CS 4341: Homework 3

Adam Camilli (aocamilli@wpi.edu)

February 25, 2018

Decision Tree:

1. **Problem 18.6:** Consider the following data set comprised of three binary input attributes (A_1, A_2, A_3) and one binary output:

Example	A_1	A_2	A_3	Output y
$\mathbf{x}_1$	1	0	0	0
$\mathbf{x}_2$	1	0	1	0
$\mathbf{x}_3$	0	1	0	0
$\mathbf{x}_4$	1	1	1	1
$\mathbf{x}_5$	1	1	0	1

Use the algorithm in Figure 18.5 (p. 702) to learn a decision tree for these data. Show the computations made to determine the attribute to split at each node.

function DECISION-TREE-LEARNING(*examples*, *attributes*, *parent_examples*) **returns** a tree

if *examples* is empty **then return** PLURALITY-VALUE(*parent_examples*)
else if all *examples* have the same classification **then return** the classification
else if *attributes* is empty **then return** PLURALITY-VALUE(*examples*)
else
  $A \leftarrow \text{argmax}_{a \in attributes}$ IMPORTANCE(a, *examples*)
  tree $\leftarrow$ a new decision tree with root test A
  for each value v_k of A **do**
    $exs \leftarrow \{e : e \in examples \text{ and } e.A = v_k\}$
    subtree $\leftarrow$ DECISION-TREE-LEARNING(exs, *attributes* $- A$, *examples*)
    add a branch to *tree* with label $(A = v_k)$ and subtree *subtree*
  return *tree*

Figure 18.5 The decision-tree learning algorithm. The function IMPORTANCE is described in Section 18.3.4. The function PLURALITY-VALUE selects the most common output value among a set of examples, breaking ties randomly.

Calculate the value of the remainder for each attribute in the first split, since provided info is before that.

$$A_1 = \frac{4}{5} \cdot \left(\frac{-2}{4} \log_2 \left(\frac{2}{4} \right) \right) + \frac{1}{5} \left(-0 - \frac{1}{1} \log_2 \left(\frac{1}{1} \right) \right) = 0.8$$

$$A_2 = \frac{3}{5} \cdot \left(\frac{-2}{3} \log_2 \left(\frac{2}{3} \right) \right) + \frac{2}{5} \left(-0 - \frac{2}{2} \log_2 \left(\frac{2}{2} \right) \right) \approx 0.551$$

$$A_3 = \frac{2}{5} \cdot \left(\frac{-1}{2} \log_2 \left(\frac{1}{2} \right) \right) + \frac{3}{5} \left(\frac{-1}{3} \log_2 \left(\frac{1}{3} \right) - \frac{2}{3} \log_2 \left(\frac{2}{3} \right) \right) \approx 0.951$$

Since A_2 is the minimum after the first split, the values for $A_2 = 0$ are classified as zero and so X_1, X_2 are ignored. The remaining X_3, X_4, X_5 are considered, all of which have $A_2 = 1$. Perform split on A_2, and then calculate remainders for A_1, A_3:

$$A_1 = \frac{2}{3} \cdot \left(\frac{-1}{1} \log_2 \left(\frac{1}{1} \right) - 0 \right) + \frac{1}{3} \left(-0 - \frac{1}{1} \log_2 \left(\frac{1}{1} \right) \right) = 0$$

$$A_3 = \frac{1}{3} \cdot \left(\frac{-1}{1} \log_2 \left(\frac{1}{1} \right) - 0 \right) + \frac{2}{3} \left(\frac{-1}{2} \log_2 \left(\frac{1}{2} \right) - \frac{1}{2} \log_2 \left(\frac{1}{2} \right) \right) \approx 0.667$$

Therefore we select A_1 for the split because all remaining examples are correct.

