Gradient Boosting Machine

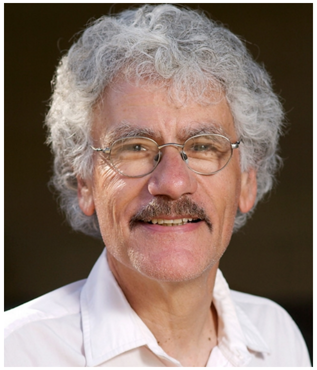
Can a set of weak learners create a single strong learner?

Machine Learning IENAC 2018

Plan

- Introduction
- Boosting et Gradient Boosting
- Arbre de régression et de classification
- Gradient Tree Boosting
- Conclusion

Introduction



- Objectif : Répondre aux questions :
- Qu'est-ce que le Boosting ?
- Qu'est-ce que le Gradient Boosting Machine (GBM) ?
- Un peu d'histoire : Origine du Boosting en 1988, suite à la question de Kearns and Valiant : "Can a set of weak learners create a single strong learner?"
- 1ère proposition d'algorithme en 1990 par Schapire
- Introduction du GBM : Amélioration du Boosting par Jerome Friedman en 1999

sources : http://statweb.stanford.edu/~jhf/

Plan

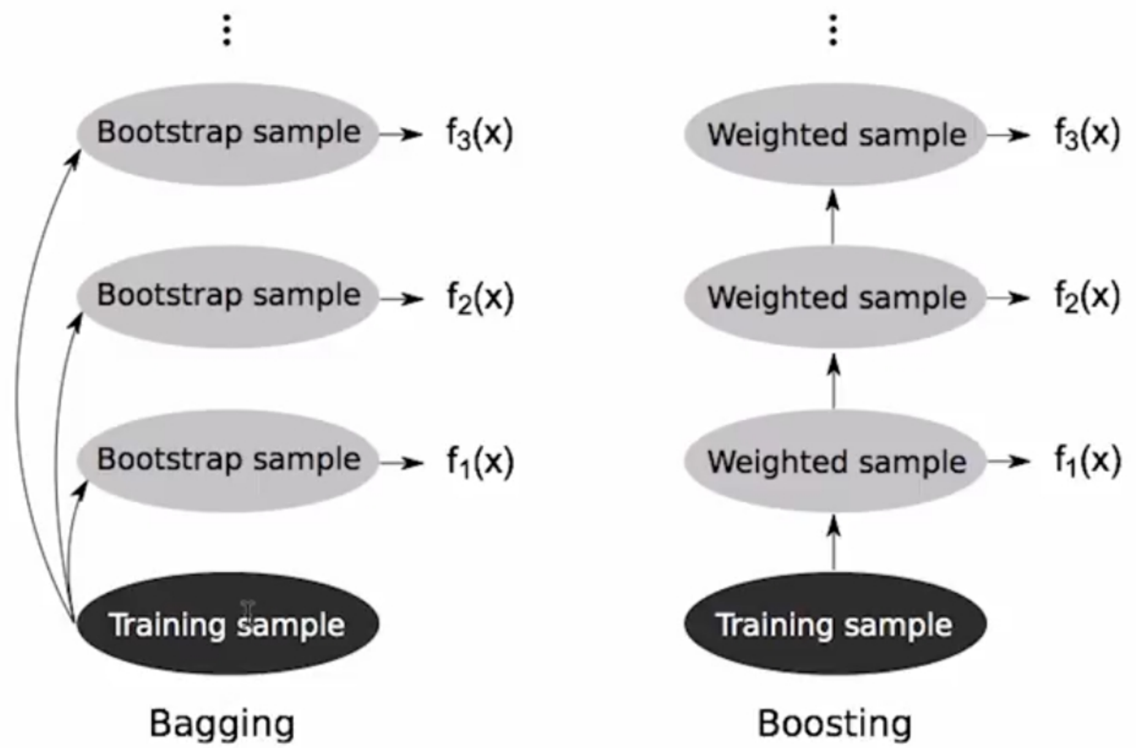
- Introduction
- Boosting et Gradient Boosting
- Arbre de régression et de classification
- Gradient Tree Boosting
- Conclusion

Boosting

- Agrégation de modèles

- élaborés séquentiellement

- différence avec les « random forest » qui en font une utilisation sumultanée.



COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Boosting : AdaBoost

Algorithm: Boosting a binary classifier

Given $(x_1, y_1), \ldots, (x_n, y_n)$, $x \in \mathcal{X}$, $y \in \{-1, +1\}$, set $w_1(i) = \frac{1}{n}$

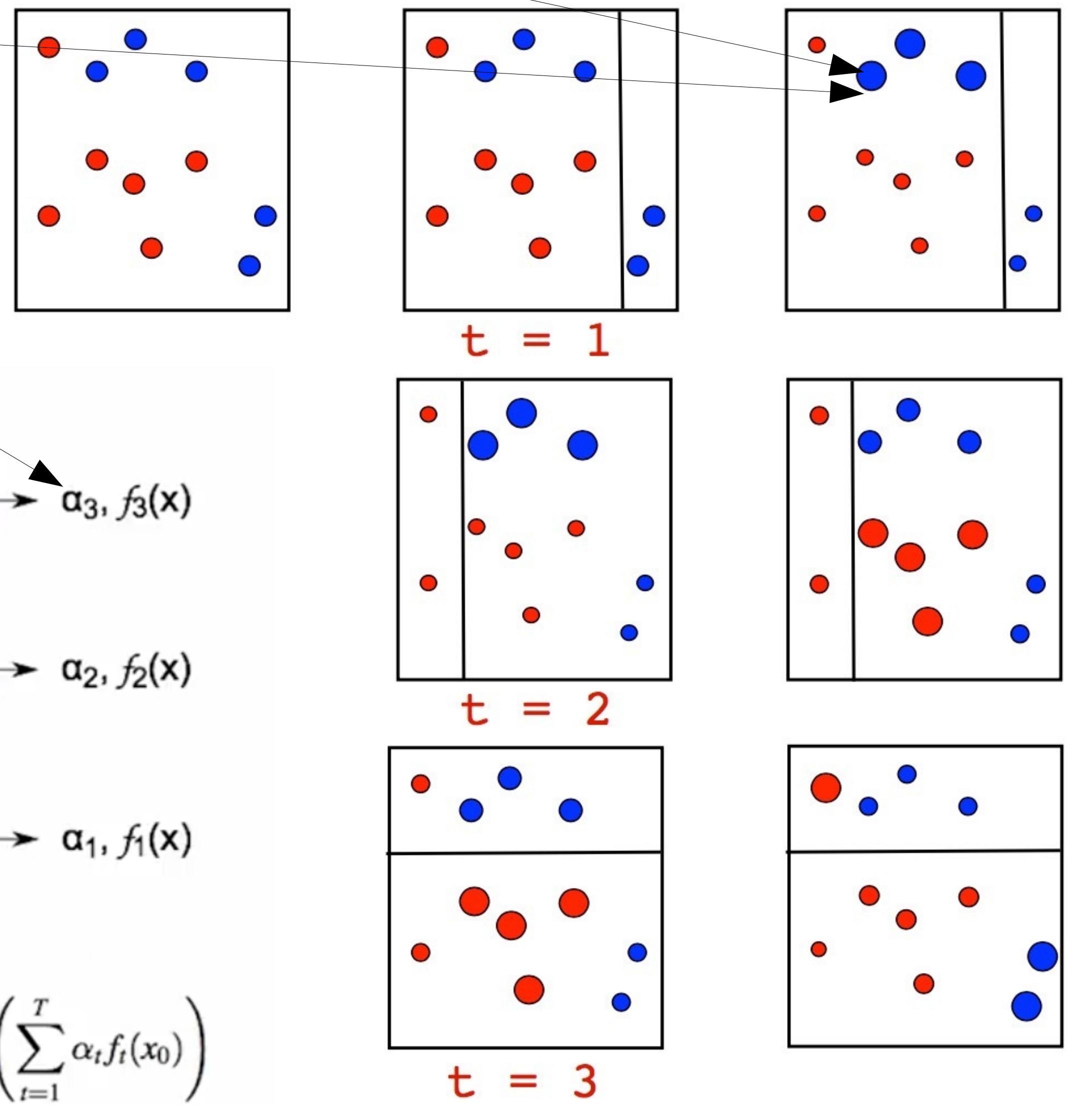
- For $t = 1, \ldots, T$
  1. Sample a bootstrap dataset $\mathcal{B}_t$ of size n according to distribution w_t. Notice we pick (x_i, y_i) with probability $w_t(i)$ and not $\frac{1}{n}$.
  2. Learn a classifier f_t using data in $\mathcal{B}_t$.
  3. Set $\epsilon_t = \sum_{i=1}^{n} w_t(i) \mathbb{1}\{y_i \neq f_t(x_i)\}$ and $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
  4. Scale $\hat{w}_{t+1}(i) = w_t(i) e^{-\alpha_t y_i f_t(x_i)}$ and set $w_{t+1}(i) = \dfrac{\hat{w}_{t+1}(i)}{\sum_j \hat{w}_{t+1}(j)}$.
- Set the classification rule to be

