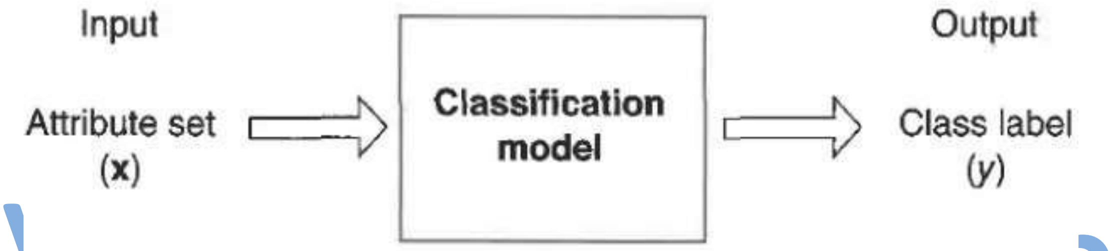
MODULE-4 CLASSIFICATION

4.1 Introduction

Classification: Definition

Classification, which is the task of assigning objects to one of several predefined categories. Given a collection of records (training set),Each record contains a set of attributes, one of the attributes is the class. Find a model for class attribute as a function of the values of other attributes. The input data for a classification task is a collection of records. Each record,



Goal: previously unseen records should be assigned a class as accurately as possible.
A test set is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

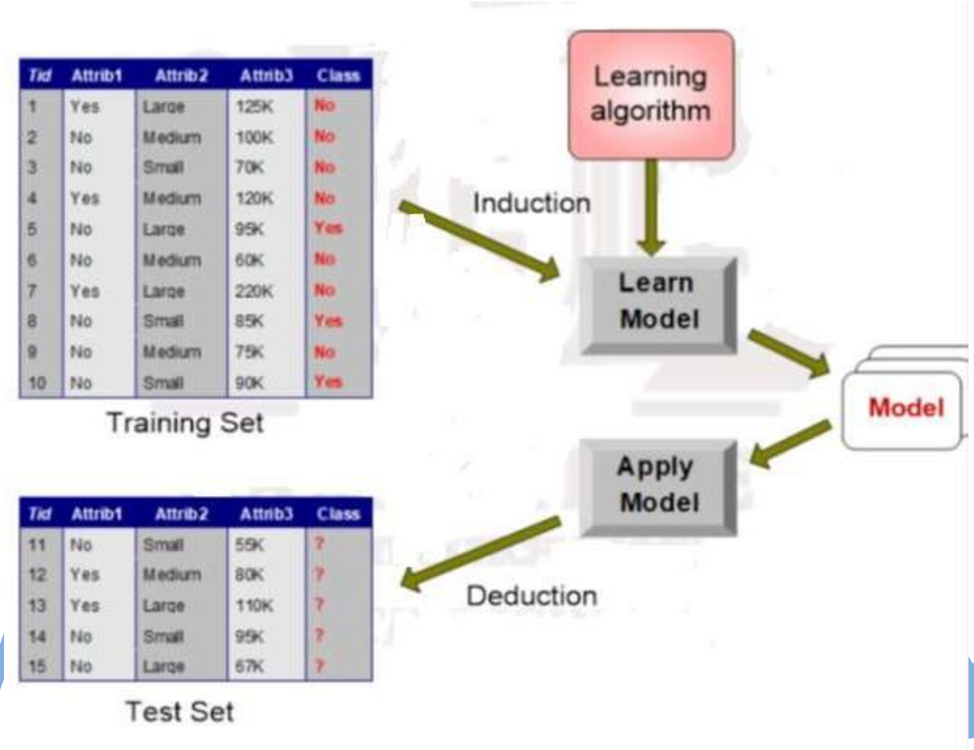
Applications:
Examples

- Detecting spam email messages based upon the messageheader and content.
- Categorizing cells as malignant or benign based upon the results of MRI scans.
- Classifying galaxies based upon their shapes.
- Categorizing news stories as finance, weather, entertainment, sports, etc
- Classifying credit card transactions as legitimate or fraudulent.

General Approach to Solving a Classification

- A classification technique (or classifier) is a systematic approach to building
- classification models from an input data set.

- Each technique employs a learning algorithm to identify a model that best fits the relationship between the attribute set and class label of the input data.
- The model generated by a learning algorithm should both fit the input data well and correctly predict the class labels of records it has never seen before.
- Therefore, a key objective of the learning algorithm is to build models with good generalization capability; i.e., models that accurately predict the class labels of previously unknown records



Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

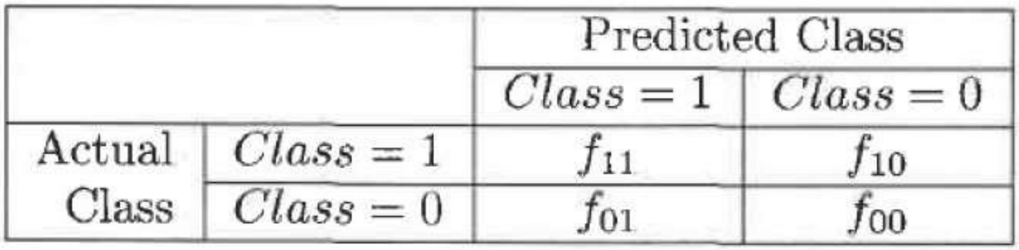
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Evaluation of the performance of a classification model:

is based on thecounts of test records correctly and incorrectly predicted by the model. Thesecounts are tabulated in a table known as a confusion matrix.

