CS 189 Introduction to Machine Learning
Spring 2023 Jonathan Shewchuk

DIS8

1 Entropy and Information

In this problem, we try to build intuition as to why entropy of a random variable corresponds to the amount of information that variable transmits. In particular, it determines the number of 0's and 1's needed to "efficiently" encode a random variable.

A coin with bias $b \in (0, 1)$ is flipped until the first head occurs, meaning that each flip gives heads with probability b. Let X denote the number of flips required. Recall that the entropy of a random variable Y is defined as:

$$H(Y) = -\sum_y \mathbb{P}(Y = y) \log(\mathbb{P}(Y = y)).$$

(a) Find the entropy $H(X)$. Assuming the logarithm in the definition of entropy has base 2, then the entropy is measured in *bits*.
Hint: The following expressions might be useful:

$$\sum_{n=0}^{\infty} b^n = \frac{1}{1-b}, \quad \sum_{n=1}^{\infty} nb^n = \frac{b}{(1-b)^2}.$$

Solution: The random variable X follows a geometric distribution with parameter b, and for all $k \in \mathbb{N}$, $\mathbb{P}(X = k) = (1-b)^{k-1}b$. By definition, its entropy (in bits) is:

$$\begin{aligned}
H(X) &= -\sum_{k=1}^{\infty} (1-b)^{k-1} b \log\Big((1-b)^{k-1} b\Big) \\
&= -\sum_{k=1}^{\infty} (1-b)^{k-1} b \left[(k-1)\log(1-b) + \log(b)\right] \\
&= -b\log(b) \sum_{k=1}^{\infty} (1-b)^{k-1} - b\log(1-b) \sum_{k=1}^{\infty} (k-1)(1-b)^{k-1} \\
&= -\frac{b\log(b)}{b} - b\log(1-b)\frac{1-b}{b^2} \\
&= \frac{-b\log(b) - (1-b)\log(1-b)}{b}.
\end{aligned}$$

(b) Let $b = \frac{1}{2}$. Find an "efficient" sequence of yes-no questions of the form, "Is X contained in the set S?", such that X is determined as fast as possible. Compare $H(X)$ to the expected number of asked questions.

Solution: First notice that, if $b = \frac{1}{2}$, $H(X) = 2$. Now we construct a sequence of questions. Encode the answer "yes" with 1 and "no" with 0. First we ask: is $X = 1$? The answer is 1 with probability $1/2$, and 0 with the same probability. Then we ask: is $X = 2$? Conditioned the first question being answered with 0, the answer is 1 again with probability $1/2$. And similarly, at every round k, we ask: is $X = k$? This, of course, implies that the answer was 0 in all previous rounds. Therefore, the expected number of questions is exactly the entropy of $H(X)$, because requiring k questions happens with probability $\left(\frac{1}{2}\right)^k$:

$$\mathbb{E}[\text{no. questions}] = \sum_{k=1}^{\infty} k\left(\frac{1}{2}\right)^k = 2 = H(X).$$

2 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.

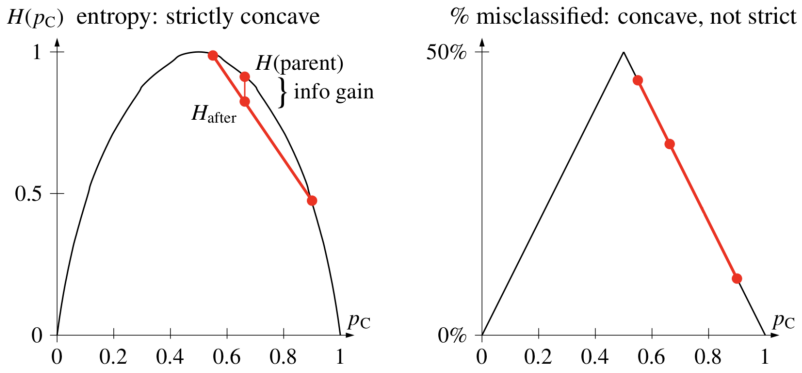
(c) Now we have 10 balls in the bag, each of which is black or white. Under what color distribution(s) is the entropy of the bag minimized? And under what color distribution(s) is the entropy maximized? Calculate the entropy in each case.
Recall: The entropy of an index set S is a measure of expected 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. Note that this is why we can also pick even simpler strictly concave functions (like $H'(p) = p(1-p)$) which will work nearly as well.

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.

3 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**, and S_r is the set of samples remaining in the right subtree after **node**. Intuitively, this is the difference between the entropy of a node's elements if no split is performed and the entropy of the node's elements if a split is performed. Thus, information gain essentially measures the reduction in entropy achieved by the split.

(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

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

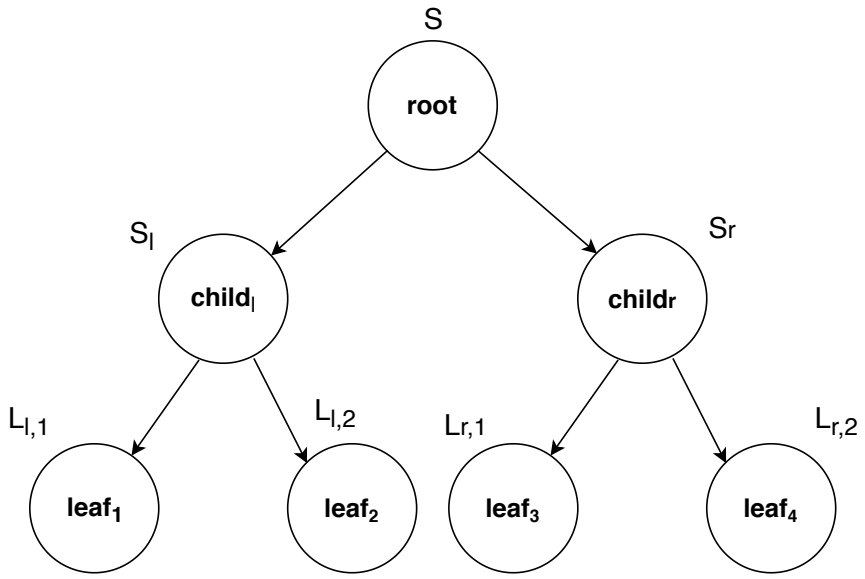
where the first two entries in every sample are features, and the last one is the label. Then, $H(S) = 1$. The first split is done based on the first feature, which gives $S_l = \{(0, 0; 0), (0, 1; 1)\}$ and $S_r = \{(1, 0; 1), (1, 1; 0)\}$; denote the corresponding nodes as $\textbf{child}_l$ and $\textbf{child}_r$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

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

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 is the depth of a decision tree related to overfitting and underfitting?

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, it might underfit. In this case, the decisions are too "coarse".

(d) 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 examples are drawn uniformly from the set of 2^{d+1} possible examples. Calculate the number of samples required before the probability of finding a contradiction in the data reaches $\frac{1}{2}$.
(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

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

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