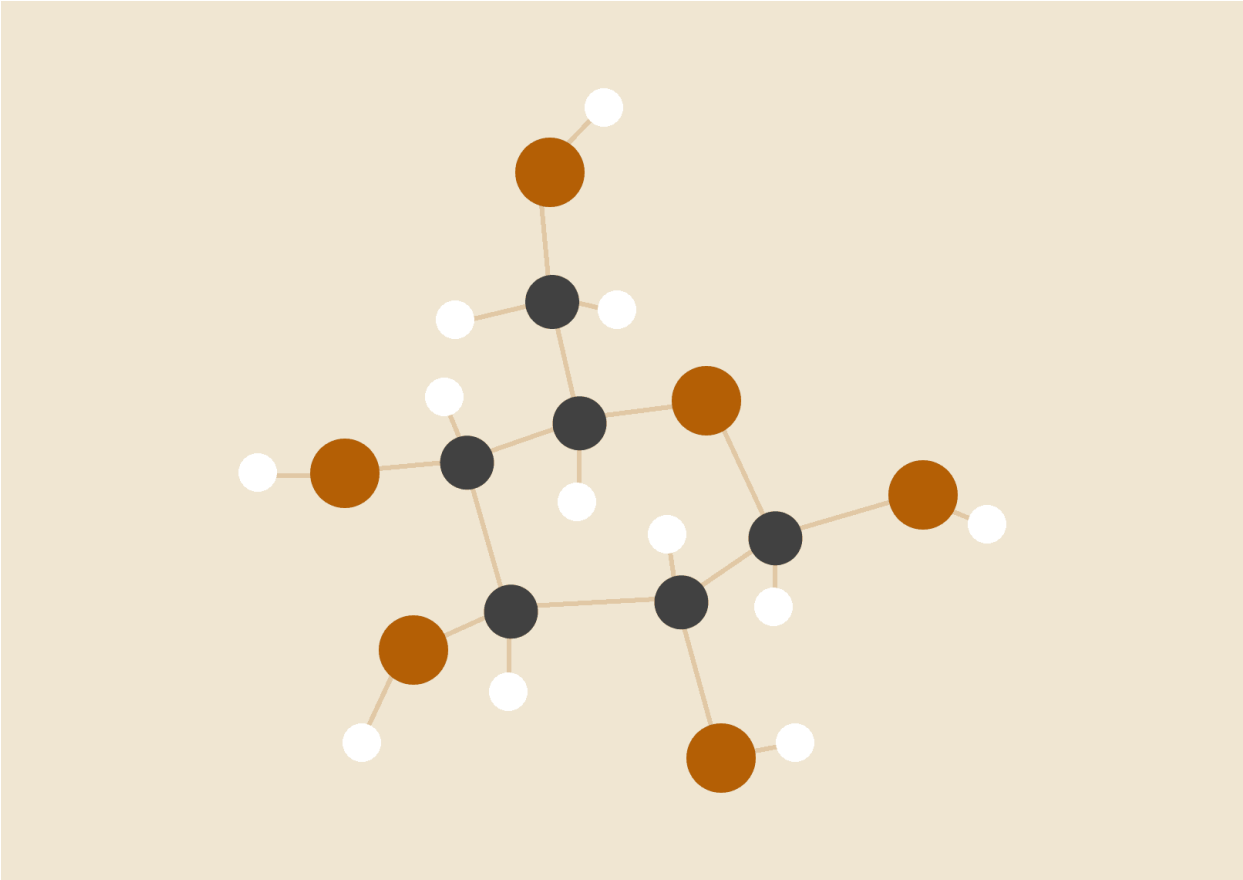
- Projet 1 -

Apprentissage Statistique

Estimation d'une fonction de régression



Israël BISMUTH - Sarah CHEMLA

30.05.2017

Master I - Mathématiques Appliqués

INTRODUCTION

Le problème de prédiction ou d'explication d'une variable Y à l'aide d'une variable X est souvent rencontré en pratique. La fonction qui fournit la meilleure prévision (en moyenne quadratique) de Y en fonction de X est l'espérance conditionnelle.

$$f(x) = \mathbf{E}\,[Y \mid X=x]$$

Cette fonction est appelée fonction de régression et son estimation à partir de n copies indépendantes du couple (X,Y) est un problème fondamental en statistique.

On considère le cas où X est un vecteur de $\mathbf{R}^d$ et Y $\in$**R**. Si l'on ne connaît pas de forme paramétrique spécifique pour la fonction f alors les méthodes d'estimation classiques (moindres carrés, maximum de vraisemblance, ..) ne peuvent être utilisés directement.

On parle de problème d'estimation non-paramétrique.

La statistique non-paramétrique s'intéresse à l'estimation, à partir d'un nombre n d'observations, d'une fonction inconnue f $\in \Theta$, où Θ est un espace fonctionnel assez large. Ces 30 dernières années, la théorie de l'estimation non-paramétrique s'est développée autour des thèmes suivants :

1. Méthodes de constructions des estimateurs,

2. Propriétés statistiques de ces estimateurs,

3. Optimalité de ces estimateurs,

4. Estimation adaptative.

On se place dans le cadre d'**un modèle de régression** :

$$Y_i = f(X_i) + \xi_i, \text{ avec } i=1,\dots,n.$$

Hypothèses :

Les X_i sont déterministes, les variables ξ_i sont i.i.d. centrées et de variance σ^2.
Les Y_i sont aléatoires, on supposera les variables ξ_i indépendantes des Y_i i.i.d. centrées et de variance σ^2.
En l'absence de toute hypothèse sur la fonction de régression f, nous sommes dans un cadre non paramétrique. Nous avons plusieurs types de méthodes d'apprentissage pour la fonction f : l'estimation par des splines, les estimateurs à noyaux et les estimateurs par projection sur des bases orthonormées.
Nous allons étudier la méthode de l'estimateur par projection :

ESTIMATEUR PAR PROJECTION

On joint les calculs manuscrits pour cette première partie :

1) Estimateur par projection

On sait que $f_{N,\upsilon} = \sum_{j=1}^{N} \upsilon_j \varphi_j(x) \quad \forall x \in \mathbb{R}^d$

On note Φ_N la matrice $n \times N$ dont la $j^{ième}$ colonne est $\Phi_{\cdot j} = (\varphi_j(X_1), \dots, \varphi_j(X_n))^t$ pour $j \in [\![1, N]\!]$.

On suppose que $\Phi_N^t \Phi_N$ est une matrice définie strictement positive

a] Calcul de l'estimateur des moindres carrés $\hat{\upsilon}_{n,N}$ de υ:

On a $Y_i = f_{N,\upsilon}(X_i) + U_i$, alors on note $Y = \Phi_N \cdot \upsilon + U$

avec $U = \begin{pmatrix} U_1 \\ \vdots \\ U_n \end{pmatrix}$ et $\Phi_N = \begin{pmatrix} \varphi_1(X_1) & \cdots & \varphi_N(X_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(X_n) & \cdots & \varphi_N(X_n) \end{pmatrix} \in \mathcal{M}_{n,N}(.)$

Donc l'estimateur des moindres carrés de υ dans le modèle approché $Y_i = f_{N,\upsilon}(X_i) + U_i$ est donné par:

$$\hat{\upsilon}_{n,N} = [\Phi_N^t \Phi_N]^{-1} \Phi_N Y \quad (*)$$

De plus, on sait que $\Phi_N^t \Phi_N$ est définie strictement positive, et est également symétrique (*), donc $\Phi_N^t \Phi_N$ est inversible et $\hat{\upsilon}_{n,N}$ est unique

On peut alors en déduire un estimateur $\hat{f}_{n,N}(x)$ de $f(x)$:

$$\hat{f}_{n,N}(x) = f_{\hat{\upsilon}_{n,N}}(x) = \sum_{j=1}^{N} [\hat{\upsilon}_{n,N}]_j \varphi_j(x)$$

(* car $(\Phi_N^t \Phi_N)^t = \Phi_N^t \Phi_N$)

b] Soit $Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}$. Montrons que $(\hat{f}_{n,N}(X_1), \dots, \hat{f}_{n,N}(X_n))^t = A_N Y$

où $A_N = \Phi_N (\Phi_N^t \Phi_N)^{-1} \Phi_N^t$ est un projecteur orthogonal sur le sev de $\mathbb{R}^n$ engendré par les colonnes de la matrice Φ_N.