2. Consider the following *Play Tennis* dataset (adapted from: Quinlan, "Induction of Decision Trees", Machine Learning, 1986):

ATTRIBUTES:	POSSIBLE VALUES:
Outlook	{Sunny, Rain, Overcast}
Temperature	{Hot, Mild, Cool}
Humidity	{High, Normal}
Wind	{Strong, Weak}
PlayTennis	{Yes, No} ← classification target

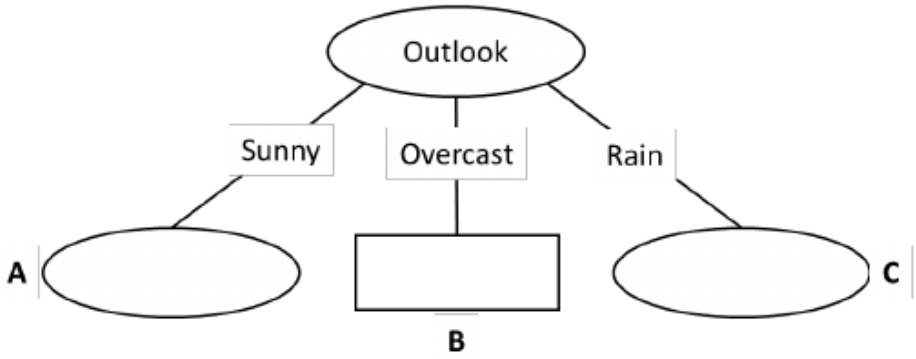
ID	**Outlook**	**Temperature**	**Humidity**	**Wind**	**PlayTennis**
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Sunny	Hot	Normal	Weak	Yes
4	Sunny	Mild	Normal	Strong	Yes
5	Sunny	Mild	Normal	Weak	No
6	Rain	Mild	High	Strong	No
7	Rain	Mild	Normal	Weak	Yes
8	Rain	Cool	Normal	Weak	Yes
9	Rain	Mild	Normal	Strong	Yes
10	Rain	Cool	Normal	Strong	No
11	Overcast	Hot	High	Weak	Yes
12	Overcast	Cool	High	Strong	Yes
13	Overcast	Mild	High	Strong	Yes
14	Overcast	Hot	Notmal	Weak	Yes

The machine learning task is to predict whether to play tennis or not, based on the data about the weather. Answer the questions in the following page.

(a) Which of the major machine learning categories (supervised, unsupervised, or reinforcement) does this problem fall under? Explain.

Since this problem already has a well-defined model with a set of variables and a domain for each, a given set of states, and does not involve any sort of rewards or incentives, it is classified as a supervised learning problem.

Suppose we are in the middle of inducing the decision tree. The current state of the decision tree is given below:



(b) What should the tree output be in the leaf box labelled B?

Since being Overcast does not preclude PlayingTennis, consider another attribute. The next one with the smallest domain are Humidity and Wind.

(c) Which data instances should be considered in node A? Write down the relevant instance ID numbers from the table.

Instances which are Overcast: 11,12,13,14

(d) Calculate the **entropy** at node A for the **Humidity** attribute. Show all the steps of your calculation. For your convenience, the logarithm in base 2 of selected values are provided.

