Decision Trees and Ensemble Learning

Wenting Tu

SHUFE, SIME

Machine Learning and Deep Lerning

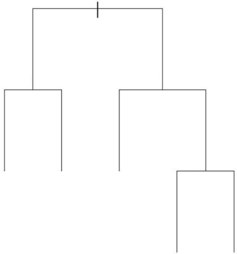
Course No. 1638

Outline

What is a Decision Tree

- Definition

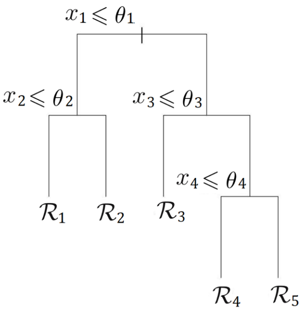
  A classical decision tree classifies data cases using a conjunction of rules organized into a binary tree structure.



What is a Decision Tree

- Definition

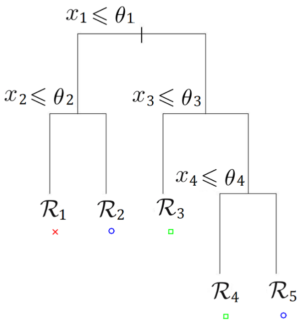
  Each internal node in a classical decision tree contains a rule of the form $x_d \leq \theta$ or $x_d = \theta$ that tests a single data dimension d against a single value θ and assigns the data case to it's left or right sub-tree according to the result.



What is a Decision Tree

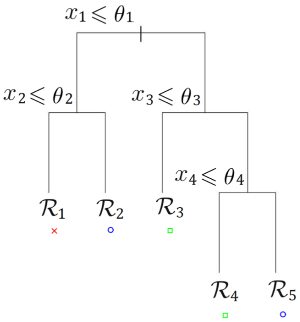
- Definition

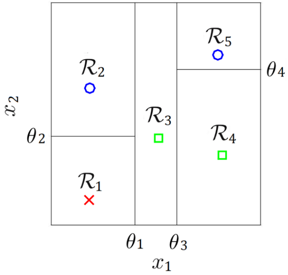
  A data case is routed through the tree from the root to a leaf. Each leaf node is associated with a class, and a data case is assigned the class of the leaf node it is routed to.



What is a Decision Tree

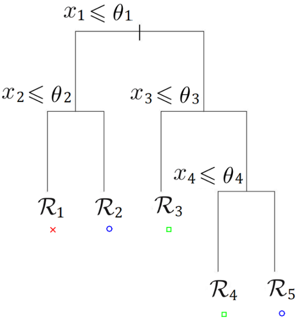
- Partitioning of the Input Space

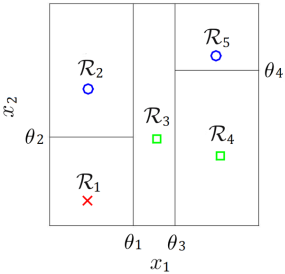




How to Learn a Decision Tree

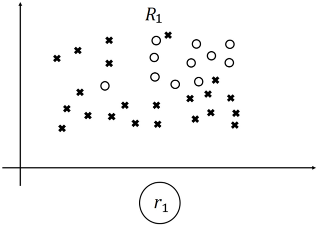
- Typically, Decision trees are learned using recursive greedy algorithms that select the variable and threshold at each node from top to bottom.





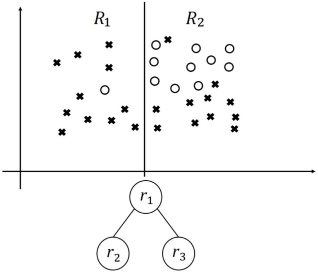
How to Learn a Decision Tree

- The learning algorithm begins with all data cases at the root of the tree.



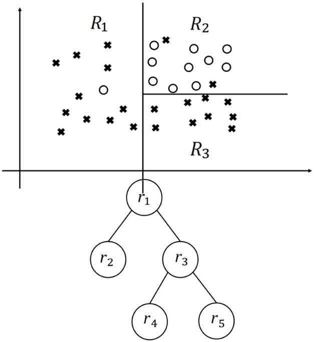
How to Learn a Decision Tree

- The algorithm selects a variable and a threshold to split on according to a heuristic.
- The algorithm applies the chosen rule and assigns each data case to the left or right sub-tree.



