AI534 — Written Homework Assignment 3 —

1. (Naive Bayes Classifier) (7 pts) Consider the following training set:

A	B	C	Y
0	1	1	0
1	1	1	0
0	0	0	0
1	1	0	1
0	1	0	1
1	0	1	1

(a) (3 pts) Learn a Naive Bayes classifier by estimating all necessary probabilities (there should be 7 independent probabilities to be estimated in total).

Prior:

Y=1	Y=0
$\frac{1}{2}$	$\frac{1}{2}$

class conditional for Y=0:

A=1	A=0	B=1	B=0	C=1	C=0
$\frac{1}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{3}$

class conditional for Y=1:

A=1	A=0	B=1	B=0	C=1	C=0
$\frac{2}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{2}{3}$

(b) (3 pts) Compute the probability $P(Y = 1|A = 1, B = 0, C = 0)$.

$$\begin{aligned} P(Y = 1|(A, B, C) = (1, 0, 0)) &= \frac{\frac{1}{2} * \frac{2}{3} * \frac{1}{3} * \frac{2}{3}}{\frac{1}{2} * \frac{2}{3} * \frac{1}{3} * \frac{2}{3} + \frac{1}{2} * \frac{1}{3} * \frac{1}{3} * \frac{1}{3}} \\ &= \frac{4}{5} \end{aligned}$$

(c) (1 pts) Suppose we know that the three features A, B and C are independent from one another, can we say that the Naive Bayes assumption is valid? (Note that the particular data set is irrelevant for this question). If your answer is yes, please explain why; if you answer is no please give an counter example.

we can say that Naive Bayes assumption is valid under the supposition that the three features A, B and C are independent from one another. According to the lecture slides, the Naive Bayes classifier operates under the assumption that the features are conditionally independent given the class label Y, meaning that:

$$\forall i, j, k, l \; P(A = i, B = j, C = k|Y = l) = P(A = i|Y = l) \cdot P(B = j|Y = l) \cdot P(C = k|Y = l)$$

Denoted as:

$$P(A, B, C|Y) = P(A|Y) \cdot P(B|Y) \cdot P(C|Y)$$

As the features A, B, and C are already independent of each other, implying that they are unconditionally independent, this independence would naturally extend to conditional independence given Y. This means:

$$P(A, B, C|Y) = \frac{P(Y|A, B, C)P(A, B, C)}{P(Y)}$$

$$\begin{aligned} P(A|Y) \cdot P(B|Y) \cdot P(C|Y) &= \frac{P(Y|A)P(A)}{P(Y)} \cdot \frac{P(Y|B)P(B)}{P(Y)} \cdot \frac{P(Y|C)P(C)}{P(Y)} \\ &= \frac{P(Y|A)P(Y|B)P(Y|C) \cdot P(A)P(B)P(C)}{P(Y)} \end{aligned}$$

So, we can guess:

$$P(Y|A,B,C) = P(Y|A)P(Y|B)P(Y|C)$$

$$P(A,B,C) = P(A)P(B)P(C)$$

Thus, these satisfies the Naive Bayes assumption.

2. (Naive Bayes learns linear decision boundary.) (10 pts) Show that the following naive Bayes classifiers learn linear decision boundary $w_0 + w_1 x_1 + w_2 x_2 + ... + w_d x_d = 0$. Express the weights using the corresponding Naive Bayes parameters. Hint: start with the decision boundary defined by $\log \frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} = 0$.

   (a) Bernoulli Naive Bayes model, where features $x_1, x_2, ..., x_d$ are binary indicating the presence/absence of words in the vocabulary.
   We can redefine $log\frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})}$ *as:*

$$log\frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} = log\frac{\frac{P(y=1)P(\mathbf{x}|y=1)}{P(\mathbf{x})}}{\frac{P(y=0)P(\mathbf{x}|y=0)}{P(\mathbf{x})}} = log\frac{P(y=1)P(\mathbf{x}|y=1)}{P(y=0)P(\mathbf{x}|y=0)} = log\frac{P(y=1)}{P(y=0)} + log\frac{P(\mathbf{x}|y=1)}{P(\mathbf{x}|y=0)}$$

