Université de Montpellier, France

Projet HMMA307 - M2 MIND

Utilisation de la médiane géométrique et de l'estimation robuste dans des espaces de Banach

Hanna Bacave

Novembre 2020

Table des matières

Introduction

Dans ce document, nous allons présenter la façon dont nous avons implémenté, grâce au logiciel Python, les travaux effectués par Stanislav Minsker *Geometric median and robust estimation in Banach spaces* [0]. Après une rapide présentation de ce qu'est l'estimateur médian, nous nous interesserons plus particulièrement aux travaux de Stanislav Minsker, avant de réaliser l'implémentation de ces recherches.

1 Présentation de la régression quantile

1.1 Médiane

A partir de la ressource [1], nous allons introduire la définition de la médiane.

Définition 1. *Soit $y_1, ..., y_n \in \mathbb{R}$, on définit la médiane par :*

$$Med_n(y_1, ..., y_n) \in \arg\min_{\mu \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} |y_i - \mu|.$$

En pratique, on utilise plutôt une écriture de la médiane comme suit :

$$Med_n(y_1, ..., y_n) = \begin{cases} y_{\lfloor \frac{n}{2} \rfloor - 1} \text{ si } n \text{ est pair} \\ \frac{y_{\lfloor \frac{n}{2} \rfloor} + y_{\lfloor \frac{n}{2} \rfloor - 1}}{2} \text{ si } n \text{ est impair} \end{cases}.$$

1.2 Quantile

A partir de la ressource [1], nous allons introduire la définition du quantile.

Définition 2. *Le quantile de niveau α est défini par :*

$$\forall \alpha \in]0, 1], \ q_\alpha(y_1, ..., y_n) = \inf\{t \in \mathbb{R} : F_n(t) \geq \alpha\}.$$

où

$$F_n(t) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{y_i \leq t\}}.$$

Remarque 1. *Le quantile de niveau $\alpha = 0.5$ est la médiane.*

Définition 3. *On définit la fonction de perte de niveau α par :*

$$l_\alpha : \begin{array}{ccc} \mathbb{R} & \longrightarrow & \mathbb{R} \\ x & \longmapsto & \begin{cases} -(1-\alpha)x \ si \ x \geq 0 \\ \alpha x \ si \ x \leq 0 \end{cases} \end{array} .$$

1.3 Régression quantile

On dispose de
— $y_1, ..., y_n \in \mathbb{R}$ observations,
— $x_1, ..., x_n \in \mathbb{R}^p$ variables explicatives.

Définition 4. *Soit $\alpha \in]0, 1[$, on appelle régression quantile les coefficients :*

$$\beta^\alpha \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} l_\alpha(y_i - x_i^T \beta),$$

où

$$X = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix} \in \mathbb{R}^{nxp}.$$

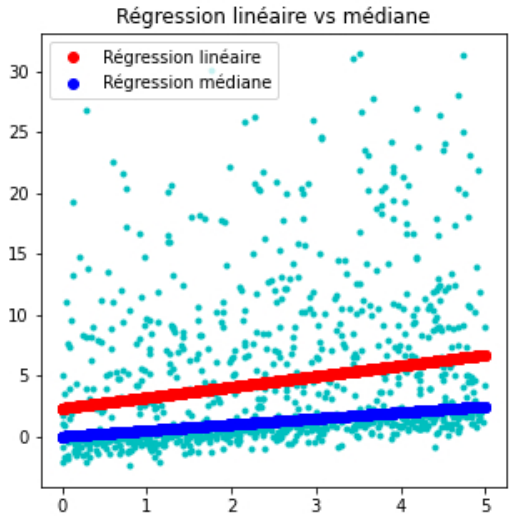
Remarque 2. *La régression quantile est efficace sur des jeux de données :*
— *ayant une distribution non symétrique par rapport à la moyenne*
— *ayant une nature hétéroscédastique.*
En fait, la régression quantile est moins sensible aux points aberrants que d'autre régression.

Exemple 1. *Après avoir simulé des données dont la représentation est non symétrique par rapport à la moyenne et portant des points abérrants, nous allons comparer la régression médiane avec une régression linéaire présentée dans la Figure 1. On observe que la régression linéaire accorde plus de poids aux valeurs extrêmes ce qui lui confère un résultat moins efficace que pour la regression quantile.*

2 Régression quantile en grande dimension avec des données sparses

Dans cette section, nous allons présenter la façon dont nous avons implémenté la méthode de régression quantile en grande dimension avec des données sparses, présentée dans la ressource [0]. Ensuite, nous présenterons le moyen utilisé pour comparer les régressions entre elles.

FIGURE 1 – Comparaison de la régression linéaire avec la régression quantile sur un jeu de données simples



2.1 Implémentation pratique

On définit :
— $x_1, ..., x_n \in \mathbb{R}^D$,
— $\lambda_0 \in \mathbb{R}^D$.

