



Conformal prediction II

Rémi Vaucher

Doctorant, et prof à ses heures perdues
Laboratoire ERIC UR 3083

Un certain jour de l'année





Contents

- 1 Prédiction conforme, premier contact: split conformal prediction
 - Un brin de cadre
 - L'algorithme en "gros"
 - Une idée de pourquoi la loi des erreurs n'intervient plus
 - Exemple simple : une régression linéaire
- 2 Prediction conforme, deuxième contact : le cas des régressions
 - Split conformal prediction
 - Conformal Quantile Regression
- 3 Le cas des algorithmes de classification
- 4 Full Conformal Prediction





Le contexte

Considérons que nous tentons de créer un algorithme de prédiction à partir d'un couple de variables $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$.

Pour cela, on dispose d'un jeu de données: $(X_i, Y_i), 1 \leq i \leq n$.
Nous allons les séparer en 3 sets:

Train

Calibration

Test

Le problème principal est : peut on prédire **avec confiance**, un label Y_{n+1} pour un **nouveau** point X_{n+1} .





Contexte

Comment y arriver : En se donnant un niveau de confiance $1 - \alpha$ (ou un niveau de rejet α), construisons un ensemble de prédiction $\mathcal{C}_\alpha(X_{n+1})$ vérifiant:

$$\mathbb{P} [Y_{n+1} \in \mathcal{C}_\alpha(X_{n+1})] \geq 1 - \alpha \quad (1)$$

La première chose qui saute aux yeux, c'est que l'ensemble de prédiction est dépendant de X_{n+1} , et donc d'une nouvelle donnée.





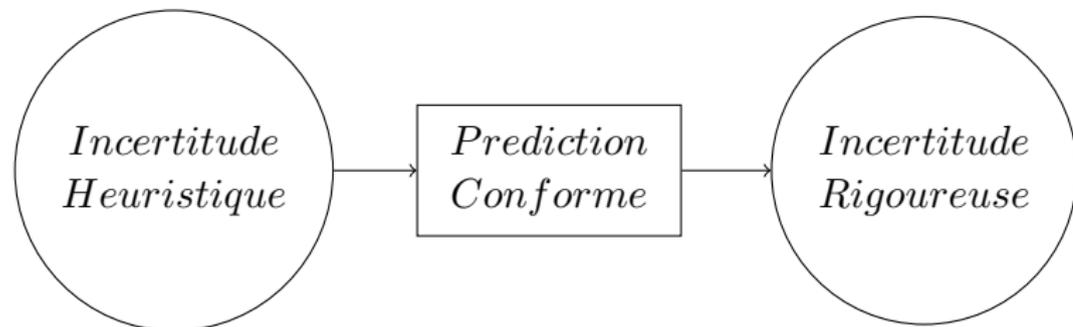
Contexte

On va imposer 4 "restrictions" pour la construction de ces ensembles:

- 1 $C_\alpha(X)$ doit être le "plus petit" possible.
- 2 Indépendance au modèle.
- 3 Indépendance à la loi des données.
- 4 Valide avec un nombre de données fini.

La notion d'incertitude

L'intérêt de cet algorithme est de pouvoir s'appliquer à tout type de modèle de prédiction \hat{f} :



La première étape est donc de comprendre le type d'incertitude à laquelle on a affaire.



La fonction score de conformité

Cette notion est la **plus importante** du principe de prédiction conforme, nous le verrons plus loin.

Nous devons définir un **score de conformité** s . Deux exemples (bien connus) de score de conformités:

- 1 Les écarts au carré : $s(X_i, Y_i) = \|Y_i - \hat{f}(X_i)\|^2$
- 2 Les écarts absolus : $s(X_i, Y_i) = |Y_i - \hat{f}(X_i)|$

La caractéristique principale d'un score de conformité: plus c'est grand, moins la prédiction est précise.





