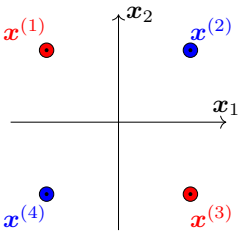
Arbre de Décision & Forêt Aléatoire

L3 FLEX, Machine Learning, Université Paul Sabatier

Romain Thoreau
romain.thoreau@cnes.fr

Mars 2025

Arbres de décision



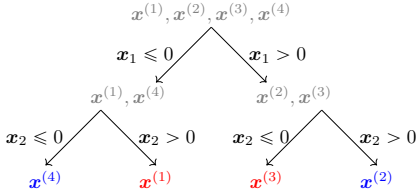


Figure: Illustration d'un arbre de décision pour le problème de classification XOR

Optimisation d'un arbre de décision

- On considère une base d'apprentissage annotée $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)}) | i \in \{1, \dots, N\}\}$ où $\boldsymbol{x}^{(i)} \in \mathbb{R}^D$.
- On représente un arbre de décision par une collection de M nœuds notés Q_m contenant n_m échantillons de la base d'apprentissage. Q_1 contient tous les échantillons de la base.

Optimisation d'un arbre de décision

- On considère une base d'apprentissage annotée $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)}) | i \in \{1, \dots, N\}\}$ où $\boldsymbol{x}^{(i)} \in \mathbb{R}^D$.
- On représente un arbre de décision par une collection de M nœuds notés Q_m contenant n_m échantillons de la base d'apprentissage. Q_1 contient tous les échantillons de la base.
- Chaque nœud a deux "enfants" paramétrés par $\boldsymbol{\theta}_m = \{j_m, t_m\}$:

$$\begin{cases} Q_m^{gauche}(\boldsymbol{\theta}_m) = \{(\boldsymbol{x}, y) \in Q_m | \boldsymbol{x}_{j_m} \leqslant t_m\} \\ Q_m^{droite}(\boldsymbol{\theta}_m) = Q_m \backslash Q_m^{gauche}(\boldsymbol{\theta}_m) \end{cases}$$

Optimisation d'un arbre de décision

- On considère une base d'apprentissage annotée $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)}) | i \in \{1, \dots, N\}\}$ où $\boldsymbol{x}^{(i)} \in \mathbb{R}^D$.
- On représente un arbre de décision par une collection de M nœuds notés Q_m contenant n_m échantillons de la base d'apprentissage. Q_1 contient tous les échantillons de la base.
- Chaque nœud a deux "enfants" paramétrés par $\boldsymbol{\theta}_m = \{j_m, t_m\}$:

$$\begin{cases} Q_m^{gauche}(\boldsymbol{\theta}_m) = \{(\boldsymbol{x}, y) \in Q_m | \boldsymbol{x}_{j_m} \leqslant t_m\} \\ Q_m^{droite}(\boldsymbol{\theta}_m) = Q_m \backslash Q_m^{gauche}(\boldsymbol{\theta}_m) \end{cases}$$

- Les nœuds de l'arbre sont optimisés de manière récursive en minimisant la fonction objective $G(Q_m, \boldsymbol{\theta})$:

$$G(Q_m, \boldsymbol{\theta}) = \frac{n_m^{gauche}}{n_m} H(Q_m^{gauche}(\boldsymbol{\theta}_m)) + \frac{n_m^{droite}}{n_m} H(Q_m^{droite}(\boldsymbol{\theta}_m))$$

où H est un critère de segmentation.

Optimisation d'un arbre de décision : critère de segmentation

▸ Pour un problème de classification, il existe deux critères principaux, fonction de la proportion p_{mc} du nombre d'échantillons appartenant à la classe c dans le nœud m :

- **Critère de Gini**

$$H(Q_m) = \sum_{c=1}^{C} p_{mc}(1 - p_{mc})$$

- **Entropie**

$$H(Q_m) = \sum_{c=1}^{C} p_{mc} \log \frac{1}{p_{mc}}$$

avec la convention $p_{mc} \log p_{mc} = 0$ si $p_{mc} = 0$ puisque $\lim_{p \longrightarrow 0^+} p \log \frac{1}{p} = 0$ [MacKay, 2003].