x	1/2	1/3	2/3	1/4	3/4	1
$\log_2(x)$	-1	-1.6	-0.6	-2	-0.4	0

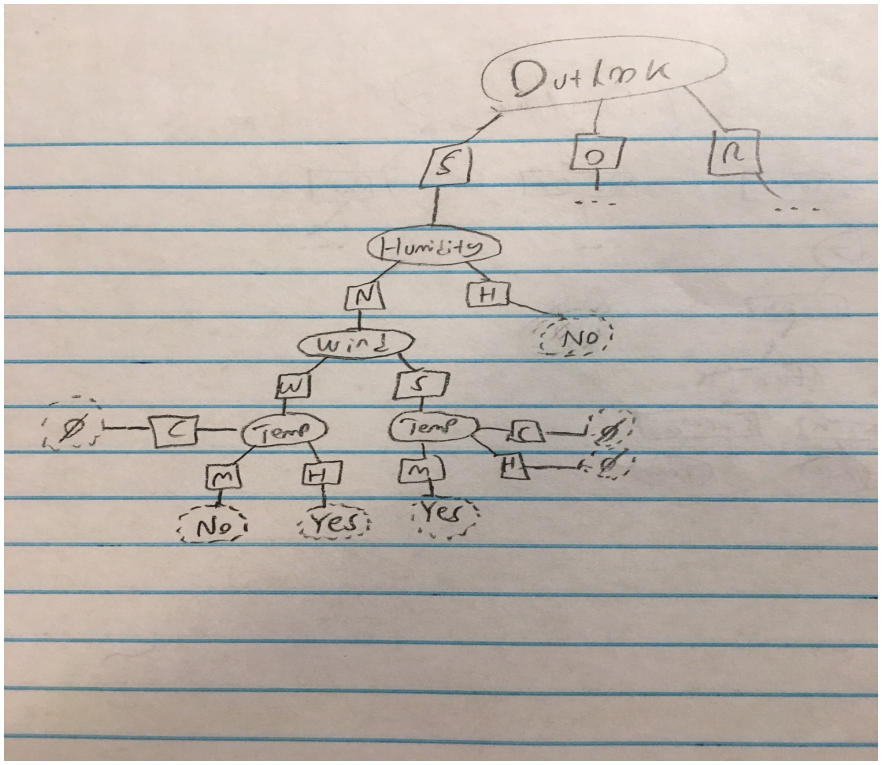
$$\text{Entropy} = -\sum_i P_i \log_2 P_i$$

$$\text{E(Sunny) + E(Sunny,High,Normal)} = -\frac{2}{5}(-1.32) - \frac{3}{5}(-0.74) + 0 + \frac{3}{3}(0) \approx 0.972$$

(e) We have already calculated the entropies of Temperature and Wind at node A, and they are both 0.96. Based on these values, and based on the result you obtained previously, which attribute should be used to split node A? If needed, break any ties arbitrarily.

All three have equal probabilities (two values, split two and three) at node A. Since Humidity and Wind have the smallest domains, arbitrarily choose between them (i.e. Wind affects tennis more).

(f) Write your chosen attribute in the blank space of node A in the tree given in the previous page. Now extend the appropriate number of branches from node A and label them properly. For each branch that leads to a leaf, draw the leaf in the tree on the previous page and mark it with the output that the leaf should produce. Show your work on the tree on the previous page.



Naïve Bayes

3. Use the Naive Bayes network in the class handout to predict the Risk of the following data instance. Show your work.

Credit History = bad; Debt = high; Collateral = adequate; and Income = ¿35.

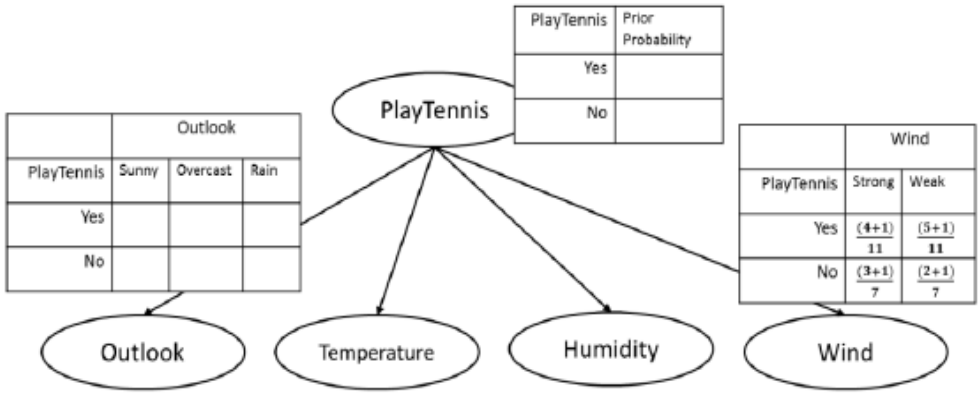
$$v = low : P(low) \cdot P(bad|low) \cdot P(high|low) \cdot P(adequate|low) \cdot P(> 35|low)$$
$$v = \frac{3}{17} \cdot \frac{1}{8} \cdot \frac{3}{7} \cdot \frac{3}{7} \cdot \frac{6}{8} \cdot = \frac{162}{53312} \approx 0.003034$$
$$v = mod : P(mod) \cdot P(bad|mod) \cdot P(high|mod) \cdot P(adequate|mod) \cdot P(> 35|mod)$$
$$v = \frac{5}{17} \cdot \frac{3}{7} \cdot \frac{2}{6} \cdot \frac{2}{6} \cdot \frac{2}{7} \cdot = \frac{120}{29988} \approx 0.004002$$
$$v = high : P(high) \cdot P(bad|high) \cdot P(high|high) \cdot P(adequate|high) \cdot P(> 35|high)$$
$$v = \frac{6}{17} \cdot \frac{3}{8} \cdot \frac{5}{7} \cdot \frac{1}{7} \cdot \frac{1}{8} \cdot = \frac{90}{53312} \approx 0.001689$$

