1 Decision Tree

1.1 Decision Tree Model

Decision tree can be used in both classification and regression. Decision tree can be seen as:

1. A set of if-then rules
2. Conditional probability distribution of classes when given features, $P(Y|X)$, Y is the random variable of class, X is the random variable of feature.

1.2 Problem

Dataset, (N is number of instances):

$$D = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\} \tag{1}$$

Instance, (d is number of feature):

$$x_i = \{x_i^1, x_i^2, ..., x_i^d\} \tag{2}$$

Labels, (K is number of classes):

$$y_i \in \{1, 2, ..., K\} \tag{3}$$

Loss function is regularized maximum likelihood function.

Learning algorithm of decision tree is heuristic algorithm. Recursively find best feature to split and then split the node. Always find a sub-optimal solution.

Learning algorithm of decision tree includes

1. selecting features
2. generating decision tree
3. pruning decision tree

1.3 selecting features

Entropy:

$$H(X) = -\sum_{i=1}^{n} p_i \log(p_i) \tag{4}$$

$$P(X = x_i) = p_i, i = 1, 2, ..., N \tag{5}$$

$$0 \leq H(X) \leq \log(N) \tag{6}$$

Conditional Entropy:

$$H(Y|X) = \sum_{i=1}^{N} p_i H(Y|X = x_i) \tag{7}$$

When we get entropy and conditional entropy from data estimation especially maximum likelihood estimation, the entropy and conditional entropy called empirical entropy and empirical conditional entropy.

Information gain:

$$\begin{aligned} g(D, A) &= H(D) - H(D|A) \\ &= H(D) - \sum_{i=1}^{n} \frac{|D_i|}{D} H(D_i) \end{aligned} \tag{8}$$

Information gain ratio:

$$g_R(D, A) = \frac{g(D, A)}{H_A(D)} \tag{9}$$

$$H_A(D) = \sum_{i=1}^{n} \frac{|D_i|}{D} \log_2(\frac{|D_i|}{D}) \tag{10}$$

n is the number of values that feature A can take.

1.4 ID3

Training data D, feature set A, threshold ϵ, is equal to selecting model by maximum likelihood.

1. If all instances in D belong to one class C_k, the tree has only one node and we assign C_k to the node, then return the tree T.

2. If A is $\emptyset$, the tree has only one node and we assign C_k that has most instances to the node, then return the tree T.

3. Else, compute information gain for each feature, choose the max one A_g.

4. If information gain is less than threshold ϵ, we assign C_k that has most instances to the node, then return the tree T.

5. If information gain is greater than or equal to threshold ϵ, we split the the data D by every value of $A_g = a_i$ and get D_i, we assign assign C_k that has most instances in D_i to node, construct child node gat every T_i, return tree T.

6. For ith node, we use D_i as training data, $A - \{A_g\}$ as feature set, recursively call $1 \sim 5$, return tree T_i

1.5 C4.5

Training data D, feature set A, threshold ϵ, is equal to selecting model by maximum likelihood.

1. If all instances in D belong to one class C_k, the tree has only one node and we assign C_k to the node, then return the tree T.

2. If A is $\emptyset$, the tree has only one node and we assign C_k that has most instances to the node, then return the tree T.

3. Else, compute information gain ratio for each feature, choose the max one A_g.

4. If information gain ratio is less than threshold ϵ, we assign C_k that has most instances to the node, then return the tree T.

5. If information gain ratio is greater than or equal to threshold ϵ, we split the the data D by every value of $A_g = a_i$ and get D_i, we assign assign C_k that has most instances in D_i to node, construct child node gat every T_i, return tree T.

6. For ith node, we use D_i as training data, $A - \{A_g\}$ as feature set, recursively call $1 \sim 5$, return tree T_i

1.6 Pruning

Pruning is usually by minimizing the loss function or cost function of the whole tree.