Table 4.2. Confusion matrix for a 2-class problem.



		Predicted Class	
		$Class = 1$	$Class = 0$
Actual Class	$Class = 1$	f_{11}	f_{10}
	$Class = 0$	f_{01}	f_{00}

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}. \quad (4.1)$$

Equivalently, the performance of a model can be expressed in terms of its **error rate**, which is given by the following equation:

$$\text{Error rate} = \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}. \quad (4.2)$$

Most classification algorithms seek models that attain the highest accuracy, or equivalently, the lowest error rate when applied to the test set.

Classification Techniques:

- Decision Tree based Methods
- Rule-based Methods
- Memory based reasoning
- Neural Networks
- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines

4.2 Decision Tree Induction

The tree has three types of nodes:

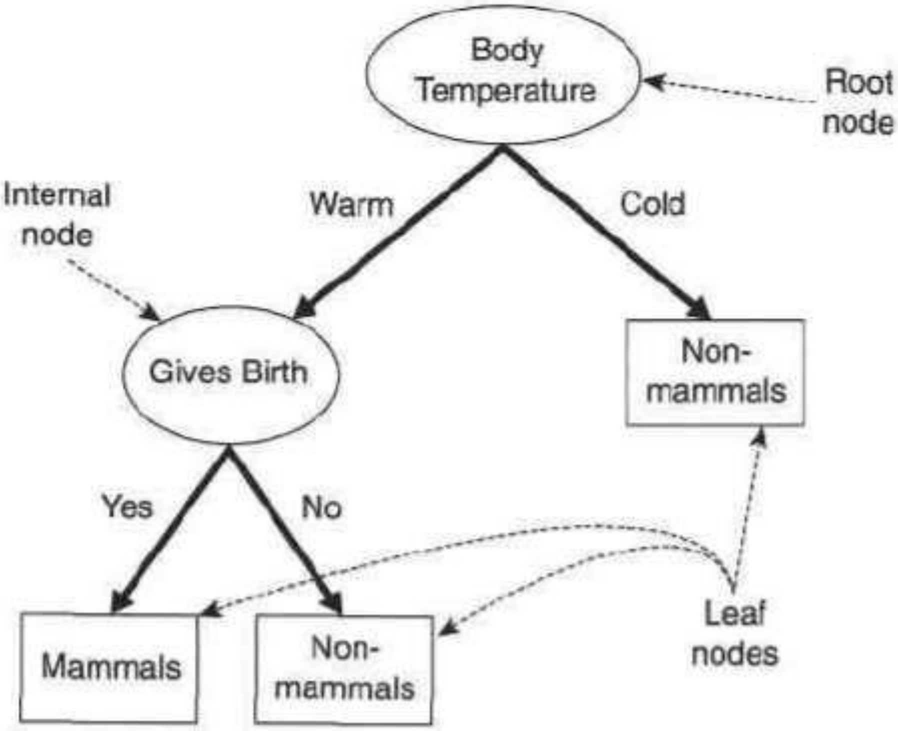
A ***root node*** that has no incoming edges and zero or more outgoing edges.

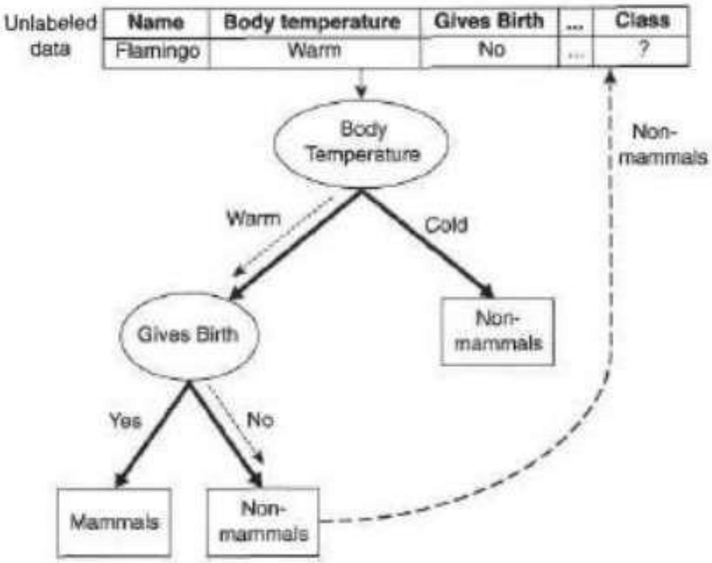
Internal nodes, each of which has exactly one incoming edge and two or more outgoing edges.

Leaf or terminal nodes, each of which has exactly one incoming edge and no outgoing edges.

In a decision tree, each leaf node is assigned a class label. The non terminal nodes, which include the root and other internal nodes, contain attribute test conditions to separate records that have different characteristics

In principle, there are exponentially many decision trees that can be constructed from a given set of attributes.





