CS 189/289A Introduction to Machine Learning
Fall 2024 Jennifer Listgarten, Saeed Saremi

DIS8

1 The Classical Bias-Variance Tradeoff

Consider a random variable X, which has unknown mean μ and unknown variance σ^2. Given n iid realizations of training samples $X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n$ from the random variable, we wish to estimate the mean of X. We will call our estimate of μ the random variable $\hat{X}$, which has mean $\hat{\mu}$. There are a few ways we can estimate μ given the realizations of the n samples:

1. Average the n samples: $\frac{x_1+x_2+\ldots+x_n}{n}$.
2. Average the n samples and one sample of 0: $\frac{x_1+x_2+\ldots+x_n}{n+1}$.
3. Average the n samples and n_0 samples of 0: $\frac{x_1+x_2+\ldots+x_n}{n+n_0}$.
4. Ignore the samples: just return 0.

In the parts of this question, we will measure the *bias* and *variance* of each of our estimators. The *bias* is defined as

$$\mathbb{E}[\hat{X} - \mu]$$

and the *variance* is defined as

$$\text{Var}[\hat{X}].$$

(a) What is the bias of each of the four estimators above?

(b) What is the variance of each of the four estimators above?

(c) Suppose we have constructed an estimator $\hat{X}$ from some samples of X. We now want to know how well $\hat{X}$ estimates a new independent sample of X. Denote this new sample by X'. Derive a general expression for $\mathbb{E}[(\hat{X} - X')^2]$ in terms of σ^2 and the bias and variance of the estimator $\hat{X}$. Similarly, derive an expression for $\mathbb{E}[(\hat{X} - \mu)^2]$. Compare the two expressions and comment on the differences between them.

(d) It is a common mistake to assume that an unbiased estimator is always "best." Let's explore this a bit further. Compute $E[(\hat{X} - \mu)^2]$ for each of the estimators above.

(e) Demonstrate that the four estimators are each just special cases of the third estimator, but with different instantiations of the hyperparameter n_0.

(f) What happens to bias as n_0 increases? What happens to variance as n_0 increases?

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:

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

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

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

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

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

(b) Draw the graph of entropy $H(p_c)$ when there are only two classes C and D, with $p_D = 1 - p_C$. Is the entropy function strictly concave, concave, strictly convex, or convex? Why? What is the significance?

Hint: For the significance, recall the information gain.

(c) 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. Precisely, consider the set

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

where the first two entries in every sample are features, and the last one is the label.

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

3 Curse of Dimensionality in Nearest Neighbor Classification

We have a training set: $(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(n)}, y^{(n)})$, 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 positive ϵ:

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

What is the minimum number of training points we need for inequality (1) to hold (assuming the training points are well spread)? How does this lower bound depend on the dimension d?

Hint: Think about the volumes of the hyperspheres in d dimensions.