Prediction Uncertainty Based On Classification Agianst Unmodelled Input Space

Xiaozhe Gu
Energy Research Institute (ERI@N), Singapore

I. Basic Decision Tree

- Criterion: Gini Index

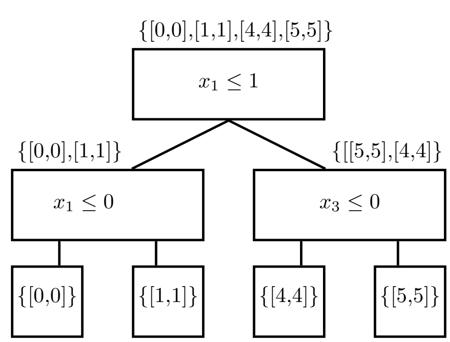
$$I_G(p) = \sum_{i=1}^{n} p_i(1-p_i) = 1 - \sum_{i=1}^{n} p_i^2$$

- Split Value for feature j: $\mathbf{X}_j \in \{\mathbf{X}_{1,j}, \mathbf{X}_{2,j}, \ldots, \mathbf{X}_{n,j}\}$. We do not need to consider the value in empty space.
- Gini Index After Split by feature i and value s $R_1(x_i, s) = \{x | x_i \le s\}$, and $R_1(x_i, s) = \{x | x_i > s\}$,$s \in \mathbf{X}_j$:

$$\begin{aligned}
IG_L(x_i, s) = 1 - \left(\frac{R_1(x_i, s)}{R_1(x_i, s) + E_1}\right)^2 - \left(\frac{E_1}{R_1(x_i, s) + E_1}\right)^2 \\
E_1 = E \times \left(\frac{s - x_i^{\text{min}}}{x_i^{\text{max}} - x_i^{\text{min}}}\right) \\
IG_R(x_i, s) = 1 - \left(\frac{R_2(x_i, s)}{R_2(x_i, s) + E_2}\right)^2 - \left(\frac{E_2}{R_1(x_i, s) + E_2}\right)^2 \\
E_2 = E \times \left(\frac{x_i^{\text{max}} - s}{x_i^{\text{max}} - x_i^{\text{min}}}\right) \\
E = R \times c, \text{ where c is a hyper paramter} \\
IG_{\text{gain}}(x_i, s) = \frac{R_1 + E_1}{R + E} IG_L + \frac{R_2 + E_2}{E + R} IG_R \\
x_i, s = \arg\min_{x_i, s} IG_{\text{gain}}(x_i, s)
\end{aligned}$$

- Node Representation after spilt: the limit of each feature in the split data: $[x_i^{\text{min}}, x_i^{\text{max}}]$ for each i. Any x that not include in these rules are consider empty.

Fig. 1. Data Limitation as Node



  - Furture work: But $x_i^{\text{min}}, x_i^{\text{max}}$ can be manually set to avoid the case that X_i is amostly uniformaly distributed. Thus $\mu(x_{i,k+1} - x_{i,k}))$ is a useful information to determine the node limitations. For now, just let bagging solve these problems.
- For category feature, for example, $X_i \in \{1, 2, 3, 4, 5\}$, we do not need to consider these features.

A. *Issues*

- Suppose for node 0, the split feature index is i, with data $\{1, 2, 4, 7, 10\}$ and s=4, then the left is $x_i \leq 4$ and the right is $x_i > 4 \Rightarrow x_i \geq 7$. Thus, we `node 0` should restore `feature i, sl=4,sr=7`.
- The edge limit of each split is stored in the data structure, and the leafs.
- Stop building when `n_sampling=1` , and at this time, the confident region is around the data point x
- **Problem:** how to determine the `n_sampling` after each split is a big question. Suppose $x_1 = 2x_2 + b + \sigma^2$, then `n_empty >>n_sampling`

B. *Ideas*

:

- Also consider the feature importance. For example, feature 1 is important when $x_1 \in [0, 10]$, then, as we consider the split feature, this feature could has higher priority
- Consider the problem to classify 100 points in 10 dimension.
- consider the variance of data in the split? If it has small variance, then reduce the empty samples?