Tree Models: Decision Tree, Boosting Tree, GBDT

Richard Liu

School of Mathematics, XMU

July 12, 2020

Content

Source

- Pang Ning Tan, et al. *Introduction to Data Mining*
- Hang Li, *Statistical Learning Methods*

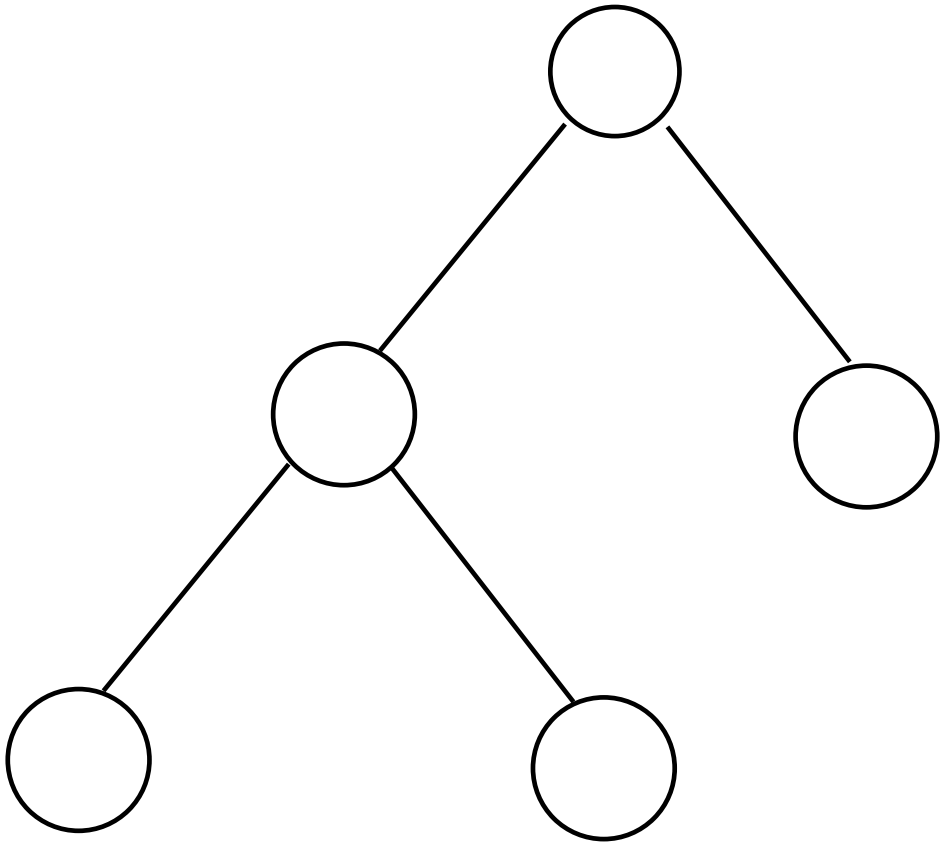
Section 1

Introduction

Introduction

- Tree: A collection of "if-then" rules.
- Readable but impossible to get an optimal result (NP-hard)
- Terminologies: Root nodes, Leaf nodes, Internal nodes, Edges

