Echantillonnage optimal pour l'apprentissage

Rapport

Membre du projet :
Jules GOUY
Aurelien MOUNIER
Benjamin PINEAU

Professeur
Charles Miranda
charles.miranda@ec-nantes.fr
Anthony Nouy
anthony.nouy@ec-nantes.fr

Table des matières

1 Introduction

1.1 Motivation

On veut estimer la fonction $u : X \longrightarrow \mathbb{R}$ à partir de données $(y_i)_{i\in[1,n]}$ qui sont des observations de u aux points $(x_i)_{i\in[1,n]}$ de X. Pour se faire, on part du principe que u peut être "bien approximé" dans un espace V_m de fonction définies partout sur X, tel que $\dim V_m = m$. Dans $L^2(X, d\rho)$, où $d\rho$ est une mesure de probabilité sur X, le meilleur estimateur est donné par la projection orthogonale de u sur V_m :

$$P_m u = \underset{v\in V_m}{\operatorname{argmin}} \|u - v\|$$

Cependant, cette solution n'est en général pas calculable avec une observation finie d'observation $y^i = u(x^i)$.

1.2 Objectif

L'objectif est alors de trouver une méthode pour approcher u avec un nombre finie d'observation. Qui plus est, il est préférable de "bien" choisir ces observations. On introduit alors la méthode des moindres carrés pondérés qui consiste à considérer l'estimateur

$$u_W = \underset{v\in V_m}{\operatorname{argmin}} \frac{1}{n} \sum_{u=1}^{n} w^i \left|v(x^i) - y^i\right|^2 \tag{1}$$

où w^i représente des poids à ajuster. On peut reformuler ce problème dans le cas sans bruit (ie $y^i = u(x^i)$) par

$$\underset{v\in V_m}{\operatorname{argmin}} \|u - v\|_n \tag{2}$$

où

$$\|v\|_n = \left(\frac{1}{n} \sum_{i=1}^{n} w^i \left|v(x^i)\right|^2\right)^{1/2}$$

2 Théorie de l'optimal sampling

2.1 Approche des moindres carrés standard

2.1.1 Cadre d'étude

On cherche donc à savoir, pour un espace V_m et une mesure $d\rho$ donnés, comment choisir de manière optimale les échantillons y^i ainsi que les poids w^i afin que l'erreur $\|u - v\|$ soit le plus proche possible de $\|u - P_m u\|$ qui est la plus petite erreur qu'il est théoriquement possible d'atteidre.

On se place ici dans le cas où l'on tire aléatoirement (et indépendamment) les x^i selon une certaine densité de probabilité $d\mu$ définie sur X. Une approche possible, décrite dans [2], et de considérer le cas simple :

$$d\mu = d\rho \quad \text{et} \quad w^i = 1 \, \forall i \in [1, n]$$

2.1.2 Fonction de Christoffel

Nous avons introduit la quantité $P_m u$ qui est la projection orthogonale de u sur V_m. Sa forme intégrale est

$$\Big(P_m u\Big)(x) = \int_X K(x, y) u(y) d\rho(y)$$

or, pour une base orthonormée $(L_1, \dots L_m)$ de V_m, on sait que

$$\Big(P_m u\Big)(x) = \sum_{j=1}^{m} \langle u, L_j \rangle \, L_j(x) \quad \text{avec} \quad \langle u, L_j \rangle = \int_X u(y) L_j(y) d\rho(y)$$

Ce qui permet de trouver une expression du noyau intégrale de P_m :

$$K(x, y) = \sum_{j=1}^{m} L_j(x) L_j(y)$$

et en particulier, on peut alors définir la diagonale du noyau intégrale $x \mapsto k_m(x) = \sum_{i=1}^{m} |L_j(x)|^2$, connue également sous la forme de la fonction de Christoffel :

$$\frac{1}{k_m(x)} = \min_{v \in V_m, v(x)=1} \|v\|^2 \tag{3}$$

La base $(L_1, \dots L_m)$ étant orthnormée, il est immédiat que

$$\int_X k_m d\rho = m$$

On définit enfin la quantité

$$K_m = \|k_m\|_{L^\infty} = \sup_{x \in X} \sum_{i=1}^{m} |L_j(x)|^2$$

2.1.3 Un théorème intéressant

Un théorème présenté dans [2] et rappelé dans [1] est le suivant :

Théorème 1 *Pour tout $r > 0$ si m et n sont tels que*

$$K_m \leq \kappa \frac{n}{ln(n)} \quad \kappa = \frac{1 - ln(2)}{2 + 2r} \tag{4}$$

alors on a les résultats suivants :

1. *La matrice* $\mathbf{G}$ *satisfait :*

$$\mathbb{P}\Big(\|\mathbf{G} - \mathbf{I}\|^2 > \frac{1}{2}\Big) \leq 2n^{-r}$$

2. *Si* $u \in L^2(X, d\rho)$*, alors dans le cas sans bruit, l'estimateur* u_W *vérifie* $\|u - u_W\| \leq (1+\sqrt{2})e_m(u)_\infty$ *avec probabilité* $1 - 2n^{-r}$ *où* $e_m(u)_\infty = \min_{v \in V_m} \|u - v\|_{L^\infty}$

3. *si* $u \in L^2(X, d\rho)$ *est bornée en norme infinie, alors dans le cas sans bruit,*

$$\mathbb{E}\left(\|u - u_T\|^2\right) \leq (1 + \varepsilon(n))e_m(u)^2 + 8\tau^2 n^{-r}$$

où $\varepsilon(n) := \frac{4\kappa}{\ln(n)} \to 0$ *lorsque* $n \to +\infty$

La condition 4 implique qu'il faut un nombre minimum d'échantillon pour approcher u. D'après la définition de k_m, on a $K_m \geq m$, ce qui implique à minima $n \geq m\ln(m)$. Cependant, il y a de nombreux exembles où la condition est beaucoup plus restrictive. En particulier, si l'on considère $X = [-1, 1]$, $V_m = \mathbb{P}_{m-1}$ et ρ et la mesure uniforme sur $[-1, 1]$, un choix possible pour la base $(L_j)_j$ est la base formée par les polynomes de Legendre normalisés. Alors

$$K_m = \sum_{i=1}^{m} |L_j(1)|^2 = \sum_{i=1}^{m} \frac{2n+1}{2} = m^2 + \frac{m}{2}$$

Ce qui impose asymptotiquement que n doit au moins être de l'ordre de $m^2\ln(m)$. Pour $m = 10$, on passe alors d'une vingtaine d'échantillons nécessaire à plus de 200 ! Dans le cas où les domaines d'études sont non bornés (par exemple $X = \mathbb{R}$ muni de la mesure gaussienne), on peut se retrouver dans le cas $K_m = \infty$. Ce théorème présente donc en réalité de nombreux inconvénients.

2.2 Echantillonnage optimal

On retourne alors à la définition générale de l'estimateur pondéré des moindres carrés. On va maintenant utiliser une mesure $d\mu$ qui diffère de la mesure d'origine $d\rho$ sur X, et qui est telle que

$$w d\mu = d\rho$$

En supposant que la famille $(L_j)_j$ correspond toujours à une famille orthonormée de V_m, on peut redéfinir la fonction de Christoffel par

$$x \mapsto k_{m,w}(x) = \sum_{j=1}^{m} w(x) \, |L_j(x)|^2$$

et il vient alors

$$K_{m,w} = \|k_{m,w}\|_{L^\infty}$$

On a pu voir précédemment que c'est une quantité que l'on va chercher à controler. Dans cette optique, on remarque que si l'on prend

$$w = \frac{m}{k_m} = \frac{m}{\sum_{j=1}^{m} |L_j|^2}$$

Alors

$$k_{m,w} = wk_m = m, \quad K_{m,w} = m, \quad d\mu = \frac{k_m}{m} d\rho \tag{5}$$

et l'on peut énoncer le théorème suivant :

Théorème 2 *Pour tout $r > 0$, si m et n vérifient*

$$K_{m,w} = m \leq \kappa \frac{n}{\ln n}, \text{ avec } \kappa := \frac{1 - \ln 2}{2 + 2r}$$

alors on a les résultats suivants :

1. *La matrice* $\mathbf{G}$ *satisfait*

$$\mathbb{P}\Big(\|\mathbf{G} - \mathbf{I}\|_2 > \frac{1}{2}\Big) \leq 2n^{-r}$$

2. *Si $u \in L^2(X, d\rho)$ est bornée en norme infinie, alors dans le cas sans bruit,*

$$\mathbb{E}\left(\|u - u_T\|^2\right) \leq (1 + \epsilon(n))e_m(u)^2 + 8r^2 n^{-r}$$

   où $\epsilon(n) = \frac{4\kappa}{\ln(n)} \to 0$ lorsque $n \to +\infty$.

3. *Si $x \in L^\infty(X, d\rho)$, alors dans le cas sans bruit,*

$$\|u - u_W\| \leq (1 + \sqrt{2})e_m(u)_\infty$$

   avec probabilité $1 - 2n^{-r}$.

4. *Si $u \in L^2(X, d\rho)$, alors dans le cas sans bruit,*

$$\mathbb{E}\left(\|u - u_C\|^2\right) \leq (1 + \epsilon(n))e_m(u)^2 + 2\|u\|^2 n^{-r}$$

   où $\epsilon(n) = \frac{4\kappa}{\ln(n)} \to 0$ lorsque $n \to +\infty$.

Par construction, on remarque que la mesure d'échantillonage (ainsi que les poids) dépend de V_m. On notera donc à présent $d\mu = d\mu_m$. Le document [1] propose de se concentrer dans le cas où $d\rho$ est un produit tensoriel et V_m est généré par un produit tensoriel de bases. Nous allons expliciter les hypothèses et mettre au point un algorithme afin générer n mesures indépendantes identiquement distribuées selon $d\mu_m$.

2.3 Elaboration de l'algorithme

2.3.1 Hypothèses

Nous avons abordé précédemment la nécessité d'avoir une mesure d'échantillonage optimal $d\mu_m$ pour générer notre échantillon $x^1, \dots x^n$ avec la méthode pondérée des moindres carrés.