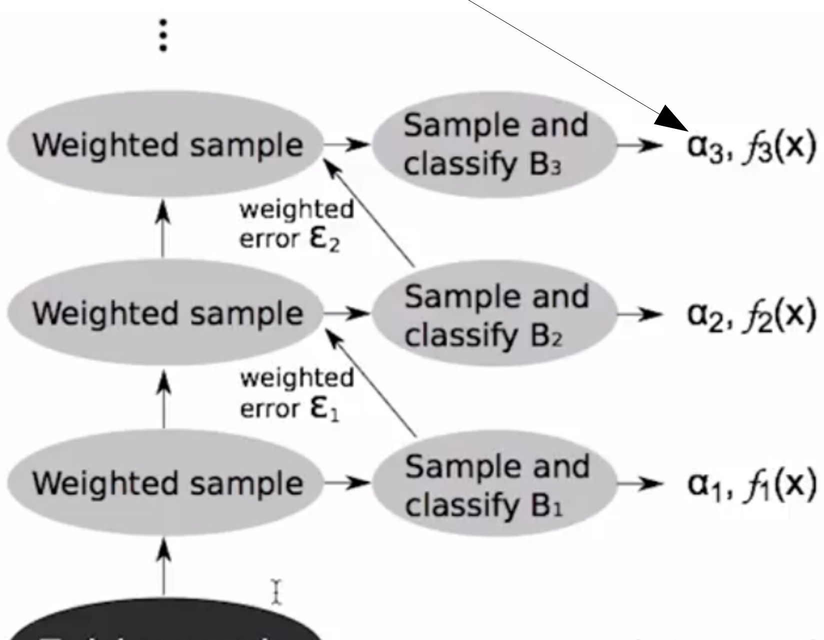
$$f_{boost}(x_0) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t f_t(x_0)\right).$$

Boosting

- Correction des poids w des données à chaque itération. Un poids plus élevé est attribué aux données mal classées par le modèle.