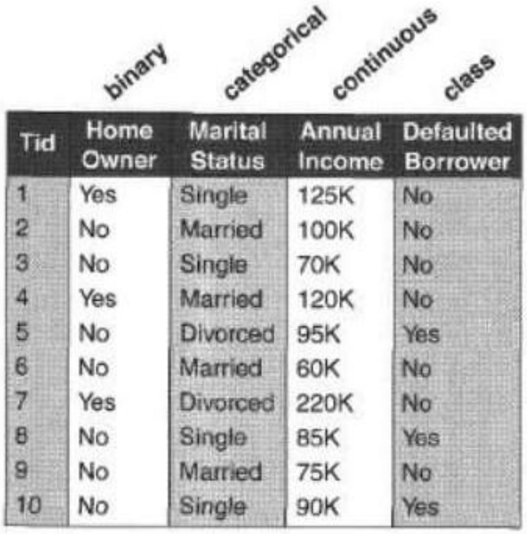
Unlabeled data	Name	Body temperature	Gives Birth	...	Class
	Flamingo	Warm	No	...	?

Figure 4.5. Classifying an unlabeled vertebrate. The dashed lines represent the outcomes of applying various attribute test conditions on the unlabeled vertebrate. The vertebrate is eventually assigned to the `Non-mammal` class.

Hunt's Algorithm

In Hunt's algorithm, a decision tree is grown in a recursive fashion by partitioning the training records into successively purer subsets. Let Di be the set of training records that are associated with

node t and y= {“y1,y2,y3….yc"} be the class labels. The following is a recursive definition of Hunt's algorithm.
Step 1: If all the records in Dt belong to the same class yt, then t is a leaf node labeled as yt.
Step 2: If Di contains records that belong to more than one class, an attribute test condition is selected to partition the records into smaller subsets. A child node is created for each outcome of the test condition and the records in Dt are distributed to the children based on the outcomes. The algorithm is then recursively applied to each ***child node.***


binary
categorical
continuous
class

Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Figure 4.6. Training set for predicting borrowers who will default on loan payments.

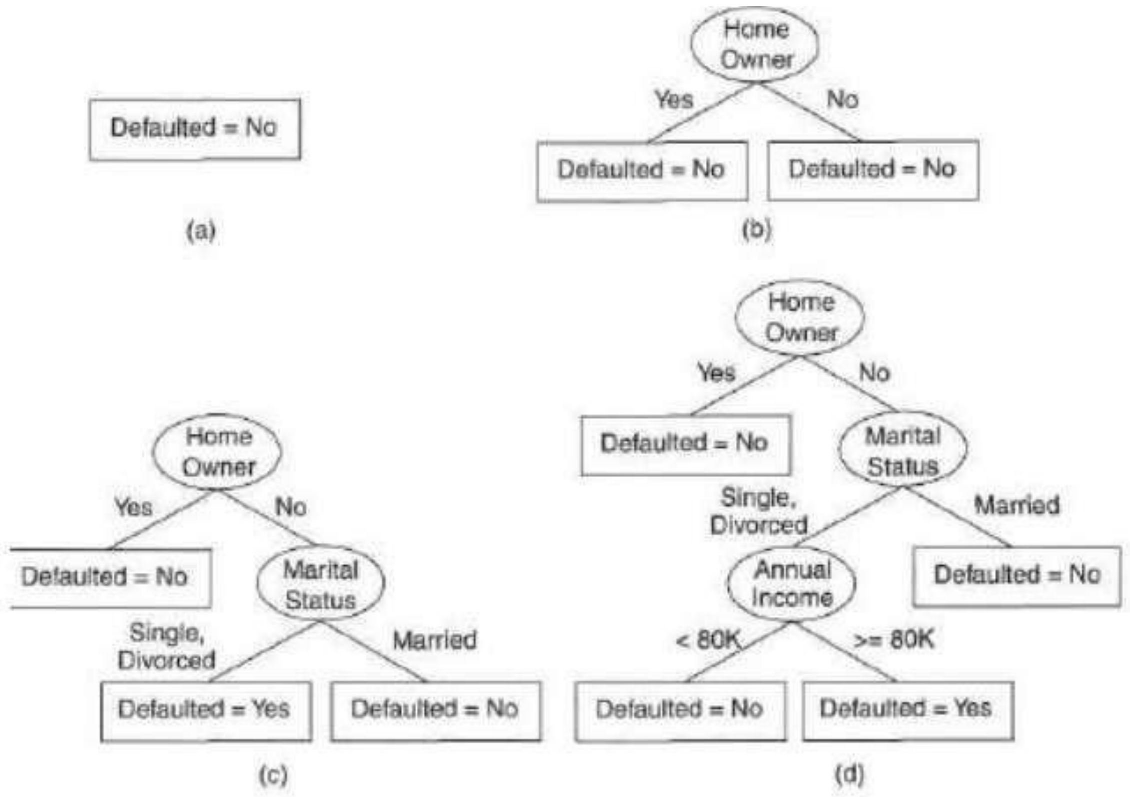


Figure 4.7. Hunt's algorithm for inducing decision trees.

To illustrate how the algorithm works, consider the problem of predicting whether a loan applicant will repay her loan obligations or become delinquent, subsequently defaulting on her loan.

The initial tree for the classification problem contains a single node with class label Defaulted = No (see Figure 4.7a), which means that most of the borrowers successfully repaid their loans. The tree, however, needs to be refined since the root node contains records from both classes.

The records are subsequently divided into smaller subsets based on the outcomes of the Home Owner test condition as shown in Figure 4.7(b). The justification for choosing this attribute test condition will be discussed later. For now, we will assume that this is the best criterion for splitting the data at this point.

Hunt's algorithm is then applied recursively to each child of the root node. From the training set given in Figure 4.6, notice that all borrowers who are home owners successfully repaid their loans. The left child of the root is therefore a leaf node labeled Defaulted = No (see Figure 4.7(b)).

