L2-Boosting et k-plus proches voisins

Michael WEIL, Xing WEI

29 novembre 2014

1 Introduction

Le L2-boosting est un algorithme itératif de regression ou de classification. Il consiste de manière générale à modifier un estimateur obtenu à une étape de l'itération en lui ajoutant une fonction de son résidu afin d'obtenir l'estimateur dans l'étape suivante. Différents lisseurs peuvent être utilisés, mais dans notre projet nous utiliserons un même estimateur. L'estimateur de base choisi pour ce projet semble à premiere vue être assez intuitif : l'estimateur des k-plus proches voisins, ainsi que sa variante, l'estimateur des k-plus proches voisins mutuels. Cependant, cet estimateur de base ne semble pas être adapté à l'algorithme du L2-boosting.

Dans un premier temps, nous présenterons succinctement l'algorithme ainsi que l'estimateur des plus proches voisin. Ensuite nous montrerons à l'aide de simulations puis de manière théorique que ces estimateurs ne sont pas adaptés au L2-boosting.

2 Théorie : L2-boosting & lisseur des k-plus proches voisins

2.1 L2-boosting

Avant de détailler l'algorithme du L2-boosting, il est nécessaire d'évoquer le boosting. Le boosting est un domaine en apprentissage statistique très utilisé pour la classification. Il repose sur des algorithmes d'itérations adaptatives visant à réduire le biais. L'itération se fait sur des classifieurs faibles. Un classifieur faible est un classifieur classant deux ensembles mieux que le hasard ne pourrait le faire.

Le boosting fait apprendre au classifieur faible pour être ajouté à un "classifieur fort". En l'ajoutant, il est pondéré en fonction de sa capacité à bien classer les données. Ces données sont aussi pondérées : les exemples mal appris par le classifieur sont "boostés" en augmentant leurs poids, tandis que les exemples bien appris perdent du poids. Ainsi, les prochains classifieurs faibles prendront mieux en compte les exemples boostés.

Parmi les algorithmes les plus célèbres, nous pouvons entre autres citer l'algorithme du AdaBoost. Dans le cadre de problèmes de régressions, les algorithmes de boosting sont moins utilisés.

Maintenant, détaillons le principe du L2-boosting.

Le but est d'estimer la fonction $F : \mathbf{R}^d \longrightarrow \mathbf{R}$ en minimisant $\mathbf{E}[(Y - F(X))^2]$ avec les données $(X_i, Y_i)_{i=1...n}$.

Dans le cadre de notre projet, prenons comme échantillon des points de la fonction pertubée :

$$Y_i = F(X_i) + \epsilon_i, i = 1, ...n,$$

$$\epsilon_1, ..., \epsilon_n \ i.i.d. \ avec \ \mathbf{E}[\epsilon_i] = 0, Var(\epsilon_i) = \sigma^2$$

Écrivons sous forme vectorielle :

$$Y = F(X) + \xi \quad Y, \xi \in \mathbf{R}^n$$

L'estimateur de base peut être représenté par un operateur linéaire $\mathcal{S} : \mathbf{R}^n \longrightarrow \mathbf{R}^n$. La fonction estimée $\hat{F} = \mathcal{S}Y$. Le biais de l'estimateur $\mathcal{S}$ est :

$$\begin{aligned} b &= \mathbf{E}[\hat{F} - F] \\ &= \mathbf{E}[\mathcal{S}Y - Y] \\ &= -\mathbf{E}[(\mathcal{I} - \mathcal{S})Y] \end{aligned}$$

Alors nous pouvons estimer le biais b en lissant le résidu $R = (\mathcal{I} - \mathcal{S})Y$. Pour cela, nous appliquons le même estimateur $\mathcal{S}$:

$$\begin{aligned} \hat{b} &= -\mathcal{S}R \\ &= -\mathcal{S}(\mathcal{I} - \mathcal{S})Y \end{aligned}$$

Donc nous pouvons corriger le biais en soustrayant le $\hat{b}$. Cela conduit à une autre estimation de F :

$$\begin{aligned} \hat{F}_2 &= \hat{F} - \hat{b} \\ &= \mathcal{S}Y + \mathcal{S}(\mathcal{I} - \mathcal{S})Y := \mathcal{S}_2 Y \end{aligned}$$

Nous pouvons ensuite refaire la même chose pour l'estimateur $\mathcal{S}_2$. Ainsi, l'algorithme du L2-boosting se présente comme ceci :

1. On obtient l'estimation : $\hat{F}_k = \mathcal{S}_k Y$
2. On applique le lisseur $\mathcal{S}$ au résidu : $\hat{F}_{k+1} = \mathcal{S}_k Y + \mathcal{S}(Y - \mathcal{S}_k Y) := \mathcal{S}_{k+1} Y$
3. On itère le procédé.