Par définition de $\hat{f}_{n,N}(x)$ et par l'estimateur trouvé dans la question précédente, on sait que:

$$\hat{f}_{n,N}(x) = \sum_{j=1}^{N} [\hat{\upsilon}_{n,N}]_j \varphi_j(x) = \hat{\upsilon}_{n,N}^t \varphi(x) = \varphi^t(x) \hat{\upsilon}_{n,N} \quad \forall x \in \mathbb{R}^d \text{ où } \varphi(x) = \begin{pmatrix} \varphi_1(x) \\ \vdots \\ \varphi_N(x) \end{pmatrix}$$
$$= \varphi^t(x) \cdot [\Phi_N^t \Phi_N]^{-1} \Phi_N Y \quad \text{par } (*)$$

Donc on a:

$$\begin{pmatrix} \hat{f}_{n,N}(X_1) \\ \vdots \\ \hat{f}_{n,N}(X_n) \end{pmatrix} = \begin{pmatrix} \varphi^t(X_1)\left[\Phi_N^t\Phi_N\right]^{-1}\Phi_N^t Y \\ \vdots \\ \varphi^t(X_n)\left[\Phi_N^t\Phi_N\right]^{-1}\Phi_N^t Y \end{pmatrix} = \begin{pmatrix} \varphi^t(X_1) \\ \vdots \\ \varphi^t(X_n) \end{pmatrix} \left[\Phi_N^t\Phi_N\right]^{-1}\Phi_N^t Y$$

Or par définition, on a $\varphi_j(X_i) = [\Phi_N]_{ij}$ et par définition de Φ_N, on a:

$$\begin{pmatrix} \hat{f}_{n,N}(X_1) \\ \vdots \\ \hat{f}_{n,N}(X_n) \end{pmatrix} = \Phi_N\left[\Phi_N^t\Phi_N\right]^{-1}\Phi_N^t Y = A_N Y$$

- Montrons maintenant que A_N est un projecteur orthogonal sur le sev de $\mathbb{R}^n$ engendré par les colonnes de la matrice Φ_N:

- projecteur: montrons que $A_N^2 = A_N$:

$$A_N^2 = A_N^t A_N = \left[\Phi_N\left[\Phi_N^t\Phi_N\right]^{-1}\Phi_N^t\right]^t\left[\Phi_N\left[\Phi_N^t\Phi_N\right]^{-1}\Phi_N^t\right]$$

$$A_N^2 = \left(\Phi_N\left(\left[\Phi_N^t\Phi_N\right]^{-1}\right)^t\Phi_N^t\right).\left(\Phi_N\left[\Phi_N^t\Phi_N\right]^{-1}\Phi_N^t\right)$$

$$A_N^2 = \Phi_N\left[\Phi_N^t\Phi_N\right]^{-1}\Phi_N^t\Phi_N\left[\Phi_N^t\Phi_N\right]^{-1}\Phi_N^t$$

$$A_N^2 = \Phi_N\left[\Phi_N^t\Phi_N\right]^{-1}\Phi_N^t$$

$$A_N^2 = A_N$$

- orthogonalité: On dit qu'un projecteur est orthogonal si sa matrice dans une base orthonormée est symétrique.

Ici, la matrice en question est A_N. A_N est bien dans une base orthonormée $(\varphi_1, \varphi_2, \dots)$ car Φ_N l'est.

Montrons que A_N est symétrique:

$$A_N^T = \left(\Phi_N\left[\Phi_N^t\Phi_N\right]^{-1}\Phi_N^t\right)^t = \Phi_N^{tt}\left(\left[\Phi_N^t\Phi_N\right]^{-1}\right)^t\Phi_N^t = \Phi_N\left[\Phi_N^t\Phi_N\right]^{-1}\Phi_N^t = A_N$$

Donc A_N est bien symétrique dans une base orthonormée.

A_N est bien un projecteur orthogonal sur le sous-espace vectoriel de $\mathbb{R}^n$ engendré par les colonnes de ϕ_N.

c) On a la matrice $\frac{1}{n}\phi_N^t\phi_N$ donnée par :

$$\frac{1}{n}\phi_N^t\phi_N = \frac{1}{n}\begin{pmatrix} \varphi_1(X_1) & \cdots & \varphi_1(X_n) \\ \vdots & & \vdots \\ \varphi_N(X_1) & \cdots & \varphi_N(X_n) \end{pmatrix}\begin{pmatrix} \varphi_1(X_1) & \cdots & \varphi_N(X_1) \\ \vdots & & \vdots \\ \varphi_1(X_n) & \cdots & \varphi_N(X_n) \end{pmatrix}$$

$$= \frac{1}{n}\begin{pmatrix} \sum_{i=1}^n \varphi_1^2(X_i) & \sum_{i=1}^n \varphi_1(X_i)\varphi_2(X_i) & \cdots & \sum_{i=1}^n \varphi_1(X_i)\varphi_N(X_i) \\ \sum_{i=1}^n \varphi_2(X_i)\varphi_1(X_i) & \sum_{i=1}^n \varphi_2^2(X_i) & & \vdots \\ \vdots & & \ddots & \\ \sum_{i=1}^n \varphi_N(X_i)\varphi_1(X_i) & \cdots & & \sum_{i=1}^n \varphi_N^2(X_n) \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{n}\sum_{i=1}^n \varphi_1^2(X_i) & \frac{1}{n}\sum_{i=1}^n \varphi_1(X_i)\varphi_2(X_i) & \cdots & \frac{1}{n}\sum_{i=1}^n \varphi_1(X_i)\varphi_N(X_i) \\ \frac{1}{n}\sum_{i=1}^n \varphi_2(X_i)\varphi_1(X_i) & \frac{1}{n}\sum_{i=1}^n \varphi_2^2(X_i) & & \vdots \\ \vdots & & \ddots & \\ \frac{1}{n}\sum_{i=1}^n \varphi_N(X_i)\varphi_1(X_i) & \cdots & & \frac{1}{n}\sum_{i=1}^n \varphi_N^2(X_N) \end{pmatrix}$$

OR, comme $X_i \overset{iid}{\sim} \mathcal{U}_{[0,1]^d}$, on a en faisant tendre $n \to +\infty$, par la loi des grands nombres

$$\frac{1}{n}\sum_{i=1}^n \varphi_k^2(X_i) \xrightarrow[n\infty]{\mathbb{P}} \mathbb{E}[\varphi_k^2(X_i)] \quad \forall k \in [\![1, N]\!]$$

or $\mathbb{E}[\varphi_k^2(X_i)] = \int_{[0,1]^d} \varphi_k^2(x)\,dx = 1$ car $X_i \overset{iid}{\sim} \mathcal{U}_{[0,1]^d}$ et $(\varphi_k)_{k\geq 1}$ est une BOH de $L^2([0,1]^d)$

De plus, on a $\forall \ell \neq k$,

$$\frac{1}{n}\sum_{i=1}^{n} \varphi_k(X_i)\varphi_\ell(X_i) \xrightarrow[n\to\infty]{\mathbb{P}} \mathbb{E}[\varphi_k(X_i)\varphi_\ell(X_i)] = \int_{[0,1]^d} \varphi_k(x)\varphi_\ell(x)\,dx = 0$$

car $(\varphi_k)_{k\geq 1}$ est une BON de $L^2([0,1]^d)$.