For the right child, we need to continue applying the recursive step of Hunt's algorithm until all the records belong to the same class. The trees resulting from each recursive step are shown in Figures 4.7(c) and (d).

Design Issues of Decision Tree Induction:

A learning algorithm for inducing decision trees must address the following two issues.

1) Should the training records be split?

Each recursive step of the tree-growing process must select an attribute test condition to divide the records into smaller subsets. To implement this step, the algorithm must provide a method for specifying the test condition for different attribute types as well as an objective measure for evaluating the goodness of each test condition.

3) How should the splitting procedure stop?

A stopping condition is needed to terminate the tree-growing process. A possible strategy is to continue expanding a node until either all the records belong to the same class or all the records have identical attribute values. Although both conditions are sufficient to stop any decision tree induction algorithm, other criteria can be imposed to allow the tree-growing procedure to terminate earlier.

Methods for Expressing Attribute Test Conditions:

Decision tree induction algorithms must provide a method for expressing an attribute test condition and its corresponding outcomes for different attribute types.

Binary Attributes: The test condition for a binary attribute generates two potential outcomes, as shown in Figure 4.8.

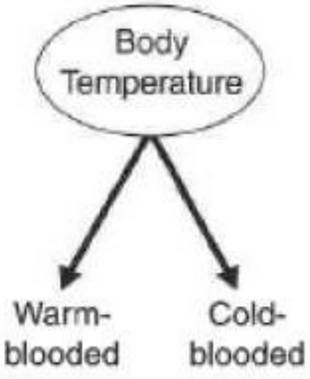
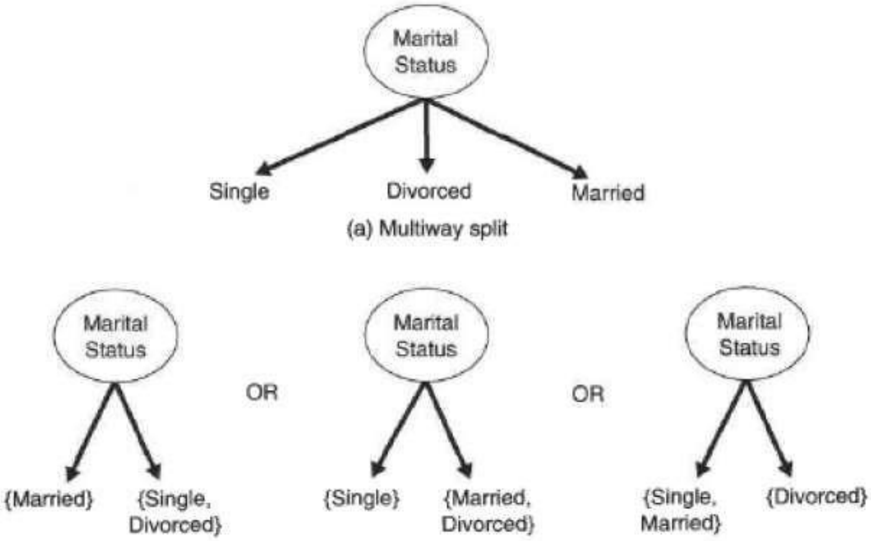


Figure 4.8. Test condition for binary attributes.

Nominal Attributes :Since a nominal attribute can have many values, its test condition can be expressed in two ways, as shown in Figure 4.9. For a multiway split (Figure 4.9(a)), the number of outcomes depends on the number of distinct values for the corresponding attribute. For example, if an attribute such as marital status has three distinct values-single, married, or divorced-its test condition will produce a three-way split.

Figure 4.9(b) illustrates three different ways of grouping the attribute values for marital status into two subsets.



(a) Multiway split

(b) Binary split {by grouping attribute values}

Figure 4.9. Test conditions for nominal attributes.

Ordinal Attributes: Ordinal attributes can also produce binary or multiway splits. Ordinal attribute values can be grouped as long as the grouping does not violate the order property of the attribute values. Figure 4.10 illustrates various ways of splitting training records based on the Shirt Size attribute.

The groupings shown in Figures 4.10(a) and (b) preserve the order among the attribute values, whereas the grouping shown in Figure a.10(c) violates this property because it combines the attribute values Small and Large into the same partition while Medium and Extra Large are combined into another partition.

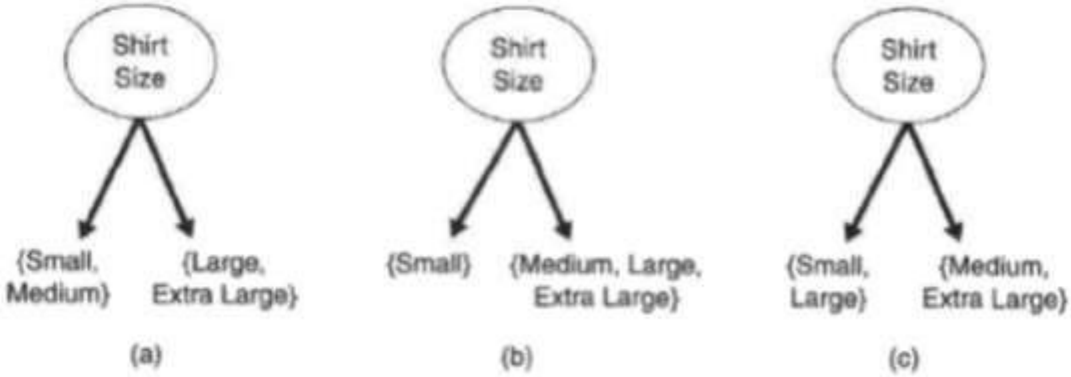


Figure 4.10. Different ways of grouping ordinal attribute values.