Une relation de récurrence se dégage de cet algorithme :

$$\hat{F}_n = (\mathcal{I} - (\mathcal{I} - \mathcal{S})^n)Y$$

L'idée du L2-boosting est de partir d'un estimateur de très grand biais, d'itérer ce procédé pour réduire le biais pour ainsi obtenir un "bon estimateur".

2.2 Estimateur des k-plus proches voisins

Pour notre projet, nous étudions le procédé du L2-Boosting en utilisant les lisseurs des k-plus proches voisins (*kppv*) et des des k-plus proches voisins mutuels dont voici les définitions.

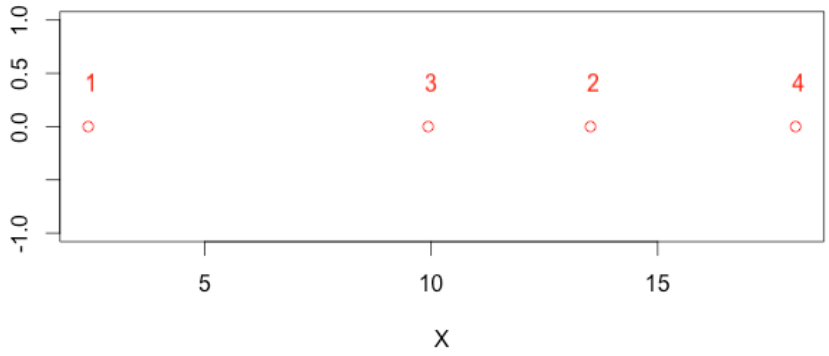
Le lisseur kppv est défini par la matrice $\mathcal{S} \in \mathbf{R}^{n \times n}$ suivante

$$\mathcal{S}_{ij} = \begin{cases} \frac{1}{k} \text{ si j est un des k-plus proches voisins de i} \\ 0 \text{ sinon} \end{cases}$$

Par convention un point est lui même un des ses k-plus proches voisins. Donc

$$\mathcal{S}_{ii} = \frac{1}{k}$$

Illustrons à l'aide d'un exemple :



La matrice des k-plus proches voisins avec k=2 est :

$$\mathcal{S} = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}$$

Nous remarquons que la matrice $\mathcal{S}$ n'est pas nécessairement symétrique. Il en va de même pour la matrice d'adjacence associée

$$\mathcal{M} = k\mathcal{S}$$

La matrice $\mathcal{S}$ est stochastique : la somme sur chaque ligne vaut 1. Cependant la somme sur chaque colonne ne vaut pas nécessairement 1. Cette somme mesure pour chaque point le nombre d'appartenances à un voisinage. Une nombre important d'appartenances à un voisinage est communément appelé "hub". Un déséquilibre entre ces valeurs manifesté par la présence d'hubs peut engendrer des problèmes de classification.

Pour remédier au manque de symetrie et à la présence d'éventuels hubs nous pouvons considérer la matrice des k-plus proches voisins *mutuels* : (i, j) sont voisins mutuels si i est voisin de j et réciproquement.

Nous pouvons définir la matrice d'adjacence de k plus proches voisins mutuels comme ceci

$$\tilde{\mathcal{M}} = \mathcal{M} * {}^t\mathcal{M}$$

avec $*$ le produit d'Hadamard :

$$(A * B)_{ij} = A_{ij}B_{ij}$$

Dans l'exemple précedent, la matrice d'adjacence est la suivante :

$$\tilde{\mathcal{M}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La matrice $\tilde{\mathcal{M}}$ est bien symétrique. La somme sur une colonne i donnée est égale à la somme sur la ligne i. Notons k_i la somme de chaque ligne de $\tilde{\mathcal{M}}$. Ces valeurs sont inférieures aux valeurs des hubs de la matrice $\mathcal{M}$. En effet, un point appartient désormais au voisinage d'un autre point si ce dernier est aussi un de ses k-plus proches voisins.

