CS 189/289A Introduction to Machine Learning
Fall 2023 Jennifer Listgarten, Jitendra Malik **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. $\|\mathbf{x}\|_2 \le 1$. 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:

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

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.

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 vol(B) indicate the volume of object B, then we have

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

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

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

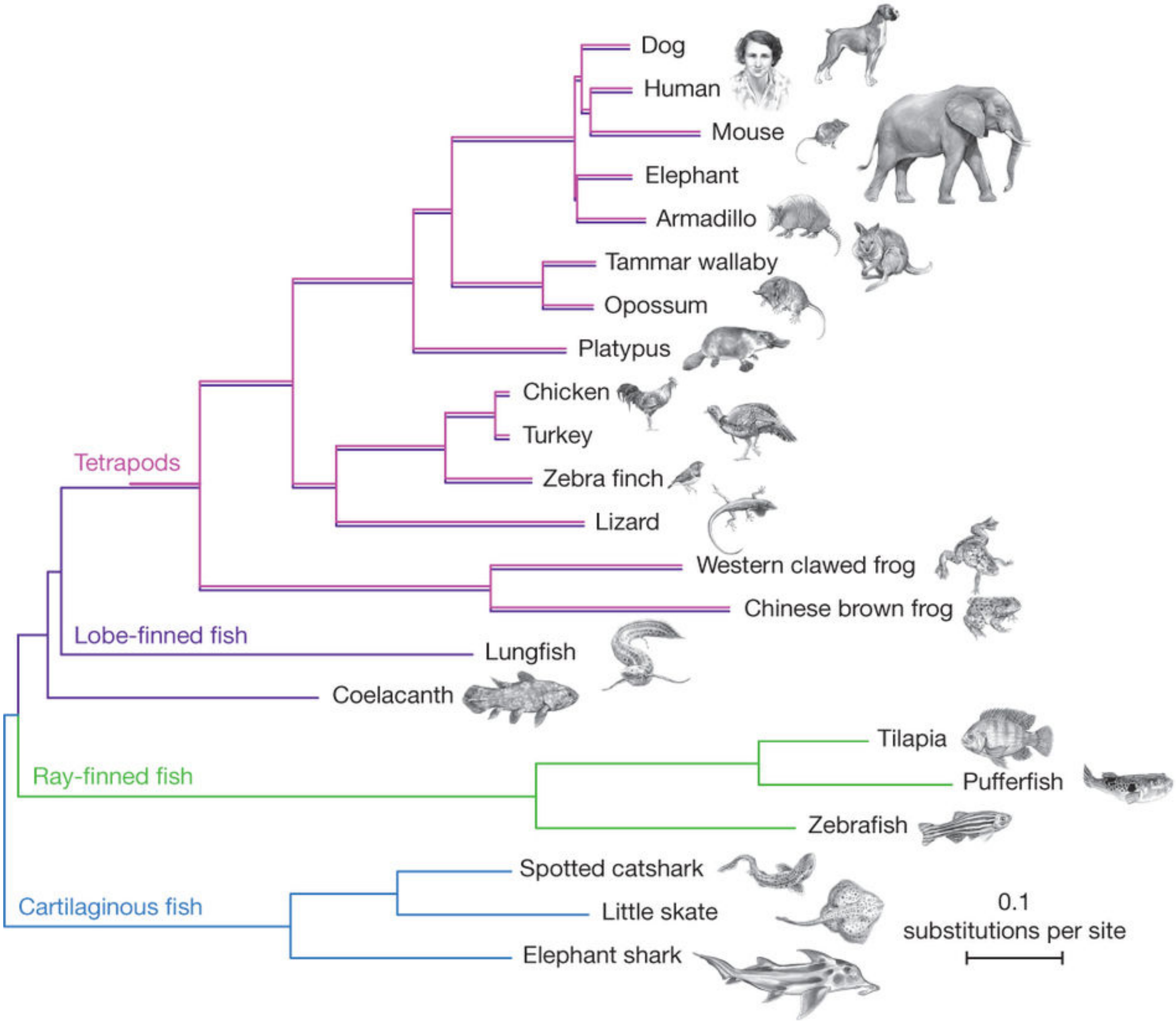
Where the constant c is dependent on the formula for the volume of a hypersphere in d dimensions.