[1] Voir `https://scikit-learn.org/stable/modules/tree.html#tree-mathematical-formulation` pour d'autres critères

Optimisation d'un arbre de décision : critère de segmentation

▸ Pour un problème de classification, il existe deux critères principaux, fonction de la proportion p_{mc} du nombre d'échantillons appartenant à la classe c dans le nœud m :

- **Critère de Gini**

$$H(Q_m) = \sum_{c=1}^{C} p_{mc}(1 - p_{mc})$$

- **Entropie**

$$H(Q_m) = \sum_{c=1}^{C} p_{mc} \log \frac{1}{p_{mc}}$$

avec la convention $p_{mc} \log p_{mc} = 0$ si $p_{mc} = 0$ puisque $\lim_{p \longrightarrow 0^+} p \log \frac{1}{p} = 0$ [MacKay, 2003].

▸ Pour un problème de régression[1], il est courant d'utiliser l'erreur moyenne au carré par rapport à la moyenne des échantillons $\bar{y}_m = \frac{1}{n_m} \sum_{i \in Q_m} y^{(i)}$:

$$H(Q_m) = \frac{1}{n_m} \sum_{i \in Q_m} (y^{(i)} - \bar{y}_m)^2$$

[1] Voir `https://scikit-learn.org/stable/modules/tree.html#tree-mathematical-formulation` pour d'autres critères

Optimisation du critère de segmentation

▸ Pour un nœud Q_m contenant n_m échantillons, et pour une caractéristique j_m, t_m^* est le seuil qui minimise le critère $G(Q_m, \boldsymbol{\theta})$ parmi les $(n_m - 1)$ seuils possibles :

$$t_m^* = \underset{t_m \in \mathcal{T}_m(j_m)}{\arg\min} \; G(Q_m, t_m, j_m)$$

où :

$$\mathcal{T}_m(j_m) = \left\{ \frac{\boldsymbol{x}_{j_m}^{(1)} + \boldsymbol{x}_{j_m}^{(2)}}{2}, \ldots, \frac{\boldsymbol{x}_{j_m}^{(N-1)} + \boldsymbol{x}_{j_m}^{(N)}}{2} \right\}$$

où l'on suppose que les $x_{j_m}^{(i)}$ sont ordonnés par ordre croissant.

Optimisation du critère de segmentation

▸ Pour un nœud Q_m contenant n_m échantillons, et pour une caractéristique j_m, t_m^* est le seuil qui minimise le critère $G(Q_m, \boldsymbol{\theta})$ parmi les $(n_m - 1)$ seuils possibles :

$$t_m^* = \underset{t_m \in \mathcal{T}_m(j_m)}{\arg\min} \; G(Q_m, t_m, j_m)$$

où :

$$\mathcal{T}_m(j_m) = \left\{ \frac{\boldsymbol{x}_{j_m}^{(1)} + \boldsymbol{x}_{j_m}^{(2)}}{2}, \ldots, \frac{\boldsymbol{x}_{j_m}^{(N-1)} + \boldsymbol{x}_{j_m}^{(N)}}{2} \right\}$$

où l'on suppose que les $x_{j_m}^{(i)}$ sont ordonnés par ordre croissant.

▸ La complexité de l'optimisation d'un nœud, pour j_m fixé, est égale à la complexité du tri, soit $O(n_m \log\, n_m)$.

Optimisation du critère de segmentation

▸ Pour un nœud Q_m contenant n_m échantillons, et pour une caractéristique j_m, t_m^* est le seuil qui minimise le critère $G(Q_m, \boldsymbol{\theta})$ parmi les $(n_m - 1)$ seuils possibles :

$$t_m^* = \underset{t_m \in \mathcal{T}_m(j_m)}{\arg\min} \; G(Q_m, t_m, j_m)$$

où :

$$\mathcal{T}_m(j_m) = \left\{ \frac{\boldsymbol{x}_{j_m}^{(1)} + \boldsymbol{x}_{j_m}^{(2)}}{2}, \ldots, \frac{\boldsymbol{x}_{j_m}^{(N-1)} + \boldsymbol{x}_{j_m}^{(N)}}{2} \right\}$$

où l'on suppose que les $x_{j_m}^{(i)}$ sont ordonnés par ordre croissant.

▸ La complexité de l'optimisation d'un nœud, pour j_m fixé, est égale à la complexité du tri, soit $O(n_m \log\, n_m)$.

▸ Le tri est répété D fois pour chaque caractéristique, ce qui conduit à une complexité en $O(D \cdot N \log\, N)$ pour le premier nœud de l'arbre.