Nous pouvons aussi considérer le lisseur des kppv-mutuels :

$$\tilde{\mathcal{S}} = \mathcal{W}\tilde{\mathcal{M}}$$

avec $\mathcal{W} = diag(\frac{1}{k_1}, \frac{1}{k_2}, ..., \frac{1}{k_n})$.

Dans notre exemple précedent,

$$\tilde{\mathcal{S}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Nous testerons d'une part le L2-boosting tout d'abord avec la matrice du lisseur kppv $\mathcal{S}$, et ensuite avec la matrice du lisseur kppv-mutuels $\tilde{\mathcal{S}}$.

3 Echec du L2-Boosting avec k-plus proches voisins

Pour tester l'algorithme du L2-boosting à la fois avec les lisseurs kppv et kppv-mutuels, nous avons effectué des simulations sur le logiciel R.

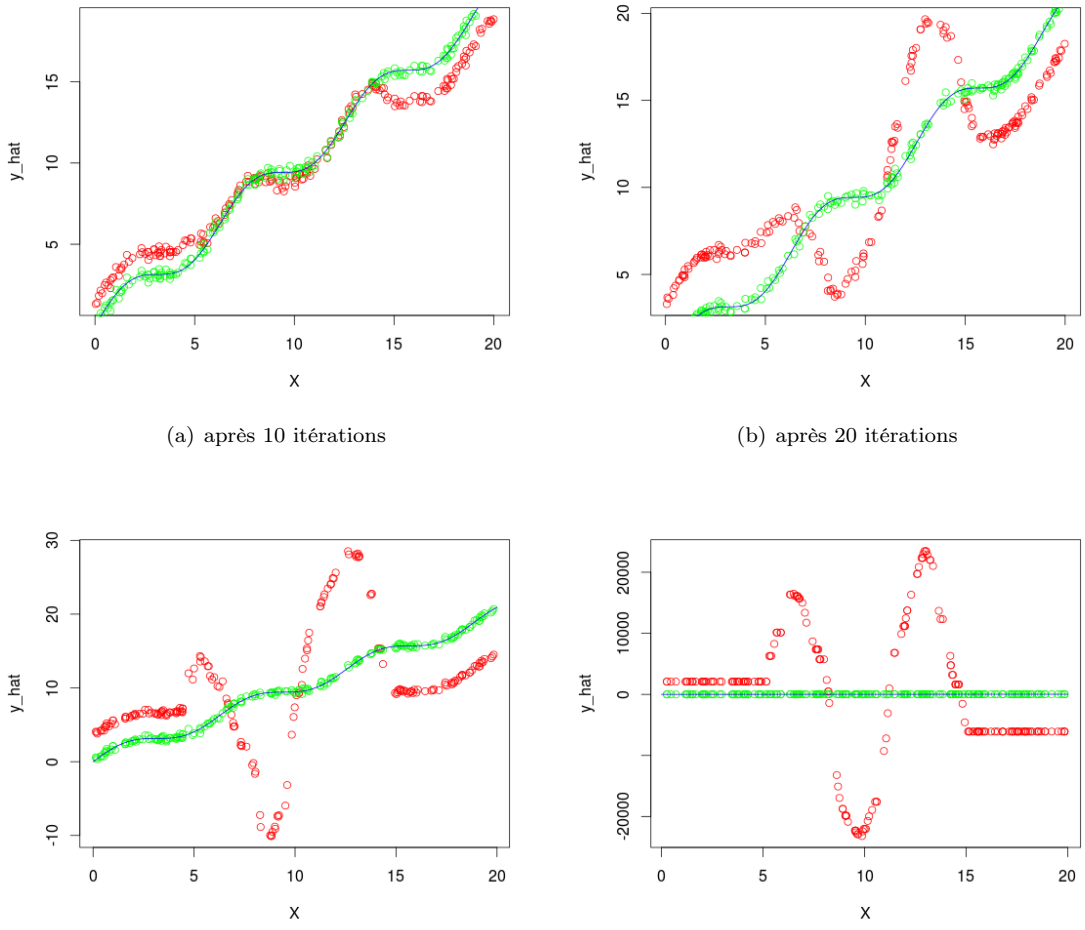
Nous avons testé le boosting dans le cas de la régression d'une fonction en dimension 1. Pour cela nous avons pris les données suivantes

$$(X_i, Y_i)_{i=1,...n}$$

avec $n = 100$, X_i sont des valeurs prises uniformement au hasard dans l'intervalle [0,20] et

$$Y_i = \sin(X_i) + X_i + \epsilon_i, \epsilon_1, ..., \epsilon_n \ i.i.d. \ avec \ \mathbf{E}[\epsilon_i] = 0, Var(\epsilon_i) = 0.04$$

On itère le procédé de l'algorithme avec k=$\frac{n}{2}$. Le graphique 1 et 2 montre que que la regression diverge quand on applique L2-boosting en kppv ou kppv-mutuels.