On suppose ici que X est le produit cartésien d'espaces univariés X_i de sorte que $X = \times_{i=1}^{d} X_i$. On rappelle également que le produit tensoriel de deux mesures μ_1 et μ_2 définies sur deux espaces mesurés $(\Omega_1, \mathcal{F}_1, \mu_1)$ et $(\Omega_1, \mathcal{F}_1, \mu_1)$ n'est autre que la mesure $\mu_1 \otimes \mu_2$ définie sur l'espace produit $\Omega_1 \times \Omega_2$ muni de la tribu produit $\mathcal{F}_1 \otimes \mathcal{F}_2$ telle que

$$(\mu_1 \otimes \mu_2)(A \times B) = \mu_1(A)\mu_2(B), \quad \forall A \in \mathcal{F}_1, B \in \mathcal{F}_2$$

Ainsi sur X on peut considérer la mesure

$$d\rho = \bigotimes_{i=1,\dots,d} d\rho_i$$

où chaque $d\rho_i(t) = \rho_i(t)dt$ est la mesure définie sur X_i. Ceci implique $\forall x = (x_1, \dots, x_d) \in X$, $\rho(x) = \prod_{i=1}^{d} \rho_i(x_i)$

Enfin, on considère la notation suivante : pour une base orthonormale $(\phi_j^i)_{j\geq 0}$ dans $L^2(X_i, d\rho_i)$, on peut alors définir la base tensorielle

$$L_\nu(x) = \prod_{i=1}^{d} \phi_{\nu_i}^i(x_i), \quad \nu \in \mathbb{N}_0^d \tag{6}$$

Remarque : $\nu \in \mathbb{N}_0^d$ est un vecteur de dimension d donc chaque composante appartient à $\mathbb{N}_0 = \{0, 1, 2, \dots\}$ l'ensemble des entiers naturels incluant zéro. On désigne alors la collection multi-index $\Lambda \subset \mathbb{N}_0^d : \#(\lambda) = m$ par une famille de m éléments ν, où chaque ν est un vecteur de taille d dans $\mathbb{N}$, *ie* $\nu \in \mathbb{N}_0^d$, $\forall \nu \in \Lambda$.

On peut alors considérer l'espace

$$V_m = \text{Vect}\Big\{L_\nu \ : \ \nu \in \Lambda\Big\}$$

Remarque : les valeurs des vecteurs $\nu \in \mathbb{N}_0^d$ ne servent qu'à indicer les différentes bases. Ainsi, on considère que l'on réorganise les indices (par exemple dans l'ordre lexicographique) de sorte que $(L_\nu)_{\nu\in\Lambda}$ devient $(L_j)_{j\in[1,m]}$

2.3.2 Echantillonnage selon μ_m

On rappelle au vu de 5 que

$$\mu_m(x) = \frac{k_m}{m}\rho(x) = \frac{1}{m}\sum_{i=1}^{m} |L_i(x)|^2 \rho(x)$$

Contrairement à ρ, μ_m ne peut s'exprimer sous forme de produit. On va alors utiliser les densités marginales pour pouvoir générer des échantillons selon μ_m. On rappelle que pour $Z = (X, Y)$, où X et Y sont de densité marginale respective f_X et f_Y et Z de loi conjointe f_Z, alors on a

$$\begin{cases} f_X(x) = \int_\Omega f_Z(x, y)dy \\ f_Y(y) = \int_\Omega f_Z(x, y)dx \end{cases}$$

On introduit de nouveau quelques notations. Pour $x = (x_1, \dots, x_d)$ et pour $A \subseteq \{1, \dots, d\}$, on pose $x_A = (x_i)_{i\in A}$ et $\bar{A} = \{A, \dots, d\}\backslash A$, de sorte que $A^q = \{1, \dots, q\} \Rightarrow \bar{A^q} = \{q + 1, \dots, d\}$. On pose également :

$$dx_A = \bigotimes_{i\in A} dx_i, \quad d\rho_A = \bigotimes_{i\in A} d\rho_i, \quad \rho_A(x_A) = \prod_{i\in A} \rho_i(x_i), \quad X_A = \underset{i\in A}{\times} X_i$$

Avec ces notations, on peut considérer la densité marginale de μ_m relative aux q premières variables :

$$\psi_q(x_{A^q}) = \int_{X_{A^q}} \mu_m(x_1, \dots, x_d)dx_{A^q} \tag{7}$$

En reprenant l'expression de μ_m et en considérant l'othonormalité de la base, on peut aisément calculer les densités marginales :

$$\begin{aligned}
\psi_q(x_{A^q}) &= \int_{X_{\bar{A}^q}} \mu_m(x_1, \dots, x_d)dx_{\bar{A}^q} \\
&= \int_{X_{\bar{A}^q}} \frac{1}{\#(\Lambda)} \sum_{\nu\in\Lambda} \left(\prod_{i=1}^{q} \left|\phi^i_{\nu_i}(x_i)\right|^2 \prod_{i=q+1}^{d} \left|\phi^i_{\nu_i}(x_i)\right|^2 \right) \rho(x_1, \dots, x_d)dx_{\bar{A}^q} \\
&= \frac{1}{\#(\Lambda)} \rho_{A^q}(x_{A^q}) \sum_{\nu\in\Lambda} \prod_{i=1}^{q} \left|\phi^i_{\nu_i}(x_i)\right|^2 \left[\int_{X_{\bar{A}^q}} \left(\prod_{i=q+1}^{d} \left|\phi^i_{\nu_i}(x_i)\right|^2 \right) \rho_{\bar{A}^q}(x_{\bar{A}^q})dx_{\bar{A}^q} \right] \\
&= \frac{1}{\#(\Lambda)} \rho_{A^q}(x_{A^q}) \sum_{\nu\in\Lambda} \prod_{i=1}^{q} \left|\phi^i_{\nu_i}(x_i)\right|^2 \left[\prod_{i=q+1}^{d} \int_{X_i} \left|\phi^i_{\nu_i}(x_i)\right|^2 \rho_i(x_i)dx_i \right] \\
&= \frac{1}{\#(\Lambda)} \rho_{A^q}(x_{A^q}) \sum_{\nu\in\Lambda} \prod_{i=1}^{q} \left|\phi^i_{\nu_i}(x_i)\right|^2
\end{aligned}$$

ce qui donne une expression simple des densités marginales :

$$\psi_q(x_{A^q}) = \frac{1}{\#(\Lambda)} \rho_{A^q}(x_{A^q}) \sum_{\nu\in\Lambda} \prod_{i=1}^{q} \left|\phi^i_{\nu_i}(x_i)\right|^2 \tag{8}$$

A partir de cela, on peut en déduire une méthode nous permettant de générer n échantillons $x^k = (x^k_1, \dots, x^k_d) \in X$ à partir d'une densité μ_m. Cet algorithme procède

de manière séquentielle en échantillonnant d'abord la première coordonnée x_1^k à partir de la densité univariée suivante :

$$\varphi_1(t) = \psi_1(t) = \frac{\rho_1(t)}{\#(\Lambda)} \sum_{\nu \in \Lambda} |\phi_{\nu_1}^1(t)|^2$$

Ensuite, pour chaque coordonnée x_q^k avec $q \geq 2$, on échantillonne à partir de la densité conditionnelle suivante :

$$\varphi_q\left(t | x_{A_{q-1}}^k\right) = \frac{\rho_q(t) \prod_{j=1}^{q-1} |\phi_{\nu_j}^j(x_j^k)|^2}{\sum_{\nu \in \Lambda} \prod_{j=1}^{q-1} |\phi_{\nu_j}^j(x_j^k)|^2}$$

Cela signifie que l'on conditionne chaque nouvelle coordonnée sur les $q-1$ précédentes. Ce processus séquentiel permet de capturer les dépendances entre les dimensions tout en générant des réalisations indépendantes et identiquement distribuées selon la densité cible μ_m.

3 Expérimentation de l'algorithme

Dans cette partie, nous allons essayer d'implémenter l'algorithme 1 proposé dans [1] Pour tester sa cohérence et son efficacité, nous essaierons de reproduire la figure

Algorithm 1 Echantillonnage pour μ_m

INPUT : $n, d, \Lambda, \rho_i, (\phi_j^i)_{j \geq 0}$ for $i = 1, \ldots, d$.
OUTPUT : $x^1, \ldots, x^n \overset{\text{i.i.d.}}{\sim} \mu_m$.
for $k = 1$ à n **do**
  $\alpha_\nu \leftarrow (\#(\Lambda))^{-1}$, for any $\nu \in \Lambda$.
  Tirer x_1^k selon $t \mapsto \varphi_1(t) = \rho_1(t) \sum_{\nu \in \Lambda} \alpha_\nu |\phi_{\nu_1}^1(t)|^2$.
  for $q = 2$ à d **do**
    $\alpha_\nu \leftarrow \frac{\prod_{j=1}^{q-1} |\phi_{\nu_j}^j(x_j^k)|^2}{\sum_{\nu \in \Lambda} \prod_{j=1}^{q-1} |\phi_{\nu_j}^j(x_j^k)|^2}$, for any $\nu \in \Lambda$.
    Tirer x_q^k selon $t \mapsto \varphi_q(t) = \rho_q(t) \sum_{\nu \in \Lambda} \alpha_\nu |\phi_{\nu_q}^q(t)|^2$.
  end for
  $x^k \leftarrow (x_1^k, \ldots, x_d^k)$.
end for

1 de [1]. On se placera dans $L^2([-1,1], d\rho)$ dans cette partie.