Optimisation du critère de segmentation

▸ Pour un nœud Q_m contenant n_m échantillons, et pour une caractéristique j_m, t_m^* est le seuil qui minimise le critère $G(Q_m, \boldsymbol{\theta})$ parmi les $(n_m - 1)$ seuils possibles :

$$t_m^* = \underset{t_m \in \mathcal{T}_m(j_m)}{\arg\min} \; G(Q_m, t_m, j_m)$$

où :

$$\mathcal{T}_m(j_m) = \left\{ \frac{\boldsymbol{x}_{j_m}^{(1)} + \boldsymbol{x}_{j_m}^{(2)}}{2}, \ldots, \frac{\boldsymbol{x}_{j_m}^{(N-1)} + \boldsymbol{x}_{j_m}^{(N)}}{2} \right\}$$

où l'on suppose que les $x_{j_m}^{(i)}$ sont ordonnés par ordre croissant.

▸ La complexité de l'optimisation d'un nœud, pour j_m fixé, est égale à la complexité du tri, soit $O(n_m \log\, n_m)$.

▸ Le tri est répété D fois pour chaque caractéristique, ce qui conduit à une complexité en $O(D \cdot N \log\, N)$ pour le premier nœud de l'arbre.

▸ En utilisant un algorithme efficace, les implémentations comme celles de scikit-learn trient une seule fois les données, ce qui permet de conserver une complexité en $O(D \cdot N \log\, N)$.

Prédictions d'un arbre de décision

- Les nœuds appartenant au dernier niveau de l'arbre sont appelés les feuilles. Soit on impose une hauteur maximale, soit l'arbre est aussi haut qu'il faut pour que chaque feuille ne contienne qu'un seul échantillon.
- Pour de la classification, la prédiction d'une donnée $\boldsymbol{x}^*$

Prédictions d'un arbre de décision

▸ Les nœuds appartenant au dernier niveau de l'arbre sont appelés les feuilles. Soit on impose une hauteur maximale, soit l'arbre est aussi haut qu'il faut pour que chaque feuille ne contienne qu'un seul échantillon.

▸ Pour de la classification, la prédiction d'une donnée $\boldsymbol{x}^*$ est la classe la plus représentée dans la feuille à laquelle $\boldsymbol{x}^*$ appartient.

Prédictions d'un arbre de décision

- Les nœuds appartenant au dernier niveau de l'arbre sont appelés les feuilles. Soit on impose une hauteur maximale, soit l'arbre est aussi haut qu'il faut pour que chaque feuille ne contienne qu'un seul échantillon.
- Pour de la classification, la prédiction d'une donnée $\boldsymbol{x}^*$ est la classe la plus représentée dans la feuille à laquelle $\boldsymbol{x}^*$ appartient.
- Pour de la régression, la prédiction d'une donnée $\boldsymbol{x}^*$

Prédictions d'un arbre de décision

▸ Les nœuds appartenant au dernier niveau de l'arbre sont appelés les feuilles. Soit on impose une hauteur maximale, soit l'arbre est aussi haut qu'il faut pour que chaque feuille ne contienne qu'un seul échantillon.

▸ Pour de la classification, la prédiction d'une donnée $\boldsymbol{x}^*$ est la classe la plus représentée dans la feuille à laquelle $\boldsymbol{x}^*$ appartient.

▸ Pour de la régression, la prédiction d'une donnée $\boldsymbol{x}^*$ est la moyenne des $y^{(i)}$ de la feuille à laquelle $\boldsymbol{x}^*$ appartient.

Méthodes d'ensemble : principe général

- Idée de la "sagesse des foules" développée, entre autres, par [Galton, 1907] :
  - A un marché, un concours consiste à estimer le poids d'un bœuf,
  - Parmi les 787 participants, l'estimation du meilleur a une erreur supérieure à un centième...
  - Alors que la médiane a une erreur inférieure à un millième.

Biaise et variance d'un ensemble de modèles

▸ On suppose qu'on dispose de K bases d'apprentissage indépendantes $\{\mathcal{D}_1, \ldots, \mathcal{D}_K\}$ à partir desquelles on optimise K modèles $\{f_{\hat{\theta}_1}, \ldots, f_{\hat{\theta}_K}\}$, constituant un ensemble de modèles.

Pour faire une prédiction, l'ensemble de modèles moyenne les prédictions des K modèles :

$$f_{\hat{\theta}}(\boldsymbol{x}) = \frac{1}{K} \sum_{k=1}^{k} f_{\hat{\theta}_k}(\boldsymbol{x})$$

▸ *On suppose que les K modèles ont le même biais et la même variance. Calculer le biais et la variance de l'ensemble de modèles.*

Bagging (bootstrap aggregating)

- En pratique, on ne dispose que d'une seule base d'apprentissage $\mathcal{D} = \{(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(N)}, y^{(N)})\}$!