L'ensemble des scores de conformités

Nous allons créer l'ensemble des **scores de calibrations**

$$S = \{s(X_1, Y_1), s(X_2, Y_2), \dots, s(X_n, Y_n)\}.$$

ATTENTION : cet ensemble n'est pas créé grâce aux données d'entraînement, mais grâce aux données de calibrations. On va

calculer \hat{q} comme le quantile d'ordre $\frac{n+1}{n}(1 - \alpha)$ de l'ensemble S .



Les ensembles de prédictions

Armés de ce quantile, nous créons l'ensemble suivant:

$$\mathcal{C}(X_{new}) = \{y | s(X_{new}, y) \leq \hat{q}\}$$



garantie théorique



Définition

On dit que les variables $(X_i, Y_i)_{i=1}^n$ sont **échangeables** si, pour toute permutations σ de $\{1, \dots, n\}$

$$\mathcal{L}((X_1, Y_1), \dots, (X_n, Y_n)) = \mathcal{L}((X_{\sigma(1)}, Y_{\sigma(1)}), \dots, (X_{\sigma(n)}, Y_{\sigma(n)}))$$

Où \mathcal{L} est la loi jointe.

Un échantillon de variables aléatoire (continu ou discret) est automatiquement échangeable (par définition). Des relevés de série temporelle ne le sont pas.





garanties théoriques



Théorème



On considère que $(X_1, Y_1), \dots, (X_n, Y_n), (X_{new}, Y_{new})$ sont échangeable. Alors:

$$\mathbb{P}[Y_{new} \in \mathcal{C}(X_{new})] \geq 1 - \alpha$$



De l'importance de la fonction score.

L'implication de la fonction score dans la création de $\mathcal{C}(X)$ prouve bien l'importance de cette fonction. En fait, à la condition que la fonction score s **ordonne correctement** les erreurs (en terme de magnitude), alors l'ensemble de prédiction s'adaptera bien : il sera petit pour les prédictions faciles, et plus grand pour les prédictions plus difficiles.



De l'importance de l'échangeabilité

Prenons des données $\{(X_1, Y_1), \dots, (X_n, Y_n), (X_{n+1}, Y_{n+1})\}$ i.i.d. (donc échangeables). Nous n'avons aucune idée de la loi de ces données.

Notre modèle est sensé approcher la loi des données, mais il n'y arrivera pas avec exactitude, ce qui entraine que nous ne pouvons pas connaitre avec certitude la loi des erreurs.

De la même manière, nous ne pouvons pas connaitre la loi des scores de conformités $s(X_i, Y_i)$ (qui représentent dans les cas les plus simples nos erreurs).





De l'importance de l'échangeabilité

Par contre, la fonction score est une fonction:

$$s : \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \rightarrow \mathbb{R}$$

$$(X_i, Y_i) \mapsto s(X_i, Y_i)$$

Nous ne pouvons certes pas connaître la loi de la variable $s_i = s(X_i, Y_i)$, mais nous savons comment se comporte la loi des **rangs**:

$$r_i = \mathbf{rang}(s_i) \sim \mathcal{U}(\llbracket 1; n + 1 \rrbracket).$$

Partant de là, la création des quantiles devient simple et automatique.





On bascule sur R!





Premiers algorithmes

Les deux modèles suivants s'appliquent uniquement dans les cas de régressions. Toutefois, les deux algorithmes ne s'appliquent pas du tout dans les mêmes conditions!





Algorithme 1

- 1 Séparer les données d'entraînement en deux : **entraînement** et **calibration**.
- 2 Entraîner un modèle \hat{f} sur les données d'**entraînement**.
- 3 Calculer les scores de conformités:

$$S = \{|Y_i - \hat{f}(X_i)|, (X_i, Y_i) \in \text{Calibration}\}$$

- 4 Calculer le quantile d'ordre $\frac{n+1}{n}(1 - \alpha)$ sur S . On le notera $q_{1-\alpha}(S)$.
- 5 Pour une nouvelle donnée X_{n+1} , calculer:

$$\mathcal{C}(X_{n+1}) = \left[\hat{f}(X_{n+1}) - q_{1-\alpha}(X_{n+1}); \hat{f}(X_{n+1}) + q_{1-\alpha}(X_{n+1}) \right]$$





Algorithme 1

Cet algorithme est:

- Simple
- Rapide
- Mais non adaptatif



Conformal Quantile Regression

L'objectif serait maintenant d'avoir une couverture adaptative, et donc **conditionnelle**:

- Non conditionnel: $\mathbb{P} [Y_{n+1} \in \mathcal{C}(X_{n+1})]$
- Conditionnel : $\mathbb{P} [Y_{n+1} \in \mathcal{C}(X_{n+1}) | X_{n+1}]$

Malheureusement, nous perdons les garanties théoriques. Mais nous allons quand même nous en sortir!