3.1 Implémentation de l'algorithme

On commence par implémenter l'algorithme dans le cas unidimensionnel (ie $d = 1$). On se rend compte dans [1] qu'il est impératif de se munir d'une base

orthonormale $L_1, \dots L_m$ dans $L^2([-1,1], d\rho)$ de l'espace V_m. Il est assez naturel de prendre $V_m = \mathbb{P}_{m-1}$, et il est alors possible de produire une base convenable avec les polynômes de Legendre.
Pour $n \in \mathbb{N}$ et $x \in [-1,1]$, on pose

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n}((x^2-1)^n), \quad e_n = \sqrt{\frac{2n+1}{2}} P_n \tag{9}$$

Alors $(e_n)_{n\in\mathbb{N}}$ est une base hilbertienne de $L^2([-1,1], d\rho)$ et convient pour notre problème. Sachant cela, il est ensuite possible grâce aux objets `Polynomial` de Python de générer des polynomes et de les évaluer en certains points x. (x pouvant être un vecteur).
Ensuite, on définit la densité selon laquelle on veut générer nos $x_1, \dots x_n$. Dans le cadre d'un problème unidimentionnel ($d = 1$), il suffit de simuler la réalisation d'une variable aléatoire de densité

$$\varphi_1(t) = \rho_1(t) \sum_{\nu \in \Lambda} \alpha_\nu \left| \phi^1_{\nu_1}(t) \right|^2 \tag{10}$$

où ρ_1 est une densité au choix. Nous prendrons par exemple la densité de la loi uniforme. Ainsi, il faut un moyen pour simuler une variable aléatoire qui suivrait cette loi de densité φ_1. La méthode de la transformée inverse semble inaccessible, et on se tourne donc vers la méthode de rejet, que l'on rappelle ci-après :

Méthode de rejet

Soit X une variable aléatoire de densité de probabilité f par rapport à la mesure de Lebesgue. La première composante du vecteur aléatoire de loi uniforme sur le domaine $B = (x,z) : 0 \leq z \leq f(x)$ situé sous le graphe de la fonction f suit la loi de X. On suppose que l'on sache simuler des échantillons de Y de densité g, et qu'il existe $c > 0$ tel que $f(x) < cg(x)$. Alors la méthode du rejet consiste à :

- Générer une réalisation x de la variable Y de densité g
- Générer une réalisation $z = cg(x)u$ avec u tiré selon la loi uniforme $\mathcal{U}(0,1)$
- Si $z < f(x)$, accepter x comme réalisation de X

Il faut donc trouver une enveloppe à notre fonction φ_1. Une méthode brutale (et nous allons le voir assez peu optimal), et de prendre g la densité de $\mathcal{U}(-1,1)$ et $c = 2 \max_{x\in[-1,1]} \varphi_1(x)$. C'est cependant une première approche qui permet de faire rapidement des tests. On exécute la fonction `sequential_conditional_sampling` avec ρ_1 la densité de la loi uniforme sur $[0,1]$ et on trace les points obtenus, ainsi que la courbe sur $[-1,1]$ de φ_1 et cg. On obtient la Figure 2 :
Les résultats semblent satisfaisant (les points sont tous en dessous de la courbe de

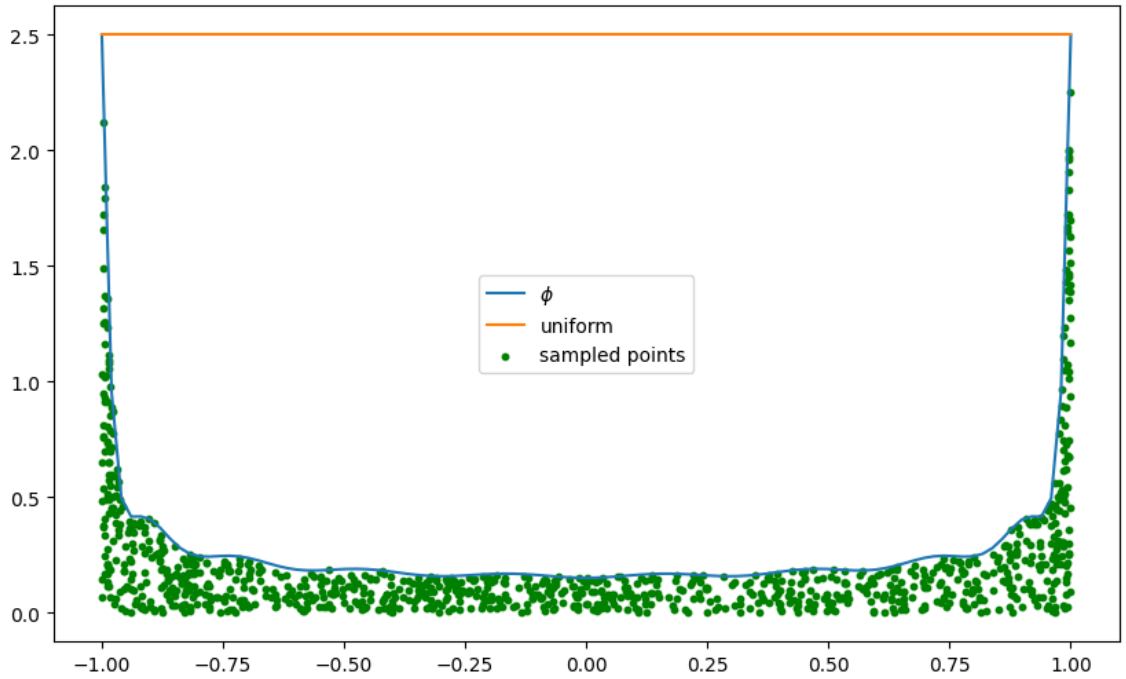


FIGURE 2 – Visualisation de la méthode du rejet

φ_1), mais on remarque que la densité de la loi uniforme n'est clairement pas optimale pour faire la méthode du rejet, car l'air entre cette densité et φ_1 est très importante. Ainsi, de nombreuses itérations sont "inutiles" (ie $cg(x) > f(x)$) et sont donc rejetées. Une piste d'amélioration de l'algorithme est donc de trouver une autre enveloppe de φ_1. Dans un premier temps, nous nous contenterons de cela pour faire un test sur la matrice de Gram.

3.2 Test sur la matrice de Gram

On souhaite utiliser l'algorithme mis en place précédemment pour mettre en lumière les relations entre la matrice de Gram, le nombre d'échantillons et la dimension de V_m (ie la taille de la base). Il est expliqué dans [1] que l'on peut étendre le problème unidimensionnel

$$\underset{v \in V_m}{\operatorname{argmin}} \|u - v\|_n$$

au problème

$$\mathbf{G}v = d$$

où $\mathbf{G}$ est la matrice de Gram telle que $\mathbf{G}_{i,j} = \langle L_i, L_j \rangle_n$ et d est le vecteur "data" tel que $d_j = \frac{1}{n} \sum_{i=1}^{n} w^i u(x^i) L_j(x^i)$. On rappelle que

$$\langle u, v \rangle_n = \frac{1}{n} \sum_{i=1}^{n} w^i u(x^i) v(x^i) \tag{11}$$

Le document [1] illustre le fait que pour n et m satisfaisant le Corollaire 2.2 (de [1]), la probabilité $\mathbf{P}(\text{cond}(\mathbf{G}) \leq 3)$ est supérieure à $\mathbf{P}(\|\mathbf{G} - I_m\| \leq \frac{1}{2})$ et ainsi dans le cadre de ce même corollaire, c'est à dire pour n et m vérifiant

$$m \leq \kappa \frac{n}{ln(n)}, \text{ avec } \kappa = \frac{1 - ln(2)}{2 + 2r} \text{ et } r > 0$$

on peut estimer la probabilité

$$\mathbf{P}(\text{cond}(\mathbf{G}) \leq 3) \geq 1 - 2n^{-r} \tag{12}$$

L'idée est de faire varier n et m sur des plages assez grandes et d'estimer la probabilité $\text{cond}(\mathbf{G}) \leq 3$. Pour calculer cette probabilité, on procède par méthode itérative : on calcule par exemple 100 fois $\text{cond}(\mathbf{G})$ et on fait la moyenne.
On obtient le résultat présent sur la Figure 3

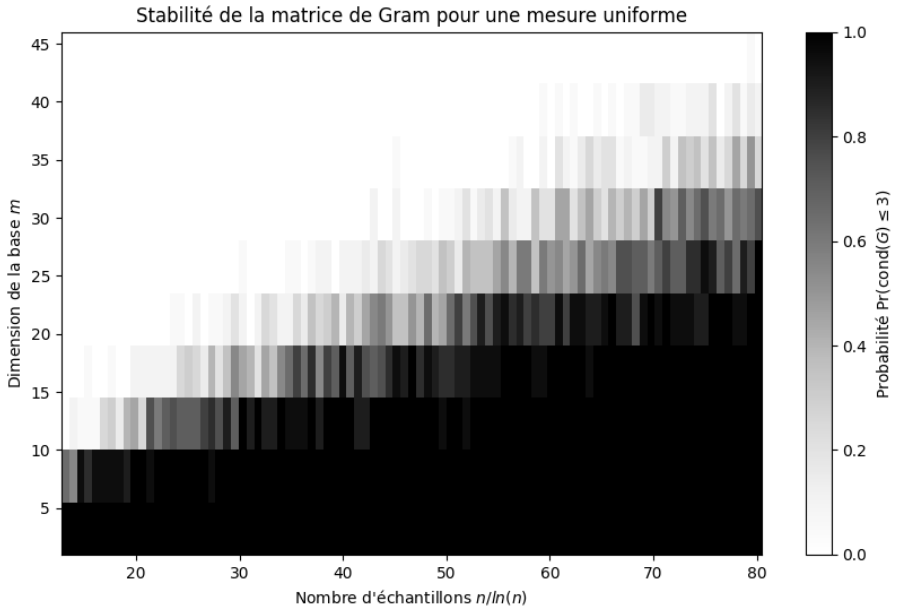


FIGURE 3 – Calcul de la probabilité 12 pour différentes valeurs de n et m

La Figure 3 permet de mettre en évidence la relation nécessaire entre n et m pour que le conditionnement de la matrice de Gram soit inférieur à 3. Par exemple pour $n/\ln(n) = 70$, $r = 1$ et $m = 10$, alors on a clairement $m < \kappa\frac{n}{ln(n)}$ et donc la probabilité est 1. Au contraire lorsque $n/\ln(n) = 30$, $r = 1$ et $m = 45$, cette condition n'est plus respecté et la probabilité est 0. On peut également constater des probabilités intermédiaires pour des valeurs de m et $n/\ln(n)$ proches.