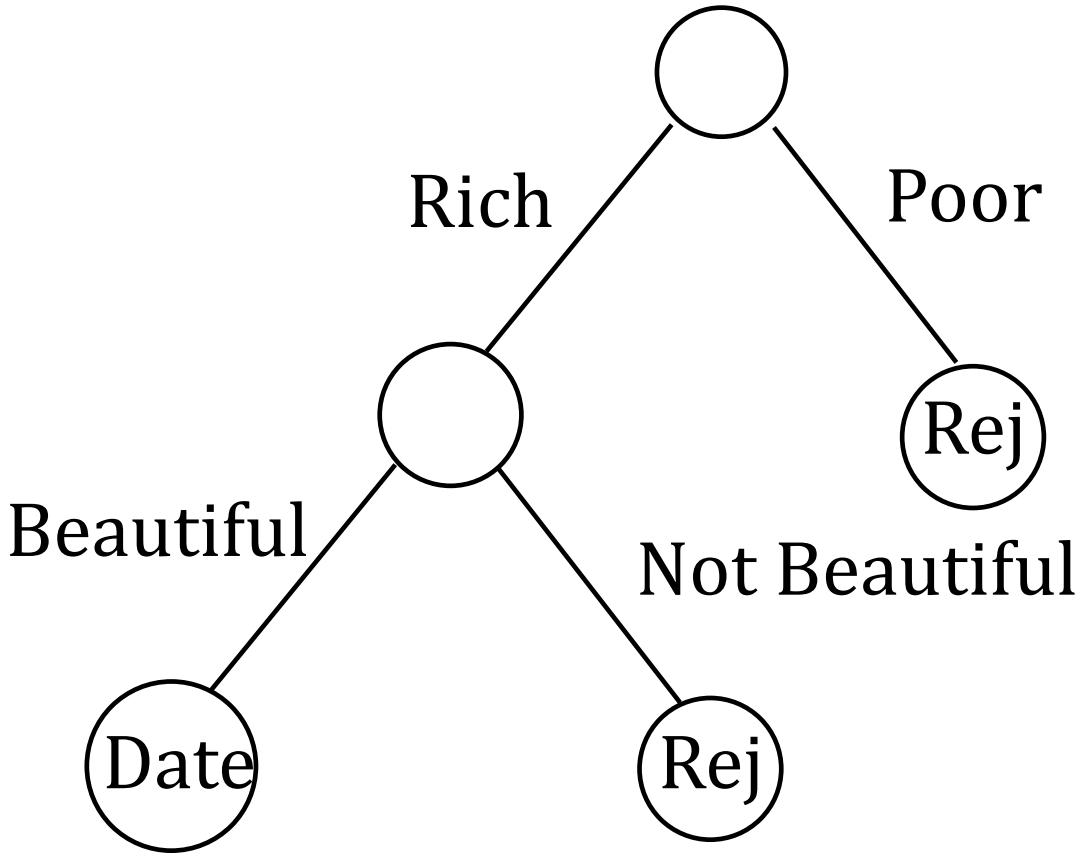
Graph of a Tree



Section 2

Decision Tree

Decision Tree: A Case



Decision Tree: Idea

- How to make decision?
  - Data helps.
  - Gain information, or reduce uncertainty! But how?

Definition 1: Entropy

Suppose X is a discrete random variable with probability distribution $P(X = x_i) = p_i, i = 1, 2, \cdots, n$, then we define entropy as $H(X) = -\sum_{i=1}^{n} p_i \log p_i$ and define $0 \log 0 = 0$

- Reasonable?
- Come from Information Theory

Decision Tree: Idea

Definition 2: Conditional Entropy

Suppose X, Y has a joint probability distribution $P(X = x_i, Y = y_j) = p_{ij}, i = 1, \cdots, n; j = 1, \cdots, m$, then we have $H(Y|X) = \sum_{i=1}^{n} p_i H(Y|X = x_i)$, where $p_i = P(X = X_i)$.

Definition 3: Information Gain

Define $g(D, A) = H(D) - H(D|A)$ as the information gain of feature A w.r.t. dataset D.

- Greedy Algorithm: Only do a "local" optimization problem.

Subsection 1

ID3, C4.5 Algorithm

ID3 Algorithm

- Choose a feature with the highest information gain, on which "make decision" relies.
- If the information gain is less than a threshold ϵ or the data is "totally pure" and no need to do split, stop.

Case Study

表 5.1 贷款申请样本数据表

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

Case Study

- Suppose A_1 means "age", A_2 means "job", A_3 means "house" and A_4 means "credit", then we have

$$g(D, A_1) = H(D) - [\frac{5}{15}H(D_1) + \frac{5}{15}H(D_2) + \frac{5}{15}H(D_3)]$$

where $H(D) = -\frac{9}{15}\log_2\frac{9}{15} - \frac{6}{15}\log_2\frac{6}{15} = 0.971$,
$H(D_1) = H(D_2) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971$,
$H(D_3) = -\frac{1}{5}\log_2\frac{1}{5} - \frac{4}{5}\log_2\frac{4}{5} = 0.722$. So we have
$g(D, A_1) = 0.971 - 0.888 = 0.083$.

- By the same ways we have $g(D, A_2) = 0.324$, $g(D, A_3) = 0.420$ and $g(D, A_4) = 0.363$. This means the first feature for us to consider is A_3.

Case Study

- After the first step, we could find that, if $A_3 = True$ (a man has his own house), then they are all labeled as "True", which means we do not need to further split that node. Therefore we only need to consider the case with $A_3 = False$.
- Suppose the subset is denoted by D_2, then $H(D_2) = -\frac{3}{9}\log\frac{3}{9} - \frac{6}{9}\log\frac{6}{9} = 0.918$. Moreover, we have $g(D_2, A_1) = H(D_2) - H(D_2|A_1) = 0.251$, $g(D_2, A_2) = 0.918, g(D_2, A_4) = 0.474$, so we choose A. Stop because $H(D_2|A_2) = 0$.

C4.5 Algorithm

Definition 4: Information Gain Ratio

We define $g_R(D, A) = \frac{g(D,A)}{H(D)}$ as the information gain ratio.

- Compared with ID3, the C4.5 algorithm only changes the criterion from information gain to information gain ratio.