   The term $log\frac{P(\mathbf{x}|y=1)}{P(\mathbf{x}|y=0)}$ *can also be redefined as:*

$$log\frac{P(\mathbf{x}|y=1)}{P(\mathbf{x}|y=0)} = log\frac{\prod_i^d P(x_i|y=1)}{\prod_i^d P(x_i|y=0)} = \sum_{i=1}^{d} log\frac{P(x_i|y=1)}{P(x_i|y=0)}$$

   Then, features $x_1, x_2, ...x_d$ *are binary (presence/absence), meaning that:*

$$log\frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} = log\frac{P(y=1)}{P(y=0)} + \sum_{i=1}^{d} x_i log\frac{P(x_i=1|y=1)}{P(x_i=1|y=0)} + \sum_{i=1}^{d}(1-x_i)log\frac{P(x_i=0|y=1)}{P(x_i=0|y=0)}$$

   If feature x_i *is equal to 0 (absence), the weight* w *will be* w_0*. On the other hand, if feature* x_i *is equal to 1 (presence), the weight* w *will be* w_i*. From the equation above, this indicates:*

$$w_0 = log\frac{P(y=1)}{P(y=0)} + \sum_{i=1}^{d} log\frac{P(x_i=0|y=1)}{P(x_i=0|y=0)}$$

$$\sum_{i=1}^{d} w_i = \sum_{i=1}^{d}(log\frac{P(x_i=1|y=1)}{P(x_i=1|y=0)} - log\frac{P(x_i=0|y=1)}{P(x_i=0|y=0)})$$

   Thus, we conclude:

$$w_0 + \sum_{i=1}^{d} w_i x_i$$

$$= log\frac{P(y=1)}{P(y=0)} + \sum_{i=1}^{d} log\frac{P(x_i=0|y=1)}{P(x_i=0|y=0)} + \sum_{i=1}^{d}(log\frac{P(x_i=1|y=1)}{P(x_i=1|y=0)} - log\frac{P(x_i=0|y=1)}{P(x_i=0|y=0)})x_i$$

   (b) Multinomial Naive Bayes Model, where $x_1, ..., x_d$ representing counts of words $w_1, ..., w_d$ in the vocabulary. Express the weights using the Naive Bayes parameters: the class priors $P(y=1), P(y=0)$ and the class conditionals: $p(w_i|y=1)\ and\ p(w_i|y=0)$
   Similarly, we can redefine $log\frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})}$ *as:*

$$log\frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} = log\frac{P(y=1)}{P(y=0)} + log\frac{P(\mathbf{x}|y=1)}{P(\mathbf{x}|y=0)}$$

The term $log\frac{P(\mathbf{x}|y=1)}{P(\mathbf{x}|y=0)}$ *can also be redefined as:*

$$log\frac{P(\mathbf{x}|y=1)}{P(\mathbf{x}|y=0)} = log\frac{\prod_i^d P(w_i|y=1)^{x_i}}{\prod_i^d P(w_i|y=0)^{x_i}} = \sum_{i=1}^{d} x_i log\frac{P(w_i|y=1)}{P(w_i|y=0)}$$

Then, we get:

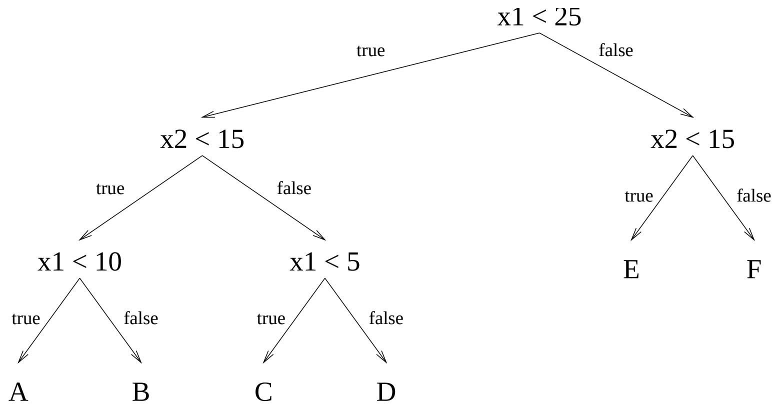
$$log\frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} = log\frac{P(y=1)}{P(y=0)} + \sum_{i=1}^{d} x_i log\frac{P(w_i|y=1)}{P(w_i|y=0)}$$

Thus, we conclude:

$$w_0 + \sum_{i=1}^{d} w_i x_i$$

$$= log\frac{P(y=1)}{P(y=0)} + \sum_{i=1}^{d} log\frac{P(w_i|y=1)}{P(w_i|y=0)} x_i$$

3. (6 pts) Consider the following decision tree:



(a) (2 pts) Draw the decision boundaries defined by this tree. Each leaf of the tree is labeled with a letter. Write this letter in the corresponding region of input space.