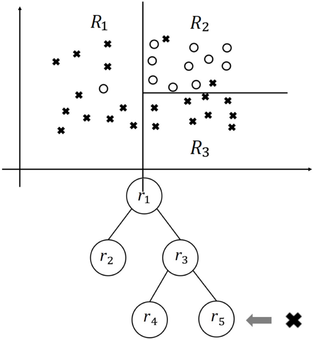
How to Learn a Decision Tree

- The algorithm then recurses on the child nodes until a given stopping condition is satisfied.



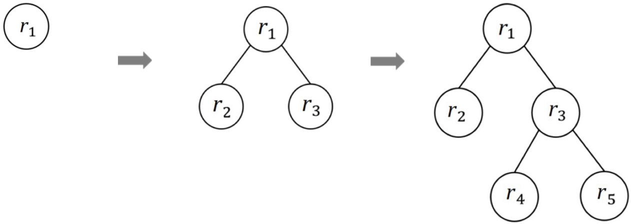
How to Learn a Decision Tree

- The algorithm then recurses on the child nodes until a given stopping condition is satisfied.
- When the stopping condition is satisfied, the current node is a leaf in the tree. It is typically assigned a label that corresponds to the most common label of the data cases it contains.



How to Learn a Decision Tree

- Top-down greedy algorithm
  - Start with a single leaf node containing all data
  - Loop through the following steps:
    - Pick the leaf to split that reduces uncertainty the most.
    - Figure out the split rule on one of the dimensions.



Decision Function for Classification Trees

$$y\left(x_n\right) = \sum_{m=1}^{M} c_m \cdot \mathbb{I}\left\{x_{n \in R_m}\right\}$$

or

$$\widehat{p}\left(\mathcal{C}_k | x_n\right) = \sum_{m=1}^{M} p_k^m \cdot \mathbb{I}\left\{x_{n \in R_m}\right\}$$

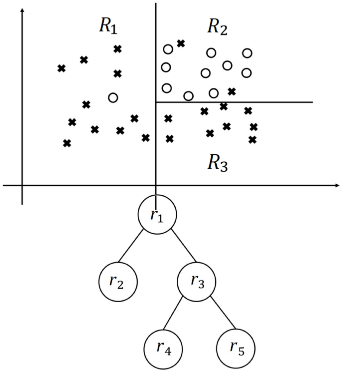
Solution for Classification Trees

$$\{R_1, \ldots, R_M\} \longrightarrow c_m \text{ or } p_k^m?$$

$$c_m = \max_k \sum_{x_n \in R_m} \mathbb{I}\{t_n = k\}$$

$$p(\mathcal{C}_k|\mathbf{x}) = \sum_{m=1}^{M} p_k^m \cdot \mathbb{I}\{\mathbf{x} \in R_m\}$$

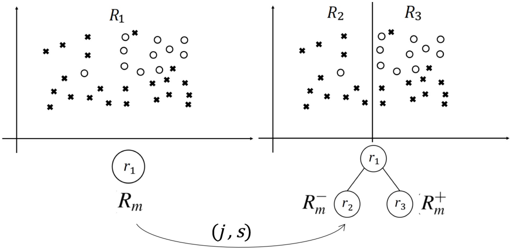
$$p_k^m = \frac{1}{|R_m|} \sum_{x_n \in R_m} \mathbb{I}\{t_n = k\}$$



Solution for Classification Trees

$$p_k^m \longrightarrow \{R_1, \ldots, R_M\}?$$

which (j, s) is best?

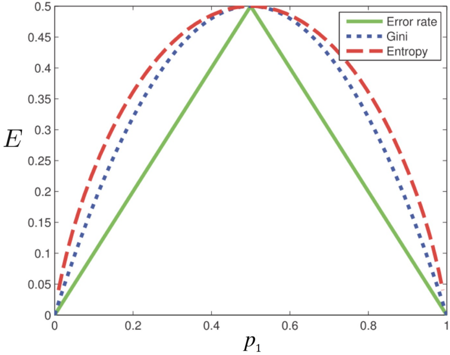


$$R^-(j, s) = \{x_i \in R | x_i(j) \leq s\}$$
$$R^+(j, s) = \{x_i \in R | x_i(j) > s\}$$

Loss Function for Classification Trees

Choices for $E(R_m)$, given $p_k^m = \frac{1}{|R_m|} \sum_{x_n \in R_m} \mathbb{I}\{t_n = k\}$

· Classification error $1 - \max_k p_k^m$
· Entropy $-\sum_k p_k^m \ln p_k^m$
· Gini index $1 - \sum_k (p_k^m)^2$



