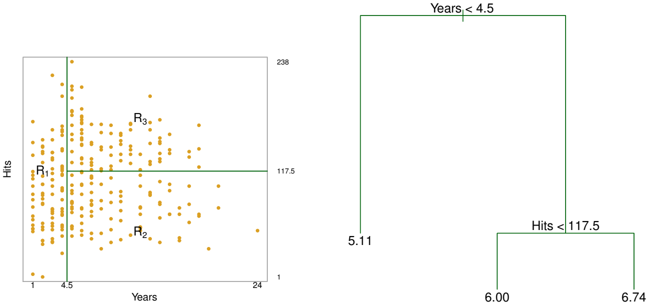
ST720 Data Science

Tree-Based Method

Seung Jun Shin (sjshin@korea.ac.kr)

Department of Statistics, Korea University

Tree Model I



(a) Partitioned Predictor Space (b) Fitted Model

Tree Model II

- A good partition yields similar response values in a same node and different values in deferent nodes.
- Impurity:
  - Regression $\Rightarrow$ $\sum_i (y_i^\ell - \bar{y}^\ell)^2$
  - Classification $\Rightarrow$ $\sum_k p_k^\ell (1 - p_k^\ell)$ or $\sum_k p_k^\ell \log p_k^\ell$
- Sequentially detect the partition that minimizes total impurities.

Tree Model III

- Recursive binary splitting

ex Toy Example: Regression Tree

x	1	2	3	4
y	0	2	10	12

1. $x \leq 1$ vs $x > 1$:
   $(0-0)^2 + (2-8)^2 + (10-8)^2 + (12-8)^2 = 56$
2. $x \leq 2$ vs $x > 2$:
   $(0-1)^2 + (2-1)^2 + (10-11)^2 + (12-11)^2 = 4$
3. $x \leq 3$ vs $x > 3$:
   $(0-4)^2 + (2-4)^2 + (10-4)^2 + (12-12)^2 = 56$

Tree Model IV

ex Toy Example: Binary Classification

x	1	2	3	4
y	1	1	1	2

1. $x \le 1$ vs $x > 1$: $1 \cdot 0 + 0 \cdot 1 + \frac{2}{3} \cdot \frac{1}{3} + \frac{1}{3} \cdot \frac{2}{3} = \frac{2}{3}$
2. $x \le 2$ vs $x > 2$: $1 \cdot 0 + 0 \cdot 1 + \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2}$
3. $x \le 3$ vs $x > 3$: $1 \cdot 0 + 0 \cdot 1 + 0 \cdot 1 + 1 \cdot 0 = 0$

Tree Model V

- Size of tree is a tuning parameter.
- Where to stop?
- Pruning: Fit full tree and then cut.
- Objective function

$$\text{Total Impurity of } T + \lambda |T|$$

- `tree` / `rpart` package available in R

Random Forest I

- Tree is a weak leaner with large variance.
- Reduce the variance by repeated learning.
- Bootstrapping and then Fitting
- Ensemble is a popular idea in ML

"Combine the outputs of may "weak" classifiers to produce a powerful committee!!"

Random Forest II

- From $(\mathbf{y}, \mathbf{X})$, generate bootstrap sample $(\mathbf{y}_b^*, \mathbf{X}_b^*), b = 1, 2, \cdots, B$.
- Fit a tree model from each of $(\mathbf{y}_b^*, \mathbf{X}_b^*)$ denoted by $T_b^*, b = 1, \cdots, B$.
- Prediction Rule from $\{T_1^*, \cdots, T_B^*\}$
  - Regression: Average
  - Classification: Majority Vote
- This is known as Bootstrap aggregating, "Bagging".

Random Forest III

- Random Forest (RF)is an improved version of Bagging.
- When fitting a tree model from each of bootstrap samples, randomly pick a subset of variables when splitting.
- This additional variation increases the variability of T_b^*.
- Bagging is a special case of RF. (Pick p variables for the splitting).
- `randomForest` package available.

Boosting I

1. Initialize the observation weights $w_i = 1/n, i = 1, \cdots, n$.
2. For $m = 1$ to M:
   2.1 Fit a classifier (eg, stump) denoted by $T_m(\mathbf{x})$ using weights w_i.
   2.2 Compute the error rate

$$\text{err}_m = \frac{\sum_{i=1}^n w_i \mathbb{1}\{y_i \neq T_m(\mathbf{x}_i)\}}{\sum_{i=1}^n w_i}$$

   2.3 Compute the contribution of $T_m(\mathbf{x})$ for ensemble:

$$\alpha_m = \log\{(1 - \text{err}_m)/\text{err}_m\}$$

   2.4 Update weight as $w_i \quad \leftarrow \quad w_i \exp[\alpha_m \mathbb{1}\{y_i \neq T_m(\mathbf{x}_i)\}]$.
3. Output

$$f(\mathbf{x}) = \sum_{m=1}^{M} \alpha_m T_m(\mathbf{x})$$

Boosting II

- Boosting can be equivalently rewritten as the ERM problem with $L(u) = \exp(-u)$:

$$\min_f \sum_{i=1}^{n} \exp(-y_i f(\mathbf{x}_i)),$$

where

$$f(\mathbf{x}) = \sum_{m=1}^{M} \alpha_m T_m(\mathbf{x})$$

- M is a tuning parameter and overfits as $M \to 0$.

Boosting III

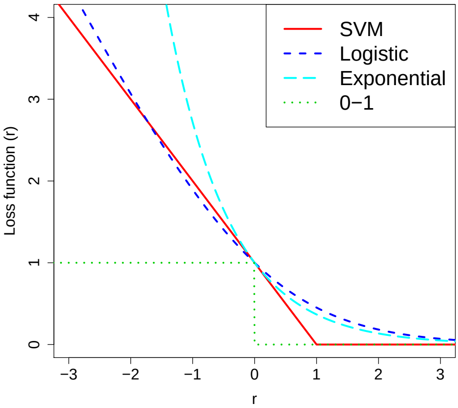


Figure: Exponential loss

Boosting IV

- Extension to other loss function is natural.
  (ex. Logit-Boosting, L_2-Boosting)
- Gradient Decent Algorithm can be applied to the optimization.
  (AdaBoosting algorithm is a Steepest Decent Algrotithm.)
- We call this Gradient Boosting.
- `gbm` / `xgboost` package are available.