Continuous Attributes: For continuous attributes, the test condition can be expressed as a comparison test (A < V) or (A>=V ,) with binary outcomes, or a range query with outcomes of the form Vi<=A<Vi+1, for i=1,2…k. The difference between these approaches is shown in Figure 4.11.

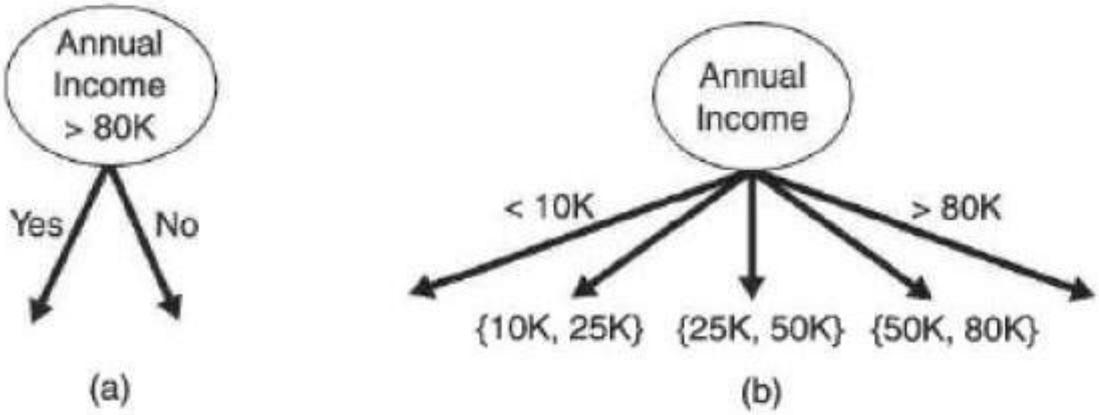


Figure 4.11. Test condition for continuous attributes.

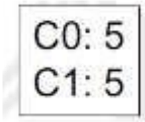
How to determine the Best Split:

Greedy approach:

– Nodes with homogeneous class distribution are preferred

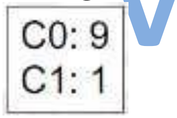
Need a measure of node impurity:

Non-homogeneous, High degree of impurity



C0: 5
C1: 5

Homogeneous, Low degree of impurity



C0: 9
C1: 1

Measures of Node Impurity:

- Gini Index
- Entropy
- Misclassification error

$$\begin{aligned}\text{Entropy}(t) &= -\sum_{i=0}^{c-1} p(i|t)\log_2 p(i|t),\\ \text{Gini}(t) &= 1-\sum_{i=0}^{c-1}[p(i|t)]^2,\\ \text{Classification error}(t) &= 1-\max_i[p(i|t)],\end{aligned}$$

where c is the number of classes and $0\log_2 0 = 0$ in entropy calculations.

Where **p(i|t)** denote the fraction of records belonging to class **i** at a given node **t** and where c is the number of classes.

The measures developed for selecting the best split are often based on the degree of impurity of the child nodes. The smaller the degree of impurity, the more skewed the class distribution.

Node N_1	Count
Class=0	0
Class=1	6

$$\begin{aligned}\text{Gini} &= 1-(0/6)^2-(6/6)^2 = 0\\ \text{Entropy} &= -(0/6)\log_2(0/6)-(6/6)\log_2(6/6) = 0\\ \text{Error} &= 1-\max[0/6,6/6] = 0\end{aligned}$$

Node N_2	Count
Class=0	1
Class=1	5

$$\begin{aligned}\text{Gini} &= 1-(1/6)^2-(5/6)^2 = 0.278\\ \text{Entropy} &= -(1/6)\log_2(1/6)-(5/6)\log_2(5/6) = 0.650\\ \text{Error} &= 1-\max[1/6,5/6] = 0.167\end{aligned}$$

Node N_3	Count
Class=0	3
Class=1	3

$$\begin{aligned}\text{Gini} &= 1-(3/6)^2-(3/6)^2 = 0.5\\ \text{Entropy} &= -(3/6)\log_2(3/6)-(3/6)\log_2(3/6) = 1\\ \text{Error} &= 1-\max[3/6,3/6] = 0.5\end{aligned}$$

Node N1 has the lowest impurity value, followed by N2 and N3.

To determine how well a test condition performs, we need to compare the degree of impurity of the parent node (before splitting) with the degree of impurity of the child nodes (after splitting). The larger their difference, the better the test condition. The gain, is a criterion that can be used to determine the goodness of a split.

$$\Delta = I(\text{parent}) - \sum_{j=1}^{k}\frac{N(v_j)}{N}I(v_j), \qquad (4.6)$$

where $I(\cdot)$ is the impurity measure of a given node, N is the total number of records at the parent node, k is the number of attribute values, and $N(v_j)$ is the number of records associated with the child node, v_j. Decision tree

Characteristics of Decision Tree Based Classification:

Advantages :

- Decision tree induction is a nonparametric approach for building classification models. In other words, it does not require any prior assumptions regarding the type of probability distributions satisfied by the class and other attributes.
- Finding an optimal decision tree is an NP-complete problem
- Techniques developed for constructing decision trees are computationally inexpensive, making it possible to quickly construct models even when the training set size is very large. Once a decision tree has been built, classifying a test record is extremely fast, with a worst- case complexity of O(W), where ,W.irs the maximum depth of the tree.
- Decision trees, especially smaller-sized trees, are relatively easy to interpret.
- Decision tree algorithms are quite robust to the presence of noise.
- The presence of redundant attributes does not adversely affect the accuracy of decision trees.

