Prediction Uncertainty Based On Classification Agianst Unmodelled Input Space

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

I. Random Forest for Separating the space

In this section, we propose a decision tree that can be used to separate regions that higher data density from other such regions by sparse regions in which most error could exist. One important reason we choose a decision tree classifier rather than other machine learning classifiers (e.g., neural networks) is that we do not need to sample data points for the empty space which can be very hard in high dimension space.

A. Decision Tree Construction

A commonly used criterion (or purity function) for choosing the best cut is the Gini impurity. Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset.

$$I_G(p) = \sum_{k=1}^{K} p_k(1-p_k) = 1 - \sum_{k=1}^{K} p_k^2$$

, where p_k denotes the fraction of items labeled k in the set. In our case, we are only interested in separating the training samples and the empty space, and the impurity of a subspace can be calculated as follows

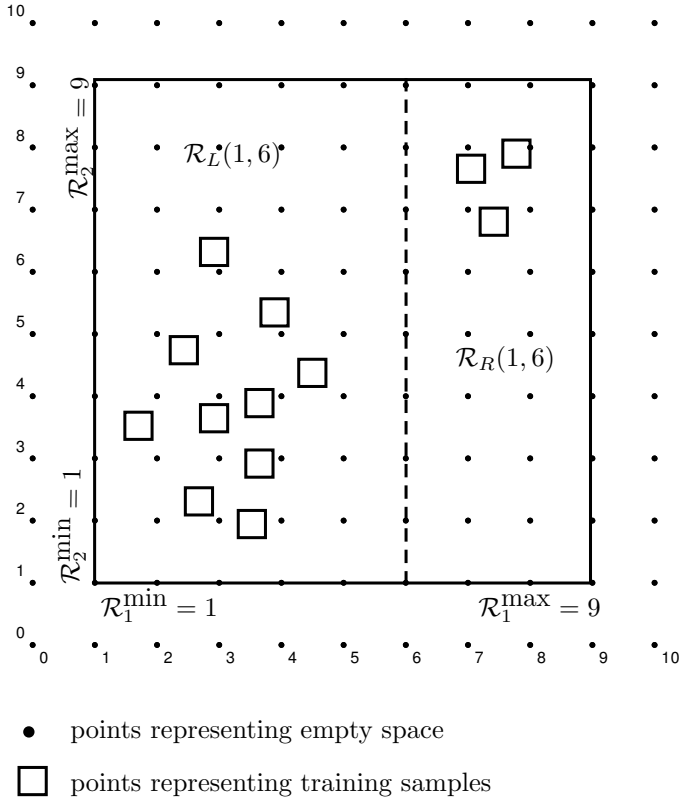
$$I_G(\mathcal{R}) = 1 - \left(\frac{\mathbb{R}}{\mathbb{R}+\mathbb{E}}\right)^2 - \left(\frac{\mathbb{E}}{\mathbb{R}+\mathbb{E}}\right)^2$$

Here $\mathbb{R}$ and $\mathbb{E}$ denote the number of training samples and samples representing empty space in region $\mathcal{R}$. Suppose we cut $\mathcal{R}$ into two subspaces $\mathcal{R}_L(i,s)$ and $\mathcal{R}_R(i,s)$ by feature i and value s, then new Gini impurity for the left child and right child can be calculated as follows

$$IG_L(i,s) = 1 - \left(\frac{\mathbb{R}_L(i,s)}{\mathbb{R}_L(i,s)+\mathbb{E}_L}\right)^2 - \left(\frac{\mathbb{E}_L}{\mathbb{R}_L(i,s)+\mathbb{E}_L}\right)^2$$
$$\mathbb{E}_L = \mathbb{E} \times \left(\frac{s - \mathcal{R}_i^{\min}}{\mathcal{R}_i^{\max} - \mathcal{R}_i^{\min}}\right)$$
$$IG_R(i,s) = 1 - \left(\frac{\mathbb{R}_R(i,s)}{\mathbb{R}_R(i,s)+\mathbb{E}_R}\right)^2 - \left(\frac{\mathbb{E}_R}{\mathbb{R}_R(i,s)+\mathbb{E}_R}\right)^2$$
$$\mathbb{E}_R = \mathbb{E} \times \left(\frac{\mathcal{R}_i^{\max} - s}{\mathcal{R}_i^{\max} - \mathcal{R}_i^{\min}}\right)$$

