CS 189/289A Introduction to Machine Learning
Fall 2022 Jennifer Listgarten, Jitendra Malik

DIS8

1 Decision Trees

Consider constructing a decision tree on data with d features and n training points where each feature is real-valued and each label takes one of m possible values. The splits are two-way, and are chosen to maximize the information gain. We only consider splits that form a linear boundary parallel to one of the axes. We will only consider a standalone decision tree and not a random forest (hence no randomization). Recall the definition of information gain:

$$IG(\textbf{node}) = H(S) - \frac{|S_l|H(S_l) + |S_r|H(S_r)}{|S_l| + |S_r|},$$

where S is set of samples considered at **node**, S_l is the set of samples remaining in the left subtree after **node**, S_r is the set of samples remaining in the right subtree after **node**, and $H(S)$ is the entropy over a set of samples:

$$H(S) = -\sum_{i=1}^{C} p_i \log(p_i)$$

Where C is the number of classes, and p_i is the proportion of samples in S labeled as class i.

(a) Prove or give a counter-example: In any path from the root to a leaf, the same feature will never be split on twice.

Solution: False. Example: one dimensional feature space with training points of two classes x and o arranged as xxxooooxxx.

(b) Prove or give a counter-example: The information gain at the root is at least as much as the information gain at any other node.
Hint: Think about the XOR function.

Solution: False. Consider the XOR function, where the samples are

$$S = \{(0,0;0),(0,1;1),(1,0;1),(1,1;0)\},$$

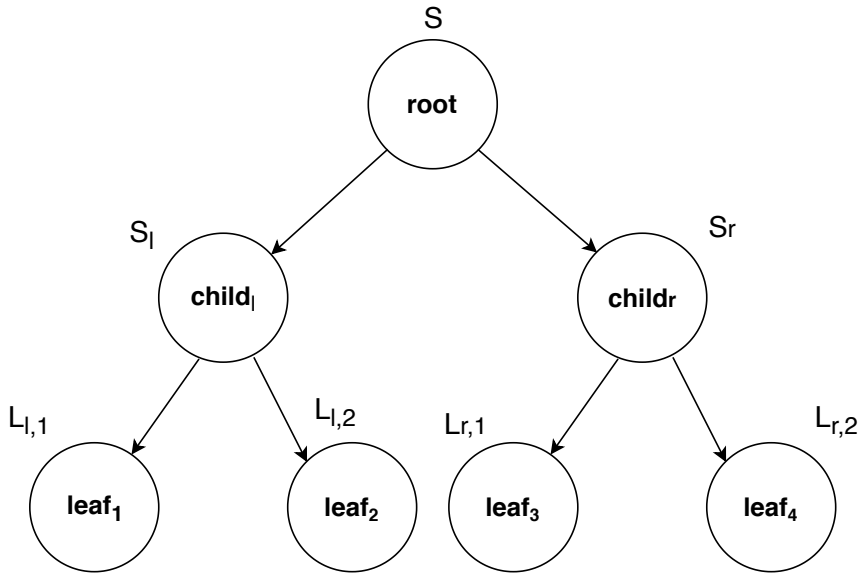
where the first two entries in every sample are features, and the last one is the label. Then, $H(S) = 1$. The first split is done based on the first feature, which gives $S_l = \{(0,0;0),(0,1;1)\}$ and $S_r = \{(1,0;1),(1,1;0)\}$; denote the corresponding nodes as $\textbf{child}_l$ and $\textbf{child}_r$ respectively. This gives $H(S_l) = 1$ and $H(S_r) = 1$. The information gain of the first split is:

$$IG(\textbf{root}) = H(S) - \frac{|S_l|H(S_l) + |S_r|H(S_r)}{|S_l| + |S_r|} = 0.$$

Now we further split S_l and S_r according to the second feature, which gives 4 leaves of 1 sample each. Denote the leaf samples corresponding to S_r as $L_{r,1}$ and $L_{r,2}$, and accordingly denote by $L_{l,1}$ and $L_{l,2}$ the leaves corresponding to S_l. Now we have

$$IG(\textbf{child}_l) = H(S_l) - \frac{1 \cdot H(L_{l,1}) + 1 \cdot H(L_{l,2})}{1+1} = 1,$$

and analogously $IG(\textbf{child}_r) = 1$. Therefore, the information gain at each of the child nodes is 1, while the information gain at the root is 0.