Ainsi on a bien que $\frac{1}{n}\phi_N^t \phi_N \xrightarrow[n\to\infty]{\mathbb{P}} I_{N\times N}$

Par la question b], on a :

$$\hat{f}_{n,N}(x) = \sum_{j=1}^{N} [\hat{\alpha}_{n,N}]_j \varphi_j(x) = \varphi^t(x)[\phi_N^t \phi_N]^{-1}\phi_N Y$$

En remplaçant $\phi_N^t\phi_N$ par l'approximation $n.I_{N\times N}$, on a :

$$\hat{f}_{n,N}(x) = \varphi^t(x)\, n.\, I_{N\times N}^{-1}\, \phi_N.Y = \varphi^t(x)\frac{1}{n}\phi_N Y$$

or $\tilde{v}_j = \frac{1}{n}\sum_{i=1}^{n} Y_i \varphi_j(X_i) = \frac{1}{n}\phi_N Y$

Donc on a bien : $\boxed{\hat{f}_{n,N}(x) = \sum_{j=1}^{N} \varphi_j(x)\tilde{v}_j}$ avec $\tilde{v}_j$ comme défini ci-dessus.

d] Montrons que $\tilde{v}_j$ est l'estimateur par la méthode des moments de v_j :

Soit $Y_i = f_{N,v}(X_i) + U_i$, Par la loi des grands nombres : $\frac{1}{n}\sum_{i=1}^{n} Y_i\varphi_j(X_i) \xrightarrow[n\to\infty]{\mathbb{P}} \mathbb{E}[Y_i\varphi_j(X_i)]$

On cherche $\tilde{v}_j$ tel que $\mathbb{E}[Y_i\varphi_j(X_i)] = \frac{1}{n}\sum_{i=1}^{n} Y_i\varphi_j(X_i)$ $\boxed{\begin{matrix}(X_i)_i \text{ et}\\ (Y_i)_i \text{ i.i.d}\end{matrix}}$

On a :

$$\begin{aligned}
\mathbb{E}[Y_i\varphi_j(X_i)] &= \mathbb{E}[(f_{N,v}(X_i) + U_i)\varphi_j(X_i)] \\
&= \mathbb{E}[f_{N,v}(X_i)\varphi_j(X_i)] + \mathbb{E}[U_i\varphi_j(X_i)] \quad \text{par linéarité de l'espérance} \\
&= \mathbb{E}\Big[\sum_{k=1}^{N} v_k \varphi_k(X_i)\varphi_j(X_i)\Big] + \mathbb{E}\Big[\underbrace{\mathbb{E}[U_i|X_i]}_{=0 \text{ car on suppose } \mathbb{E}[U_i|X_i]=0}\varphi_j(X_i)\Big] \qquad v_j \\
&= \sum_{k=1}^{N} \int_{[0,1]^d} v_k \varphi_k(x)\varphi_j(x)\,dx \\
&= \sum_{k=1}^{N} v_k \underbrace{\int_{[0,1]} \varphi_k(x)\varphi_j(x)\,dx}_{\circledast}
\end{aligned}$$

OR $(\varphi_k)_k$ BON de $L^2([0,1]^d)$ donc $\circledast = \begin{cases} 1 \text{ si } k=j \\ 0 \text{ sinon} \end{cases}$

Donc $\mathbb{E}[Y_i \varphi_j(X_i)] = \hat{v}_j = \frac{1}{n}\sum_{i=1}^{n} Y_i \varphi_j(X_i)$

de $\tilde{v}_j$ est bien l'estimateur par la méthode des moments de v_j

e] On suppose maintenant $Y_i = f(X_i) + U_i$ où $\begin{cases} U_i \perp\!\!\!\perp X_i \ \forall i \in [\![1,n]\!] \\ U_i \text{ iid} \end{cases}$

on suppose que la variance $\sigma^2 = \mathbb{E}[U_i]$ existe et est connue.

on veut calculer le biais $b_{n,N}(x)$ de l'estimateur $\hat{f}_{n,N}(x)$:

$$\mathbb{E}[\hat{f}_{n,N}] = \mathbb{E}\left[\sum_{j=1}^{N} \varphi_j(x)\tilde{v}_j\right] = \sum_{j=1}^{N} \varphi_j(x)\,\mathbb{E}[\tilde{v}_j] \quad \text{par linéarité de l'intégrale}$$

$$= \sum_{j=1}^{N} \varphi_j(x)\,\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} Y_i \varphi_j(X_i)\right]$$

$$= \frac{1}{n}\sum_{j=1}^{N} \varphi_j(x) \sum_{i=1}^{n} \mathbb{E}[Y_i \varphi_j(X_i)] = \sum_{j=1}^{N} \varphi_j(x)\,\mathbb{E}[Y_1 \varphi_j(X_1)] \quad \text{car } (X_i, Y_i) \text{ iid}$$

$$= \sum_{j=1}^{N} \varphi_j(x)\,\mathbb{E}[f(X_1)\varphi_j(X_1)] + \varphi_j(x)\,\mathbb{E}[U_1 \cdot \varphi_j(X_1)] \quad \text{car } Y_i = f(X_i) + U_i$$

$$\overset{U_1 \perp\!\!\!\perp X_1}{=} \sum_{j=1}^{N} \mathbb{E}[f(X_1)\varphi_j(X_1)]\varphi_j(x) + \underbrace{\mathbb{E}[U_1]}_{=0} \cdot \mathbb{E}[\varphi_j(X_1)]\varphi_j(x)$$

car $\sigma^2 = \mathbb{E}(U_1^2) - \underbrace{\mathbb{E}(U_1)^2}_{=0} = \mathbb{E}(U_1^2)$

$$= \sum_{j=1}^{N} \mathbb{E}\left[\sum_{k=1}^{\infty} v_k \varphi_k(X_1)\varphi_j(X_1)\right]\varphi_j(x) \quad \text{car } f(X_1) = \sum_{k=1}^{\infty} v_k \varphi_k(X_1)$$

$$= \sum_{j=1}^{N}\sum_{k=1}^{\infty} v_k \underbrace{\mathbb{E}[\varphi_k(X_1)\varphi_j(X_1)]}\varphi_j(x)$$

$$= \begin{cases} 1 & \text{si } k=j \\ 0 & \text{sinon} \end{cases} \quad \text{car } (\varphi_k)_k \text{ BON de } L^2([0,1]^d)$$

$$= \sum_{j=1}^{N} v_j \varphi_j(x)$$

$$= f_{N,v}(x)$$

Quand $N \to +\infty$, $f_{N,v}(x) \to f(x)$, donc le biais tend vers 0 quand $N \to +\infty$.

f] Pour toute fonction $h \in L^2([0,1]^d)$, on note $\|h\| = \left[\int_{[0,1]^d} h^2(x)\,dx\right]^{1/2}$

Montrons que le risque quadratique $R(\hat{f}_{n,N}, f) = \mathbb{E}[\|\hat{f}_{n,N} - f\|^2]$

est borné par $\sum_{j=N+1}^{\infty} v_j^2 + \frac{N(\|f\|_\infty^2 + \sigma^2)}{n}$

$$R(\tilde{f}_{n,N}, f) = \mathbb{E}\left[\|\tilde{f}_{n,N} - f\|^2\right] = \mathbb{E}\left[\|\tilde{f}_{n,N} - \mathbb{E}(\tilde{f}_{n,N}) + \mathbb{E}(\tilde{f}_{n,N}) - f\|^2\right]$$

$$= \mathbb{E}\left[\|\tilde{f}_{n,N} - \mathbb{E}[\tilde{f}_{n,N}]\|^2\right] + \underbrace{2\mathbb{E}\left[(\tilde{f}_{n,N} - \mathbb{E}[\tilde{f}_{n,N}]).[\mathbb{E}[\tilde{f}_{n,N}] - f]\right]}_{(*)}$$
$$+ \mathbb{E}\left[\|\mathbb{E}[\tilde{f}_{n,N}] - f\|^2\right]$$

$$(*) = 2(\mathbb{E}[\tilde{f}_{n,N}] - f)\,\mathbb{E}\left[(\tilde{f}_{n,N} - \mathbb{E}(\tilde{f}_{n,N})\right] = 2(\mathbb{E}(\tilde{f}_{n,N}) - f)\underbrace{[\mathbb{E}[\tilde{f}_{n,N}] - \mathbb{E}[\tilde{f}_{n,N}]]}_{=0}$$
$$= 0$$

$$\Rightarrow R(\tilde{f}_{n,N}, f) = \mathbb{E}\left[\|\tilde{f}_{n,N} - f\|^2\right] = \mathbb{E}\left[\|\tilde{f}_{n,N} - \mathbb{E}[\tilde{f}_{n,N}]\|^2\right] + \mathbb{E}\left[\|\mathbb{E}(\tilde{f}_{n,N}) - f\|^2\right]$$