- Pondération des modèles

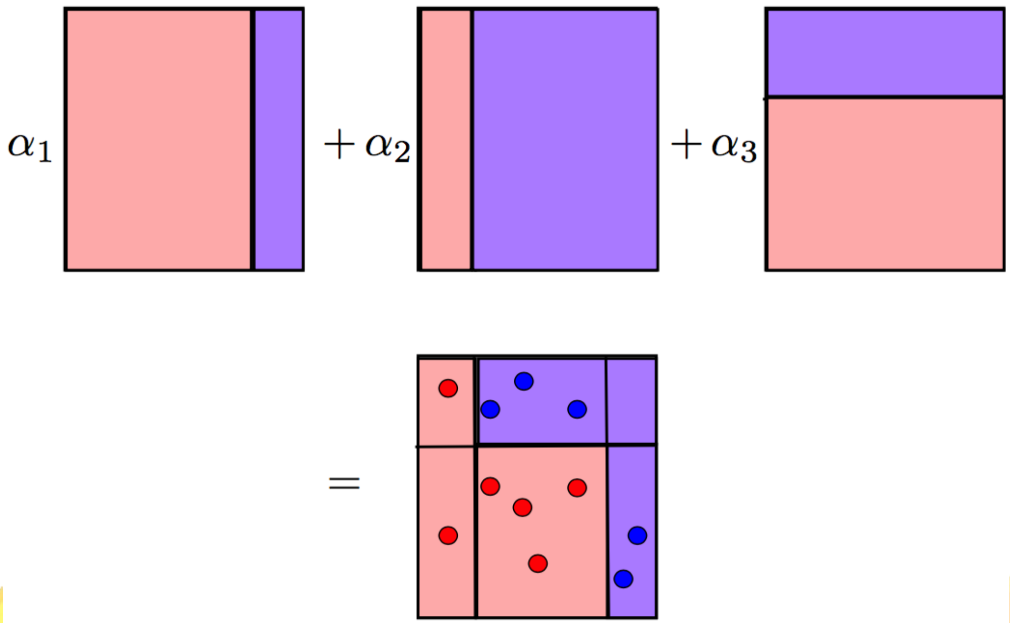




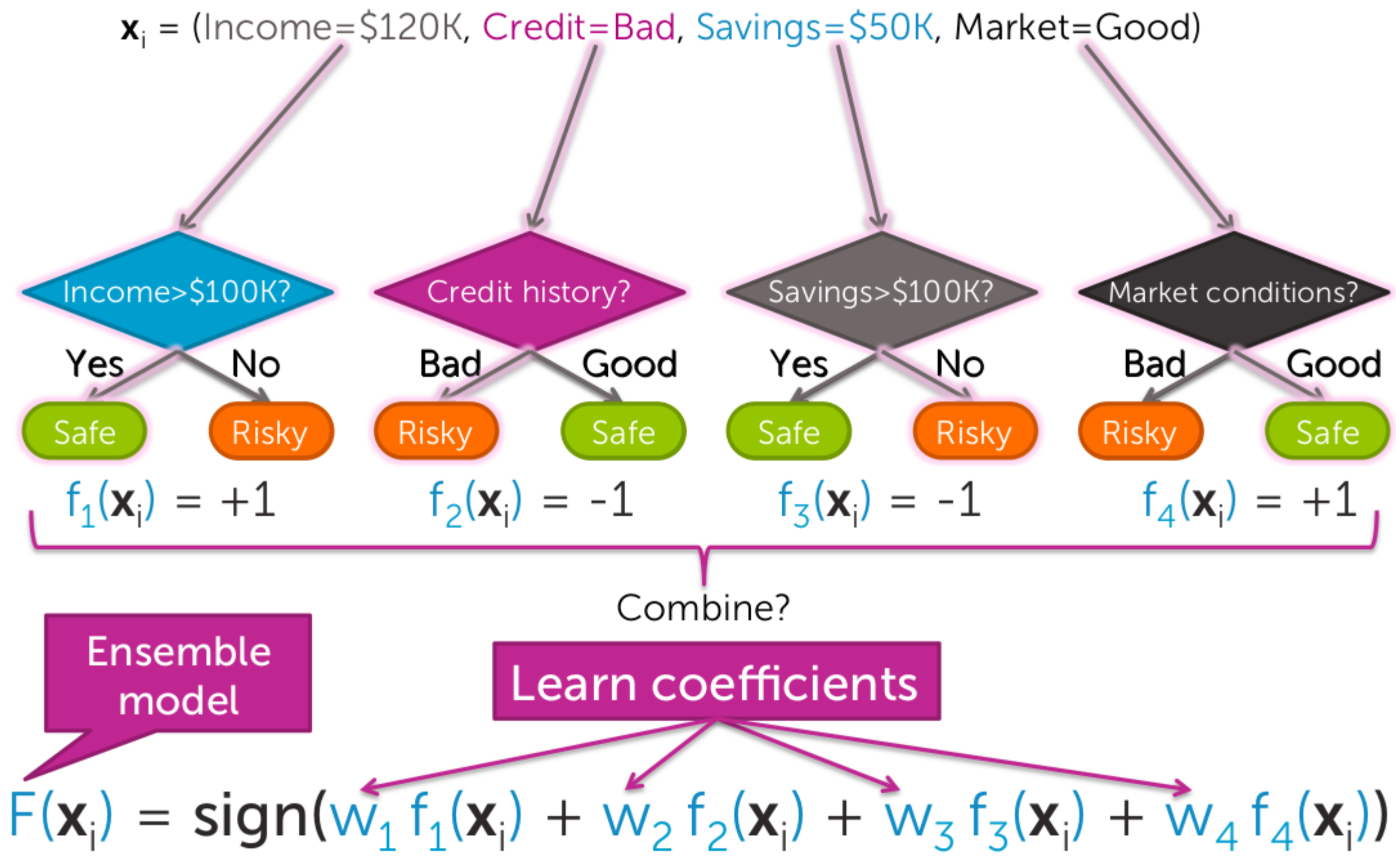
$$f_{boost}(x_0) = \mathrm{sign}\left(\sum_{t=1}^{T} \alpha_t f_t(x_0)\right)$$

Boosting

- chaque modèles peu performant i.e. « weak classifier », ici des « stumps » : arbres binaires de profondeur 1 échouent.
- Mais un vote pondéré des modèles successifs fournit un classification correcte.