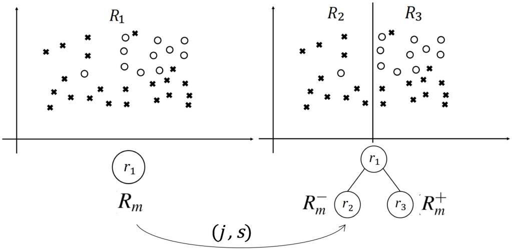
Solution for Classification Trees

$$p_k^m \longrightarrow \{R_1, \ldots, R_M\}?$$

$$p_k^m, p_k^{m+}, p_k^{m-} \longrightarrow E\left(R_m\right), E\left(R_m^+\right), E\left(R_m^-\right)$$

$$L(j,s) = \left(p_{R_m^-} \cdot E\left(R_m^-\right) + p_{R_m^+} \cdot E\left(R_m^+\right)\right) - E\left(R_m\right)$$

$$(j^*, s^*) = \underset{(j,s)}{\arg\min}\, L(j,s)$$

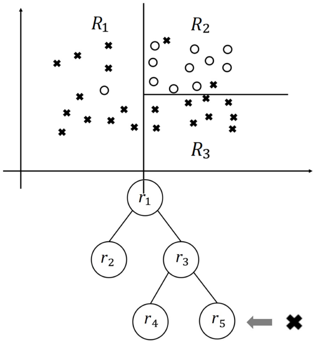


$$R^-(j,s) = \{x_i \in R | x_i(j) \le s\}$$

$$R^+(j,s) = \{x_i \in R | x_i(j) > s\}$$

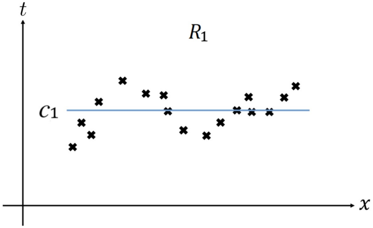
Stopping Criteria

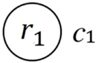
- The main stopping criteria used are all data cases assigned to a node have the same label, the node is at the maximum allowable depth...



Decision Function for Regression Trees

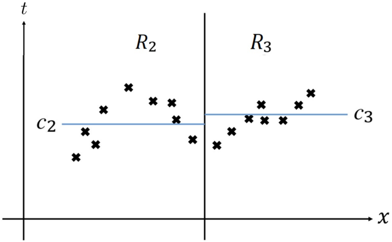
$$y(x_n) = \sum_{m=1}^{M} c_m \cdot \mathbb{I}\{x_{n \in R_m}\}$$

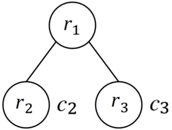




Decision Function for Regression Trees

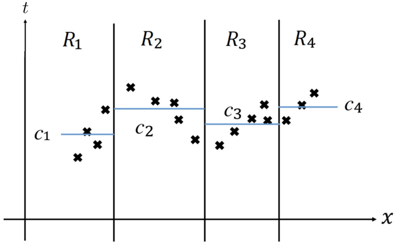
$$y(x_n) = \sum_{m=1}^{M} c_m \cdot \mathbb{I}\{x_{n \in R_m}\}$$

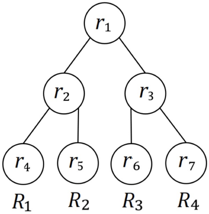




Decision Function for Regression Trees

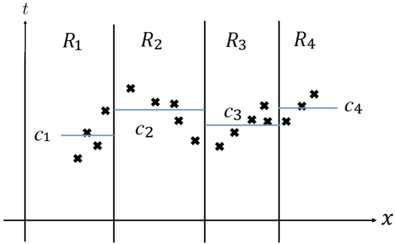
$$y(x_n) = \sum_{m=1}^{M} c_m \cdot \mathbb{I}\{x_{n \in R_m}\}$$

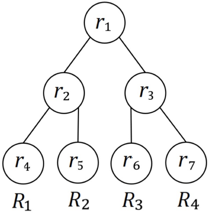




Solution for Regression Trees

$$\{R_1, \ldots, R_M\} \longrightarrow c_m$$





$$\begin{aligned} c_m &= \arg\min_{c_m} \sum_{x_{n \in R_m}} (t_n - c_m)^2 \\ &= \text{ave}\left(t_n | x_{n \in R_m}\right) \end{aligned}$$