(a) après 10 itérations

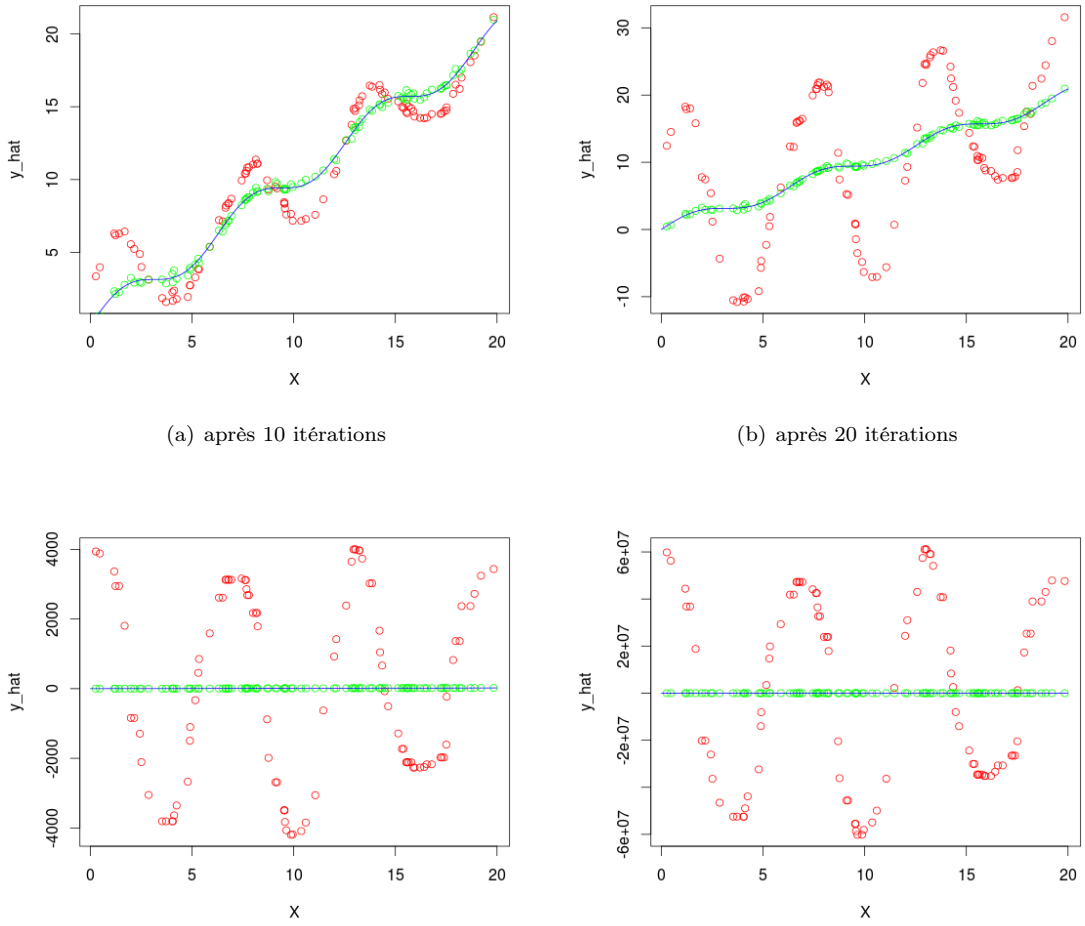
(b) après 20 itérations

(c) après 50 itérations

(d) après 100 itérations

FIGURE 1 – Échec du L2-boosting avec kppv. La régression est représentée en rouge, la fonction à approcher est en vert.

Les résultats des simulations réalisées sous-entendent une impossibilité de se servir de la méthode du L2-boosting avec les lisseurs des k-plus proches voisins. Il faut confirmer ces résultats avec la théorie. Pour cela, nous allons étudier les lisseurs des k-plus proches voisins et k-plus proches voisins mutuels.



(a) après 10 itérations

(b) après 20 itérations

(c) après 50 itérations

(d) après 100 itérations

FIGURE 2 – Échec du L2-boosting avec kppv-mutuels.