- $\mathbb{E}\left[\|\tilde{f}_{n,N} - \mathbb{E}(\tilde{f}_{n,N})\|^2\right] = \mathbb{E}\left[\int (\tilde{f}_{n,N} - \mathbb{E}[\tilde{f}_{n,N}])^2(x)\,dx\right]$

$$= \int_{[0,1]^d} \mathbb{E}\left[(\tilde{f}_{n,N} - \mathbb{E}(\tilde{f}_{n,N}))^2\right](x)\,dx \quad \text{par Fubini}$$

* $\mathbb{E}\left[(\tilde{f}_{n,N} - \mathbb{E}[\tilde{f}_{n,N}])^2\right] = \text{Variance}(\tilde{f}_{n,N}) = V(\tilde{f}_{n,N})$

OR $V(\tilde{f}_{n,N}) = V\left(\sum_{j=1}^{N} \tilde{v}_j \varphi_j(x)\right) = V\left(\sum_{j=1}^{N} \frac{1}{n} \sum_{i=1}^{n} Y_i \varphi_j(X_i) \varphi_j(x)\right)$

(X_i, Y_i) i.i.d et $Y_i = f(X_i) + U_i$ et $U_i \perp\!\!\!\perp X_i \Rightarrow \text{Cov}(X_i, U_i) = 0$

$$= \frac{1}{n} V\left(\sum_{j=1}^{N} f(X_1)\varphi_j(X_1)\varphi_j(x)\right) + \frac{1}{n} V\left(U_1 \sum_{j=1}^{N} \varphi_j(X_1)\varphi_j(x)\right)$$

$$\leq \frac{1}{n}\,\mathbb{E}\left[\sum_{j=1}^{N} f(X_1)\varphi_j(X_1)\varphi_j(x)\right]^2 + \frac{1}{n}\,\mathbb{E}\left[U_1^2\left(\sum_{j=1}^{N} \varphi_j(X_1)\varphi_j(x)\right)^2\right]$$

$$\leq \frac{\|f\|_\infty^2}{n}\,\mathbb{E}\left[\left(\sum_{j=1}^{N} \varphi_j(X_1)\varphi_j(x)\right)^2\right] + \frac{1}{n}\cdot\sigma^2\cdot\mathbb{E}\left[\left(\sum_{j=1}^{N} \varphi_j(X_1)\varphi_j(x)\right)^2\right]$$

$$\leq \frac{\|f\|_\infty^2 + \sigma^2}{n}\,\mathbb{E}\left[\left(\sum_{j=1}^{N} \varphi_j(X_1)\varphi_j(x)\right)^2\right]$$

Donc $\int_{[0,1]^d} \mathbb{E}\left(\left(\sum_{j=1}^{N} \varphi_j(X_1)\varphi_j(x)\right)^2\right) dx$

$$= \int_{[0,1]^d} \mathbb{E}\left[\sum_{j=1}^{N} \varphi_j^2(X_1)\varphi_j^2(x) + \sum_{\substack{k,j=1 \\ k \neq j}}^{N} \varphi_k(X_1)\varphi_j(X_1)\varphi_k(x)\varphi_j(x)\right] dx$$

D'où $\int_{[0,1]^d} \mathbb{E}\left[\left(\sum_{j=1}^{N} \varphi_j(X_1)\varphi_j(x)\right)^2\right] dx$

$= \int_{[0,1]^d} \sum_{j=1}^{N} \underbrace{\mathbb{E}[\varphi_j^2(X_1)]}_{=1 \text{ car } (\varphi_k)_k \text{ BON de } L^2([0,1]^d)} \varphi_j^2(x)\,dx + \sum_{\substack{j,k=1 \\ j\neq k}}^{N} \mathbb{E}[\varphi_j(X_1)\varphi_k(X_1)] \underbrace{\int_{[0,1]^d} \varphi_j(x)\varphi_k(x)\,dx}_{=0 \text{ car } (\varphi_k)_k \text{ BON}}$

$= \sum_{j=1}^{N} \underbrace{\int_{[0,1]^d} \varphi_j^2(x)\,dx}_{=1 \text{ car } (\varphi_k)_k \text{ BON de } L^2([0,1]^d)} = N$

Ainsi $\int_{[0,1]^d} V(\tilde{f}_{n,N})\,dx \leq \frac{\|f\|_\infty^2 + \sigma^2}{n} \cdot N$

- $\|\mathbb{E}[\tilde{f}_{n,N}] - f\|^2 = \int_{[0,1]^d} (\mathbb{E}[\tilde{f}_{n,N}] - f)^2\,dx$

$= \underbrace{\int_{[0,1]^d} \left[\mathbb{E}\left[\sum_{j=1}^{N} (\tilde{v}_j - v_j)\varphi_j(x)\right]\right]^2 dx}_{①} - \underbrace{2\int_{[0,1]^d} \mathbb{E}\left[\sum_{j=1}^{N} (\tilde{v}_j - v_j)\varphi_j(x)\right] \sum_{j=N+1}^{\infty} v_j\varphi_j(x)\,dx}_{②=0} + \underbrace{\int_{[0,1]^d} \left(\sum_{j=N+1}^{+\infty} v_j\varphi_j(x)\right)^2 dx}_{③}$

② $= 2 \sum_{j=1}^{N} \sum_{k=N+1}^{\infty} \mathbb{E}[\tilde{v}_j - v_j] v_k \underbrace{\int_{[0,1]^d} \varphi_j(x)\varphi_k(x)\,dx}_{=0 \text{ car } j\neq k \text{ et } (\varphi_k)_k \text{ BON de } L^2([0,1]^d)} = 0$

③ $= \int_{[0,1]^d} \sum_{j=N+1}^{+\infty} v_j\varphi_j(x) \sum_{k=N+1}^{+\infty} v_k\varphi_k(x)\,dx = \int \sum_{j=N+1}^{+\infty} v_j^2\varphi_j^2(x)\,dx \text{ car } \int_{[0,1]^d} \varphi_j(x)\varphi_k(x)\,dx = 0 \ \forall j\neq k$

$= \sum_{j=N+1}^{\infty} v_j^2 \int_{[0,1]^d} \varphi_j^2(x)\,dx = \sum_{j=N+1}^{+\infty} v_j^2$

① $= \int_{[0,1]^d} \mathbb{E}\left[\sum_{j=1}^{N} (\tilde{v}_j - v_j)\varphi_j(x)\right] \mathbb{E}\left[\sum_{k=1}^{N} (\tilde{v}_k - v_k)\varphi_k(x)\right] dx$

$= \int_{[0,1]^d} \sum_{j=1}^{N} \varphi_j^2(x)\,\mathbb{E}^2[\tilde{v}_j - v_j]\,dx \quad \text{car} \quad \int_{[0,1]^d} \varphi_j(x)\varphi_k(x) = 0 \ \forall j\neq k$

$$\textcircled{5} = \sum_{j=1}^{N} \mathbb{E}^2[\tilde{v}_j - v_j] \underbrace{\int_{[0,1]^d} \varphi_j^2(x)\,dx}_{=1 \text{ car } (\varphi_k)_k \text{ BON de } L^2([0,1]^d)}$$

OR

$$\sum_{j=1}^{N} \mathbb{E}^2[\tilde{v}_j - v_j] = \sum_{j=1}^{N} \left[\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} Y_i \varphi_j(X_i)\right] - v_j\right]^2$$

car (X_i, Y_i) iid

$$= \sum_{j=1}^{N} \left[\underbrace{\mathbb{E}[Y_1 \varphi_j(X_1)]}_{=v_j} - v_j\right]^2 = 0$$

$\Rightarrow \textcircled{5} = 0$

D'où $R(\tilde{f}_{n,N}, f) \leq \frac{N(\|f\|_\infty^2 + \tau^2)}{n} + \sum_{j=N+1}^{+\infty} v_j^2$