Hence, the predicted value is Risk=moderate

4. Consider the same PlayTennis dataset of Problem 2, which is reproduced here for your convenience (though the data instances appear in a different order):

New ID	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	Normal	Weak	Yes
2	Sunny	Mild	Normal	Strong	Yes
3	Rain	Mild	Normal	Weak	Yes
4	Rain	Cool	Normal	Weak	Yes
5	Rain	Mild	Normal	Strong	Yes
6	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	High	Strong	Yes
8	Overcast	Mild	High	Strong	Yes
9	Overcast	Hot	Normal	Weak	Yes
10	Sunny	Hot	High	Weak	No
11	Sunny	Hot	High	Strong	No
12	Sunny	Mild	Normal	Weak	No
13	Rain	Mild	High	Strong	No
14	Rain	Cool	Normal	Strong	No

(a) From the dataset above, the following Naïve Bayes model is built. Your job is to fill out the likelihood tables given below. Some tables have already been filled for your convenience. Remember that 1 is added to all the counts to avoid the problem of having a probability that is equal to 0 (Laplace smoothing).



	Temperature		
PlayTennis	Hot	Mild	Cool
Yes	$\frac{3+1}{12}$	$\frac{4+1}{12}$	$\frac{2+1}{12}$
No	$\frac{2+1}{8}$	$\frac{2+1}{8}$	$\frac{1+1}{8}$

	Humidity	
PlayTennis	High	Normal
Yes		
No		

	Outlook		
PlayTennis	Sunny	Outcast	Rain
Yes	$\frac{2+1}{5}$	$\frac{4+1}{4}$	$\frac{2+1}{5}$
No	$\frac{3+1}{5}$	$\frac{0+1}{4}$	$\frac{3+1}{5}$

	Humidity	
PlayTennis	High	Normal
Yes	$\frac{3+1}{6}$	$\frac{6+1}{8}$
No	$\frac{3+1}{6}$	$\frac{2+1}{8}$

PlayTennis	Prior Probability
Yes	$\frac{9+1}{15}$
No	$\frac{6+1}{15}$

(b) Use the Naïve Bayes model constructed in the previous page to classify the following new instance.

That is, determine which of the two values of PlayTennis (Yes or No) has the highest probability given the outlook, temperature, humidity and wind values of the given instance. **Show all your work and explain your answer.**

Outlook	Temperature	Humidity	Wind	PlayTennis
Sunny	Hot	Normal	Strong	?

$$v = No : P(No) \cdot P(Sunny|No) \cdot P(Hot|No) \cdot P(Normal|No) \cdot P(Strong|No)$$
$$v = \frac{7}{15} \cdot \frac{4}{5} \cdot \frac{3}{8} \cdot \frac{4}{6} \cdot \frac{4}{7} \cdot = \frac{1344}{25200} = \frac{4}{75} \approx 0.0533$$
$$v = Yes : P(Yes) \cdot P(Sunny|Yes) \cdot P(Hot|Yes) \cdot P(Normal|Yes) \cdot P(Strong|Yes)$$
$$v = \frac{10}{15} \cdot \frac{3}{5} \cdot \frac{7}{8} \cdot \frac{4}{6} \cdot \frac{5}{7} \cdot = \frac{4200}{25200} = \frac{1}{6} \approx 0.1667$$

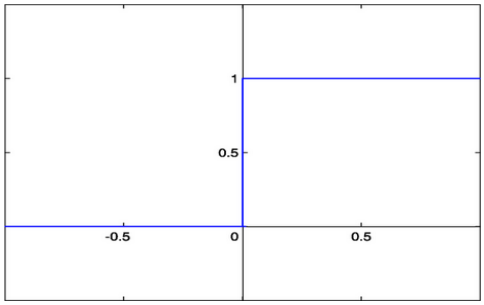
Yes has the highest probability.