4 Étude de l'échec du kppv-mutuels

4.1 Étude du spectre de $\tilde{\mathcal{S}}$

Nous voyons que le L2-boosting avec k-plus proches voisins aboutit à une regression qui explose après plusieurs itérations. Selon l'article [2], si la valeur propre de la matrice $\mathcal{I} - \mathcal{S}$ n'est pas entre 0 et 1, alors le L2-boosting ne peut pas converger.

Notons qu'avec k-plus proches voisins mutuels, la matrice de l'estimateur s'écrit comme ceci :

$$\tilde{\mathcal{S}} = \mathcal{W}\tilde{\mathcal{M}}$$

où la matrice $\mathcal{W} = diag(\frac{1}{k_1}, \frac{1}{k_2}, ..., \frac{1}{k_n})$ est une matrice diagonale, et chaque k_i est la somme de ligne i de $\tilde{\mathcal{M}}$. Alors nous avons la proposition suivante :

Proposition 1 *Les valeurs propres de la matrice d'adjacence $\tilde{\mathcal{M}}$ sont de même signe que celles de la matrice de lissage $\tilde{\mathcal{S}}$.*

Pour démontrer cette proposition, montrons d'abord que les valeurs propres de $\mathcal{W}\tilde{\mathcal{M}}$ sont égaux aux valeurs propres de $\sqrt{\mathcal{W}}\tilde{\mathcal{M}}\sqrt{\mathcal{W}}$.

Étudions les polynômes caracteristiques des deux matrices :

$$\begin{aligned} det(\mathcal{W}\tilde{\mathcal{M}} - \lambda\mathcal{I}) &= det(\sqrt{\mathcal{W}}\sqrt{\mathcal{W}}\tilde{\mathcal{M}} - \lambda\mathcal{I}) \\ &= det(\sqrt{\mathcal{W}}) \times det(\sqrt{\mathcal{W}}\tilde{\mathcal{M}} - \lambda\sqrt{\mathcal{W}}^{-1}) \\ &= det(\sqrt{\mathcal{W}}) \times det(\sqrt{\mathcal{W}}\tilde{\mathcal{M}}\sqrt{\mathcal{W}} - \lambda\mathcal{I}) \times det(\sqrt{\mathcal{W}})^{-1} \\ &= det(\sqrt{\mathcal{W}}\tilde{\mathcal{M}}\sqrt{\mathcal{W}} - \lambda\mathcal{I}) \end{aligned}$$

Ensuite, montrons que les valeurs propres de $\sqrt{\mathcal{W}}\tilde{\mathcal{M}}\sqrt{\mathcal{W}}$ et de $\tilde{\mathcal{M}}$ ont le même signe. Pour cela, on étudie le quotient de Rayleigh de la matrice.

Definition 1 *Soit $\mathcal{A}$ une matrice symétrique. Alors le quotient de Rayleigh R est défini comme ceci*

$$\forall x \in \mathbf{R}^n,\ R(x) = \frac{{}^t x\mathcal{A}x}{{}^t xx}$$

Le quotient de Rayleigh est très pratique car il a un lien avec le spectre de la matrice symétrique associée.

Proposition 2 *$\forall x \in \mathbf{R}^n$, $\lambda_{min} \leq R(x) \leq \lambda_{max}$ avec λ_{min} et λ_{max} respectivement la plus petite et la plus grande valeur propre de A.*

Pour un vecteur x donné, le quotient de Rayleigh pour la matrice $\sqrt{\mathcal{W}}\tilde{\mathcal{M}}\sqrt{\mathcal{W}}$ s'écrit :

$$R(x) = \frac{{}^t x\sqrt{\mathcal{W}}\tilde{\mathcal{M}}\sqrt{\mathcal{W}}x}{{}^t xx}$$

En posant $y = \sqrt{\mathcal{W}}x$, le quotient vaut :

$$\begin{aligned} R(x) &= \frac{{}^t y\tilde{\mathcal{M}}y}{{}^t yy}\frac{{}^t yy}{{}^t xx} \\ &= \frac{{}^t y\tilde{\mathcal{M}}y}{{}^t yy}\frac{{}^t x\mathcal{W}x}{{}^t xx} \end{aligned}$$

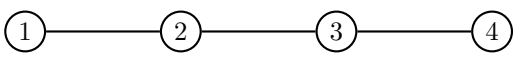
Mais comme les valeurs propres de $\mathcal{W}$ (les $\frac{1}{k_i}$) sont positives, on conclut que les valeurs propres de $\sqrt{\mathcal{W}}\tilde{\mathcal{M}}\sqrt{\mathcal{W}}$ et de $\tilde{\mathcal{M}}$ ont le même signe.