Algorithme 2

- 1 Séparer les données d'entraînement en deux : **entraînement** et **calibration**.
- 2 Entraîner **deux** modèles $\widehat{QR}_{1-\alpha/2}$ et $\widehat{QR}_{\alpha/2}$ sur les données d'**entraînement**.

- 3 Calculer les scores de conformités:

$$S = \left\{ \max \left(\widehat{QR}_{\alpha/2} - Y_i, Y_i - \widehat{QR}_{1-\alpha/2} \right), (X_i, Y_i) \in \text{Calibration} \right\}$$

- 4 Calculer le quantile d'ordre $\frac{n+1}{n}(1-\alpha)$ sur S . On le notera $q_{1-\alpha}(S)$.
- 5 Pour une nouvelle donnée X_{n+1} , calculer:

$$\mathcal{C}(X_{n+1}) = \left[\widehat{QR}_{\alpha/2}(X_{n+1}) - q_{1-\alpha}(S); \widehat{QR}_{1-\alpha/2}(X_{n+1}) + q_{1-\alpha}(S) \right]$$



Algorithme 2

Cet algorithme est:

- Simple
- Adaptatif
- Rapide
- Mais ne repose pas sur un modèle de prédiction précis.



Contexte

On considère un ensemble de données $(X_1, Y_1), \dots, (X_n, Y_n)$ dont les labels sont en nombre fini $Y \in \{C_1, \dots, C_k\}$.

On suppose avoir créé un modèle \hat{f} tel que

$$\hat{f}(X) = \{\hat{p}(C_1), \hat{p}(C_2), \dots, \hat{p}(C_k)\}$$

... où les $\hat{p}(c_i)$ représentent les probabilités estimées d'appartenance de X à chaque classe. Un modèle de classification est un modèle de la forme

$$g \circ \hat{f}(X) = \max_C \{p(C_1), \dots, p(C_k)\}$$





Contexte

L'objectif est d'avoir un ensemble, le plus petit possible, regroupant les classes les plus probables pour une entrée X .





Algorithme 1

- 1 Séparer les données d'entraînement en deux : **entraînement** et **calibration**.
- 2 Entraîner un modèle d'estimation de probabilité des classes \hat{f} sur les données d'**entraînement**.
- 3 Calculer les scores de conformités:

$$S = \{s(X_i, Y_i) = 1 - \hat{f}(X_i)_{Y_i}, (X_i, Y_i) \in \text{Calibration}\}$$

- 4 Calculer le quantile d'ordre $1 - \alpha$ sur S . On le notera $q_{1-\alpha}(S)$.
- 5 Pour une nouvelle donnée X_{n+1} , calculer:

$$\mathcal{C}(X_{n+1}) = \{C_i, s(X_{n+1}, C_i) = 1 - \hat{f}(X_i)_{C_i} \leq q_{1-\alpha}(S)\}$$





Exemple 1

On crée un classifieur sur des images de Chiens, Chats et Tigre (les probabilités seront données dans cet ordre).