Solution for Regression Trees

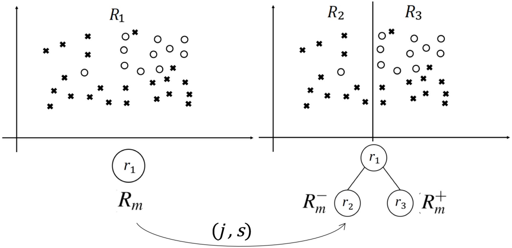
$$c_m \longrightarrow (R_1, \ldots, R_M)$$

$$E(R_m) = \sum_{x_{n \in R_m}} (t_n - c_m)^2 = \sum_{x_{n \in R_m}} (t_n - \text{ave}(t_n \mid x_{n \in R_m}))^2$$

$$c_m, c_{m+}, c_{m-} \longrightarrow E(R_m), E\left(R_m^+\right), E\left(R_m^-\right)$$

$$L(j,s) = \left(p_{R_m^-} \cdot E\left(R_m^-\right) + p_{R_m^+} \cdot E\left(R_m^+\right)\right) - E(R_m)$$

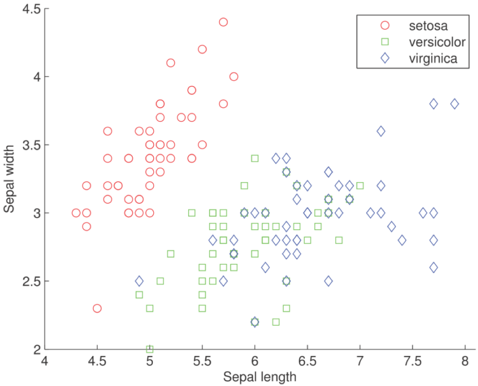
$$(j^*, s^*) = \underset{(j,s)}{\arg\min}\, L(j,s)$$



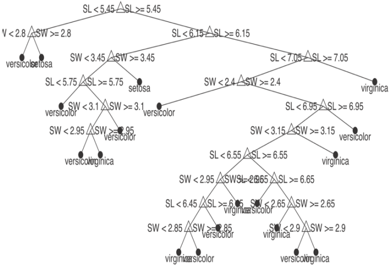
$$R^-(j,s) = \{x_i \in R | x_i(j) \leq s\}$$

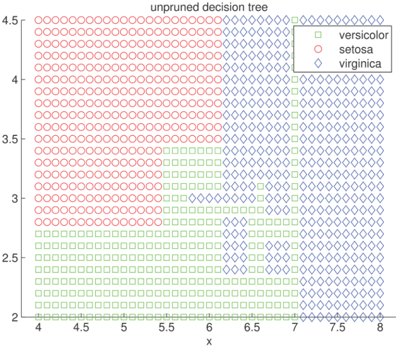
$$R^+(j,s) = \{x_i \in R | x_i(j) > s\}$$