Donc nous devons prouver l'existence d'une valeur propre strictement négative de $\tilde{\mathcal{M}}$. Nous allons étudier la matrice $\tilde{\mathcal{M}}$.

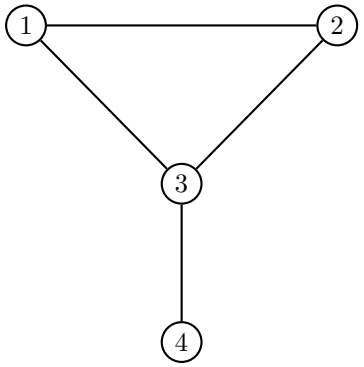
4.2 Etude avec les graphes associés aux matrices d'adjacences

La matrice d'adjacence $\tilde{\mathcal{M}}$ des k-plus proches voisins mutuels peuvent-être vus comme une matrice associée à un graphe non orienté.

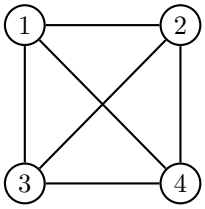
Par exemple prenons pour n=4 la matrice $\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$ est associée au graphe suivant :



La matrice $\begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$ est associée au graphe :



Dans ces deux exemples les matrices ont une valeur propre strictement négative. Par contre la matrice $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$ associée au graphe



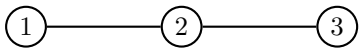
n'a pas de valeurs propres strictement négatives.

Dans le graphe, remarquons que tous les chemins sont de longueur 1 (c'est un graphe complet). Les graphes precédents ont des chemins de longueur 2.

Nous pouvons emmetre l'hypothèse suivante :

Proposition 3 *La matrice d'adjacence associée à un graphe ayant un chemin de longueur supérieure ou égale à 2 admet au moins une valeur propre strictement négative.*

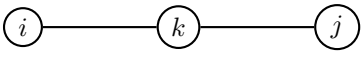
Étudions maintenant le cas plus simple qui vérifie la proposition : un graphe avec 3 sommets et 2 arêtes :



La matrice d'adjacence associée est $\tilde{\mathcal{M}}_0 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$.

Et nous pouvons calculer et trouver une valeur propre négative égale à $1 - \sqrt{2}$. Le vecteur propre associé est ${}^t(1, -\sqrt{2}, 1)$.

Maintenant considérons un graphe où il y a un (plus court) chemin de longueur 2 entre le sommet i et le sommet j, i.e. il y a une arête (i, k) et une arête (k, j), mais pas d'arête (i, j) :



Alors si on extrait les i, j, k-ème colonnes et lignes de $\tilde{\mathcal{M}}$, nous obtenons une matrice 3×3 identique à la matrice $\tilde{\mathcal{M}}_0$.

Considérons à nouveau le quotient de Rayleigh, pour un vecteur x de dimension N, le quotient est :

$$R(x) = \frac{{}^t x \tilde{\mathcal{M}} x}{{}^t x x}$$

Alors nous pouvons choisir un x où ${}^t(x_i, x_k, x_j)$ est égal à vecteur propre associé au valeur propre $1-\sqrt{2}$ de $\tilde{\mathcal{M}}_0$ (par exemple ${}^t(1, -\sqrt{2}, 1)$) , et tous les autres composantes sont nulles, alors le quotient $R(x)$ est négatif, donc nous avons démontré la proposition 3.

Pour n quelconque, nous pouvons diagonaliser la matrice par blocs. Chaque bloc correspond à une composante connexe du graphe.

Intuitivement, pour ne pas avoir de valeurs propres strictements negatives, il suffirait que tous les sous-graphes associés aux composantes connexes du graphe soient complets. Ce cas peut-il se produire pour n grand ? D'après l'article [1], en prenant k plus grand que $\log(n)$, des composantes complètes sont "connectées". Ainsi il existe une composante connexe du graphe non complète.