Remarque : Comprenons d'où vient l'équation 12

$$\mathbf{P}(\text{cond}(\mathbf{G}) \leq 3) \geq 1 - 2n^{-r}$$

Le nombre de conditionnement d'une matrice symétrique définie positive $\mathbf{G}$ est donné par :

$$\text{cond}(\mathbf{G}) = \frac{\lambda_{\max}(\mathbf{G})}{\lambda_{\min}(\mathbf{G})}, \tag{13}$$

où $\lambda_{\max}$ et $\lambda_{\min}$ sont respectivement la plus grande et la plus petite valeur propre de $\mathbf{G}$.
On sait que la norme spectrale vérifie $\sqrt{\rho(A^*A)} = \|A\|_2$ donc lorsque $A \in \mathcal{S}(\mathbb{R})$, $\rho(A) = \|A\|_2$. Comme par définition la matrice de Gram est symétrie, $\mathbf{G} - \mathbf{I} \in \mathcal{S}\mathbb{R}$. Ainsi

$$\|\mathbf{G} - \mathbf{I}\|_2 = \max_i |\lambda_i(\mathbf{G}) - 1|.$$

On suppose que :

$$\|\mathbf{G} - \mathbf{I}\|_2 \leq \frac{1}{2},$$

ce qui implique que toutes les valeurs propres de $\mathbf{G}$ sont dans l'intervalle :

$$\lambda_i(\mathbf{G}) \in \left[1 - \frac{1}{2}, 1 + \frac{1}{2}\right] = \left[\frac{1}{2}, \frac{3}{2}\right].$$

On a :

$$\text{cond}(\mathbf{G}) = \frac{\lambda_{\max}(\mathbf{G})}{\lambda_{\min}(\mathbf{G})} \leq \frac{\frac{3}{2}}{\frac{1}{2}} = 3.$$

Nous avons ainsi montré que :

$$\|\mathbf{G} - \mathbf{I}\|_2 \leq \frac{1}{2} \implies \text{cond}(\mathbf{G}) \leq 3. \tag{14}$$

Enfin, on tire du théorème 4 que la matrice $\mathbf{G}$ satisfait

$$\mathbb{P}\Big(\|\mathbf{G} - \mathbf{I}\|_2 > \frac{1}{2}\Big) \leq 2n^{-r}$$

et

$$\left(\|\mathbf{G} - \mathbf{I}\|_2 \leq \frac{1}{2} \implies \text{cond}(\mathbf{G}) \leq 3\right) \implies \left(\{\text{cond}(\mathbf{G}) \leq 3\} \subset \{\|\mathbf{G} - \mathbf{I}\|_2 \leq \frac{1}{2}\}\right)$$

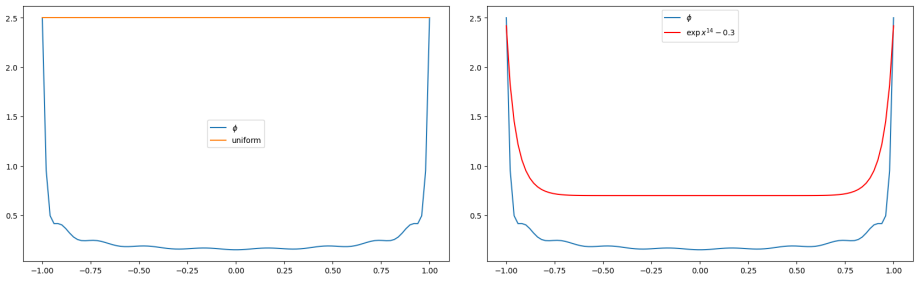
d'où l'intéret pour l'équation

$$\mathbf{P}(\text{cond}(\mathbf{G}) \leq 3) \geq 1 - 2n^{-r}$$

3.3 Amélioration de la compléxité temporelle

3.3.1 Méthode de rejet : Changement de l'enveloppe

Comme évoqué précédemment, le choix de la fonction uniforme comme enveloppe de la densité pour la méthode du rejet n'est clairement pas optimale (Cf Figure 4a)



(a) Enveloppe uniforme (b) Enveloppe $\exp x^{14} - 0.3$

FIGURE 4 – Optimisation de l'enveloppe de φ_1

Par exemple, la fonction utilisée sur la Figure 4b "épouse" mieux la densité φ_1, et semple donc plus adaptée. A titre de comparaison, le temps d'exécution du programme `sequential_conditional_sampling` est de 1.85 seconde pour $\exp x^{14} - 0.3$ et 6.59 secondes pour la densité uniforme. (avec les paramètres $n = 10000$, $m = 9$). Cette différence de calcul est non négligeable.

Cependant, si l'on change la base, cette fonction ne convient plus (on peut s'en persuader en augmentant fortement la taille de la base des polynomes de Legendre). Il semble donc difficile de trouver une enveloppe qui convienne à toute les bases, alors que $c \times \mathcal{U}$ convient toujours car cela revient à tracer un rectangle autour de φ_1 de hauteur égale au maximum de cette fonction.

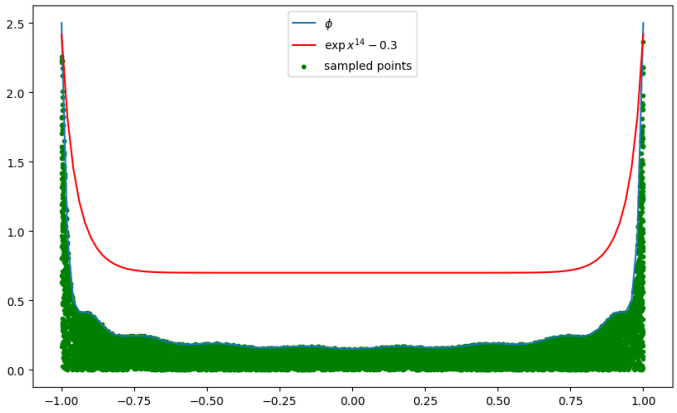


FIGURE 5 – Visualisation de la méthode du rejet pour $\exp x^{14} - 0.3$

3.3.2 Méthode d'inversion et interpolation

Une autre technique permettant de simuler des variables aléatoires de loi donnée est d'utiliser leur fonction de répartition. On considère la méthode d'inversion suivante :

Méthode de la transformée inverse

Soit X une variable aléatoire à valeur dans $\mathbb{R}$ de fonction de répartition F. On définit l'inverse généralisé de F comme

$$F^-(u) = \inf\{x : F(x) \geq u\}$$

Alors si U suit la loi uniforme sur $[0, 1]$, on a

$$F^-(U) = X$$

A priori, on n'a pas accès à la fonction de répartition de la loi de densité φ_1, mais on sait que $F(t) = \mathbf{P}(X \leq t) = \int_{-\infty}^{t} \varphi_1(x)\,dx$. On peut calculer cette intégrale grâce à des bibliothèques Python comme `cumulative_trapezoid`. On estime ensuite F^- avec la bibliothèque `interp1d`, qui permet de faire le calcul de F^- avec une technique d'interpolation. Cette méthode est beaucoup plus rapide que les 2 autres : le temps d'éxécution est de 0.002 seconde pour les mêmes choix de paramètres.

3.3.3 Méthode de discrétisation

C'est la méthode que nous utiliserons par la suite pour faire les tests. On discrétise l'intervalle en n intervalles plus petits. Pour 1 point de chacun de ces intervalles, on calcule la densité selon laquelle on veut échantilloner et on la multiplie par dt qui est la taille de nos petits intervalles. On renormalise pour avoir une probabilité. Il suffit alors de tirer aléatoirement un x dans l'intervalle global avec la probabilité calculée précédemment.

3.4 Approche de fonctions

On peut estimer des fonctions où l'on sait que la solution approchée doit être exacte, comme des polynômes par exemple. Prenons $u(x) = x^2$. La figure 6 donne le résultat de l'algorithme, et l'on peut constater que l'approximation est exacte. L'erreur pour l'optimal sampling est calculé avec la norme $(u, v)_n$ introduite précédemment, et l'erreur pour l'approximation des moindres carrrés classique est calculé en norme 2. Dans les 2 cas, l'erreur est de 10^{-16}.

Des estimations d'autres polynômes donnent également des solutions exactes. On peut alors regarder ce qu'il se passe avec des fonctions plus exotiques :

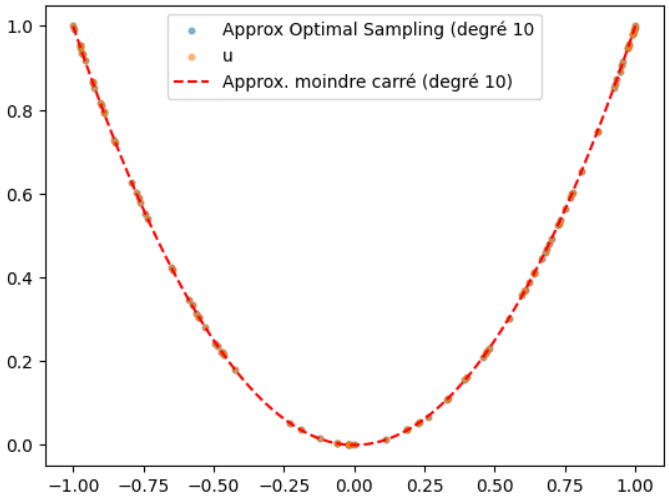


FIGURE 6 – Estimation de x^2

$$u(x) = \sin(x^2) + \frac{\log(|x|+1)}{(x^2+1)} - \exp(-\cos(x))$$

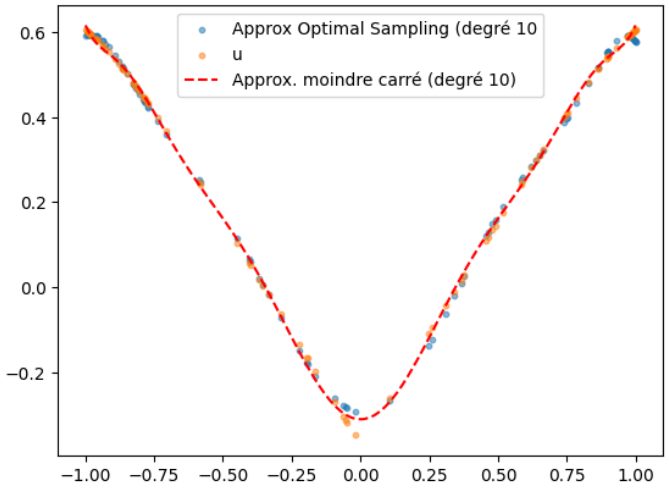


FIGURE 7 – Estimation de $\sin(x^2) + \frac{\log(|x|+1)}{(x^2+1)} - \exp(-\cos(x))$

L'erreur pour l'optimal sampling est $e = 0.015$, et celle pour les moindres carrés classique est $e = 0.017$. L'optimal sampling produit donc une meilleur approximation que les moindres carrés.