$R(\tilde{f}_{n,N}, f)$ était le risque quadratique, si on connaissait la fonction f, on voudrait minimiser ce risque, donc on choisirait un N tel que $\frac{N(\|f\|_\infty^2 + \tau^2)}{n} + \sum_{j=N+1}^{\infty} v_j^2$ soit minimum.

ie on cherchait à minimiser $N \mapsto \sum_{j=N+1}^{+\infty} v_j^2 + N.\frac{(\|f\|_\infty^2 + \tau^2)}{n}$

On remarque que
- $\sum_{j=N+1}^{\infty} v_j^2$ est décroissante par rapport à N
- $N\frac{(\|f\|_\infty^2 + \tau^2)}{n}$ est croissant par rapport à N

Donc on choisit le N qui donne $\left|\sum_{j=N+1}^{\infty} v_j^2\right| = \left|\frac{N(\|f\|_\infty^2 + \tau^2)}{n}\right|$

9] On suppose que f est bornée, ie $\sup_x |f(x)| \leq M$

On suppose de plus que l'on connait un entier $k > 0$ et un réel $L > 0$ tq $\sum_{j=1}^{\infty} j^{2k} v_j^2 \leq L$. Mq $\sum_{j>N} v_j^2 \leq L N^{-2k}$: ($j > N \Leftrightarrow \frac{j}{N} > 1$)

$$\sum_{j>N} v_j^2 \leq \sum_{j>N} \frac{j^{2k}}{N^{2k}} v_j^2 \leq \sum_{j=1}^{+\infty} \frac{j^{2k}}{N^{2k}} v_j^2 \leq \frac{L}{N^{2k}} = L N^{-2k}$$

Or on avait montré dans la question précédente que :

$$R(\tilde{f}_{n,N}, f) \leq N\frac{(\|f\|_\infty^2 + \tau^2)}{n} + \underbrace{\sum_{j=N+1}^{\infty} v_j^2}_{\sum_{j>N}^{\infty} v_j^2 \leq LN^{-2k}}$$

Donc $R(\tilde{f}_{n,N}, f) \leq \underbrace{\frac{N(M+\tau^2)}{n} + LN^{-2k}}_{A(N)}$

On cherche à trouver le N qui minimise $N \mapsto A(N)$:

A est dérivable par rapport à N et on a $A'(N) = \frac{M+\tau^2}{n} - 2kLN^{-2k-1}$

$$A(\tilde{N}) = 0 \iff \frac{M+\tau^2}{n} = 2k\tilde{N}^{-2k-1} \cdot L \iff \tilde{N}^{k+1} = \frac{2kLn}{M+\tau^2}$$

$$\iff \tilde{N} = \left(\frac{2kLn}{M+\tau^2}\right)^{\frac{1}{2k+1}}$$

et $A''(N) = 2(2k+1)kLN^{-2k-2} \geq 0$ Donc $\tilde{N}$ est un minimum

$\tilde{N}$ minimise $R(\hat{f}_{n,N}, f)$

h] On suppose que par un entier naturel $N_0 < n$, le vecteur $(f(X_1), \dots, f(X_n))^t$ appartient à l'espace vectoriel engendré par les vecteurs $\{(\varphi_j(X_1), \dots, \varphi_j(X_n))^t ; 1 \leq j \leq N_0\}$. (H)

On veut montrer que $\hat{\tau}^2_{N_0}$ est un estimateur sans biais, ie $\mathbb{E}[\hat{\tau}^2_{N_0}] = \tau^2_{N_0}$:

$$\mathbb{E}[\hat{\tau}^2_{N_0}] = \mathbb{E}\left[\frac{1}{n-N_0}\|(I_{n\times n} - A_{N_0})Y\|^2\right] = \frac{1}{n-N_0}\mathbb{E}\left[\|(I_{n\times n} - A_{N_0}).Y\|^2\right]$$

On a :

$(I_{n\times n} - A_{N_0})Y = Y - A_{N_0}Y$ où $A_{N_0} = \Phi_{N_0}(\Phi_{N_0}^t\Phi_{N_0})^{-1}\Phi_{N_0}^t Y$ par la question b)

$= Y - \Phi_{N_0}(\Phi_{N_0}^t\Phi_{N_0})^{-1}\Phi_{N_0}^t Y$ or $Y = \Phi_{N_0}v + u$

$= \Phi_{N_0}v + U - \Phi_{N_0}\hat{v}_{n,N_0}$ (d'après la question a), $\hat{v}_{n,N} = (\Phi_N^t\Phi_N)^{-1}\Phi_N Y$)

d'après l'hypothèse (H)

Alors

$$(I_{nxn} - A_{N_0})Y = \Phi_{N_0}(\theta - \hat{\theta}_{n,N_0}) + U = \Phi_{N_0}\left(\theta - \left(\theta + (\Phi_{N_0}^t \Phi_{N_0})^{-1}\Phi_{N_0}^t U\right)\right) + U$$

car

$$\hat{\theta}_{n,N_0} - \theta = (\Phi_{N_0}^t \Phi_{N_0})^{-1}\Phi_{N_0}^t Y - \theta \overset{(Y = \Phi_{N_0}\theta + U)}{=} \underbrace{(\Phi_{N_0}^t \Phi_{N_0})^{-1}(\Phi_{N_0}^t \Phi_{N_0})}_{I_{N_0 \times N_0}}\theta + (\Phi_{N_0}^t \Phi_{N_0})^{-1}\Phi_{N_0}^t U - \theta$$

$$= (\Phi_{N_0}^t \Phi_{N_0})^{-1}\Phi_{N_0}^t U$$

Donc $[I_{nxn} - A_{N_0}]Y = \Phi_{N_0}\left(-(\Phi_{N_0}^t \Phi_{N_0})^{-1}\Phi_{N_0}^t \cdot U\right) + U = -A_{N_0} \cdot U + U$

$$= [I_{nxn} - A_{N_0}]U$$

Ainsi, on a :

$$\mathbb{E}\left[\|(I_{nxn} - A_{N_0})Y\|^2\right] = \mathbb{E}\left[\|Y - A_{N_0}Y\|^2\right]$$

$$= \mathbb{E}\left[\|(I_{nxn} - A_{N_0})U\|^2\right]$$

$$= \mathbb{E}\left[U^t (I_{nn} - A_{N_0})^t (I_{nn} - A_{N_0}) U\right]$$

or A_{N_0} est un projecteur Donc $\begin{cases} (I_n - A_{N_0}) = (I_{nn} - A_{N_0})^t \\ (I_{nn} - A_{N_0})^2 = (I_{nn} - A_{N_0}) \end{cases}$

Dès

$$\mathbb{E}\left[\|(I_{nn} - A_{N_0})Y\|^2\right] = \mathbb{E}\left[U^t (I_{nn} - A_{N_0}) U\right] \overset{(*)}{=} \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}\left[U_i U_j (\delta_{ij} - a_{ij})\right]$$

où $\begin{cases} \delta_{ij} = \begin{cases} 1 \text{ si } i=j \\ 0 \text{ sinon} \end{cases} \\ a_{ij} = (A_{N_0})_{ij} \end{cases}$ et comme $U_i \perp\!\!\!\perp U_j \ \forall i \neq j$, $\mathbb{E}[U_i] = 0$

on a

$$(*) = \sum_{i=1}^{n} \mathbb{E}\left[U_i^2 (1 - a_{ii})\right] = \sigma^2 \sum_{i=1}^{n} (1 - a_{ii}) = \sigma^2 \left(n - \mathrm{Tr}(A_{N_0})\right) = \sigma^2 (n - N_0)$$

car $\mathrm{Tr}(A_{N_0}) = N_0$ car A_{N_0} était un projecteur, sa trace est égale à la dimension de l'espace sur lequel on projette, ie

$$\mathrm{Tr}(A_{N_0}) = \dim\left(\mathrm{Im}(\Phi_{N_0})\right) = \min(N_0, n) = N_0 \text{ car } N_0 < n.$$

car $\Phi_{N_0} \in \mathcal{M}_{n,N_0}(\mathbb{R})$

SIMULATION

On codera cette partie en Python.