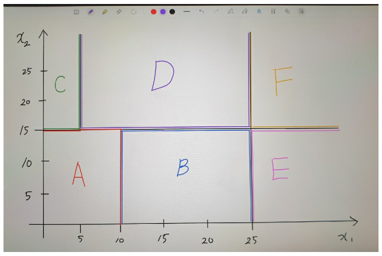


Figure 1: Decision boundaries

(b) (2 pts) Give another decision tree that is syntactically different but defines the same decision boundaries. This demonstrates that the space of decision trees is syntactically redundant.

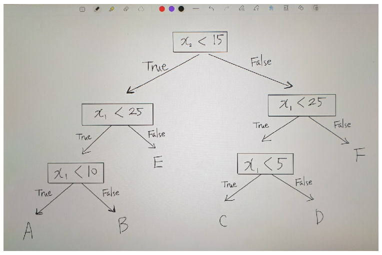


Figure 2: Syntactically different decision tree

(c) (2pts) How does this redundancy influence learning (does it make it easier or harder to find an accurate tree)?

The redundant space of decision trees can make it harder to find an optimal, accurate tree. Multiple syntactically different trees that represent the same decision boundaries increase the search space learning algorithm finds.

4. (6 pts) In the basic decision tree algorithm (assuming we always create binary splits), we choose the feature/value pair with the maximum information gain as the test to use at each internal node of the decision tree. Suppose we modified the algorithm to choose at random from among those feature/value combinations that had non-zero mutual information, and we kept all other parts of the algorithm unchanged.

(a) (2 pts) What is the maximum number of leaf nodes that such a decision tree could contain if it were trained on m training examples?

The maximum number of leaf nodes is m since each internal node splits one subset of data into two in a binary tree. This means that the maximum number of leaf nodes occurs when every training example is separated into its own leaf.

(b) (2 pts) What is the maximum number of leaf nodes that a decision tree could contain if it were trained on m training examples using the original maximum mutual information version of the algorithm? Is it bigger, smaller, or the same as your answer to (a)?

The maximum number of leaf nodes when using maximum mutual information version of algorithm is as same as the answer to (a), which is m. The maximum mutual information means the tree chooses splitting strategy that results in the greatest reduction in uncertainty about the target variable. However, even the algorithm uses optimal splitting strategy, the tree could split until each example is isolated in the worst-case scenario (m leaf nodes). Thus, splitting method cannot determine the number of leaf nodes.

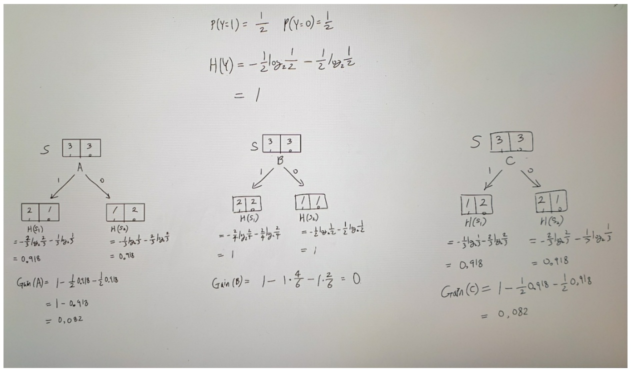
(c) (2 pts)How do you think this change (using random splits vs. maximum information mutual information splits) would affect the testing accuracy of the decision trees produced on average? Why?

The change using random splits would reduce testing accuracy on average compared to using maximum information gain splits since random splits are less effective at creating well-separated, meaningful partitions of data. Random splits do not prioritize reducing uncertainty as much as possible, causing to make the tree too shallow or deep instead. On the other hand, maximum information gain splits focus on creating well-separated subsets of data at each step, reducing entropy efficiently.