BOOSTING



$\mathbf{x}_i$ = (Income=$120K, Credit=Bad, Savings=$50K, Market=Good)
Income>$100K?
Yes
No
Safe
Risky
Credit history?
Bad
Good
Risky
Safe
Savings>$100K?
Yes
No
Safe
Risky
Market conditions?
Bad
Good
Risky
Safe
$f_1(\mathbf{x}_i) = +1$
$f_2(\mathbf{x}_i) = -1$
$f_3(\mathbf{x}_i) = -1$
$f_4(\mathbf{x}_i) = +1$
Combine?
Ensemble model
Learn coefficients
$F(\mathbf{x}_i) = \text{sign}(w_1 f_1(\mathbf{x}_i) + w_2 f_2(\mathbf{x}_i) + w_3 f_3(\mathbf{x}_i) + w_4 f_4(\mathbf{x}_i))$

Boosting

- Algorithme

  1) Initialiser le modèle initial F_0

  2) for m = 1 à M

     – Déterminer les paramètres qui minimisent la fonction de perte

     $$(\beta_m, \mathbf{a}_m) = \arg\min_{\beta,\mathbf{a}} \sum_{i-1}^{N} \Psi(y_i, F_{m-1}(\mathbf{x}_i) + \beta h(\mathbf{x}_i; \mathbf{a}))$$

     – Mettre à jour le modèle

     $$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \beta_m h(\mathbf{x}; \mathbf{a}_m)$$

  3) Renvoyer le modèle final

  $$F(\mathbf{x}) = \sum_{m=0}^{M} \beta_m h(\mathbf{x}; \mathbf{a}_m)$$

Gradient Boosting

- L’algorithme du Gradient Boosting est basé sur celui du Boosting mais le calcul des paramètres optimaux est différent.

- Déterminer la direction de descente optimale à l’aide de la fonction de perte quadratique

- Calcul du pas qui minimise la fonction de perte.

Gradient Boosting

ALGORITHM 1 (Gradient_Boost).