Dans un premier temps, on importe toutes les bibliothèques dont on aura besoin pour travailler en Python :

```
from math import sin
from math import cos
from math import pi
from random import random
from random import gauss
import matplotlib.pyplot as plt
import numpy as np
```

On se place désormais dans le cas unidimensionnel (d=1) et on choisit comme base orthonormée de L2([0,1]) la base trigonométrique $\phi 1(x) = 1$et :

$$\phi j(x) = \left\{\sqrt{2}cos(2k\pi x)\ si\ k = (j+1)/2 \in Z\ ,\ \sqrt{2}sin(2k\pi x)\ si\ k = j/2 \in Z\right\} j = 1, 2,...$$

On veut vérifier que la méthode de sélection automatique du niveau de troncature donne des résultats satisfaisants.

Pour cela, on pose n=100 et on génère n variables iid $X_1, \ldots, X_n$ de loi uniforme sur [0,1]

```
def vecteur_uniforme_0_1(n):
    X =[]
    for i in range (n):
        x = random()
        X.append(x)
    return X
```

On choisit $f(x) = (x^2 2^{(x-1)} - (x - 0.5)^3).\sin(10.x)$, $\sigma = 0.2$. On code la fonction f telle que :

```
def f(x):
    a = x**2
    b = 2**(x-1)
    c = (x-0.5)**3
    y = (a*b-c)*sin(10*x)
    return y
```

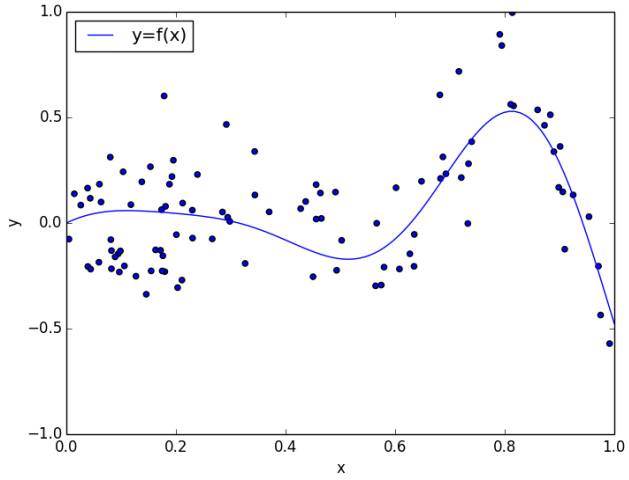
On veut maintenant calculer le vecteur $Y = (f(X_1, \ldots, f(X_n))^t + \sigma.\xi$ où ξest un vecteur gaussien $N(0, I_{n,n})$.

```
def graphe_f(xmin, xmax, ymin, ymax, n):
    x=np.linspace(xmin, xmax, n)
    y=[]
    for i in range (n):
        y.append(f(x[i]))
    plt.plot(x, y, label='y=f(x)')
    plt.legend(loc='upper left')
    plt.xlabel('x')
    plt.ylabel('y')
    plt.axis([xmin ,xmax, ymin, ymax])

def vecteur_gaussien(m, sigma, n):
    X=[]
    for i in range (n):
        X.append(gauss(m, sigma))
    return X

def vecteur_Y(X, m, sigma, n):
    Y1 = []
    for i in range (n):
        Y1.append(f(X[i]))
    Y2 = vecteur_gaussien(m, sigma, n)
    Y=[]
    for i in range (n):
       Y.append(Y1[i]+Y2[i])
    return Y
```

On trace maintenant le nuage des points (X_i, Y_i), i = 1, 2, …, n avec la courbe de f dans le même repère orthogonal :

```
def graphe_X_Y_f(X, Y, xmin, xmax, ymin, ymax, n):
    graphe_f(xmin, xmax, ymin, ymax, n)
    plt.scatter(X, Y)
```

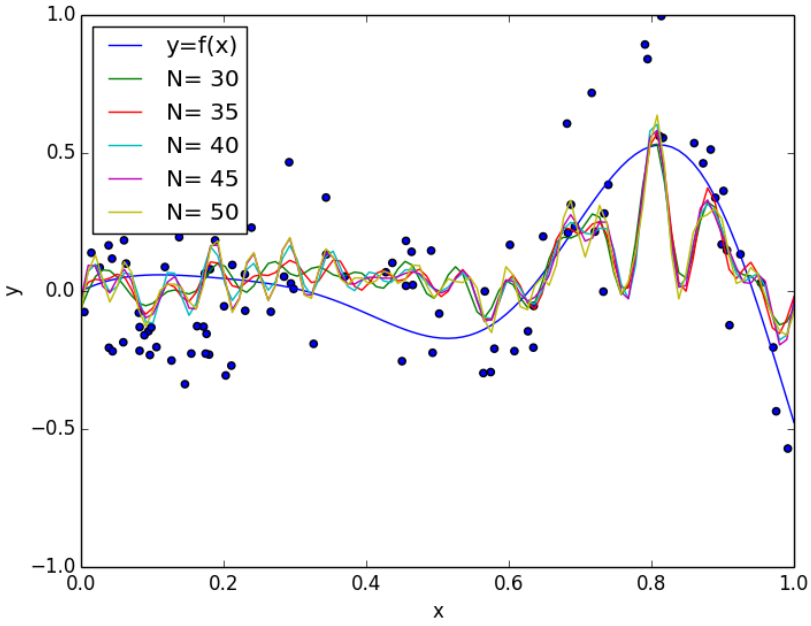
On obtient le graphe ci-dessous :



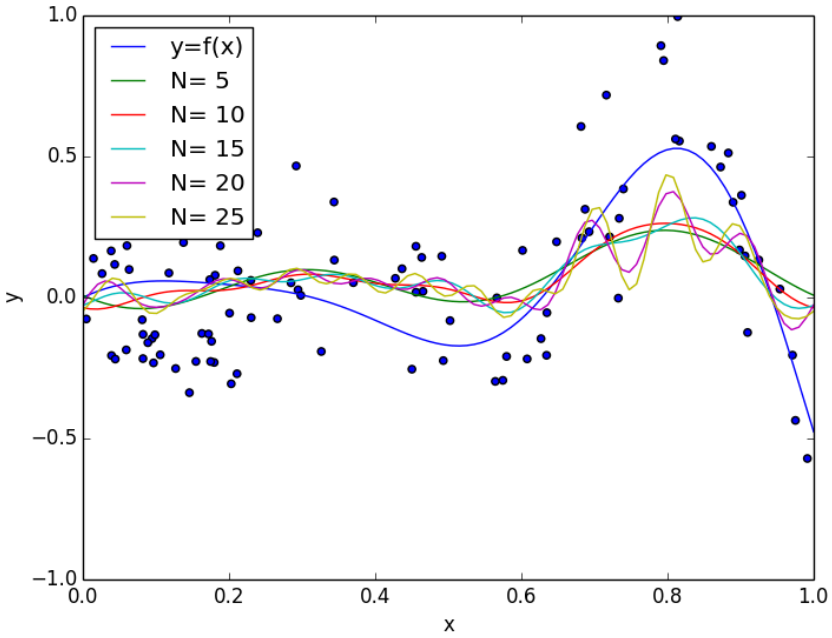
Maintenant, on calcule l'estimateur $f^*_{n,N}$ de f pour N = 5, 10, 15, 20, … , 50 et on trace la courbe superposée de f et du nuage des points $\{(X_i, Y_i)\}$.

On détermine visuellement la valeur de N qui correspond au meilleur estimateur.