4 Extension de $L^2([-1,1],d\rho)$ à $L^2([a,b],d\rho)$

Une famille classique est constituée des polynômes de Legendre, définis sur $[-1,1]$, mais ils peuvent être adaptés à $[a,b]$. Si $P_n(x)$ sont les polynômes de Le-

	$[-5,20]$	$[-5,30]$	$[-5,40]$
Erreur Optimal Sampling	2.14×10^{-13}	2.83×10^{-13}	2.01×10^{-12}
Erreur Moindres Carrés Classique	1.63×10^{-10}	1.09	52.8

TABLE 1 – Erreur d'approximation en fonction de l'intervalle

gendre sur $[-1,1]$, une base orthonormée pour $L^2([a,b],d\rho)$ peut être obtenue par un changement de variable $x \to \tilde{x} = \frac{2x-(b+a)}{b-a}$, qui ramène $[a,b]$ à $[-1,1]$.
Les fonctions orthonormées deviennent :

$$\phi_n(x) = \sqrt{\frac{2n+1}{b-a}} P_n\left(\frac{2x-(b+a)}{b-a}\right) \tag{15}$$

Ensuite, on peut ramener de nouveau l'intervalle [-1, 1] à l'intervalle [a, b] avec l'application

$$\tilde{x} \to x = \frac{(\tilde{x}+1)(b-a)}{2} + a$$

On peut par exemple vérifier que la fonction polynomiale $P(x) = x^2 - \frac{1}{10}x^3 + 3x$ est exacte sur un intervalle donné ([-3, 6] sur la figure 8)

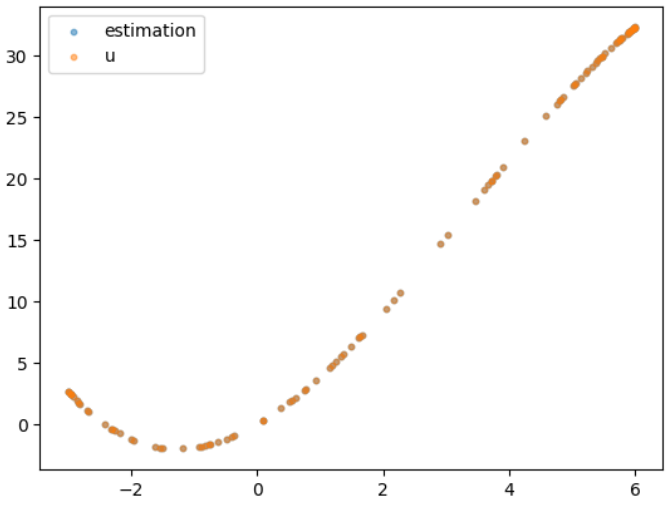
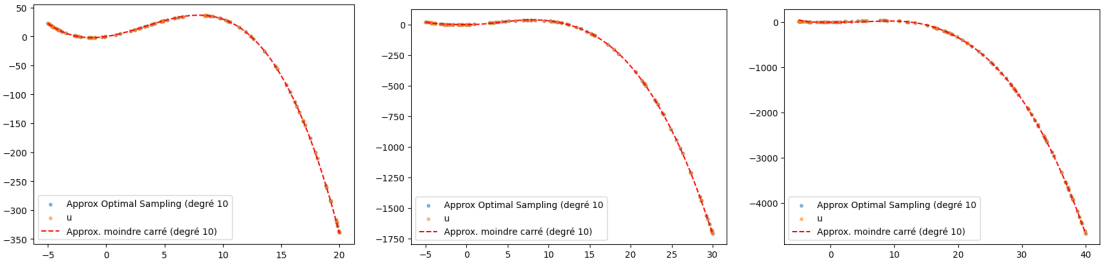


FIGURE 8 – Estimation de $x^2 - \frac{1}{10}x^3 + 3x$

Remarque : Il semble que l'optimal sampling donne de meilleurs approximations pour les grands intervalles : on a répertorié les erreurs dans le Tableau 1, et les différents graphes sont représenté Figure 9

Remarque : Si l'on diminue m, on peut également diminuer n tant que l'on vérifie



(a) Intervalle [-5, 20] (b) Intervalle [-5, 30] (c) Intervalle [-5, 40]

FIGURE 9 – Erreur d'approximation de $x^2 - \frac{1}{10}x^3 + 3x$

$m \leq \kappa \frac{n}{ln(n)}$. Il semble donc y avoir un travail de réflexion en amont de l'approximation pour choisir convenablement m et n.

5 Application aux réseaux de neurones

5.1 Motivation

5.1.1 Contexte général

On se donne un ensemble X et ρ une mesure de probabilité. On définit alors $\mathcal{H} = \mathcal{H}(X, \rho)$ un espace de Hilbert de fonctions définies sur X, doté de la norme $\|.\|$. On définie la perte attendue par

$$\mathcal{L}(v) = \int l(v, x) \mathrm{d}\rho(x) \tag{16}$$

où $l : \mathcal{H} \times X \longrightarrow \mathbb{R}$ est une fonction de perte. Il est classique de s'intéreser au problème d'optimisation

$$\min_{v \in \mathcal{M}} \mathcal{L}(v)$$

avec $\mathcal{M} \subseteq \mathcal{H}$ est un espace de classe de modèles, c'est à dire l'ensemble des fonctions candidates parmi lesquels on cherche la meilleure solution. Pour résoudre ce problème, le document [4] propose une méthode itérative en linéarisant localement $\mathcal{M}$.
On se donne un espace de probabilité $(\Omega, \Sigma, \mathbb{P})$ muni de la filtration $(\mathcal{F}_t)_{t \geq 0}$. On remarquera que cette filtration dépend des itérés jusqu'à l'étape t. On suppose alors que pour chaque étape $t \in \mathbb{N}$, il existe un espace $\mathcal{F}_t$-mesurable de dimention d_t (la dimension dépend donc de l'étape t) qui permet d'approximer $\mathcal{M}$ locallement autour de l'irété u_t. On notera cette espace $\mathcal{T}_t$. Par exemple, dans le cas où l'espace des classes de modèles $\mathcal{M}$ est de la forme $\mathcal{M} = F(\theta) : \theta \in \mathbb{R}^d$ avec $F : \mathbb{R}^d \longrightarrow \mathcal{H}$ est une fonction différentiable, l'espace $\mathcal{T}_t$ autour de $u_t = F(\theta_t)$ est la combinaison linéaire des fonctions $\partial_k F(\theta_t), k \in [1, d]$.
On rappelle que une fonction f est Fréchet différentiable en x si il existe un opérateur linéaire borné A tel que

$$\lim_{\|h\| \to 0} \frac{\|f(x+h) - f(x) - Ah\|}{\|h\|} = 0$$

Si un tel opérateur existe, il est unique et on le note $Df(x)$ $(= A)$ et est appelé la dérivé de Fréchet. Ainsi, si $\mathcal{L}$ admet une dérivé de Fréchet en u_t, on peut définir le gradient $g_t = \nabla \mathcal{L}(u_t) \in \mathcal{H}$ grâce au théorème de représentation. L'idée est alors la suivante : on introduit l'estimateur P_t^n du projecteur orthogonal $P_t : \mathcal{H} \to \mathcal{T}_t$. On applique alors un step size s_t et l'on obtient $\bar{u}_{t+1} = u_t - s_t P_t^n g_t \in \mathcal{T}_t$. Comme u_t n'est pas nécessairement dans $\mathcal{M}$, il faut appliquer une "rétractation" R de sorte que $u_{t+1} = R(\bar{u}_{t+1})$ soit dans $\mathcal{M}$. Une illustration de cet algorithme est représenté Figure 10

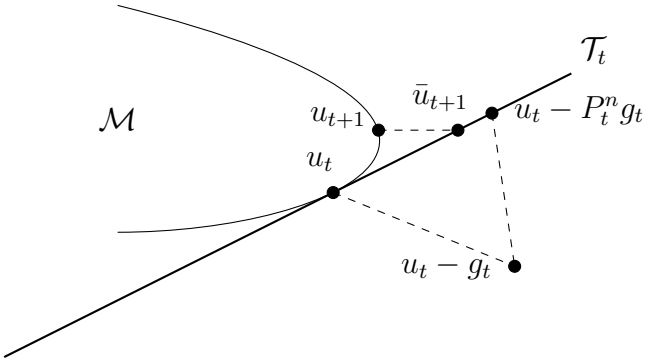


FIGURE 10 – Illustration de l'algorithme

5.1.2 Espace linéaire pour les réseaux de neuronnes

Nous allons étudier cette méthode sur des réseaux de neurones, et plus particulièrement les réseaux peu profond. On se place dans l'espace de Hilbert $\mathcal{H} = \mathcal{L}^2(\rho)$ ($\rho \rightsquigarrow \mathcal{U}(X)$ si $X = [-1, 1]$ ou $\rho \rightsquigarrow \mathcal{N}(0, 1)$ si $X = \mathbb{R}$ par exemple). On considère une fonction d'activation différentiable (la fonction sigmoïde par exemple). On définit le réseau de neuronne peu profond $\Phi_\theta : X = \mathbb{R}^n \to \mathbb{R}$ de taille $m \in \mathbb{N}$ et de paramètre $\theta = (A_1, b_1, A_0, b_0) \in \mathbb{R}^{1\times m} \times \mathbb{R}^{m\times n} \times \mathbb{R}^m$ par :

$$\Phi_{(A_1,b_1,A_0,b_0)}(x) = A_1\sigma(A_0x + b_0) + b_1 \tag{17}$$

Il vient alors que l'espace de classe de modèles peut être défini par

$$\mathcal{M} = \{\Phi_\theta : \theta \in \mathbb{R}^{1\times m} \times \mathbb{R}^{m\times n} \times \mathbb{R}^m\}$$

Les applications numériques seront effectués pour $n = 1$, mais les résultats restent valables $\forall n \in \mathbb{N}$. Pour la linéarisation, on prend $\mathcal{T}_t = \mathcal{T}_{\Phi_{\theta_t}} = \text{vect}\{\partial_{\theta_j}\Phi_\theta : j \in [1, d]\}$. Il faut donc estimer les quantités

$$\{\partial_{A_{1,1j}}\Phi_{(A_1,b_1,A_0,b_0)},\ \partial_{b_1}\Phi_{(A_1,b_1,A_0,b_0)},\ \partial_{A_{0,ij}}\Phi_{(A_1,b_1,A_0,b_0)},\ \partial_{b_{0,i}}\Phi_{(A_1,b_1,A_0,b_0)}\}.$$

Pour estimer $\partial_{A_{0,ij}}\Phi_{(A_1,b_1,A_0,b_0)}$ on peut par exemple remplacer les différentielles par une approximation par différente finie de pas $h > 0$
On trouve alors

$$\begin{aligned}\mathcal{T}^h_{\Phi_\theta} &= \text{vect}\Big(\sigma(A_0x + b_0)_j : j \in [1, m]\Big)\\ &+ \text{vect}(1)\\ &+ \text{vect}\Big(\sigma((A_0 + E_{ij}h)x + b_0) : i \in [1, n], j \in [1, m]\Big)_i\\ &+ \text{vect}\Big(\sigma(A_0x + (b_0 + e_jh))_i : j \in [1, m]\Big)\end{aligned}$$