(c) Suppose that a learning algorithm is trying to find a consistent hypothesis when the labels are actually being generated randomly. There are d Boolean features and 1 Boolean label, and n examples are drawn uniformly from the set of 2^{d+1} possible examples *with* replacement. Calculate the probability of finding a contradiction in the sampled data. For ease of computation, you don't have to consider the case where identical samples (samples with the same features and same label) are drawn from the distribution.
(A contradiction is reached if two samples with identical features but different labels are drawn.)

Solution: Suppose that we draw n samples. Each sample has d input features plus its label, so there are 2^{d+1} distinct feature vector/label examples to choose from. For each sample, there is exactly one contradictory sample, namely the sample with the same input features but the opposite label. Thus, the probability of finding no contradiction is

$$\frac{\#\text{ of sequences of non-contradictory samples}}{\#\text{ of different sequences}} = \frac{2^{d+1}(2^{d+1}-1)\ldots(2^{d+1}-n+1)}{2^{n(d+1)}} = \frac{2^{d+1}!}{(2^{d+1}-n)!2^{n(d+1)}}.$$

For example, if $d = 10$, there are 2048 possible samples, and a contradiction has probability greater than 0.5 already after 54 drawn samples.

If the sampling is done *without* replacement, we would have one fewer sample (and wouldn't have to worry about duplicates being drawn from the dataset, which we ignored in this question.) So the answer would become:

$$\frac{2^{d+1}(2^{d+1}-2)\dots(2^{d+1}-2(n-1))}{2^{d+1}(2^{d+1}-1)\dots(2^{d+1}-n+1)} = \frac{2^{d+1}!(2^{d+1}-n)!}{(\prod_{i=0}^{2(n-1)}(2^{d+1}-(2i+1)))(2^{d+1}-2(n-1)-1)!2^{d+1}!}.$$

(d) Intuitively, how does the bias-variance trade-off relate to the depth of a decision tree?

Solution: If a decision tree is very deep, the model is likely to overfit, and thereby increase variance. Intuitively, there are many conditions checked before making a decision, which makes the decision rule too fine-grained and sensitive to small perturbations; for example, if only one of the many conditions is not satisfied, this might result in a completely different prediction. On the other hand, if the tree is very shallow, this might increase bias. In this case, the decisions are too "coarse".

2 Concerns about Randomness

One may be concerned that the randomness introduced in random forests may cause trouble. For example, some features or sample points may never be considered at all. In this problem we will be exploring this phenomenon.

(a) Consider n training points in a feature space of d dimensions. Consider building a random forest with T binary trees, each having exactly h internal nodes. Let m be the number of features randomly selected (from among d input features) at each tree node. For this setting, compute the probability that a certain feature (say, the first feature) is never considered for splitting in any tree node in the forest.

Solution: The probability that it is not considered for splitting in a particular node of a particular tree is $1 - \frac{m}{d}$. The subsampling of m features at each treenode is independent of all others. There is a total of ht treenodes and hence the final answer is $(1 - \frac{m}{d})^{hT}$.

(b) Now let us investigate the possibility that some sample point might never be selected. Suppose each tree employs $n' = n$ bootstrapped (sampled with replacement) training sample points. Compute the probability that a particular sample point (say, the first sample point) is never considered in any of the trees.

Solution: The probability that it is not considered in one of the trees is $(1 - \frac{1}{n})^n$, which approaches $1/e$ as $n \to \infty$. Since the choice for every tree is independent, the probability that it is not considered in any of the trees is $(1 - \frac{1}{n})^{nT}$, which approaches e^{-T} as $n \to \infty$.

(c) Compute the values of the two probabilities you obtained in parts (b) and (c) for the case where there are $n = 50$ training points with $d = 5$ features each, $T = 25$ trees with $h = 8$ internal nodes each, and we randomly select $m = 1$ potential splitting features in each treenode. You may leave your answer in a fraction and exponentiated form, e.g., $\left(\frac{51}{100}\right)^2$. What conclusions can you draw about the concerns of not considering a feature or sample mentioned at the beginning of the problem?

Solution: $(\frac{49}{50})^{200} \approx .017$ and $(\frac{49}{50})^{1250} \approx 1.07 * 10^{-11}$. It is quite unlikely that a feature will be missed, and extremely unlikely a sample will be missed.