5 Étude de l'échec du kppv

D'abord nous allons montrer qu'il existe une valeur propre plus grande que 1 pour la matrice $(I-S)^t(I-S)$, nous pouvons étudier le quotient de Rayleigh, et montrer qu'il existe un vecteur x tel que :

$${}^t x(\mathcal{I} - \mathcal{S})^t(\mathcal{I} - \mathcal{S})x > {}^t x x$$

i.e

$${}^t x \mathcal{S}^t \mathcal{S} x > {}^t x(\mathcal{S} + {}^t\mathcal{S})x$$

Comme $\mathcal{S} = \frac{\mathcal{M}}{k}$, nous avons :

$$\begin{aligned} {}^t x \mathcal{M}^t \mathcal{M} x &> k\,{}^t x(\mathcal{M} + {}^t\mathcal{M})x \\ {}^t x(\mathcal{M}^t\mathcal{M} - k(\mathcal{M} + {}^t\mathcal{M}))x &> 0 \end{aligned}$$

Soit $\mathcal{A} = (\mathcal{M}^t\mathcal{M} - k(\mathcal{M} + {}^t\mathcal{M}))$. Remarquons $(\mathcal{M}^t\mathcal{M})_{ij} =< \mathcal{M}_i, \mathcal{M}_j >$, où $\mathcal{M}_j$ est la jème ligne de $\mathcal{M}$. Alors nous avons : $\mathcal{A}_{ii} = -k$ et $2 - 2k \leq \mathcal{A}_{ij} \leq k - 2, i \neq j$.

Pour $x = \sum x_i e_i$,

$$\begin{aligned} {}^t x A x &= {}^t(\sum_i x_i e_i) A (\sum_i x_i e_i) \\ &= \sum_i A_{ii} x_i^2 + \sum_{j \neq i} A_{ij} x_i x_j \\ &= \sum_i -k x_i^2 + \sum_{j \neq i} A_{ij} x_i x_j \end{aligned}$$

Soit $f(x) = \sum_i -k x_i^2 + \sum_{j \neq i} A_{ij} x_i x_j$ f est un polynôme en $x_1, ..., x_n$. Une étude asymptotique montre que f tend vers $-\infty$ pour $\|x\| \to \infty$. Donc f admet un maximum. Pour atteindre le maximum, il faut que les dérivés partielles soient nulles (point critique) :

$$\partial_{xi} f(x) = -2kx_i + \sum_{j \neq i} A_{ij} x_j = 0$$

Alors en ce point critique :

$$\begin{aligned} f(x) &= \sum_i -kx_i^2 + \sum_{j\neq i} A_{ij}x_i x_j \\ &= \sum_i -kx_i^2 + \sum_i x_i \sum_{j\neq i} A_{ij}x_j \\ &= \sum_i -kx_i^2 + \sum_i (x_i \times 2kx_i) \\ &= \sum_i kx_i^2 > 0 \end{aligned}$$

Le maximum étant positif, la matrice $(\mathcal{I}-\mathcal{S})^t(\mathcal{I}-\mathcal{S})$ admet une valeur propre plus grande que 1. Alors nous avons :

$$^t x(\mathcal{I}-\mathcal{S})^t(\mathcal{I}-\mathcal{S})x > {}^t xx$$

C'est à dire que :