Dans la suite, nous allons avoir besoin d'une base orthonormée de $\mathcal{T}_t$. Pour se faire, on considère une famille génératrice $(\phi_j)_{j\in[1,d]}$ de $\mathcal{T}_t$. On note G^+ la pseudo inverse au sens de Moore-Penrose de la matrice de Gram G, et c_{kj} les coefficients de la matrice $(G^+)^{\frac{1}{2}}$. On obtient alors une base orthonormée de $\mathcal{T}_t$ noté $(\psi_k)_{k\in[1,r]}$ (avec $r < d$) définis par

$$\psi_k = \sum_{j=1}^{d} c_{kj}\phi_j \tag{18}$$

En effet en considérant le produit scalaire sur $\mathcal{L}^2(\rho)$, on a

$$\begin{aligned}\langle \psi_k, \psi_k \rangle &= \int \psi_k \psi_k \mathrm{d}\rho \\ &= G^{+1/2} \int \phi_k \phi_k \mathrm{d}\rho G^{+1/2} \\ &= G^{+1/2} G G^{+1/2} = I\end{aligned}$$

5.2 Estimateurs de la projection orthogonale

Dans la majorité des cas, nous ne pourrons pas calculer directement la projection P_t sur $\mathcal{T}_t$ et il faut donc trouver des estimateurs de ce dernier. Il faudrait également trouver un "bon" estimateur, c'est à dire un estimateur pour lequel nous sommes capable de controler le biais et la variance. On sait que pour une base $(\psi_k)_{k\in[1,d_t]}$ de $\mathcal{T}_t$ de dimension d_t (on rappelle que la dimension de $\mathcal{T}_t$ dépend à priori de l'itéré u_t), alors la projection P_t du gradient g_t sur $\mathcal{T}_t$ peut être défini par

$$P_t g_t = \sum_{k=1}^{d_t} \langle g_t, \psi_k \rangle \psi_k \tag{19}$$

Pour les différents estimateurs de P_t, on va pouvoir controler le biais et la variance de sorte que

$$\begin{aligned}\mathbb{E}\left[(g_t, P_t^n g_t) \mid \mathcal{F}_t\right] &\geq c_{\text{bias },1} \left\|P_t g_t\right\|^2 - c_{\text{bias },2} \left\|P_t g_t\right\| \left\|(I - P_t)\, g_t\right\| \\ \mathbb{E}\left[\left\|P_t^n g_t\right\|^2 \mid \mathcal{F}_t\right] &\leq c_{\text{var },1} \left\|P_t g_t\right\|^2 + c_{\text{var },2} \left\|(I - P_t)\, g_t\right\|^2,\end{aligned} \tag{20}$$

où les constantes $c_{\text{bias },1/2}$ et $c_{\text{var },1/2}$ sont positives.

Remarque : Si l'on se place dans un espace de Hilbert $\mathcal{H} = \mathcal{H}(X, \rho)$ équipé du produit scalaire

$$(u, v) = \int (L_x u)^\top (L_x v) \mathrm{d}\rho(x)$$

où $L_x : \mathcal{H} \to \mathbb{R}^l$, $l \in \mathbb{N}$. Cependant, l'espace de Lebesgue $L^2(X, \rho)$ (qui est l'espace utilisé ici) correspond à $\mathcal{H}$ muni de $L_x v = v(x)$. Pour les espaces de Sobolev $H^1(X, \rho)$, on utilise $L_x v = \Big(v(x)\ \nabla v(x)\Big)^\top$. On ne considèrera que $L_x v = v(x)$ dans la suite.

5.2.1 Non-projection

On se fixe un step $t \in \mathbb{N}$. Pour estimer 16, on peut utiliser utiliser la mesure de sampling définie précédemment par

$$w_t \mathrm{d}\mu_t = \mathrm{d}\rho \tag{21}$$

et comme précédemment, on estime le produit scalaire (u, v) par

$$(u, v)_n = \frac{1}{n} \sum_{i=1}^{n} w_t(x_i) u(x_i) v(x_i) \tag{22}$$

où $x_1, \dots x_n$ sont indépendants et identiquement distribués samplé selon μ_t.
On considère $\mathcal{B}_t = \{\varphi_k\}_{k=1}^{d}$ avec $d \geq d_t$ une famile génératrice de $\mathcal{T}_t$. Alors une approximation de la projection g peut être donné par

$$P_t^n g = \sum_{k=1}^{d} \hat{\xi}_k \varphi_k \quad \text{avec} \quad \hat{\xi}_k = (g, \varphi_k)_n \tag{23}$$

Lemma 3 *Soient $x_1, \dots, x_n$ indépendants et identiquement distribuées samplé selon μ_t. Soit $G \in \mathbb{R}^{d \times d}$ tel que $G_{kl} = (\varphi_k, \varphi_l)$. On note $\lambda^*(G)$ et $\lambda_*(G)$ respectivement la plus grande valeur propre et la plus petite valeur propre de la matrice G. On pose :*

$$\mathcal{R}_t(x) = \lambda^*\Big(\sum_{k=1}^{d_t} b_k(x)^2 \Big) \text{ et } k_t = \| w_t \mathcal{R}_t \|_{L^\infty(\rho)}$$

Alors la projection définit par 23 vérifie 20 pour les constantes :

$$c_{biais,1} = \lambda_*(G), \quad c_{biais,2} = 0, \quad c_{var,1} = \frac{\lambda^*(G)(n-1) + \lambda^*(G) k_t}{n}, \quad \frac{\lambda^*(G) k_t}{n}$$

5.2.2 Quasi-projection

On comprend via le paragraphe précédent que le choix de la famille génératrice est important afin de pouvoir controler l'espérance et la variance. Dans cette section, on suppose que $\mathcal{B}_t$ est une base orthonormale. On se fixe un step $t \in \mathbb{N}$ et une base orthonormale $(\psi_k)_k$ de $\mathcal{T}_t$. On définie alors la *quasi-projection* de $g_t \in \mathcal{H}$ sur $\mathcal{T}_t$ par

$$P_t^n g_t = \sum_{k=1}^{d_t} \hat{\eta}_k \psi_k \quad \text{avec} \quad \hat{\eta}_k = (g_t, \psi_k)_n \tag{24}$$

En utilisant la linéarité de l'espérance et en faisant le calcul

$$\begin{aligned} \mathbb{E}(g_t, \psi_k)_n &= \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}\Big(w_t(x_i) g_t(x_i) \psi_k(x_i) \Big) \\ &= \frac{1}{n} \sum_{i=1}^{n} \int g_t(x) \psi_k(x) \mathrm{d}\rho(x) \\ &= (g_t, \psi_k) \end{aligned}$$

on constate que ce projecteur est sans biais. Cependant, ce n'est pas une projection : il suffit de voir que $P_t^n(P_t^n g_t) \neq P_t^n g_t$:

$$\begin{aligned} P_t^n\Big(P_t^n g_t\Big) &= \sum_{k=1}^{d_t}\left(\sum_{l=1}^{d_t}(g_t, b_k)_n (b_l, b_k)_n\right) b_k \\ &= \sum_{k=1}^{d_t}\Big[\hat{G}\hat{\eta}\Big]_k b_k \end{aligned}$$

et en général, $\hat{G} \neq I$, donc $\hat{G}\hat{\eta} \neq \hat{\eta}$ ce qui implique $P_t^n(P_t^n g_t) \neq P_t^n g_t$.

Remarque : Avec l'espace $\mathcal{T}_{\Phi_t}^h$ défini précédemment, on prend une famille génératrice φ_k et on définit la base orthonormée b qui en découle grâce à la matrice de Gram $H_k l = (\varphi_k, \varphi_l)$. Alors en écrivant

$$\eta = H^+\xi \text{ avec } \xi_k = (g, \varphi_k) \quad \text{et} \quad \hat{\eta} = H^+\hat{\xi} \text{ avec } \hat{\xi}_k = (g, \varphi_k)_n$$

ce qui permet d'exprimer la quasi-projection avec φ.

Lemma 4 *Soient $x_1, \ldots, x_n$ indépendants et identiquement distribués selon μ_t (pour un $\mathcal{T}_t$ donné). On pose*

$$\mathcal{R}_t(x) = \lambda^*\Big(\sum_{k=1}^{d_t} b_k(x)^2\Big) \text{ et } k_t = \|w_t \mathcal{R}_t\|_{L^\infty(\rho)}$$

Alors la quasi-projection P_t^n définit par l'équation 24 vérifie 20 pour les constantes

$$c_{bias\ ,1} = 1, \quad c_{bias\ ,2} = 0, \quad c_{var\ ,1} = \frac{n-1+k_t}{n}, \quad c_{var\ ,2} = \frac{k_t}{n}$$

On constate que les biais et la variance dépendent de k_t, qu'il va donc être crutial de controler. En effet, si ce n'est pas le cas on peut se retrouver dans les situations suivantes :

$$\begin{aligned} &1) k_t = d_t^2 \text{ si } w_t = 1 \text{ et } \rho \rightsquigarrow \mathcal{U} \text{ sur } X = [-1, 1] \\ &2) k_t = +\infty \text{ si } w_t = 1 \text{ et } \rho \rightsquigarrow \mathcal{N}(0, 1) \text{ sur } X = \mathbb{R} \end{aligned}$$

Il faut donc prendre une mesure de sampling μ_t et des poids w_t adaptés. Pour se faire, on va intrduire l'échantillonage de Christoffel généralisé : la fonction poids w_t qui minimise $k_t = \|w_t \mathcal{R}_t\|_L^\infty(\rho)$ est donné par

$$w_t = \|\mathcal{R}_t\|_{L^1(\rho)} \mathcal{R}_t^{-1}$$

Lemma 5 *Soit $(b_k)_k$ une base orthonormée de l'espace de dimension d_t $\mathcal{T}_t \subseteq \mathcal{H}$. On rappelle que l'on a défini*

$$\mathcal{R}_t(x) = \lambda^*\Big(\sum_{k=1}^{d_t} b_k(x)^2\Big)$$

et on note également

$$\mathcal{R}_{\mathcal{T}_t}(x) = \sup_{v\in\mathcal{T}_t} \|v(x)\|$$

la fonction de Christoffel inverse généralisée. Il vient que $\mathcal{R}_{\mathcal{T}_t} = \mathcal{R}_t$. On définit de plus

$$\tilde{\mathcal{R}}_t(x) = trace\left(\sum_{k=1}^{d_t} b_k(x)^2\right) = \sum_{k=1}^{d_t} \|b_k(x)\|^2$$

Alors $\mathcal{R}_t \leq \tilde{\mathcal{R}}_t$

Par définition, $\left\|\tilde{\mathcal{R}}_t\right\|_{L^1} = \int \tilde{\mathcal{R}}_t \mathrm{d}\rho = d_t$, donc les deux choix $w_t = \|\mathcal{R}_t\|_{L^1(\rho)} \mathcal{R}_t^{-1}$ ou $w_t = \left\|\tilde{\mathcal{R}}_t\right\|_{L^1(\rho)} {\tilde{\mathcal{R}}_t}^{-1}$ garantissent $k_t \leq d_t$.