Pruning of Decision trees

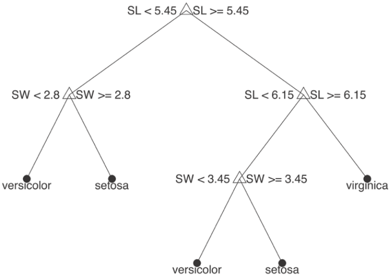


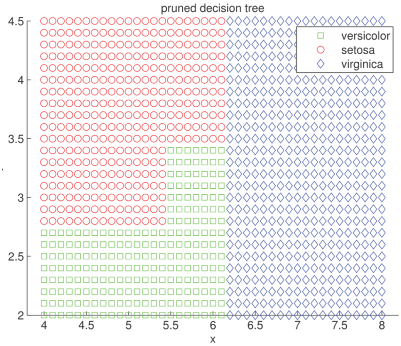
Pruning of Decision trees





Pruning of Decision trees

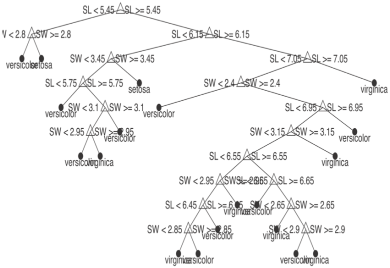


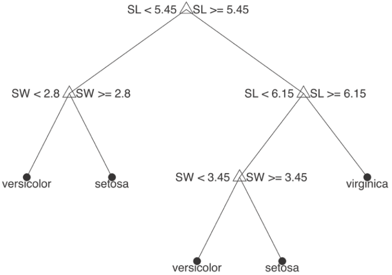


Pruning of Decision trees

- Pre-pruning/Early stopping

  Stop splitting when not statistically significant

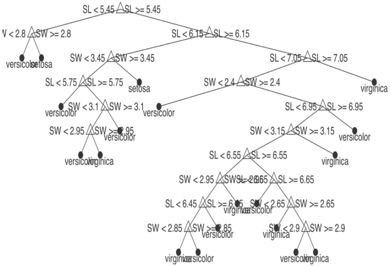


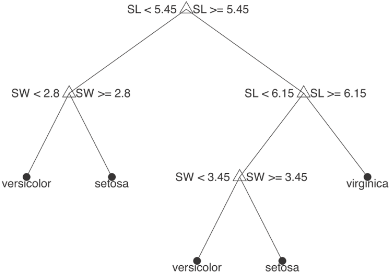


Pruning of Decision trees

- Post-pruning

  Grow, then post-prune



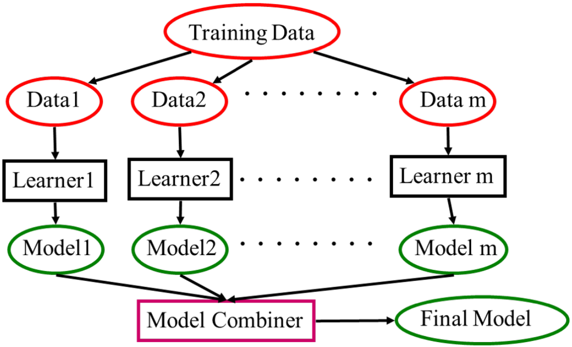


Outline

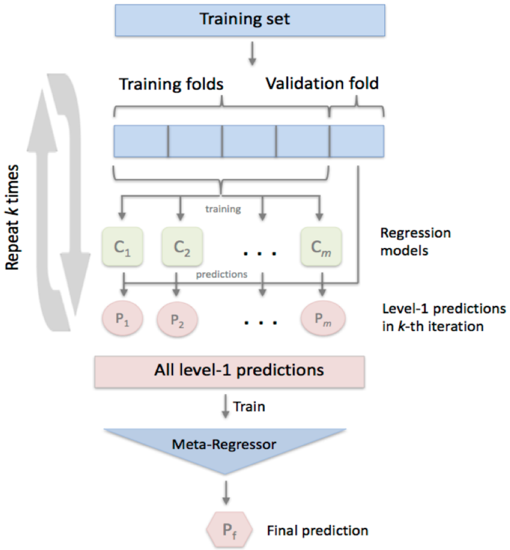
Decision Trees

Ensemble Learning

Ensemble Learning

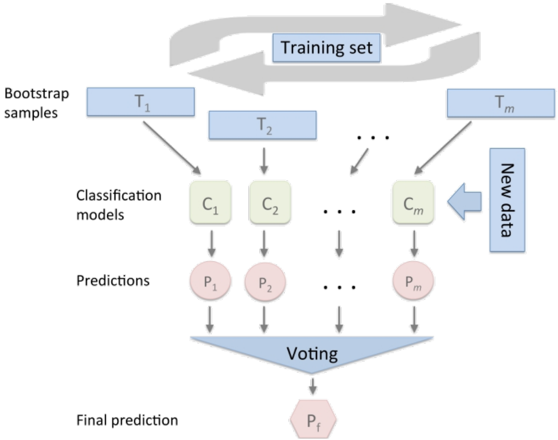


Stacking



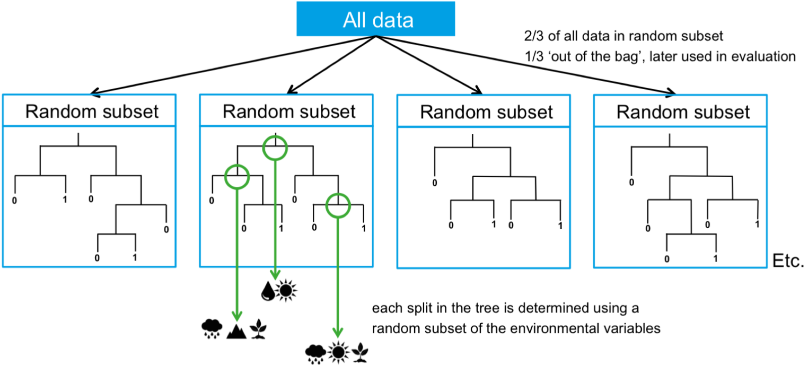
Training set
Training folds
Validation fold
Repeat k times
training
C_1
C_2
. . .
C_m
Regression models
predictions
P_1
P_2
. . .
P_m
Level-1 predictions in k-th iteration
All level-1 predictions
Train
Meta-Regressor
P_f
Final prediction