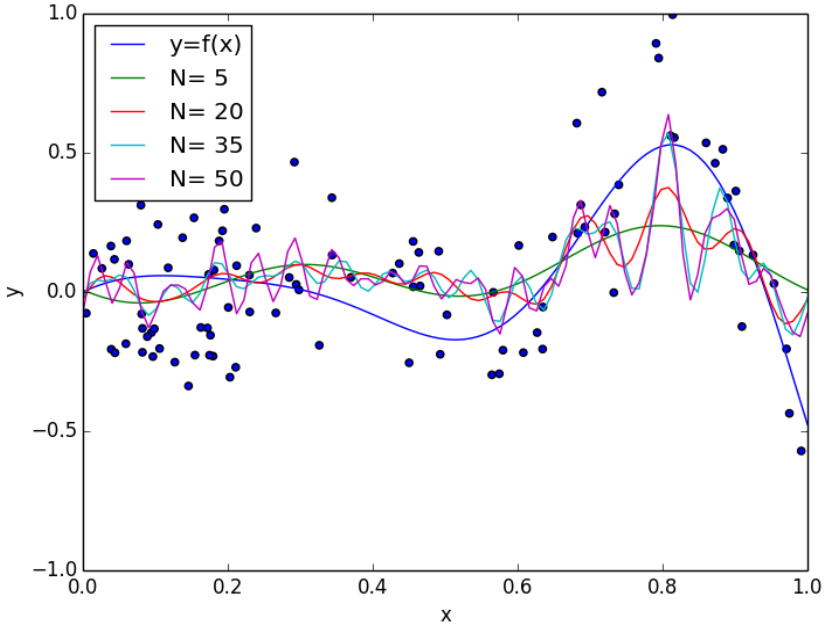
Pour N = 30, 35, 40, 45, 50 :



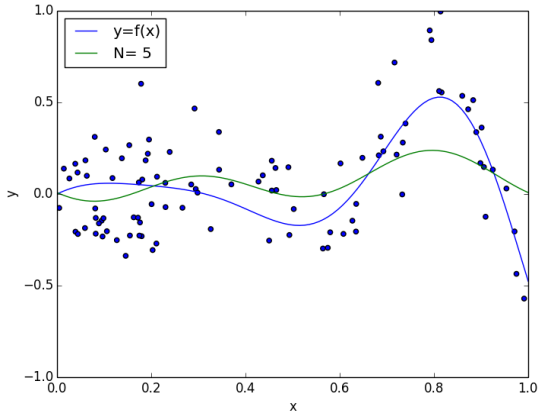
Pour N = 5, 10, 15, 25 :

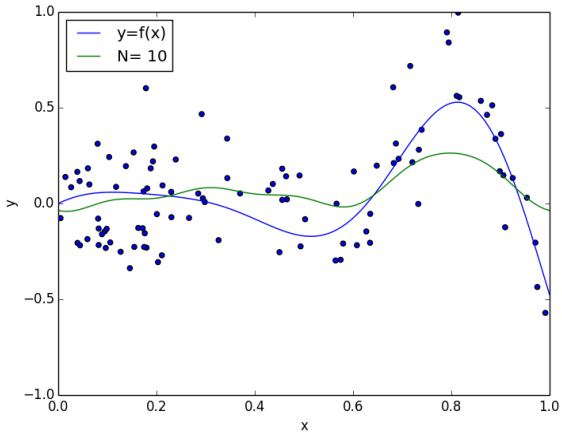


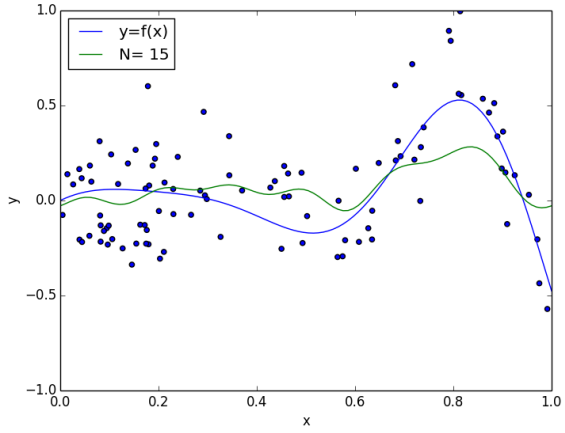
Pour N = 5, 20, 35, 50 :

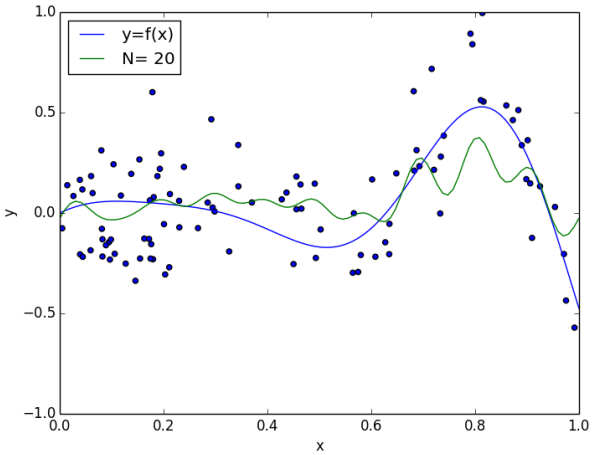


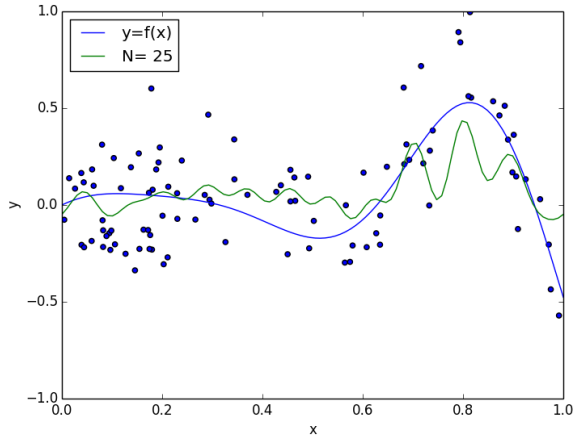
En effectuant les simulations pour les N allant de 5 à 50, on obtient :