Les scores de conformités donnent:

Cal_i										
$\hat{p}_{\text{dog}}(X_i)$	0.95	0.90	0.85	0.15	0.15	0.20	0.15	0.15	0.25	0.20
$\hat{p}_{\text{tiger}}(X_i)$	0.02	0.05	0.10	0.60	0.55	0.50	0.45	0.40	0.35	0.45
$\hat{p}_{\text{cat}}(X_i)$	0.03	0.05	0.05	0.25	0.30	0.30	0.40	0.45	0.40	0.35
S_i	0.05	0.1	0.15	0.40	0.45	0.50	0.55	0.55	0.6	0.65



Exemple 1

- Le quantile d'ordre $1 - \alpha$ (pour $\alpha = 0.1$ de S est $q_{1-\alpha}(S) = 0.65$.
- On prédit la classe d'une nouvelle image X_{new} . On obtient:

$$\hat{f}(X_{new}) = \{0.05, 0.35, 0.60\}$$

Alors:

- $s(X_{new}, \text{chien}) = 0.95$
 - $s(X_{new}, \text{chat}) = 0.65 \leq q_{1-\alpha}(S)$
 - $s(X_{new}, \text{chien}) = 0.4 \leq q_{1-\alpha}(S)$
- On obtient donc $\mathcal{C}(X_{new}) = \{\text{"chat"}, \text{"tigre"}\}$



Commentaires sur cet algorithme

- Les ensembles de prédictions produits sont de taille très raisonnables.
- Une tendance à grossir les ensembles de prédiction sur les données les plus simples.
- Une tendance à amoindrir les ensembles de prédiction sur les données les plus difficiles.



Algorithme 2

- 1 Séparer les données d'entraînement en deux : **entraînement** et **calibration**.
- 2 Entraîner un modèle d'estimation de probabilité des classes \hat{f} sur les données d'**entraînement**.
- 3 Ordonner les probabilité de chaque classe par ordre **décroissante**:

$$\hat{p}_{\sigma_i(C_1)}(X_i) \geq \hat{p}_{\sigma_i(C_2)}(X_i) \geq \dots \geq \hat{p}_{\sigma_i(C_k)}(X_i)$$





Algorithme 2

- Calculer les scores de conformités de la sorte:

$$s(X_i, Y_i) = \sum_{j=1}^{\sigma_i(Y_i)} \hat{p}_{\sigma_i(C_j)}(X_i)$$

ou sous forme algorithmique:

```

S ← 0
j ← 1
while C ≠ Yi do
    C ← Cj
    S = S +  $\hat{p}_{\sigma_i}(C)$ 
    j ← j + 1
end while
  
```





Algorithme 2

- 5 Calculer le quantile d'ordre $1 - \alpha$ sur S . On le notera $q_{1-\alpha}(S)$.
- 6 Pour une nouvelle donnée X_{n+1} , calculer:

$$\mathcal{C}(X_{n+1}) = \{\sigma_{n+1}(C_i), \quad s1 \leq i \leq r\}$$

où r est donnée par:

$$r = \arg \max_{1 \leq r \leq k} \left\{ \sum_{i=1}^r \hat{p}_{\sigma_{n+1}}(X_{n+1}) > 1 - \alpha \right\} + 1$$



Exemple 2

On se remet dans le cas du classifieur Chien/Chat/Tigre, avec un niveau $\alpha = 0.1$.

Les scores de conformités sur le set de calibration donnent :

Cal_i										
$\hat{p}_{\text{dog}}(X_i)$	0.95	0.90	0.85	0.05	0.05	0.05	0.10	0.25	0.10	0.15
$\hat{p}_{\text{tiger}}(X_i)$	0.02	0.05	0.10	0.85	0.80	0.75	0.75	0.40	0.30	0.30
$\hat{p}_{\text{cat}}(X_i)$	0.03	0.05	0.05	0.10	0.15	0.20	0.15	0.35	0.60	0.55
S_i	0.95	0.90	0.85	0.85	0.80	0.75	0.75	0.75	0.60	0.55



Exemple 2

Quand on calcule le quantile d'ordre $1 - \alpha$, on trouve

$$q_{1-\alpha}(S) = 0.95$$

On amène une nouvelle donnée dans le classifieur, et nous trouvons:

$$\hat{f}(X_{new}) = \{0.05, 0.45, 0.5\}$$

Réordonné de manière décroissante, on trouve:

$$\hat{f}(X_{new}) = \{\hat{p}(\text{"Tigre"}) = 0.5, \hat{p}(\text{"Chat"}) = 0.45, \hat{p}(\text{"Chien"}) = 0.05\}$$