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 Hierarchical Clustering for Phylogenetic Trees

A phylogenetic tree (or "evolutionary tree") is way of representing the branching nature of evolution. Early branches represent major divergences in evolution (for example, modern vertebrae diverging from modern invertebrate), while later branches represent smaller branches in evolution (for example, modern humans diverging from modern monkeys). An example is shown below.



Creating phylogenetic trees is a popular problem in computational biology. We are going to combine what we know about clustering, decision trees, and unsupervised learning.

We start with all the samples (in this case, animals) in a single cluster and gradually divide it up. This should remind you of decision trees! After k steps, we have at most 2^k clusters. Since we do not have labels, we need to find some way deciding how to split the samples (other than using entropy).

We will use the same objective as in k-means clustering to determine how good our proposed clustering is:

$$\forall i \leq k, \mu_i = \frac{1}{|S_i|} \sum_{x_j \in S_i} x_j$$

$$H(S_1, \ldots, S_k) = \sum_{i=1}^{k} \sum_{x_j \in S_i} |x_j - \mu_i|^2$$

At each iteration, we will split each cluster with more than one element into two clusters. The algorithm terminates when everything is in its own cluster.

(a) Consider the following six animals and their two features. Create the resulting decision tree.

Animal	Lifespan	Wings
Dog	12	0
Human	80	0
Mouse	2	0
Elephant	60	0
Chicken	8	2
Turkey	10	2

Solution: Root split: set a threshold of 36 on lifespan, yielding (Dog, Mouse, Chicken, Turkey) vs (Human, Elephant).

Depth 1: in the left branch (Dog, Mouse, Chicken, Turkey), set a threshold of 5 on lifespan to get (Mouse) vs (Dog, Chicken, Turkey); in the right branch (Human, Elephant), set a threshold of 70 on lifespan to get (Human) vs (Elephant).

Depth 2: from (Dog, Chicken, Turkey), set a threshold of 11 on lifespan to get (Dog) vs (Chicken, Turkey).

Depth 3: from (Chicken, Turkey) set a threshold of 9 on lifespan to get (Chicken) vs (Turkey).

At each step, we are trying to minimize the squared distance from each animal to its cluster center. Given a feature, to find the best threshold value we can start by ordering the animals on that feature (like we normally would for a decision tree) and taking the midpoints between animals as possible thresholds. Since there is only one split available for wings, for each cluster we can easily verify which feature will yield a better split by simply checking if any better split exists on lifespan. At the root, lifespan is the better feature to split on. This yields five possible threshold values. By testing each value, we can verify that the midpoint of dog and elephant yields the best objective value. Repeating the process, we can obtain the decision tree described above.

(b) Prove that an optimal clustering on $k + 1 < n$ clusters has an objective value that is at least as small as that of the optimal clustering on k clusters.

Solution: Let $\{S_1, \ldots, S_k\}$ denote a clustering with k clusters. If this clustering already has objective value 0, then, since the clustering objective is a sum of non-negative terms, each $\sum_{x_j \in S_i} |x_j - \mu_i|^2 = 0$, which implies that $x_j = \mu_i$ for all $x_j \in S_i$, for every S_i. Thus for any

$|S_i| \geq 2$ we can clearly split S_i into two clusters, each with the same mean, whose objective value is also 0.

Suppose that the clustering $\{S_1, \ldots, S_k\}$ has objective value greater than 0. Choose any cluster S_i with non-zero cost, which implies both that $|S_i| \geq 2$ and that there exists an element $x_j \neq \mu_i \in S_i$. We can construct a new clustering by splitting S_i into two clusters $\{x_j\}$ and $S_i - \{x_j\}$. The former cluster clearly has cost 0, since we can take $\mu = x_j$. The latter cluster $S_i - \{x_j\}$ has new mean $\mu_i' = \frac{1}{|S_i - \{x_j\}|} \sum_{x_k \in S_i - \{x_j\}} x_k$. The cluster $S_i - \{x_j\}$ has lower cost than S_i since

$$\begin{aligned}\sum_{x_k \in S_i - \{x_j\}} |x_k - \mu_i'|^2 &\leq \sum_{x_k \in S_i - \{x_j\}} |x_k - \mu_i|^2 \\ &\leq \sum_{x_k \in S_i} |x_k - \mu_i|^2.\end{aligned}$$

The first inequality follows from the fact that the mean of a set of points minimizes the sum of the Euclidean distances to those points. To see this, take the derivative of the objective with respect to μ and solve for the minimum. The second inequality follows from adding one additional non-negative term.