Artificial Neural Networks

5. Investigate and explain the differences among the following types of "units" (i.e., perceptrons) in terms of the activation functions that they use. In addition, provide a graphical depiction of these activation functions.

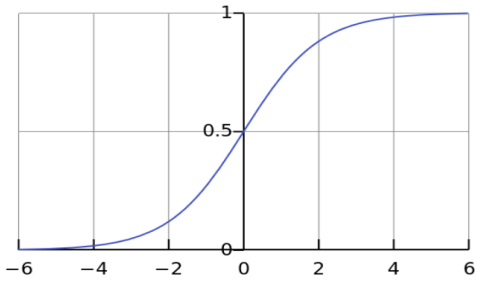
   - Step

Most often used by the initial perceptron, useful for binary classification schemes or feature identifiers.



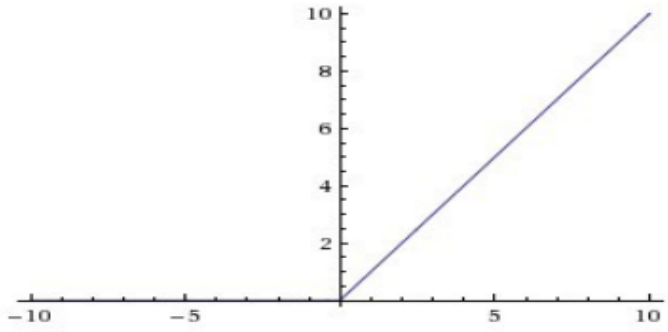
   - Sigmoid

In order to stack layers and solve problems of non-trivial complexity, non-linearity must be introduced to a model. The sigmoid function is also used for classification schemes of real-world models that are not linear in nature. Since small changes in the parameters (X-values) propagate to huge changes in Y-values, the function usually brings activations to either side of the curve. Finally, the function is bound between (0,1), meaning no input will blow up the activations.



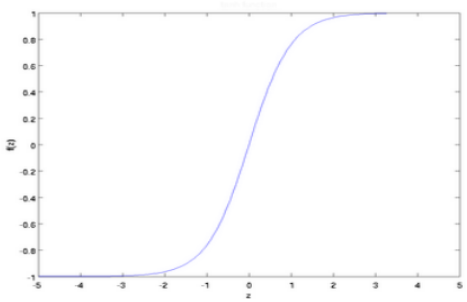
- ReLU

Again, ReLU is non linear, and so can be applied to real world scenarios. ReLU is used chiefly as an approximator, since combinations of it can approximate any function. It is not bound, and so activations can be blown up. Unlike sigmoid, however, the activations are sparse, and so most neurons in a network might not need to fire to get an output. This can be much more efficient with random initial inputs.



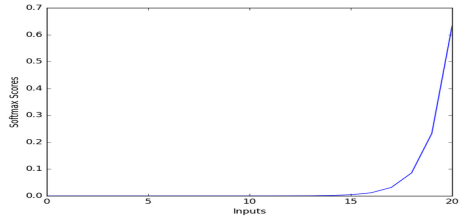
- Tanh

Tanh is essentially a scaled sigmoid function, used for the same purpose of classification. Due to its stronger gradient (steeper derivatives), tanh is even quicker to output towards 1 or 0.