Exemple 2

On somme petit à petit les probabilités jusqu'à dépasser **au moins** $q_{1-\alpha}(S)$. Nous obtenons:

$$\mathcal{C}(X_{new}) = \{ "Tigre", "Chat" \}$$



Conclusion pour la SCP

- Méthode simple et peu coûteuse qui
 - Quantifie l'incertitude d'un modèle.
 - Retourne un ensemble de prédiction
- S'adapte à **tout type** de modèles (statistiques, réseaux de neurones, random forest, ...)
- Insensible aux distributions **tant que les données sont échangeables** (exclus les time-series).
- Garanties théorique **en données finies**.
- Si la couverture **n'est pas conditionnelle**, garanties théoriques. Par contre, aucunes garanties théoriques pour les couvertures conditionnelles.





Les problèmes à résoudre

- 1 Avoir une couverture conditionnelle (traitée avec la Conformal Quantile Regression)
- 2 S'affranchir de l'échangeabilité (pour les time series).
- 3 Elucider le problème : **coût computationnel** vs. **puissance statistique**.





coût computationnel vs. puissance statistique.

En effet, nous avons étudié jusque là l'algorithme de Split Conformal Prediction, consistant à prendre des données de calibrations **en réduisant** la base d'entraînement:

- On réduit la base d'entraînement : le modèle peut en pâtir (surtout avec un split aléatoire non biaisé).
- On obtient une base de calibration qui n'est pas tant fournie que cela.

Comment faire pour contourner ces problèmes?





Full (transductive) conformal prediction: FCP

La Full Conformal Prediction permet de ne pas avoir à séparer les données d'entraînements : le modèle peut être créé avec un "maximum" de généralisation.

Il y a par contre un (très gros) inconvénient : le coût computationnel explose.

Pour autant, il ne faut pas oublier qu'historiquement, la SCP est arrivée après la FCP.





Idée de l'algorithme

On a entraîné notre modèle sur des données $(X_1, Y_1), \dots, (X_n, Y_n)$, où les $Y_i \in \mathcal{Y}$ (\mathcal{Y} peut être fini ou non).

Si on reçoit une nouvelle donnée X_{n+1} alors, en parcourant **toutes** les possibilités de \mathcal{Y} alors, on atteindra Y_{n+1} .



Algorithme 3

Pour un candidat (X_{n+1}, y) :

- 1 On sélectionne un label $y \in \mathcal{Y}$ et on entraîne le modèle \hat{f}_y sur l'ensemble de données $(X_1, Y_1), \dots, (X_n, Y_n), (X_{n+1}, y)$
- 2 On calcule les scores

$$S_y = \{s_y(X_i, Y_i), 1 \leq i \leq n\} \cup \{s_y(X_{n+1}, y)\}$$

avec $s_y(X, Y) = s(\hat{f}_y(X), Y)$

- 3 $y \in \mathcal{C}(X_{n+1})$ si $s_y(X_{n+1}, y) \leq q_{1-\alpha}(S)$



Un problème... majeur!

Le coût computationnel de cette méthode est juste
MONSTRUEUX:

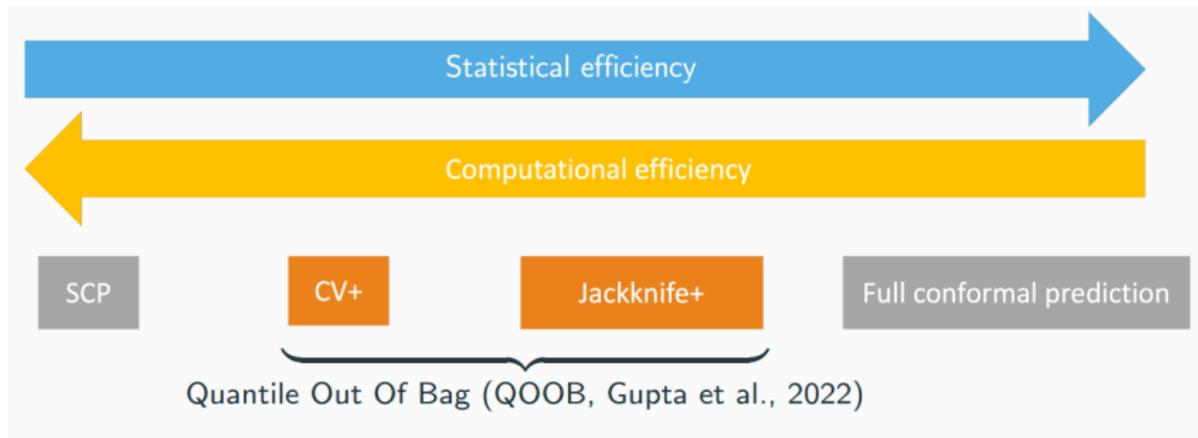
- On itère sur tous les y possibles: en classification basique, ça passe. Mais sur la plupart des modèles, c'est juste impensable.
- D'autant plus que l'on a parlé d'une **unique** données X_{n+1} . Pour être adaptatif, il faudrait parcourir plusieurs X_{n+1}





Les différentes méthodes de conformal prediction

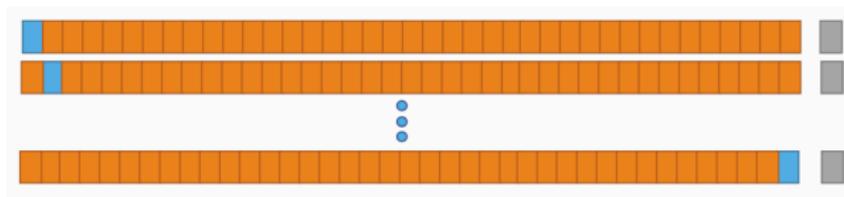
Une image (qui n'est pas de moi, j'avoue) qui explique bien le principe:





Jackknife+

La méthode du jackknife est basé sur du leave-one-out:





Algorithme Jackknife+ (ne marche que sur de la régression)

- 1 On dispose des données $\mathcal{D} = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ pour entraîner notre modèle ET calibrer la CP. On dispose aussi d'une nouvelle donnée X_{n+1}
- 2 On entraîne le modèle \hat{f}_{-i} sur $\mathcal{D} \setminus (X_i, Y_i)$
- 3 On établit les scores de conformités de la manière suivante:

$$S_{pm} = \{\hat{f}_{-i}(X_{n+1}) \pm |\hat{f}_{-i}(X_i) - Y_i|, 1 \leq i \leq n\}$$

- 4 Alors l'ensemble de prédiction à hauteur de $1 - \alpha$ est:

$$\mathcal{C}_{1-\alpha}(X_{n+1}) = [q_{1-\alpha}(S_-); q_{1-\alpha}(S_+)]$$





Algorithme CV+ (ne marche que sur de la régression)

La méthode est basé sur le principe de cross-validation:





Algorithme CV+ (ne marche que sur de la régression)

- 1 On dispose des données $\mathcal{D} = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ pour entraîner notre modèle ET calibrer la CP. On dispose aussi d'une nouvelle donnée X_{n+1}
- 2 On sépare \mathcal{D} en K sous ensemble F_1, \dots, F_K
- 3 On entraîne le modèle \hat{f}_{-k} sur $\mathcal{D} \setminus F_k$
- 4 On établit les scores de conformités de la manière suivante:

$$S_{pm} = \left\{ \{f_{-k}(X_{n+1}) \pm |\hat{f}_{-k}(X_i) - Y_i|\}_{(X_i, Y_i) \in F_k}, 1 \leq k \leq K \right\}$$

- 5 Alors l'ensemble de prédiction à hauteur de $1 - \alpha$ est:

$$\mathcal{C}_{1-\alpha}(X_{n+1}) = [q_{1-\alpha}(S_-); q_{1-\alpha}(S_+)]$$





Exercice

Mettre en place les différents algorithmes vu sur le jeu de données Iris de R.

