CS 189 Introduction to Machine Learning
Spring 2022 Marvin Zhang
DIS8

1 Curse of Dimensionality in Nearest Neighbor Classification

We have a training set: $(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(n)}, y^{(n)})$, where $\mathbf{x}^{(i)} \in \mathbb{R}^d$. To classify a new point $\mathbf{x}$, we can use the nearest neighbor classifier:

$$\text{class}(\mathbf{x}) = y^{(i^*)} \quad \text{where } \mathbf{x}^{(i^*)} \text{ is the nearest neighbor of } \mathbf{x}.$$

Assume any data point $\mathbf{x}$ that we may pick to classify is inside the Euclidean ball of radius 1, i.e. $||\mathbf{x}||_2 \leq 1$. To be confident in our prediction, in addition to choosing the class of the nearest neighbor, we want the distance between $\mathbf{x}$ and its nearest neighbor to be small, within some distance $\epsilon > 0$:

$$||\mathbf{x} - \mathbf{x}^{(i^*)}||_2 \leq \epsilon \quad \text{for all } ||\mathbf{x}||_2 \leq 1. \tag{1}$$

What is the minimum number of training points we need for inequality (1) to hold (assuming the training points are well spread to cover the maximum amount of space)**?**

How does this lower bound depend on the dimension d?

Hint: Think about the volumes of the hyperspheres in d dimensions. A d-dimensional hypersphere B with radius r has volume $\text{vol}(B) = c \cdot r^d$ for some scalar c.

Solution: Let B_0 be the ball centered at the origin, having radius 1 (inside which we assume our data lies). Let $B_i(\epsilon)$ be the ball centered at $\mathbf{x}^{(i)}$, having radius ϵ. For inequality (1) to hold, for any point $\mathbf{x} \in B_0$, there must be at least one index i such that $\mathbf{x} \in B_i(\epsilon)$. This is equivalent to saying that the union of $B_1(\epsilon), \dots, B_n(\epsilon)$ covers the ball B_0. Let $\text{vol}(B)$ indicate the volume of object B, then we have

$$\sum_{i=1}^{n} \text{vol}(B_i(\epsilon)) = n\text{vol}(B_1(\epsilon)) \geq \text{vol}(\cup_{i=1}^{n} B_i(\epsilon)) \geq \text{vol}(B_0).$$

where the last inequality holds because we are assuming the union of $B_1(\epsilon), \dots, B_n(\epsilon)$ covers the ball B_0. This implies

$$n \geq \frac{\text{vol}(B_0)}{\text{vol}(B_1(\epsilon))} = \frac{c(1^d)}{c\epsilon^d} = \frac{1}{\epsilon^d}$$

Where the constant c is dependent on the formula for the volume of a hypersphere in d dimensions.

Note that we can pick $\frac{1}{\epsilon^d}$ training points and still satisfy (1) only if all the training points are well spread (the union of $B_1(\epsilon), \dots, B_n(\epsilon)$ covers the ball B_0).

This lower bound suggests that to make an accurate prediction on high-dimensional input, we need exponentially many samples in the training set. This exponential dependence is sometimes called the *curse of dimensionality*. It highlights the difficulty of using non-parametric methods for solving high-dimensional problems.

2 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:

$$IG(\textbf{node}) = H(S) - H_{\text{after}} = H(S) - \frac{|S_\ell|H(S_\ell) + |S_r|H(S_r)}{|S_\ell| + |S_r|}$$

where S is set of samples considered at **node**, S_ℓ is the set of samples remaining in the left subtree after **node**, and S_r is the set of samples remaining in the right subtree after **node**.

(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. If false, can you modify the conditions of the problem so that this statement is true?

Solution: False. Example: one dimensional feature space with training points of two classes x and o arranged as xxxooooxxx. This statement would be true if the splits were allowed to form more complex boundaries, i.e. if the splits were not binary and linear.

(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

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

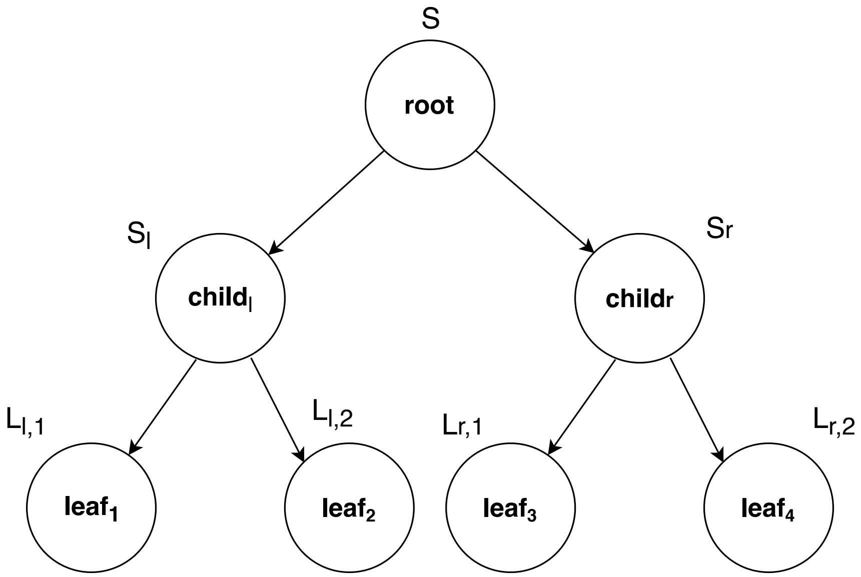
where the first two entries in every sample are features, and the last one is the label. Then, $H(S) = 1$. The first split is done based on the first feature, which gives $S_l = \{(0, 0; 0), (0, 1; 1)\}$ and $S_r = \{(1, 0; 1), (1, 1; 0)\}$; denote the corresponding nodes as $\textbf{child}_l$ and $\textbf{child}_r$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

3 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 treenode to be the available features (features outside this sampled set of size m are not allowed to make splits for this treenode).

For this setting, **compute the probability that a certain feature (say, the first feature) is never considered for splitting in any treenode 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 when training bagged decision trees. 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 (a) and (b)** for the case where there are $n = 2$ training points with $d = 2$ features each, $T = 10$ trees with $h = 4$ 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 concern you had starting the problem?

Solution: $\frac{1}{2^{40}}$ and $\frac{1}{2^{20}}$. It is quite unlikely that a feature or a sample will be missed.