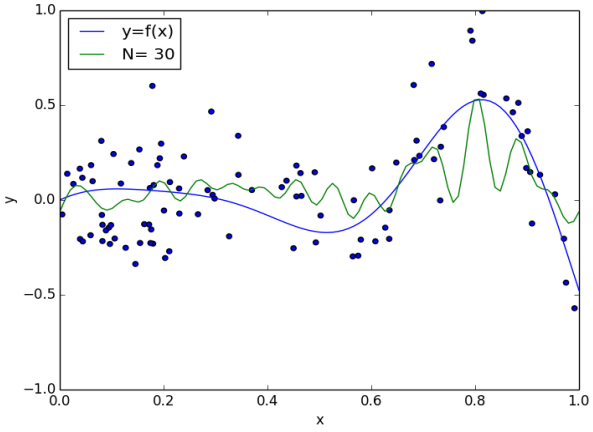


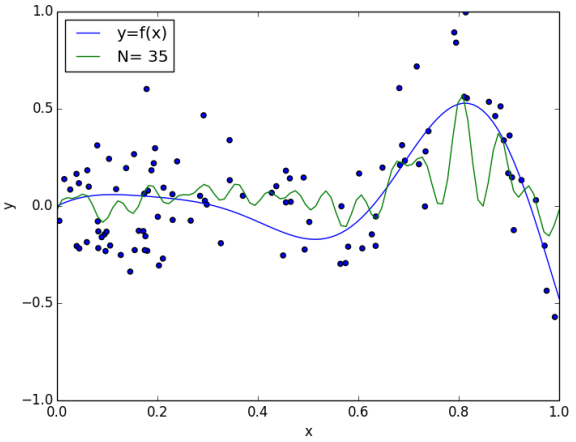


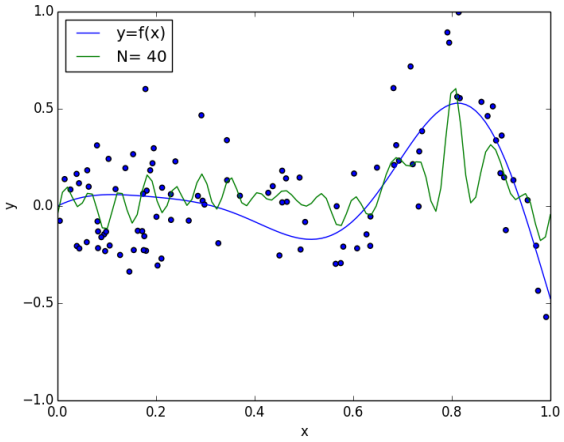


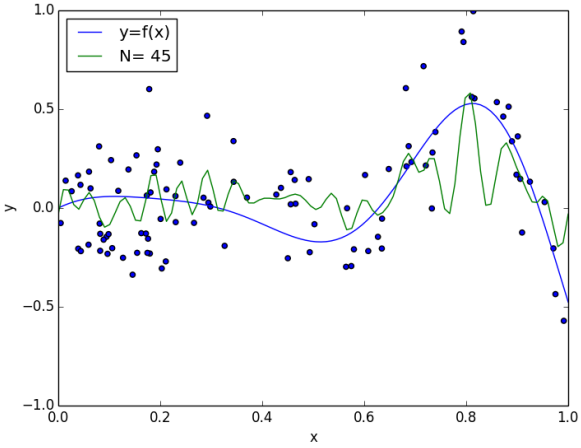


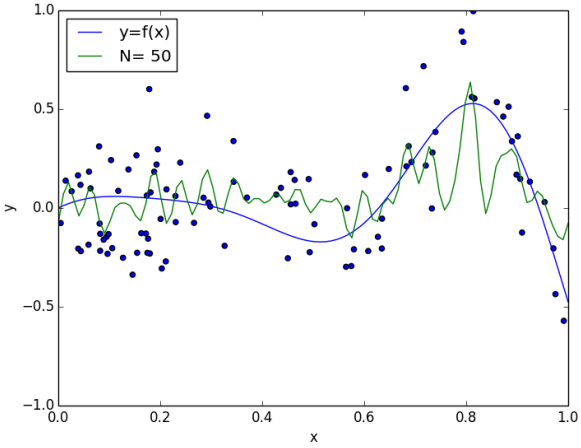












Pour les graphiques ci-dessus, on a utilisé le code :

```
def phi(j, x):
    r = 0
    if j==0:
        r = 1
    else:
        if j%2 == 1:
            r = 2**(1/2)*cos((j+1)*pi*x)
        else:
            r = 2**(1/2)*sin(j*pi*x)
    return r

def coef_vj(j, X, Y, n):
    vj = 0
    for i in range (n):
        vj += Y[i]*phi(j, X[i])
    vj /= n
```

```
    return vj


def graphe_f_tild(N, xmin, xmax, X, Y, n):
    x=np.linspace(xmin, xmax, n)
    y=[]
    for i in range (n):
        f_tild = 0
        for j in range (N):
            f_tild += coef_vj(j, X, Y, n)*phi(j, x[i])
        y.append(f_tild)
    plt.plot(x, y, label="N= "+ str(N))
    plt.legend(loc='upper left')
    plt.savefig("/Users/IsBismuth/Google
Drive/projet_apprentissage/f_tild_Nrt="+str(N))
```

Visuellement, le N qui correspond au meilleur estimateur est N = 10.

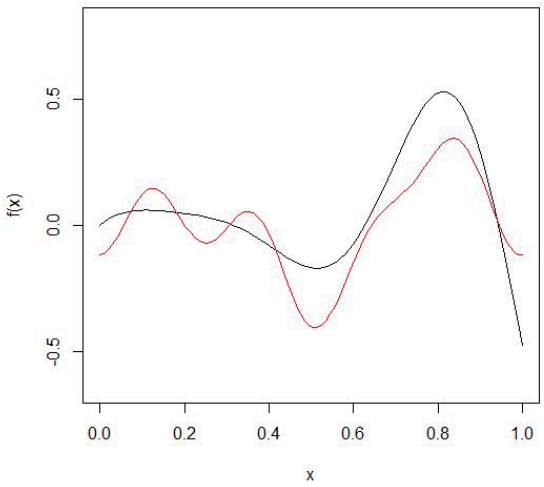
On calcule maintenant l'estimateur $\sigma^{\wedge 2}_{N0}$ pour N0 = 50 :

```
def matrice_phi(X, N, n):
    Phi=np.zeros((n, N), dtype='f')
    for i in range (n):
        for j in range (N):
            Phi[i][j] = phi(j, X[i])
    return Phi


def matrice_A(X, N, n):
    Phi = matrice_phi(X, N, n)
    Phi_t = Phi.T
    A = np.dot(Phi_t, Phi)
    A = np.linalg.inv(A)
    A = np.dot (Phi, A)
    A = np.dot (A, Phi_t)
    return A


def estimateur_sigma(X, Y, N0, n):
    A = matrice_A(X, N0, n)
    I = np.eye(n, n)
    A = I - A
    A = np.dot(A, Y)
    x = np.linalg.norm(A, 2)
    x /= (n-N0)
    return x
```

```
def estimateur_N(X, Y, N0, n):
    sigma=estimateur_sigma(X, Y, N0, n)
    MyList=[]
    MyList
    for i in range (50):
        A = matrice_A(X, i+1, n)
        I = np.eye(n, n)
        A = I - A
        A = np.dot(A, Y)
        x = np.linalg.norm(A, 2)
        x -= (n-2*(i+1))*sigma
        MyList.append(x)
    r = MyList.index(min(MyList))
    print( "l'estimateur de N est " + str((r+1)))
    return r+1
```

Si on simule 100 fois le code, on a une moyenne de 0.04 pour $\sigma^{\wedge 2}_{N0}$.
On trouve un ^N moyen autour de 10 ce qui correspond à la valeur trouvée visuellement.

On trace la courbe de l'estimateur $f^*_{n,N}$ superposée sur la courbe de f et on obtient :

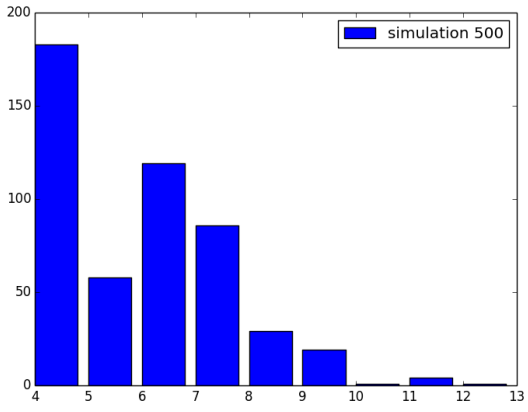


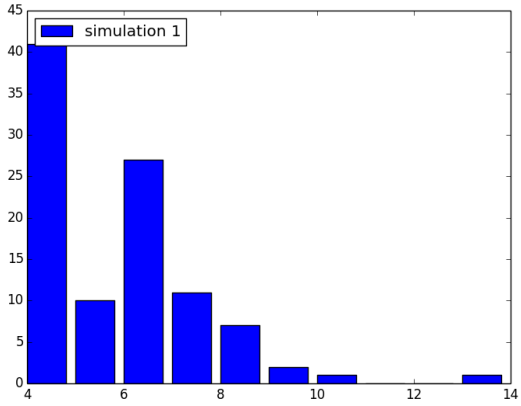
On peut voir que la courbe rouge - courbe de $f^*_{n,N}$ - est plutôt proche de la courbe noire - courbe de f.

Pour vérifier que la valeur de ^N est bien celle estimée, on réitère la simulation 100 fois et on obtient l'histogramme suivant avec le code :

```
def save_vector(mylovelyfile, X, n):
    myfile = open(mylovelyfile, "wb")
    for i in range (n):
        myfile.write(str(X[i]))
        myfile.write("\r\n")
    myfile.close()
def repartition_estimateur_N(nb_simulation, dossier):
    MyList=[]
    ListAbs=[]
    ListOrd=[]
    for i in range (nb_simulation):
        X = vecteur_uniforme_0_1(100)
        save_vector(dossier + "X"+ str(i), X, 100)
        Y = vecteur_Y(X, 0, 0.2, 100)
        save_vector(dossier + "Y"+ str(i), Y, 100)
        n_chapeau = estimateur_N(X, Y, 50, 100)
        MyList.append(n_chapeau)
        ListAbs.append(i+1)
    for i in range (1,nb_simulation+1):
        x = MyList.count(i)
        ListOrd.append(x)
    plt.bar(ListAbs, ListOrd, label = 'simulation 500')
    plt.legend(loc='upper right')
    return MyList
```

N = 500 N = 100





Nous pouvons voir sur l'histogramme que le N optimal est approximativement 8.