5.2.3 Least squares projection

Bien qu'avec des techniques d'optimal sampling nous puissions controler le biais et la variance, l'expression proposée n'est pas une projection, ce qui peut impacter la stabilité. On propose alors un projecteur pondéré des moindrs carrés basé sur l'échantillonage Christoffel généralisé introduit précédemment. On définit la matrice de Gram empirique $\hat{G}_{kl} = (b_k, b_l)_n$ et $\hat{\eta} = (g, b_k)_n$. Alors la projection des moindres carrés pondérés de $g \in \mathcal{H}$ sur $\mathcal{T}_t$ est donnée par

$$P_t^n g = \sum_{k=1}^{d_t} \hat{\hat{\eta}} b_k \quad \text{avec} \quad \hat{\hat{\eta}} = \hat{G}^+ \hat{\eta} \tag{25}$$

Cette fois, l'expression proposée représente bien une projection ! En effet,

$$\begin{aligned} P_t^n(P_t^n g_t) &= \sum_{k=1}^{d_t}\sum_{l=1}^{d_t} \hat{G}^+(\hat{G}^+(g,b_k)b_k, b_l)b_l \\ &= \sum_{k=1}^{d_t} \hat{G}^+(g,b_k)_n \sum_{l=1}^{d_t}(\hat{G}^+ b_k, b_l)_n b_l \end{aligned}$$

et de plus

$$\hat{G}_{kl} = (b_k, b_l)_n, \quad (\hat{G}^+ b_k, b_l) = \sum_{m=1}^{d_t} (\hat{G}^+_{km} b_m, b_l)_n$$

d'où

$$P_t^n(P_t^n g_t) = \sum_{k=1}^{d_t} \hat{G}^+(g, b_k)_n \hat{G}^+ \hat{G} b_k$$
$$= \sum_{k=1}^{d_t} \hat{G}^+(g, b_k)_n b_k = P_t^n g_t$$

car la pseudo inverse peut également se voir comme

$$\hat{G}^+ = \lim_{\delta \to 0} (\mathrm{G}^*\mathrm{G} + \delta I)^{-1}\mathrm{G}^* = \lim_{\delta \to 0} \mathrm{G}^*(\mathrm{G}^*\mathrm{G} + \delta I)^{-1}$$
$$\Rightarrow \mathrm{G}\mathrm{G}^+ \simeq \mathrm{G}^+\mathrm{G} \simeq I$$

Cependant, ce n'est plus un estimateur sans biais de P_t :

$$\mathbb{E}\Big(P_t^n g_t\Big) = \sum_{k=1}^{d_t} \mathbb{E}\Big(\hat{G}^+ \hat{\eta} b_k\Big)$$
$$= \sum_{k=1}^{d_t} \mathbb{E}\left(\sum_{l=1}^{d_t} (b_k, b_l)_n (g, b_k)_n b_k \right)$$
$$= \sum_{k=1}^{d_t} \sum_{l=1}^{d_t} \mathbb{E}\Big((b_k, b_l)_n (g, b_k)_n b_k\Big)$$
$$\neq \sum_{k=1}^{d_t} \sum_{l=1}^{d_t} \mathbb{E}\Big((g, b_k)_n b_k\Big) = P_t g_t \quad \text{par linéarité (fait précédemment)}$$

Contrairement à la quasi-projection, on estime également la matrice de Gram par les produits scalaires empiriques, ce qui peut causer une erreur suplémentaire. Si les échantillons sont obtenus de manière optimale, on peut montrer que

$$\mathbb{P}\Big[\left\|I - \hat{G}\right\| \leq \delta\Big] \leq 2d_t \exp\Big(\frac{-\delta^2 n}{2d_t}\Big)$$

ce qui assure que la matrice estimée est proche de celle de Gram avec grande probabilité. De ce fait la quasi-projection et la projection ne diffère pas beaucoup. Etant donné que $\left\|I - \hat{G}\right\|$ controle la qualité de l'estimateur, on peut ajouter une condition supplémentaire lors du sampling pour assurer la stailité :

$$\mathcal{S}_\delta = \left\|I - \hat{G}\right\| \leq \delta \tag{26}$$

Lemma 6 *Soit $\delta \in]0, 1[$ et $x_1, \ldots, x_n$ tirés selon μ_t indépendants et identiquement distribués et satisfaisant la condition $\mathcal{S}_\delta$ avec probabilité $\mathbb{P}(\mathcal{S}_\delta|\mathcal{F}_t) = p_{\mathcal{S}_\delta} > 0$. Soient $\mathcal{R}_t(x) = \lambda^*\Big(\sum_{k=1}^{d_t} b_k(x)^2\Big)$ et $k_t = \|w_t \mathcal{R}_t\|_{L^\infty(\rho)}$. Alors l'opérateur de projection des moindes carrés défini en 25 satisfait 20 telle que*

$$c_{bias\ ,1} = 1, \quad c_{bias\ ,2} = \frac{\sqrt{k_t}}{(1-\delta)\sqrt{p_{\mathcal{S}_\delta} n}},$$
$$c_{var\ ,1} = \frac{1}{(1-\delta)^2 p_{\mathcal{S}_\delta}} \frac{n - 1 + k_t}{n}, \quad c_{var\ ,2} = \frac{1}{(1-\delta)^2 p_{\mathcal{S}_\delta}} \frac{k_t}{n}$$

5.3 hypothèses suplémentaires

Pour analyser la convergence de l'algorithme, il est nécessaire de faire quelques hypothèses suplémentaires. On définit pour commencer

$$\mathcal{L}_{\min} = \inf_{u\in\mathcal{H}} \mathcal{L}(u), \quad \mathcal{L}_{\min,\mathcal{M}} = \inf_{u\in\mathcal{M}} \mathcal{L}(u)$$

Pour un $u \in \mathcal{M}$ donné, on se donne $\mathcal{T}_u$ l'espace linéaire défini par l'algorithme au point u, et P_u la projection orthogonale sur $\mathcal{T}_u$ de u. On rappelle que P_u^n est un estimateur de P_u et R_u est la rétractation de u (voir le schéma 10). On suppose que les propriétés suivantes sont vérifiées :

— *Borne inférieure :* Il se trouve que $\mathcal{L}_{\min,\mathcal{M}} \in \mathbb{R}$ est finie et donc pour tout $u \in \mathcal{M}$

$$\mathcal{L}_{\min,\mathcal{M}} \leq \mathcal{L}(u) \tag{B}$$

— *L-régularité :* Il existe $L > 0$ telle que $\forall u \in \mathcal{M}$ et $g \in \mathcal{T}_u$

$$\mathcal{L}(u+g) \leq \mathcal{L}(u) + (\nabla\mathcal{L}(u), g) + \frac{L}{2}\|g\|^2 \tag{LS}$$

— λ-*Polyak-Łojasiewicz :* Il existe $\lambda > 0$ telle que $\forall u \in \mathcal{M}$

$$\|\nabla\mathcal{L}(u)\|^2 \geq 2\lambda(\mathcal{L}(u) - \mathcal{L}_{\min}) \tag{PL}$$

— λ-*Polyak-Łojasiewicz forte :* Il existe $\lambda > 0$ telle que $\forall u \in \mathcal{M}$

$$\|P_u\nabla\mathcal{L}(u)\|^2 \geq 2\lambda(\mathcal{L}(u) - \mathcal{L}_{\min}) \tag{SPL}$$

— *Controle de l'erreur de rétractation :* Il existe $C_R > 0$ telle que $\forall \in \mathcal{M}$ et $g \in \mathcal{T}_u$ et tout $\beta > 0$, la rétractation $R_u : \mathcal{H} \to \mathcal{M}$ satisfait

$$\mathcal{L}(R_u(u+g)) \leq \mathcal{L}(u+g) + \frac{C_R}{2}\|g\|^2 + \beta \tag{CR}$$

5.4 Expérimentation de l'algorithme

5.4.1 Exemple simple sur la projection non biaisé

On se place dans un cas simple afin de tester si l'algorithme fonctionne correctement. On considère $\mathcal{H} = L^2(\rho)$ où $\rho = \mathcal{U}(X)$ et $X = [-1, 1]$. On veut minimiser l'erreur des moindres carrés :

$$\mathcal{L}(v) = \frac{1}{2}\|u - v\|$$

On va essayer d'approcher $u(x) = \exp(x)$ sur l'espace $\mathcal{V} = \mathbb{P}_2(x)$ et on choisit la base $\mathcal{B}_t$ constituées des 3 premiers polynomes de Legendre. On choisit $\mathcal{M} = \mathcal{V}$ et $\mathcal{T}_t = \mathcal{V}$. Il vient que $R_t(v) = v$. Dans notre cas il a été montré que $\mathcal{L}_{\min,\mathcal{M}} = 3.6 \times 10^{-4}$. Il

est clair que l'hypothèse B est vérifiée. Etant donné que $R_t(v) = v$, l'hypothèse CR est vérifiée pour $C_R = 0$. Puis

$$\begin{aligned}\mathcal{L}(u+h) &= \frac{1}{2}\Big(u-v-h, u-v-h\Big) \\ &= \frac{1}{2}\Big(u-v, u\Big) - \Big(h, u\Big) - \frac{1}{2}\Big(u-v, v\Big) + \Big(h, v\Big) + \frac{1}{2}\|h\|^2 \\ &= \mathcal{L}(v) + \Big(\nabla\mathcal{L}, h\Big) + \frac{1}{2}\|h\|^2\end{aligned}$$

permet de dire que pour $L = 1$ et $\lambda = 1$ alors LS et PL sont vérifiées. On en déduit également que

$$\nabla\mathcal{L}(v) = -\Big(u - v\Big)$$

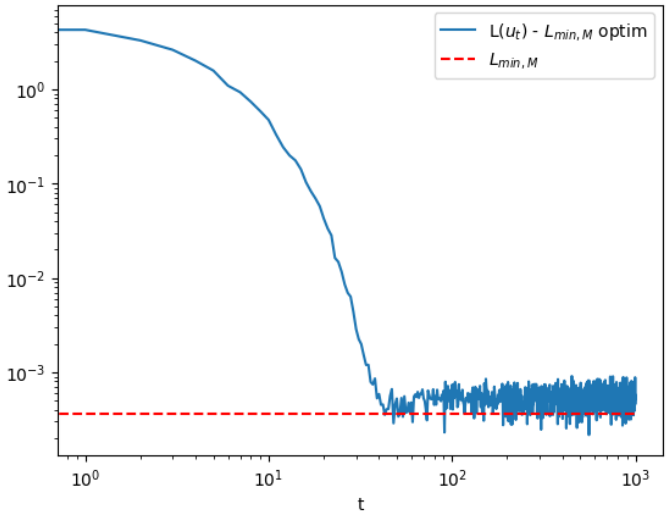
Dans notre cas, nous utiliserons la projection non biaisé définie en 24 par

$$P_t^n g_t = \sum_{k=1}^{d_t} \hat{\eta}_k \psi_k \quad \text{avec} \quad \hat{\eta}_k = (g_t, \psi_k)_n$$