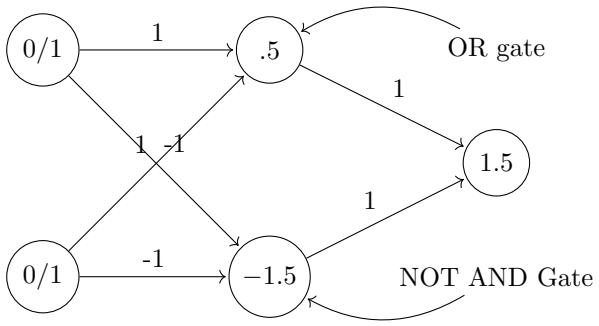
- Softmax

Softmax is often used in the final layer of a neural network classifier. It is used in multiple classification, since sigmoid cannot handle it. In addition to being bound between 0 and 1, the sum of all its outputs (probabilities) also approaches 1.



6. Problem **18.19** Construct by hand a neural network that computes the XOR function of two inputs. Be sure to specify what sort of units you are using.

Since XOR is essentially an OR with the AND case removed, it will be easiest to construct using steps of function units. An XOR function will have its network designed with OR, and the AND case will be computed with a hidden layer:



7. Problem **18.22 (a, c)** from textbook [Here h = No. of nodes in hidden layer]: Suppose you had a neural network with linear activation functions. That is, for each unit the output is some constant c times the weighted sum of the inputs.

   (a) Assume that the network has one hidden layer. For a given assignment to the weights **w**, write down equations for the value of the units in the output layer as a function of **w** and the input layer **x**, without any explicit mention of the output of the hidden layer. Show that there is a network with no hidden units that computes the same function.

A neural network with a linear activation function has an output unit that is essentially c times the input weighted sums. Therefore, if the activation function is $g(x) + d$, we are considering for each node only c_i and d_i.
If the network has a hidden layer $\mathbf{w}$ as weight, $\mathbf{x}$ as input layer without considering the output of the hidden layer. To show that there is a network without hidden units, we first suppose the output of the hidden layer is:

$$HL_j = g\Big(\sum_k w_{k,j} I_k\Big) = c\sum_k w_{k,j} I_k + d$$

Therefore the final output is:

$$O = g\Big(\sum_j w_{j,i} HL_j\Big) = c\Big(\sum_j w_{j,i}\big(\sum_k w_{k,j} I_k + d\big)\Big) + d$$

The same function can be computed as a two-layered network using a one layer perceptron with the weights

$$w_{k,j} = \sum_j w_{k,j} w_{j,i}$$

with an activation function

$$g(x) = c^2 x + d\big(1 + c\sum_j w_{j,i}\big)$$

(b) ...

(c) Suppose a network with one hidden layer and linear activation functions has n input and output nodes and h hidden nodes. What effect does the transformation in part(a) to a network with no hidden layers have on the total number of weights? Discuss in particular the case $h << n$.

This network with n input and h hidden nodes will have $2hn$ weights, whereas the reduced network has n^2 weights. When $h << n$, the original network will consist of fewer weights and so represent the input and output maps much more efficiently. Thus, a network is faster than a reduced one, which is one of the many disadvantages of using a linear activation function.

Evaluating Machine Learning Models

8. Assume that the following confusion matrix was obtained from using a machine learning model to predict the classification of 24 test instances, where the target attribute is Risk, with possible values low, moderate, and high:

```
a  b  c <-- classified as
4  0  1 | a = low
0  1  3 | b = moderate
1  2 12 | c = high
```

For each of the following metrics used in project 3, provide a formula that defines the metric, and calculate its value based on the confusion matrix above. Show your work.

- The model's classification accuracy.

Examining the matrix, we can see there were $4 + 1 + 12 = 17$ true positives (TP), and since the model does not involve rejection there are 17 true negatives (TN) as well. There are 1+3+1+2 = 7 false positives (FP), and again since there are no rejections this is the number of false negatives as well (FN).

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{17 + 17}{17 + 17 + 7 + 7} \approx 70.83\%$$

- The model's classification error (= 100% - model's classification accuracy).

$$CError = 100\% - 70.83\% = 29.167\%$$

- The model's precision for class=low.

$$PRE = \frac{TP}{TP + FP} = \frac{4}{4 + 1} = 80\%$$

- The model's recall for class=moderate.

$$REC = \frac{TP}{TP + FN} = \frac{12}{12 + 3} = 80\%$$