(c) What is the value of $H(S_1, \ldots, S_k)$ when $k = n$ (the number of samples)?

Solution: Each sample ends up in its own cluster, hence each sample is the mean of its own cluster and $H(S_1, \ldots, S_k) = \sum_{i=1}^k \sum_{x_j \in S_i} |x_j - \mu_i|^2 = \sum_{i=1}^k |x_i - x_i|^2 = 0$.

3 Surprise and Entropy

In this section, we will clarify the concepts of surprise and entropy. Recall that entropy is one of the standards for us to split the nodes in decision trees until we reach a certain level of homogeneity.

(a) Suppose you have a bag of balls, all of which are black. How surprised are you if you take out a black ball?

Solution: 0. We aren't surprised at all when events with probability 1 occur.

(b) With the same bag of balls, how surprised are you if you take out a white ball?

Solution: ∞. We are infinitely surprised when an event with probability 0 occurs.

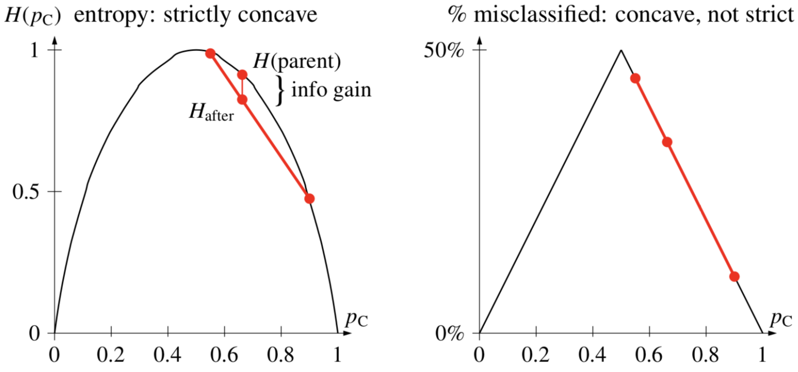
(c) Now we have 10 balls in the bag, each of which is black or white. Under what color distribution(s) is the entropy of the bag minimized? And under what color distribution(s) is the entropy maximized? Calculate the entropy in each case.
Recall: The entropy of an index set S is a measure of expected surprise from choosing an element from S; that is,

$$H(S) = -\sum_C p_C \log_2(p_C), \text{ where } p_C = \frac{|i \in S : y_i = C|}{|S|}.$$

Solution: The entropy is minimized when, for example, all the balls are black or all the balls are white. In this case the entropy is 0. The entropy is maximized when half the balls are black and half the balls are white, in which case the entropy is $-0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$.

(d) 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.

Solution: The function is strictly concave. Notice that the function $-x \log x$ is strictly concave in [0, 1], and a sum of strictly concave functions is strictly concave.



Significance: (from lecture) Suppose we pick two points on the entropy curve, then draw a line segment connecting them. Because the entropy curve is strictly concave, the interior of the line segment is strictly below the curve. Any point on that segment represents a weighted average of the two entropies for suitable weights. If you unite the two sets into one parent set, the parent set's value p_C is the weighted average of the children's p_c's. Therefore, the point directly above that point on the curve represents the parent's entropy. The information gain is the vertical distance between them. So the information gain is positive unless the two child sets both have exactly the same p_C and lie at the same point on the curve.

On the other hand, for the graph on the right, plotting the % misclassified, if we draw a line segment connecting two points on the curve, the segment might lie entirely on the curve. In that case, uniting the two child sets into one, or splitting the parent set into two, changes neither the total misclassified sample points nor the weighted average of the % misclassified. The bigger problem, though, is that many different splits will get the same weighted average cost; this test doesn't distinguish the quality of different splits well.