Ainsi que la least-squares projection définit en 25 par

$$P_t^n g = \sum_{k=1}^{d_t} \hat{\hat{\eta}} b_k \quad \text{avec} \quad \hat{\hat{\eta}} = \hat{G}^{+}\hat{\eta}$$

Dans un premier temps on prend un step size constant tel que $s_t = 1/9$:



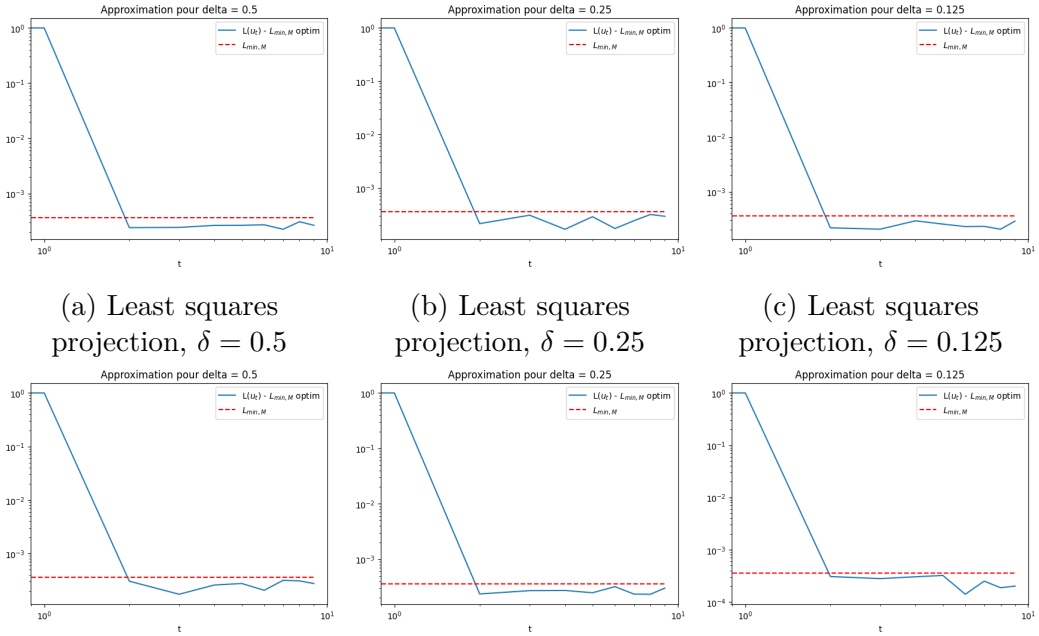
(a) Quasi-projection

FIGURE 11 – Quasi-projection : $\mathcal{L}(u_t) - \mathcal{L}_{min,\mathcal{M}}$ en fonction du nombre de step t

Lemma 7 *On suppose que la fonction perte satisfait LS, SPL et CR. On suppose également qu'à l'étape t, les* $x_1, \dots, x_n$ *sont identiques et indépendamment distribuées selon* μ_t*. Soit* $\delta \in (0, 1/2)$*,* $s_t \in \mathcal{O}(t^{\delta-1})$ *et* $\beta_t \in \mathcal{O}(t^{2\delta-2})$*. Enfin on suppose que* $\|(I - P_t)g_t\| \in l^\infty$ *presque surement. Alors* $\forall \epsilon \in (2\delta, 1)$ *on a presque surement*

$$\mathcal{L}(u_t) - \mathcal{L}_{min,\mathcal{M}} \in \mathcal{O}(t^{-1+\epsilon})$$

Ici, on ne peut pas appliquer ce lemme est donc il ne faut pas s'attendre à la convergence du modèle. On peut voir l'illustration de ce phénomène sur la Figure 11. Cependant, si l'on prend la matrice de Gram empririque proche de l'identité, c'est à dire lorsque l'on fait du sampling respectant 26, alors la least-squares projection et la quasi-projection sont très proche de la projection orthogonale. Dans ce cas, on peut utiliser un step size constant tel que $s_t = 1$ et obtenir la convergence en une seul itération : on observe ce phénomène Figure 12



(a) Least squares projection, $\delta = 0.5$

(b) Least squares projection, $\delta = 0.25$

(c) Least squares projection, $\delta = 0.125$

(d) Quasi projection, $\delta = 0.5$

(e) Quasi projection, $\delta = 0.25$

(f) Quasi projection, $\delta = 0.125$

FIGURE 12 – $\mathcal{L}(u_t) - \mathcal{L}_{min,\mathcal{M}}$ en fonction du nombre de step t

5.4.2 Application aux réseaux de neuronnes peu profonds

On sait que la linéarisation de l'espace $\mathcal{T}_{\Phi_\theta}$ peut s'exprimer par :

$$\begin{aligned}\mathcal{T}^h_{\Phi_\theta} &= \mathrm{vect}\Big(\sigma(A_0x+b_0)_j : j\in[1,m]\Big)\\ &+ \mathrm{vect}(1)\\ &+ \mathrm{vect}\Big(\sigma((A_0+E_{ij}h)x+b_0) : i\in[1,n], j\in[1,m]\Big)_i\\ &+ \mathrm{vect}\Big(\sigma(A_0x+(b_0+e_jh))_i : j\in[1,m]\Big)\end{aligned}$$

A partir de ces fonctions, il faut calculer la matrice de Gram $\mathbf{G} = \Big(\langle\phi_j,\phi_i\rangle\Big)_{i,j}$ où $(\phi_j)_{j=1,\dots d}$ est une famille génératrice de $\mathcal{T}_{\Phi_\theta}$.

Soit ρ la densité de probabilité de la mesure gaussienne standard sur $\mathbb{R}^d$, où $d\in\mathbb{N}$. Considérons deux vecteurs $\alpha,\alpha'\in\mathbb{R}^n$ et deux scalaires $\beta,\beta'\in\mathbb{R}$.
Il est possible de construire une matrice orthogonale $Q\in\mathbb{R}^{n\times n}$ telle que Qe_1 soit colinéaire à α et que Qe_2 soit proportionnel à $\alpha'-\frac{(\alpha',\alpha)}{\|\alpha\|^2}\alpha$. En posant $\gamma_1 := \alpha'^\top Qe_1$ et $\gamma_2 := \alpha'^\top Qe_2$, nous pouvons effectuer un changement de variables et réécrire l'intégrale sous la forme :

$$\begin{aligned}&\int_{\mathbb{R}^n}\sigma(\alpha^\top y+\beta)\sigma(\alpha'^\top y+\beta')\rho(y)\,dy\\ &= \int_{\mathbb{R}^n}\sigma(\alpha^\top Qx+\beta)\sigma(\alpha'^\top Qx+\beta')\rho(Qx)\,dx\\ &= \int_{\mathbb{R}^2}\sigma(\|\alpha\|x_1+\beta)\sigma(\gamma_1x_1+\gamma_2x_2+\beta')\left(\int_{\mathbb{R}^{n-2}}\rho(Qx)\,d(x_3,\dots,x_n)\right)d(x_1,x_2)\\ &= \int_{\mathbb{R}^2}\sigma(\|\alpha\|x_1+\beta)\sigma(\gamma_1x_1+\gamma_2x_2+\beta')\rho(x_1,x_2)\,d(x_1,x_2).\end{aligned}$$

Q est choisie de sorte que le premier vecteur de la nouvelle base Qe_1 soit colinéaire à α. Le vecteur Qe_2 est construit pour être dans le plan engrandré par α et α' mais orthogonal à Qe_1. Grâce à cette transformation Q on explime y dans la nouvelle base et les intégrales en dimension n se réduisent à une intégrale en dimension 2 sur x_1, x_2 car les autres dimensions sont intégrées sur une distribution gaussienne standard. On peut alors calculer l'intégrale par une quadrature de Gauss-hermite :

Quadrature de Gauss-Hermite en 2 dimensions [5]

On souhaite calculer l'intégrale gaussienne $\iint_{\mathcal{D}} f(x,y)\varphi(x)\varphi(y)\mathrm{d}x\mathrm{d}y$. On suppose qu'il n'y a pas de correaltion entre les deux variables normales standard x et y. On construit alors la quadrature de Gauss-Hermite en deux dimensions à partir d'une quadrature en une dimension qui a pour racine (z_k) et pour poids w_k, pour $k = 1, \dots m$. On fait alors l'approximation suivante :

$$\iint_{\mathcal{D}} f(x,y)\varphi(x)\varphi(y)\mathrm{d}x\mathrm{d}y \simeq \sum_{i=1}^{m}\sum_{j=1}^{m} w_i w_j f(z_i, z_j)$$

Remarque : Il faut alors supposer que α et α' sont linéairement indépendant pour appliquer cette quadrature de Gauss-Hermite.

On peut alors calculer la base orthonormée par le calcul explicité précédemment :

$$\psi_k = \sum_{j=1}^{d} c_{kj}\phi_j$$

Ensute, il nous suffit d'appliquer l'algorithme sur les paramètres du réseaux :

$$\theta_{t+1} = \theta_t - s_t P_t g_t$$

La descente se fait donc sur l'espace des paramètres.

6 Références

Références

[1] Albert Cohen, Giovanni Migliorati. Optimal weighted least-squares methods. SMAI Journal of Computational Mathematics, 2017. hal-01354003

[2] A. Cohen, M.A. Davenport, and D. Leviatan. On the stability and accuracy of least squares approximations. Found. Comput. Math., 13 :819-834, 2013.

[3] G. Leborgne. Opérateur à noyau intégral, espace de Hilbert à noyau reproduisant : introduction

[4] R. Gruhlke, A. Nouy and P. Trunschke, Optimal sampling for stochastic and natural gradient descent

[5] P. Jackel, A note on multivariate Gauss-Hermite quadrature