Bagging

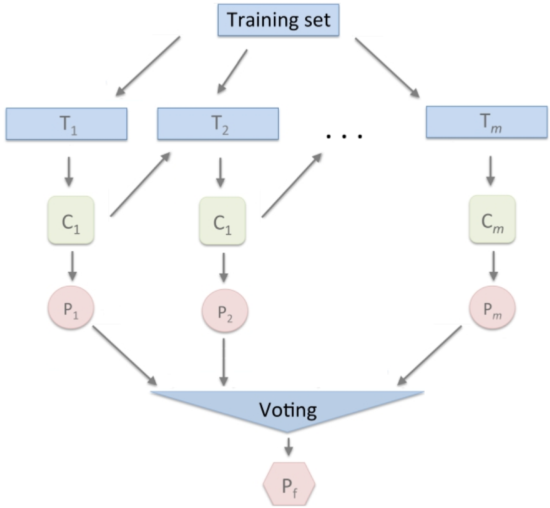


Training set
Bootstrap samples
T_1
T_2
T_m
Classification models
C_1
C_2
C_m
New data
Predictions
P_1
P_2
P_m
Voting
Final prediction
P_f

Random Forest



Boosting

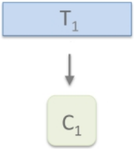


Adaboost

$$Y_M(\mathbf{x}) = \mathrm{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(\mathbf{x})\right)$$

Adaboost

1. Initialize the data weighting coefficients $\{w_n\}$ by setting $w_n^{(1)} = 1/N$ for $n = 1, \ldots, N$

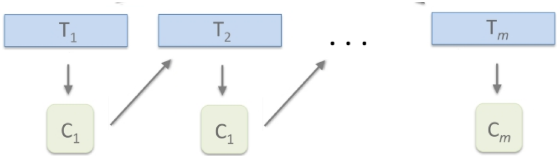


Adaboost

2. For $m = 1, \ldots, M$:
(a) Fit a classifier $y_m(\mathrm{x})$ to the training data by minimizing the weighted error function

$$J_m = \sum_{n=1}^{N} w_n^{(m)} I\left(y_m\left(\mathbf{x}_n\right) \neq t_n\right)$$

where $I\left(y_m\left(\mathbf{x}_n\right) \neq t_n\right)$ is the indicator function and equals 1 when $y_m\left(\mathbf{x}_n\right) \neq t_n$ and 0 otherwise.



Adaboost

2. For $m = 1, \ldots, M$:
(b) Evaluate the quantities

$$\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I\left(y_m\left(\mathbf{x}_n\right) \neq t_n\right)}{\sum_{n=1}^{N} w_n^{(m)}}$$

and then use these to evaluate

$$\alpha_m = \ln\left\{\frac{1-\epsilon_m}{\epsilon_m}\right\}$$

Adaboost

2. For $m = 1, \ldots, M$:

(c) Update the data weighting coefficients

$$w_n^{(m+1)} = w_n^{(m)} \exp\left\{\alpha_m I\left(y_m\left(\mathbf{x}_n\right) \neq t_n\right)\right\}$$

3. Make predictions using the final model, which is given by

$$Y_M(\mathbf{x}) = \operatorname{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(\mathbf{x})\right)$$

Gradient Boosting

- Gradient Boosting for Regression

  $J = \sum_i L\left(t_i, F\left(x_i\right)\right)$

  $L(t, F(x)) = (t - F(x))^2/2$

  $$\frac{\partial J}{\partial F\left(x_i\right)} = \frac{\partial \sum_i L\left(t_i, F\left(x_i\right)\right)}{\partial F\left(x_i\right)} = \frac{\partial L\left(t_i, F\left(x_i\right)\right)}{\partial F\left(x_i\right)} = F\left(x_i\right) - t_i$$

  $$t_i - F\left(x_i\right) = -\frac{\partial J}{\partial F\left(x_i\right)}$$

  $f\left(x_i\right) = t_i - F\left(x_i\right)$

  Update F : $F^{new}\left(x\right) = F^{old}\left(x\right) + f\left(x\right)$

Gradient Boosting

- LightGBM [link]
- Xgboost [link]

Thanks

Some images and slides are from the internet.
If related to copyright, please contact me.

tu.wenting@mail.shufe.edu.cn