5. (8 pts) Consider the following training set:

A	B	C	Y
0	1	1	0
1	1	1	0
0	0	0	0
1	1	0	1
0	1	0	1
1	0	1	1

Learn a decision tree from the training set shown above using the information gain criterion. Show your steps, including the calculation of information gain (you can skip $H(y)$ and just compute $H(y|\mathbf{x})$) of different candidate tests. You can randomly break ties (or better, choose the one that give you smaller tree if you do a bit look ahead for this problem).



P(Y=1) = 1/2   P(Y=0) = 1/2
H(Y) = -1/2 log₂ 1/2 - 1/2 log₂ 1/2
= 1
S 3 3
A
1
0
2 1
1 2
H(S₁)
= -2/3 log₂ 2/3 - 1/3 log₂ 1/3
= 0.918
H(S₂)
= -1/3 log₂ 1/3 - 2/3 log₂ 2/3
= 0.918
Gain (A) = 1 - 1/2 0.918 - 1/2 0.918
= 1 - 0.918
= 0,082
S 3 3
B
1
0
2 2
1 1
H(S₁)
= -2/4 log₂ 2/4 - 2/4 log₂ 2/4
= 1
H(S₂)
= -1/2 log₂ 1/2 - 1/2 log₂ 1/2
= 1
Gain (B) = 1 - 1·4/6 - 1·2/6 = 0
S 3 3
C
1
0
1 2
2 1
H(S₁)
= -1/3 log₂ 1/3 - 2/3 log₂ 2/3
= 0,918
H(S₂)
= -2/3 log₂ 2/3 - 1/3 log₂ 1/3
= 0,918
Gain (C) = 1 - 1/2 0,918 - 1/2 0,918
= 0,082

Figure 3: Calculation for information gain

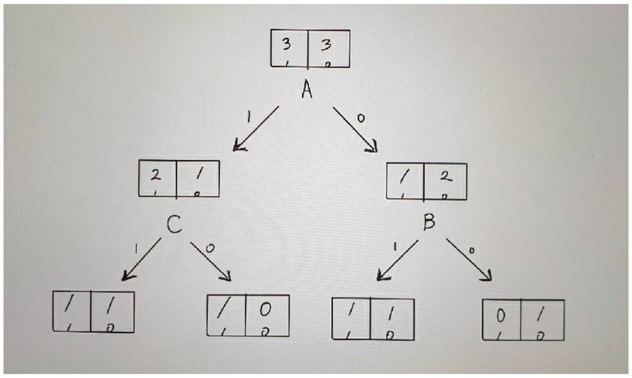


Figure 4: Decision tree based on information gain

6. (8pts) Prove that

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$$

.

Hint: you should start with the definition $H(X,Y) = -\sum_{x,y} P(x,y)\log P(x,y)$. Here we use X, Y to denote the random variables and x, y denote the values X and Y take, $P(x,y)$ is a short hand notation denoting $P(X = x, Y = y)$.

We start with the definition:

$$H(X,Y) = -\sum_{x,y} P(x,y) log P(x,y)$$

By the definition of conditional entropy:

$$H(Y|X) = -\sum_{x,y} P(x,y) logP(y|x)$$

By the definition of conditional probability:

$$P(y|x) = \frac{P(x,y)}{P(x)}$$

$$\Rightarrow logP(x|y) = logP(x,y) - logP(x)$$

Applying these to $H(Y|X)$, We get:

$$H(Y|X) = -\sum_{x,y} P(x,y)[logP(x,y) - logP(x)]$$

$$= -\sum_{x,y} P(x,y) logP(x,y) + \sum_{x,y} P(x,y) logP(x)$$

Based on the marginalization rule in probability theory, the equation $\sum_y P(x,y) = P(x)$ can be applied into the second term in $H(Y|X)$. So, we get:

$$\sum_{x,y} P(x,y) logP(x) = \sum_{x} P(x) logP(x) = -H(x)$$

Finally:

$$H(Y|X) = H(X,Y) - H(X)$$

$$\Rightarrow H(X,Y) = H(X) + H(Y|X)$$

Similarly, $H(X|Y)$ can be solved:

$$H(X|Y) = H(X,Y) - H(Y)$$

$$\Rightarrow H(X,Y) = H(Y) + H(X|Y)$$

Thus, we conclude:

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$$