$$\|^t(\mathcal{I}-\mathcal{S})x\| > \|x\|$$

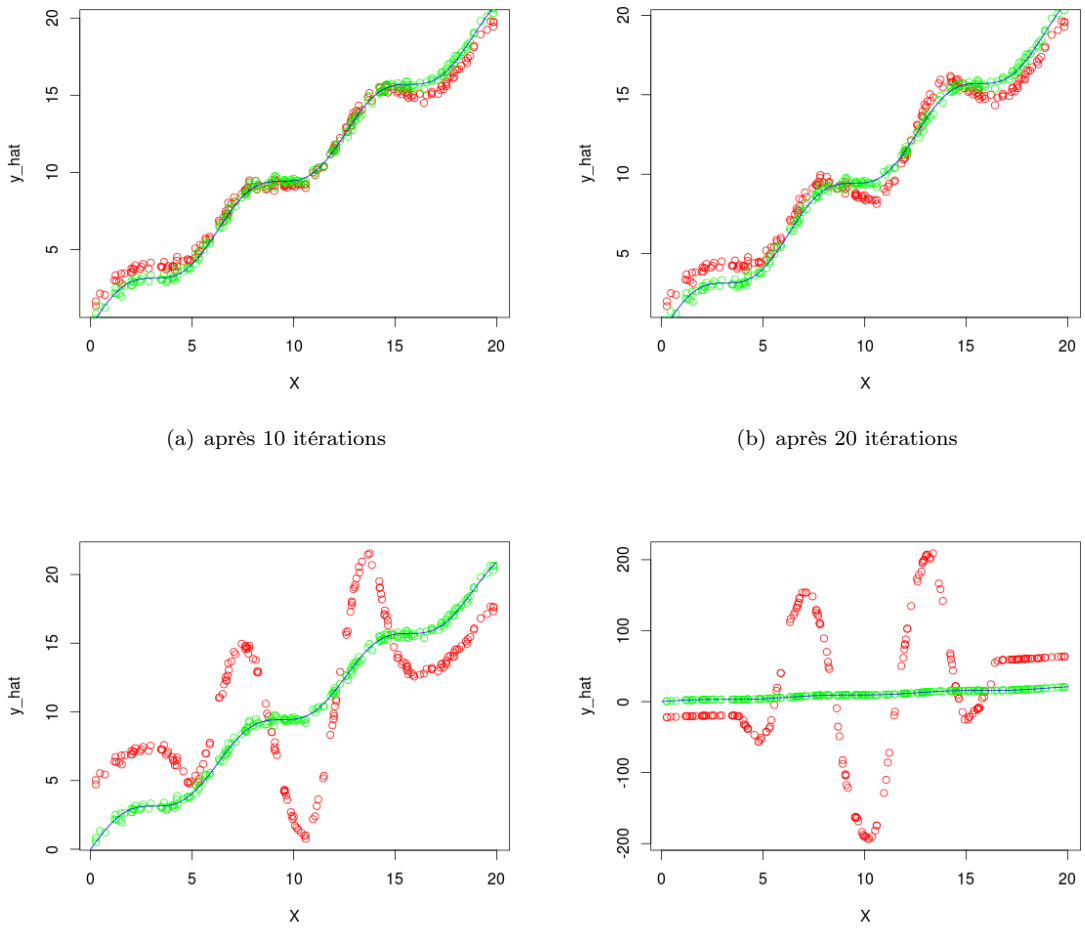
Si la matrice $(\mathcal{I}-\mathcal{S})$ est diagonalisable, elle admet une valeur propre plus grande que 1 et nous pouvons conclure. Pour $(\mathcal{I}-\mathcal{S})$ quelconque, le problème reste ouvert.

6 Conclusion

Lors ce EA, nous avons étudié la méthode de L2-boosting avec k-plus proches voisins. Un résultat de simulation suggère que le kppv ne fonctionne pas avec L2-boosting. Nous avons confirmé ces résultats avec la théorie dans le cas des k-plus proches voisins mutuels et des des k-plus proches voisins diagonalisables. L'échec de L2 boosting est due à l'existence d'une valeur propre de la matrice de lissage strictement négative.

Pour éviter ce problème, il faut éviter la valeur propre négative, par exemple, utiliser les différent lisseurs à chaque itération. Nous avons essayé d'utiliser le lisseur kppv avec k aléatoirement choisi entre $[\frac{3n}{10}, \frac{7n}{10}]$, où n correspond à la taille des données. Dans ce cas là, l'erreur explose toujours, mais moins vite (voir figure 3).

Nous pouvons aussi penser au cas où k est petit. En effet nous pouvons imaginer que les lisseurs kppv-mutuels auront un spectre positif, néanmoins la variance risque d'augmenter. Il faut sans doute, en choisissant la valeur de k, trouver un compromis entre stabilité et précision.



(a) après 10 itérations

(b) après 20 itérations

(c) après 50 itérations

(d) après 100 itérations

FIGURE 3 – L2-boosting de kppv avec k aléatoirement choisi à chaque itération, nous constatons que l'erreur explose moins vite.

Références

[1] M.R. Brito, E.L. Chávez, A.J. Quiroz, and J.E. Yukich. Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. *Statistics Probability Letters*, 35(1) :33 – 42, 1997.

[2] Pierre-André Cornillon, N. W. Hengartner, and E. Matzner-Løber. Recursive bias estimation for multivariate regression smoothers. *ESAIM : Probability and Statistics*, 18 :483–502, 1 2014.

[3] K. Chidananda Gowda and G. Krishna. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognition*, 10(2) :105 – 112, 1978.

[4] Buhlmann P. and Yu B. Boosting With the L2 Loss : Regression and Classification. *Journal of the American Statistical Association*, 98 :324–339, January 2003.