Bagging (bootstrap aggregating)

- En pratique, on ne dispose que d'une seule base d'apprentissage $\mathcal{D} = \{(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(N)}, y^{(N)})\}$!

Bagging

Le principe du Bagging est de simuler K bases d'apprentissage en échantillonnant M données parmi N avec remise :

$$\begin{cases} \mathcal{D}_1 = & \{(\boldsymbol{x}^{(2)}, y^{(2)}), (\boldsymbol{x}^{(10)}, y^{(10)}) \ldots, (\boldsymbol{x}^{(N)}, y^{(N)})\} \\ \quad \vdots & \\ \mathcal{D}_K = & \{(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}) \ldots, (\boldsymbol{x}^{(N-2)}, y^{(N-2)})\} \end{cases}$$

Bagging (bootstrap aggregating)

▸ En pratique, on ne dispose que d'une seule base d'apprentissage $\mathcal{D} = \{(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(N)}, y^{(N)})\}$!

Bagging

Le principe du Bagging est de simuler K bases d'apprentissage en échantillonnant M données parmi N avec remise :

$$\begin{cases} \mathcal{D}_1 = & \{(\boldsymbol{x}^{(2)}, y^{(2)}), (\boldsymbol{x}^{(10)}, y^{(10)}) \ldots, (\boldsymbol{x}^{(N)}, y^{(N)})\} \\ \quad \vdots & \\ \mathcal{D}_K = & \{(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}) \ldots, (\boldsymbol{x}^{(N-2)}, y^{(N-2)})\} \end{cases}$$

▸ En supposant que la corrélation entre la prédiction de deux modèles $f_{\hat{\theta}_k}(\boldsymbol{x})$, $f_{\hat{\theta}_l}(\boldsymbol{x})$ est égale à ρ, on peut montrer que :

$$\mathsf{Var}\Big(\frac{1}{K}\sum_{k=1}^{K} f_{\hat{\theta}_k}(\boldsymbol{x})\Big) = \frac{1}{K}(1-\rho)\mathsf{Var}(f_{\hat{\theta}_k}(\boldsymbol{x})) + \rho\mathsf{Var}(f_{\hat{\theta}_k}(\boldsymbol{x}))$$

On cherche donc à obtenir des modèles décorrélés les uns des autres !

Forêt aléatoire : recherche de diversité

- La méthode d'ensemble sans doute la plus populaire est la Forêt aléatoire [Breiman, 2001], qui est un ensemble d'arbres de décision.
- *Y a-t-il un lien entre la hauteur d'un arbre et son biais / sa variance ?*

Forêt aléatoire : recherche de diversité

▸ La méthode d'ensemble sans doute la plus populaire est la Forêt aléatoire [Breiman, 2001], qui est un ensemble d'arbres de décision.
▸ *Y a-t-il un lien entre la hauteur d'un arbre et son biais / sa variance ?*

▸ Si l'arbre est complet, il aura un biais faible et une grande variance. Au contraire, si l'arbre est élagué (hauteur limitée), alors il aura un plus grand biais et une variance faible.

Forêt aléatoire : recherche de diversité

▸ La méthode d'ensemble sans doute la plus populaire est la Forêt aléatoire [Breiman, 2001], qui est un ensemble d'arbres de décision.
▸ *Y a-t-il un lien entre la hauteur d'un arbre et son biais / sa variance ?*

▸ Si l'arbre est complet, il aura un biais faible et une grande variance. Au contraire, si l'arbre est élagué (hauteur limitée), alors il aura un plus grand biais et une variance faible.

▸ *Comment obtenir un ensemble d'arbres de décision tel que les prédictions de chaque arbre soient le moins corrélées ?*

Forêt aléatoire : recherche de diversité

▸ La méthode d'ensemble sans doute la plus populaire est la Forêt aléatoire [Breiman, 2001], qui est un ensemble d'arbres de décision.
▸ *Y a-t-il un lien entre la hauteur d'un arbre et son biais / sa variance ?*

▸ Si l'arbre est complet, il aura un biais faible et une grande variance. Au contraire, si l'arbre est élagué (hauteur limitée), alors il aura un plus grand biais et une variance faible.

▸ *Comment obtenir un ensemble d'arbres de décision tel que les prédictions de chaque arbre soient le moins corrélées ?*

▸ Pour chaque arbre et pour chaque nœud, d attributs parmi D sont tirés.


Breiman, L. (2001).
Random forests.
Machine learning, 45:5–32.


Galton, F. (1907).
Vox populi.
Nature.


MacKay, D. J. (2003).
Information theory, inference and learning algorithms.
Cambridge university press.