Définition 5. *On veut résoudre le problème d'optimisation suivant :*

$$Y_j = \lambda_0^T x_j + \epsilon_j$$

où
— ϵ_j *est un vecteur iid de moyenne* 0, *mais n'est pas gaussien ;*
— λ_0 *est un vecteur sparse, c'est-à-dire que la dimension s du support de* λ_0 *est très nettement inférieure à* D *;*
— $D >> n$

Une première solution serait de calculer l'estimateur Lasso de λ_0 :

$$\hat{\lambda}_\epsilon = \underset{\lambda \in \mathbb{R}^D}{\arg\min} \left[\frac{1}{n} \sum_{j=1}^{n} (Y_j - \lambda^T x_j)^2 + \epsilon \|\lambda\|_1 \right].$$

Or, cette solution n'est pas la plus efficace, car pour augmenter son score, il faudrait changer le modèle. Notre objectif est donc - sans changer le modèle,

ni faire de suppositions sur ϵ - de calculer le meilleur estimateur sur ce type de données. On se propose de procéder comme suit :

1. On commence par définir les quantités suivantes :
   — $t > 0$ fixé,
   — $k = \lfloor 3.5t \rfloor + 1$
   — $m = \lfloor \frac{n}{k} \rfloor$
   On divise notre échantillon $x_1, ..., x_n$ en k groupes disjoints $G_1, ..., G_k$ de taille m chacun. De plus, pour $1 \leq l \leq k$, on définit $G_l = \{(l-1)m + 1, ..., lm\}$ et
   $$\mathbb{X}_l = (x_{j_1}|...|x_{j_m}),$$
   où, $j_i = (l-1)m + i \in G_l$.
2. On calcule ensuite l'estimateur Lasso sur l'ensemble G_l, de la façon suivante :
   $$\hat{\lambda}_\epsilon^l = \underset{\lambda \in \mathbb{R}^D}{\arg\min} \left[\frac{1}{|G_l|} \sum_{j \in G_l} (Y_j - \lambda^T x_j)^2 + \epsilon \|\lambda\|_1 \right]$$
   .
3. Enfin, après avoir calculé tous les estimateurs Lasso pour $1 \leq l \leq k$, on calcule la médiane de toutes les estimations :
   $$\hat{\lambda}_\epsilon^* = med(\hat{\lambda}_\epsilon^1, ..., \hat{\lambda}_\epsilon^k),$$
   où $med()$ désigne la médiane géométrique avec la norme euclidienne sur $\mathbb{R}^D$.

Dans la suite de cette section, nous allons décrire comment nous avons implémenté la stratégie présentée plus haut.
Pour commencer, nous avons construit une classe prenant comme paramètres n, D, t et s, où

— n et D sont la taille de l'espace des variables ;
— t est la quantité permettant de créer les groupes disjoints $G_1, ..., G_k$;
— s est la sparcité de λ_0, cette quantité permet de définir la proportion de valeurs non nulles dans λ_0, qui est $\frac{s}{D}$.

Ensuite, nous avons défini les matrices avec lesquelles nous travaillons :

— La matrice $\mathbb{X}$, de taille (n,D), a été défini à l'aide de la fonction np.random.rand() ;
— le vecteur ϵ, de taille (1,n), a été défini à l'aide de la fonction np.random.rand() ;
— le vecteur Y, de taille (1,n), a été défini par le calcul $Y_j = \lambda_0^T x_j + \epsilon_j$

Ensuite, dans une boucle pour l variant entre 1 et k, nous avons :

1. construit la matrice $\mathbb{X}_l$ et une pseudo matrice $\mathbb{Y}_l$ dans laquelle nous n'avons gardé que les valeurs comprises entre $j1$ et jm ;
2. calculé l'estimateur Lasso en utilisant $\mathbb{X}_l$ et $\mathbb{Y}_l$, dont nous avons stocké les valeurs dans une matrice nommée L.

Enfin, une fois la matrice L obtenue, nous avons effectué la régression quantile de L et Y, afin d'obtenir la médiane géométrique de tous les estimateurs Lasso.

Remarque 3. *La plus grande difficulté dans cette implémentation réside dans la dernière partie. En effet, il a été difficile de savoir comment utiliser la régression quantile pour obtenir la médiane géométrique sur L. Et nous ne sommes toujours pas sûrs de ce choix.*

2.2 Comparaisons des méthodes d'estimations

Afin de connaître la performance de notre estimateur, nous avons pu calculer son R^2 et le comparer à celui d'estimateurs classiques. Nous avons choisi de comparer, avec la mesure du R^2, notre estimateur avec :

- L'estimateur Lasso ;
- L'estimateur Ridge ;
- L'estimateur Elastic-Net ;
- La régression linéaire.

Remarque 4. *Cette comparaison est à prendre avec des pincettes, car certaines valeurs du R^2 peuvent être égales à 1, en raison d'un surapprentissage sur les données.*

Pour aller plus loin, nous pouvons comparer les estimateurs à l'aide d'un histogramme de l'évolution de la valeur de l'erreur grâce à l'utilisation de la validation croisée, comme présenté dans la ressource [0].

Conclusion

La mise en place d'un *estimateur Lasso médian* permet de réduire l'erreur en cas de données sparses, sans avoir à changer le problème ou à faire de supposition sur la distribution du vecteur de bruit. L'implémentation de cet estimateur et le calcul de son R^2, nous a permis de la comparer à d'autres estimateurs naturels, comme les estimateurs Lasso, Ridge, etc.

Bibliographie

[0] *Geometric median and robust estimation in Banach spaces*, Stanislav Minsker, 2015, `https://arxiv.org/pdf/1308.1334.pdf`.
[1] *Régression Quantile*, Joseph Salmon, 2019, `http://josephsalmon.eu/enseignement/Montpellier/HMMA307/RegressionQuantile.pdf`