In the above equation, $\mathbb{R}_L(i,s)$ and $\mathbb{R}_R(i,s)$ denote the number of training samples in the region $\mathbb{R}$ has $X_i \leq s$ and $X_i \geq s$, respectively. $\mathcal{R}_i^{\min}$ and $\mathcal{R}_i^{\max}$ denote the lower and upper bound of dimension i for region $\mathcal{R}$. Since the data points representing the empty space are assumed to be uniformed distributed among $\mathbb{R}$, then $\mathbb{E}_L$ and $\mathbb{E}_R$ are in proportion to $s - \mathcal{R}_i^{\min}$ and $\mathcal{R}_i^{\max} - s$, respectively. After the region $\mathbb{R}$ is divided into $\mathbb{R}_L(i,s)$ and $\mathbb{R}_R(i,s)$, the Gini impurity is updated as follows.

Fig. 1. Tree Representation



$$IG_{\text{split}}(i,s) = \frac{\mathbb{R}_L(i,s) + \mathbb{E}_L}{\mathbb{R}+\mathbb{E}} IG_L(i,s) + \frac{\mathbb{R}_R + \mathbb{E}_R}{\mathbb{E}+\mathbb{R}} IG_R(i,s)$$

The decision tree will select to cut the space into subspaces that minimizes the Gini impurity.

$$i, s = \arg\min_{i,s} IG_{\text{split}}(i,s)$$

B. Inference

After the tree is built, the initial region $\mathcal{R}$ will be divided into multiple hyperrectangle. In Figure 2, we show an example of a constructed tree for 2-d training samples $X = \{(0,0),(1,1),(3,3),(4,4),(6,6),(7,7),(9,9),(10,10)\}$. The initial region that covers training samples is $\mathcal{R} = [0,10]^2$ and its volume is $\mathcal{R}.vol = 100$. We also assume that infinite number of data points representing the empty space are uniformly distributed among $\mathcal{R}$ with weight $\frac{\mathbb{R}}{100}$. Thus, the total weighted number of empty points is equal to $\mathbb{E} = \mathbb{R} = 10$.

The tree divides $\mathcal{R}$ into four rectangles $\mathcal{R}_i,\ i \in \{1,2,3,4\}$ by cut axis 1 at $\{2,5,8\}$. Thus, the proportion of training samples in the four rectangles are $\frac{2}{2+2}, \frac{2}{2+3}, \frac{2}{2+3}, \frac{2}{2+2}$, respectively.

Since axis 2 is not split, we can further divided $\mathcal{R}_i$ into $\mathcal{R}_{i,1}$ and $\mathcal{R}_{i,2}$ by considering the bounds of samples on axis 2. Thus, in the $R_{i,1}\ i \in \{1,2,3,4\}$, the proportion of training samples in the four rectangles are $\frac{2}{2+2/10}, \frac{2}{2+3/10}, \frac{2}{2+3/10}, \frac{2}{2+2/10}$, respectively.

Finally, we can get a list of training sample density for each hyperrectangle, e.g., $[d_1, d_2, \ldots, d_K]$. After sort it in ascending order. For example, we can discard $p\%$ hyperrectangle by choose the $p\% \times |K|_{st}$ item as threshold to predict the hyperrectangle as a confident region.

C. Stop Criterion

We also need to address when to stop building the tree. We mainly consider the following metrics:

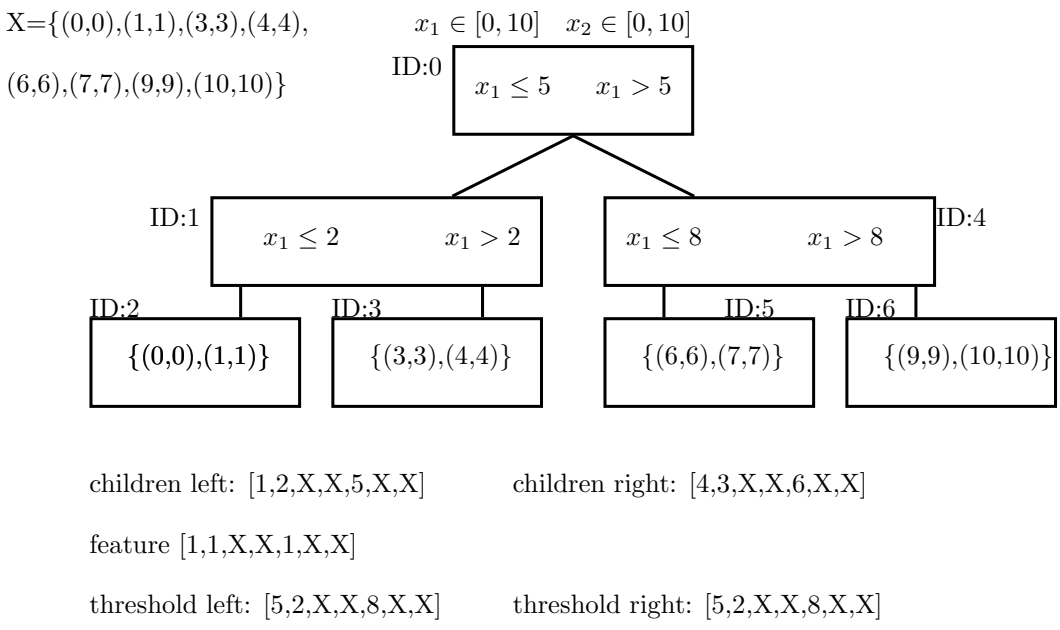


Fig. 2. Tree Representation

- `max_depth` : maximum depth of the tree
- `min_samples_leaf` : minimal samples required for a leaf node.
- `min_gain_ratio`: suppose at the first time we split the data, the gain of Gini impurity is $root\_gain = IG_{\text{split}} - IG_{\text{initial}}$. Thus, if the gain of Gini impurity declines to a certain threshold, i.e., $\frac{IG_{\text{split}} - IG_{\text{initial}}}{root_g ain} \leq \sigma$, we can stop building.

Note that in our problem, since we are only interested in separating the input space into hyperrectangles, we do not need to address the over-fit problem, and it is acceptable that each leaf even only have one sample, and the depth of the tree is very high.

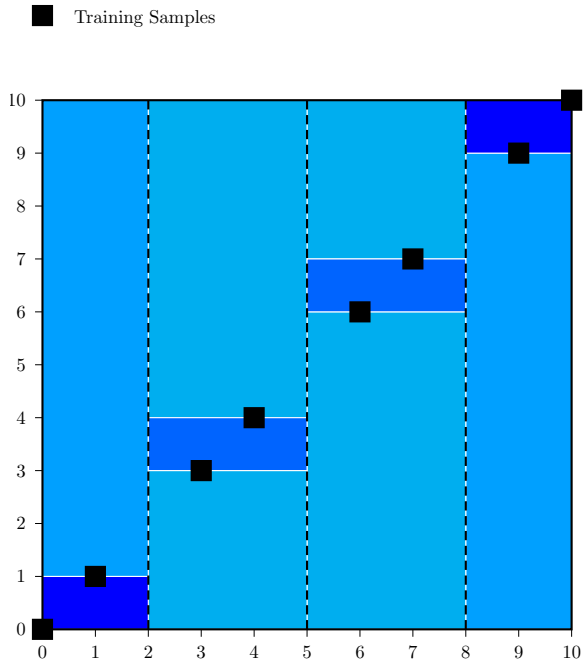
D. Build The Forests

Finally, we can build the forest based on the above tree, and the output of the forest is the probability that the base estimators vote this point as the confidence level.

II. Ideas

:

- **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`
-
- 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?
- What is the data set has a good shape (uniform distribution). In this case, the split should stop by observing that the Gini Index Increase is very small.
- try multiple features? $x_1 \in [0, 3] \wedge x_2 \in [0, 4]$ if the score gain from the two split is similar?
- Furture work: But $x_i^{\min}, x_i^{\max}$ 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.


Training Samples
0
1
2
3
4
5
6
7
8
9
10