Disadvantages:

Since most decision tree algorithms employ a top-down, recursive partitioning approach, the number of records becomes smaller as we traverse down the tree. At the leaf nodes, the number of records may be too small to make a statistically significant decision about the class representation of the nodes.

 A subtree can be replicated multiple times in a decision tree, as illustrated in Figure 4.19. This makes the decision tree more complex than necessary and perhaps more difficult to interpret. Such a situation can arise from decision tree implementations that rely on a single attribute test condition at each internal node.

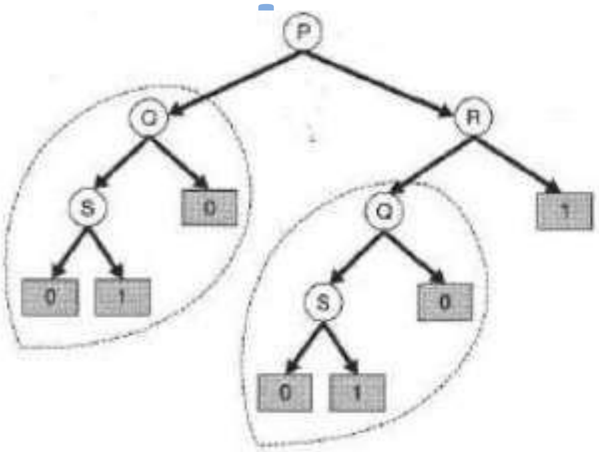
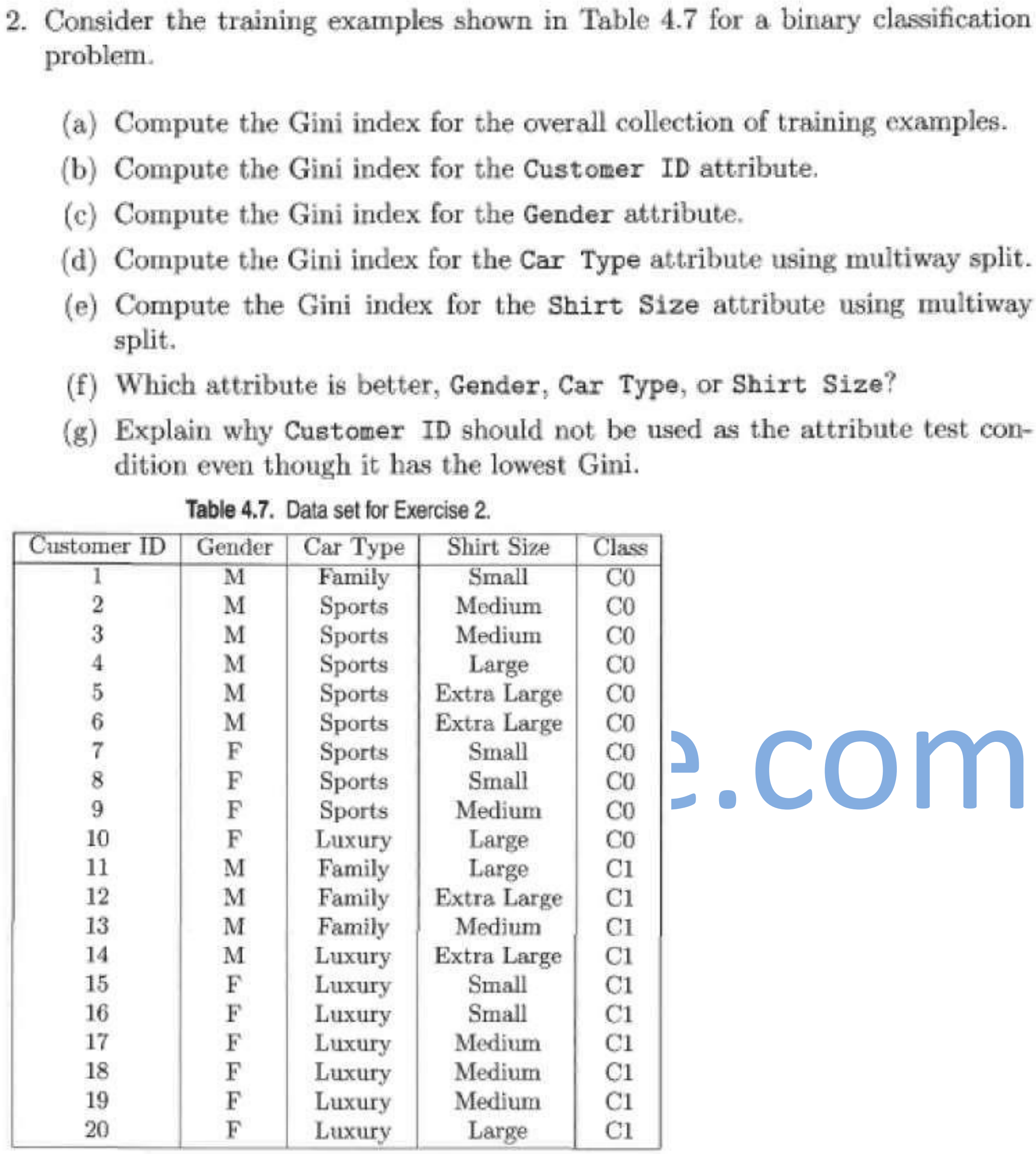


Figure 4.19. Tree replication problem. The same subtree can appear at different branches.