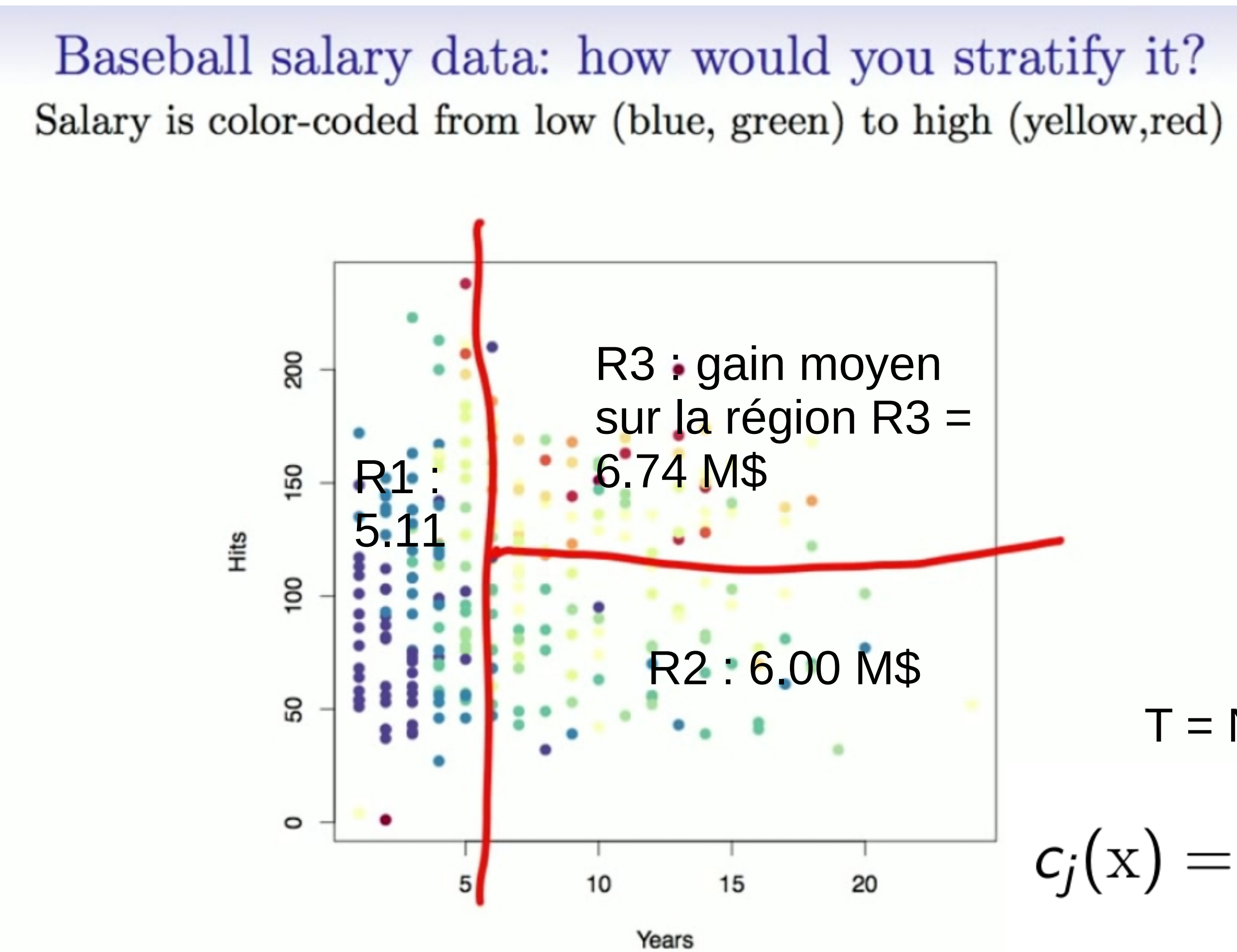
1. $F_0(\mathbf{x}) = \arg\min_\rho \sum_{i=1}^N L(y_i, \rho)$
2. For $m = 1$ to M do:
3. $\tilde{y}_i = -\left[\frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})}, \; i = 1, N$
4. $\mathbf{a}_m = \arg\min_{\mathbf{a},\beta} \sum_{i=1}^N [\tilde{y}_i - \beta h(\mathbf{x}_i; \mathbf{a})]^2$
5. $\rho_m = \arg\min_\rho \sum_{i=1}^N L(y_i, F_{m-1}(\mathbf{x}_i) + \rho h(\mathbf{x}_i; \mathbf{a}_m))$
6. $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \rho_m h(\mathbf{x}; \mathbf{a}_m)$

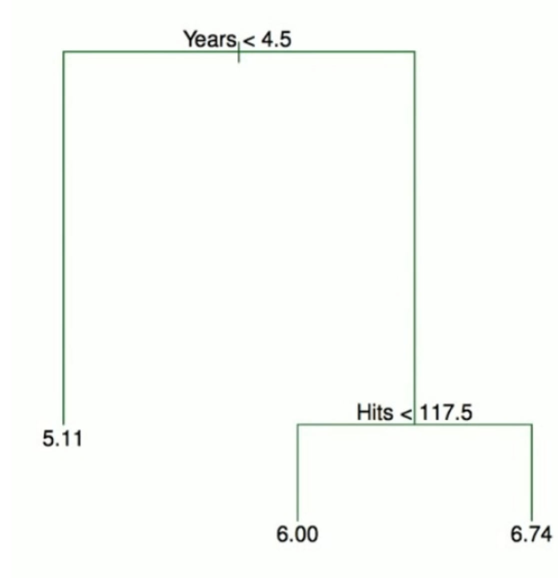
Plan

- Introduction
- Boosting et Gradient Boosting
- Arbre de régression et de classification
- Gradient Tree Boosting
- Conclusion

Arbre de régression et de classification

- Partition de l'espace des prédicteurs en régions





T = Nombre de feuilles = nombre de régions

$$c_j(\mathrm{x}) = \mathbb{E}(\mathrm{y}_n | \mathrm{x}_n \in R_j) = \frac{1}{N_j} \sum_{\mathrm{x}_n \in R_j} \mathrm{y}_n$$

Gradient Tree Boosting

- Application du Gradient Boosting aux arbres de classification et de régression

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \rho_m \sum_{j=1}^{J} b_{jm} 1(\mathbf{x} \in R_{jm})$$

