Assignment 2 EECS 6412

Daniel Marchena Parreira

2018/10/16

Question 1

To find all sequential patterns that do not only satisfy a minimum support min-sup, but also start with {a} and end with {b}, we can use the PrefixSpan algorithm multiple times. Having said that, we should first run PrefixSpan to find all sequential patterns with {a} considering a list of elements ordered alphabetically:

Example:

Sequences: <a(abc)(ac)d(cf)>, <a(abc)(a)d(cf)>, <(ad)a(bc)(ae)d>, <(ef)(ab)(df)cb>, <eg(af)cbc>
min-sup-count = 2
{a} = (a)
{b} = (d)

- **Step 1** Find length-1 sequential patterns
  - In our case we will look for a single item <a>
- **Step 2** Divide search space. The complete set of frequent sequences can be partitioned into 1 subset:
  - The ones having prefix <a>
- **Step 3** Run <a>-projected database:

  <(abc)(ac)d(cf)>, <(abc)(a)d(cf)>, <(bc)(ae)d>

  It is important to remember that we are only looking for itemsets that contain a single item, in this case "a".
- **Step 4** To filter by ending <b>-sequential patterns, we will analyze the sequences from the <a>-projected database from end to beginning by running Step 1 to 3 again. The output will be sequences starting with {a} and ending with {b}.

Question 2

a)
Number of distinct classes (m) = 2 (either + or -)
C1 = + (4 tuples)
C2 = - (6 tuples)

A (root) node N is created for the tuples in D. To find the splitting criterion for these tuples, we must compute the information gain of each attribute.

- **Step 1** We first use the database to compute the expected information needed to classify a tuple in D:

$$Info(D) = -\frac{4}{10}\log_2\left(\frac{4}{10}\right) - \frac{6}{10}\log_2\left(\frac{6}{10}\right) = 0.970_{bits}$$

- **Step 2** Next, we need to compute the expected information requirement for each attribute. Lets start with the attribute A. We need to look at the distribution of + and - tuples for each category of A (T or F):

$$Info_A(D) = \frac{7}{10}\left(-\frac{4}{7}\log_2\frac{4}{7} - \frac{3}{7}\log_2\frac{3}{7}\right) + \frac{3}{10}\left(-\frac{3}{3}\log_2\frac{3}{3}\right) = 0.689_{bits}$$

- **Step 3** Calculate the gain for such a partitioning would be:

$$Gain(A) = Info(D) - Info_A(D) = 0.970 - 0.689 = 0.281 bits$$

- **Step 4** We need to look at the distribution of + and - tuples for each category of B (T or F):

$$Info_B(D) = \frac{4}{10}\left(-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}\right) + \frac{6}{10}\left(-\frac{1}{6}\log_2\frac{1}{6} - \frac{5}{6}\log_2\frac{5}{6}\right) = 0.714_{bits}$$

- **Step 5** Calculate the gain for such a partitioning would be:

$$Gain(B) = Info(D) - Info_A(D) = 0.970 - 0.714 = 0.256 bits$$

  Because A has the highest information gain among the attributes, it is selected as the splitting attribute. Node N is labeled with A, and branches are grown for each of the attributes values.

b)
The Gini index is used in CART. Using the notation previously described, the Gini index measures the impurity of D.

- **Step 1** We first use the Database for the Gini index to compute the impurity of D:

$$Gini(D) = 1 - \left(\frac{4}{10}\right)^2 - \left(\frac{6}{10}\right)^2 = 0.480$$

- **Step 2** To find the splitting criterion for the tuples in D, we need to compute the Gini index for each attribute. Let's start with A:

$$\begin{aligned} & Gini_{A\in\{T\}}(D) = \frac{7}{10}Gini(D_1) + \frac{3}{10}Gini(D_2) \\ & = \frac{7}{10}(1 - \left(\frac{4}{7}\right)^2 - \left(\frac{3}{7}\right)^2) + \frac{3}{10}(1 - \left(\frac{3}{3}\right)^2) = 0.342 \\ & = Gini_{A\in\{F\}}(D) \end{aligned} \tag{1}$$

- **Step 3** Let's compute the Gini index for B:

$$\begin{aligned} & Gini_{B\in\{T\}}(D) = \frac{4}{10}Gini(D_1) + \frac{6}{10}Gini(D_2) \\ & = \frac{4}{10}(1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2) + \frac{6}{10}(1 - \left(\frac{1}{6}\right)^2 - \left(\frac{5}{6}\right)^2) = 0.316 \\ & = Gini_{B\in\{F\}}(D) \end{aligned} \tag{2}$$

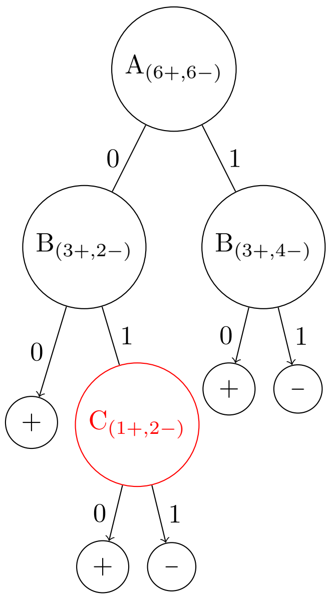
- **Step 4** The attribute B and splitting subset {T} therefore give the minimum Gini index overall, with a reduction in impurity of 0.480 - 0.316 = 0.164. The binary split "$B \in \{T?\}$" results in the maximum reduction in impurity of the tuples in D and is returned as the splitting criterion. Node N is labeled with the criterion, two branches are grown from it, and the tuples are partitioned accordingly.

Question 3

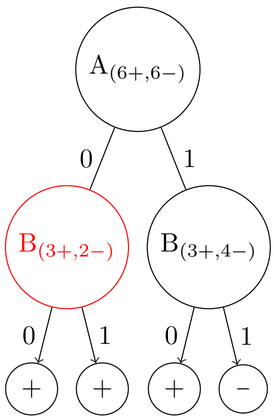
1)
From the 5 tuples available in the pruning set, the decision three got 2 tuples correctly classified and 3 incorrectly classified. That leaves us to an error rate of 60%.

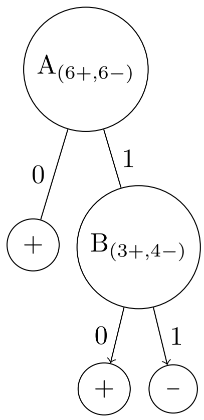
2)

- **Step 1** Consider each of the internal non-root nodes in the tree to be candidates for pruning
- **Step 2** Prune a node by removing subtree rooted at this node, making it a leaf node, and assigning it the most common classification of the training examples affiliated with this node



- **Step 3** We will start from the bottom of the tree and prune $C_{(1+,2-)}$ making it a leaf node for +. However, that will make both outcomes of $B_{(3+,2-)}$ to be equal to +, so as a matter visual simplicity we will also prune that to simplify our tree.





This will leave us with an error rate of 40% by running the pruning set. That gives us a reduction of the error rate by 20% compared to our unpruned tree. Now we can greedily remove those nodes from our decision tree.

Question 4

Attached as requested