Subsection 2

CART Algorithm

CART Algorithm

- CART: Classification and Regression Tree
- Binary Tree
- Regression: The idea is the same, because "make decision" means partitioning the dataset.
- For classification we will use Gini Index as the criterion, for regression, squared error instead.
- Suppose $D = \{(x_1, y_1), \cdots, (x_N, y_N)\}$ is the dataset and the input space has been split into M parts, named $R_1, \cdots, R_M$, and the data point's output is c_m in R_M, respectively, then the decision could be written as $f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$

Regression: Core Idea

Objective

$$\min_{j,s}\left[\min_{c_1}\sum_{x_1\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right]$$

where $R_1(j,s)=\{x|x^{(j)}\le s\}$, $R_2(j,s)=\{x|x^{(j)}>s\}$,
$\hat{c}_1=\text{ave}(y_i|x_i\in R_1(j,s))$ and $\hat{c}_2=\text{ave}(y_i|x_i\in R_2(j,s))$.

- This means we need to find j, the best feature to split, and s, the best location to partition.
- Difference with Classification?

Classification: Gini Index

Definition 5: Gini Index

Similar to definition 1, we define $\text{Gini}(X) = \sum_{i=1}^{n} p_k(1-p_k) = 1 - \sum_{i=1}^{n} 1 - p_k^2$. Moreover, given feature A, we define $\text{Gini}(D, A) = \frac{|D_1|}{|D|}\text{Gini}(D_1) + \frac{|D_2|}{|D|}\text{Gini}(D_2)$ as the Gini Index conditioned on A.

Case Study

表 5.1 贷款申请样本数据表

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

Case Study

- By similar ways we have $\text{Gini}(D, A_1 = 1) = \frac{5}{15}(2 \times \frac{2}{5} \times (1 - \frac{2}{5})) + \frac{10}{15}(2 \times \frac{7}{10}(1 - \frac{7}{10})) = 0.44$, $\text{Gini}(D, A_1 = 2) = 0.48$, $\text{Gini}(D, A_1 = 3) = 0.44$, $\text{Gini}(D, A_2 = 1) = 0.32$, $\text{Gini}(D, A_3 = 1) = 0.27$, $\text{Gini}(D, A_4 = 1) = 0.36$, $\text{Gini}(D, A_4 = 2) = 0.47$, $\text{Gini}(D, A_4 = 3) = 0.32$. So the first partition point is $A_3 = 1$.
- For the next step, similarly, we have $A_2 = 1$ is the desired partition point, stop because all the nodes do not need to be splited.

Subsection 3

Tree Pruning

Pruning for ID3 and C4.5 Algorithm

Criterion

$$C_\alpha(T) = C(T) + \alpha|T|$$

- Idea: Recursively prune the nodes from bottom to top if the new cost function decreases.
- Can be solved by dynamic programming.

CART Pruning

- As $\alpha \to \infty$, single point is the best result. Instead, $\alpha \to 0$, the original tree is the best. We want to find a series $\alpha_1 \leq \alpha_2 \leq \cdots$ to generate a series of decision trees, from which we choose one.
- Because we know $C_\alpha(T) = C(T) + \alpha|T|$, so we compare the loss functions in two cases, i.e. $C_\alpha(t) = C(t) + \alpha$ and $C_\alpha(T_t) = C(T_t) + \alpha|T_t|$, where t denotes one internal point and T_t is the tree where t is the root node. Then we want to compute $g(t) = \frac{C(t) - C(T_t)}{|T_t| - 1}$ as the criterion.
- Suppose the algorithm executes at step t, then prun the node with the least $g(t)$ and let $\alpha_{t+1} = g(t)$, continue until the tree is changed as a single point.

Section 3

Boosting Tree and GBDT

Boosting Tree

- Idea: $f_M(x) = \sum_{m=1}^{M} T(x;\Theta_m), f_0(x) = 0$, where $T(x;\Theta_m)$ is the cost function of one decision tree.

Objective

$$\hat{\Theta}_m = \arg\min_{\Theta_m} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + T(x_i;\Theta_m))$$

Boosting Tree: Regression

- When we use squared error, we have

$$L(y, f_{m-1}(x) + T(x; \Theta_m) = [r - T(x; \Theta_m)]^2$$

with $r = y - f_{m-1}(x)$. This means at each step, we only need to train a regression tree based on the residual data in the previous step.

- Similar to Classification problem (Adaboost).

GBDT

- GBDT: Gradient Boosting Decision Tree
- Idea: Use negative gradient as the approximation of residual.
- Let $f_0(x) = \arg\min_c \sum_{i=1}^{N} L(y_i, c)$, and for each step we have $r_{mi} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=f_{m-1}(x)}$ and the next step we train $T(x; \Theta_m)$ based on the residuals $\{r_{mi}\}$.

Thank you!