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \sum_{j=1}^{J} \gamma_{jm} 1(\mathbf{x} \in R_{jm})$$

Arbre de régression et de classification

- Modèle de régression : $$h_T(\mathrm{x}) = \sum_{j=1}^{|T|} c_j(\mathrm{x})\, \mathbb{1}_{R_j}(\mathrm{x})$$
-
- un joueur qui a 15 d'expérience en 1ere ligue de baseball et qui a marqué 150 points l'année précédente gagne probablement 6.74 M$/an.

$$R_{j1}(d,s) = \{\mathrm{x} = (x_1, \ldots, x_D) \in R_j \quad | \quad x_d \leq s\}$$
$$R_{j2}(d,s) = \{\mathrm{x} = (x_1, \ldots, x_D) \in R_j \quad | \quad x_d > s\}$$

- Le seuil s et la direction d selon laquelle diviser l'espace des prédicteurs est obtenu en minimisant

$$cost(d,s) = \sum_{\mathrm{x}_n \in R_{j1}(d,s)} \ell\left(\mathrm{y}_n, c_{j1}(\mathrm{x})\right) + \sum_{\mathrm{x}_n \in R_{j2}(d,s)} \ell\left(\mathrm{y}_n, c_{j2}(\mathrm{x})\right)$$

Plan

- Introduction
- Boosting et Gradient Boosting
- Arbre de régression et de classification
- Gradient Tree Boosting
- Conclusion

Gradient Tree Boosting

$F_0(\mathbf{x}) = \arg\min_{\gamma} \sum_{i-1}^{N} \Psi(y_i, \gamma).$

For $m = 1$ to M do:

$$\tilde{y}_{im} = -\left[\frac{\partial \Psi(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x}) = F_{m-1}(\mathbf{x})}, \ i = 1, N$$

$$\{R_{lm}\}_1^L = L - \text{terminal node } \mathit{tree}(\{\tilde{y}_{im}, \mathbf{x}_i\}_1^N)$$

$$\gamma_{lm} = \arg\min_{\gamma} \sum_{\mathbf{x}_i \in R_{lm}} \Psi(y_i, F_{m-1}(\mathbf{x}_i) + \gamma)$$

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu \cdot \gamma_{lm} 1(\mathbf{x} \in R_{lm})$$

endFor.

Plan

- Introduction
- Boosting et Gradient Boosting
- Arbre de régression et de classification
- Gradient Tree Boosting
- Conclusion

Conclusion

- Le boosting *stimule*, i.e. améliore, la justesse d'un « weak model ». Pour ce faire, il agrège séquentiellement des modèles peu performants qui effectuent un « vote » pondéré pour aboutir à un consensus.
- Ces « weak model », qui surperforment à peine le hasard, ne sont pas uniquement des arbres mais peuvent être des NN, etc.
- GBM est une version améliorée utilisant la technique de descente de gradient.
- Le GBM s'applique à la régression et à la classification
- Méthode performante
- Stochastic Gradient Boosting : echantilllonnage sans remise, amélioration significative de la justesse du GBM, gain de temps, limite le sur-apprentissage.
- XGBoost : implémentation efficace du GBM, offre une parallélisation efficace des calculs avec notamment la possibilité d’accéder à la carte graphique (GPU) de l’ordinateur. https://cran.r-project.org/web/packages/xgboost/vignettes/xgboostPresentation.html

Sources

- Stochastic Gradient Boosting (Friedman, 1999),
- Greedy Function Approximation : A Gradient Boosting Machine (Friedman, 1999)
- MOOC Stanford : https://lagunita.stanford.edu/courses/HumanitiesSciences/StatLearning/Winter2016/courseware/4cd5971758e84840b24d91c763df6ce8/6ad06d5d9c5740c2ade97b311e501331/
- MOOC Columbia University New York : https://courses.edx.org/courses/course-v1:ColumbiaX+CSMM.102x+3T2018/course/
- https://medium.com/open-machine-learning-course/open-machine-learning-course-topic-10-gradient-boosting-c751538131ac
- Cours ML IENAC
- The Elements of Statistical Learning (Trevor Hastie, Robert Tibshirani, Jerome Friedman)
- http://wikistat.fr/pdf/st-m-app-agreg.pdf
- https://www.math.univ-toulouse.fr/~besse/Wikistat/