Decision Tree and Extra Tree

Bo Wu

I. Notes

In this section, I review classification/regression approaches of decision tree and extra trees.

A. Decision Tree

Decision tree (DT) provides a rapid and useful solution for predicting the class or value of the target variable by learning simple decision rules inferred from training data. It is typically a top-down greedy approach to determine a rule set by recursively partitioning the training datasets into smaller subsets based on data attributes until all the subsets belong to a single class. If the dataset consists of multiple attributes then deciding which attribute to place at the root or at different levels of the tree as internal nodes is a complicated step. For solving this attribute selection problem, some common criteria such as information gain and Gini impurity are often used.

A common decision tree algorithm is ID3 (Iterative Dichotomiser 3) [1], for instance, which uses information theory, or entropy, as its attribute selection measure. In ID3, a branch with an entropy of zero or a very small number is a leaf node and a brach with entropy more than zero needs further splitting. The root node of decision tree is chosen based on the highest information gain of the attribute. Given a training dataset, D, the expected information or entropy needed to correctly classify an instance, $x_i \in D$, is given in (1), where p_i is the probability that $x_i \in D$, belongs to a class, C_i and is estimated by $|D(y = C_i)| \, / \, |D|$ where $|\cdot|$ denotes the sample number in a set.

$$H(D) = -\sum_{i=1}^{|D|} p_i \log_2 (p_i) \,. \tag{1}$$

The goal of decision tree is to iteratively partition, D, into subsets, $\{D_1, D_2, \ldots, D_n\}$, where all instances in each D_i belong to the same class, C_i, by instance attributes or features, As. Information gain is defined as the difference between the original information and the conditional information that as below.

$$\begin{aligned} IG &= H(D) - H(D|A) \\ &= H(D) - \sum_{j=1}^{|A|} \frac{|D_A|}{|D|} \times H\left(D_{A_j}\right), \end{aligned} \tag{2}$$

where A_j is the j-th distinct value of attribute A. At each newly partitioned tree level, decision tree selects the attribute that has the largest information gain (2) on the data that remains to be classified.

Another criterion is the Gini impurity used to evaluate splits in the dataset that is widely used in classification and regression tree (CART) [2]. The Gini impurity for a dataset, D, is calculated by subtracting the sum of the squared probabilities of each class from one. It favors larger partitions and easy to implement (i.e. binary trees) whereas information gain favors smaller partitions with distinct values.

$$Gini(D) = 1 - \sum_{i=1}^{C} (p_i)^2 \,. \tag{3}$$

The Gini impurity of a split of D into subsets D_1 and D_2 by the attribute A is defined as

$$Gini(D|A) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2). \tag{4}$$

Different from the information gain criterion, decision tree selects the attribute that has the smallest Gini impurity (4) on the data to grow the tree.

B. Extra Trees

Since decision tree applies hard thresholds regarding data attributes to grow the tree, it may lack of generalization. Extremely Randomized Trees, or Extra Trees [3], is an ensemble of decision trees algorithm that is widely used to mitigate decision tree's drawbacks. It is related to other ensembles of decision trees algorithms such as bootstrap aggregation (bagging) [4] and random forest [5]. Like random forest, the Extra Trees algorithm will randomly sample the features at each split point of a decision tree making the decision trees in the ensemble less correlated. Unlike bagging and random forest that develop each decision tree from a bootstrap sample of the training dataset, the Extra Trees algorithm fits each decision tree on the whole training dataset. Unlike random forest, which uses a greedy algorithm to select an optimal split point, the Extra Trees algorithm selects a split point fully at random. Extra Trees can often achieve as-good or better performance than the random forest algorithm.

Extra Trees works by creating a large number of unpruned decision trees from the whole training dataset in a top-down recursive way. Predictions are made by averaging the prediction of the decision trees in the case of regression or using majority voting in the case of classification. It is also easy to use given that it has just few key hyperparameters. There are three main hyperparameters to tune in the algorithm: K, the number of decision trees in the ensemble, i_{min}, the number of input features to randomly select and consider for each split point, and n_{min}, the minimum number of samples required in a node to create a new split point.

References

[1] J. R. Quinlan, "Induction of decision trees," *Mach. Learn*, vol. 1, no. 1, pp. 81–106, 1986.

[2] R. J. Lewis, "An introduction to classification and regression tree (cart) analysis," in *Annual meeting of the society for academic emergency medicine in San Francisco, California*, vol. 14. Citeseer, 2000.

[3] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” *Mach. Learn*, vol. 63, no. 1, pp. 3–42, 2006.

[4] J. R. Quinlan *et al.*, “Bagging, boosting, and c4. 5,” in *Aaai/iaai, Vol. 1*, 1996, pp. 725–730.

[5] L. Breiman, “Random forests,” *Mach. Learn*, vol. 45, no. 1, pp. 5–32, 2001.