1. Draw the full decision tree for the parity function of four Boolean attributes, A, B, C, and D. Is it possible to simplify the tree?

Exercises:



2. Consider the training examples shown in Table 4.7 for a binary classification problem.

   (a) Compute the Gini index for the overall collection of training examples.

   (b) Compute the Gini index for the Customer ID attribute.

   (c) Compute the Gini index for the Gender attribute.

   (d) Compute the Gini index for the Car Type attribute using multiway split.

   (e) Compute the Gini index for the Shirt Size attribute using multiway split.

   (f) Which attribute is better, Gender, Car Type, or Shirt Size?

   (g) Explain why Customer ID should not be used as the attribute test condition even though it has the lowest Gini.

Table 4.7. Data set for Exercise 2.

Customer ID	Gender	Car Type	Shirt Size	Class
1	M	Family	Small	C0
2	M	Sports	Medium	C0
3	M	Sports	Medium	C0
4	M	Sports	Large	C0
5	M	Sports	Extra Large	C0
6	M	Sports	Extra Large	C0
7	F	Sports	Small	C0
8	F	Sports	Small	C0
9	F	Sports	Medium	C0
10	F	Luxury	Large	C0
11	M	Family	Large	C1
12	M	Family	Extra Large	C1
13	M	Family	Medium	C1
14	M	Luxury	Extra Large	C1
15	F	Luxury	Small	C1
16	F	Luxury	Small	C1
17	F	Luxury	Medium	C1
18	F	Luxury	Medium	C1
19	F	Luxury	Medium	C1
20	F	Luxury	Large	C1

(a) Compute the Gini index for the overall collection of training examples.

Answer:

Gini $= 1 - 2 \times 0.5^2 = 0.5$.

(b) Compute the Gini index for the `Customer ID` attribute.

Answer:

The gini for each `Customer ID` value is 0. Therefore, the overall gini for `Customer ID` is 0.

(c) Compute the Gini index for the `Gender` attribute.

Answer:

The gini for `Male` is $1 - 2 \times 0.5^2 = 0.5$. The gini for `Female` is also 0.5. Therefore, the overall gini for `Gender` is $0.5 \times 0.5 + 0.5 \times 0.5 = 0.5$.

(d) Compute the Gini index for the `Car Type` attribute using multiway split.

Answer:

The gini for `Family` car is 0.375, `Sports` car is 0, and `Luxury` car is 0.2188. The overall gini is 0.1625.

(e) Compute the Gini index for the `Shirt Size` attribute using multiway split.

Answer:

The gini for `Small` shirt size is 0.48, `Medium` shirt size is 0.4898, `Large` shirt size is 0.5, and `Extra Large` shirt size is 0.5. The overall gini for `Shirt Size` attribute is 0.4914.

(f) Which attribute is better, `Gender`, `Car Type`, or `Shirt Size`?

Answer:

`Car Type` because it has the lowest gini among the three attributes.

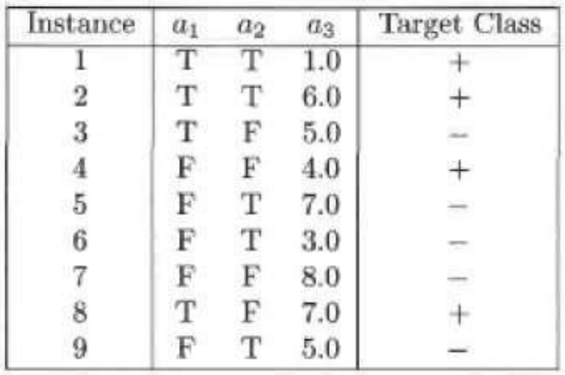
(g) Explain why `Customer ID` should not be used as the attribute test condition even though it has the lowest Gini.

Answer:

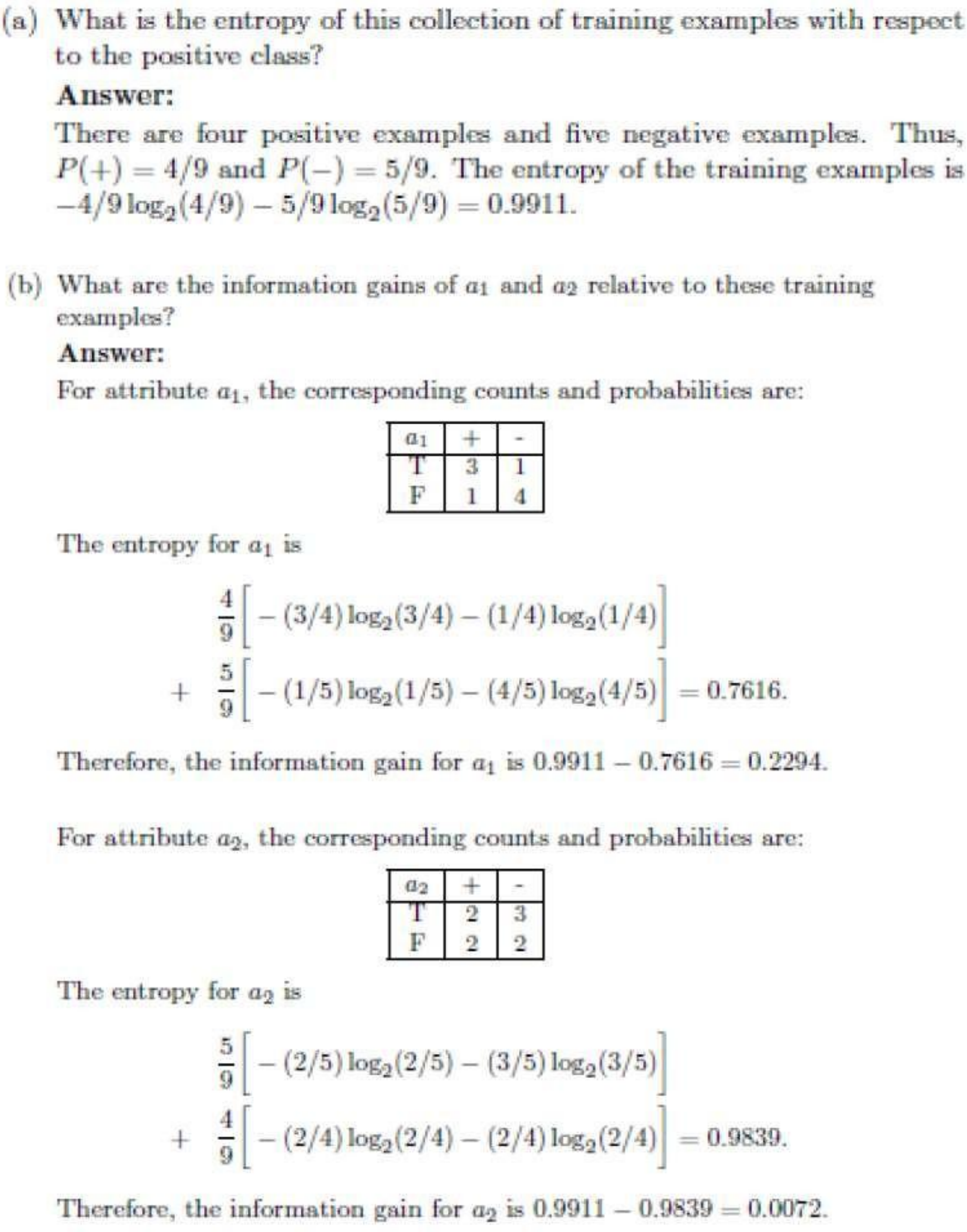
The attribute has no predictive power since new customers are assigned to new `Customer IDs`.

3. Consider the training examples shown in Table 4.8 for a binary classification problem.

Table 4.8. Data set for Exercise 3.



Instance	a_1	a_2	a_3	Target Class
1	T	T	1.0	+
2	T	T	6.0	+
3	T	F	5.0	−
4	F	F	4.0	+
5	F	T	7.0	−
6	F	T	3.0	−
7	F	F	8.0	−
8	T	F	7.0	+
9	F	T	5.0	−



(a) What is the entropy of this collection of training examples with respect to the positive class?

Answer:

There are four positive examples and five negative examples. Thus, $P(+) = 4/9$ and $P(-) = 5/9$. The entropy of the training examples is $-4/9\log_2(4/9) - 5/9\log_2(5/9) = 0.9911$.

(b) What are the information gains of a_1 and a_2 relative to these training examples?

Answer:

For attribute a_1, the corresponding counts and probabilities are:

a_1	+	-
T	3	1
F	1	4

The entropy for a_1 is

$$\begin{aligned} &\frac{4}{9}\Big[-(3/4)\log_2(3/4) - (1/4)\log_2(1/4)\Big] \\ + \; &\frac{5}{9}\Big[-(1/5)\log_2(1/5) - (4/5)\log_2(4/5)\Big] = 0.7616. \end{aligned}$$

Therefore, the information gain for a_1 is $0.9911 - 0.7616 = 0.2294$.

For attribute a_2, the corresponding counts and probabilities are:

a_2	+	-
T	2	3
F	2	2

The entropy for a_2 is

$$\begin{aligned} &\frac{5}{9}\Big[-(2/5)\log_2(2/5) - (3/5)\log_2(3/5)\Big] \\ + \; &\frac{4}{9}\Big[-(2/4)\log_2(2/4) - (2/4)\log_2(2/4)\Big] = 0.9839. \end{aligned}$$

Therefore, the information gain for a_2 is $0.9911 - 0.9839 = 0.0072$.

(c) For a_3, which is a continuous attribute, compute the information gain for every possible split.

Answer:

a_3	Class label	Split point	Entropy	Info Gain
1.0	+	2.0	0.8484	0.1427
3.0	-	3.5	0.9885	0.0026
4.0	+	4.5	0.9183	0.0728
5.0	-			
5.0	-	5.5	0.9839	0.0072
6.0	+	6.5	0.9728	0.0183
7.0	+			
7.0	-	7.5	0.8889	0.1022

The best split for a_3 occurs at split point equals to 2.

(d) What is the best split (among a_1, a_2, and a_3) according to the information gain?

Answer:

According to information gain, a_1 produces the best split.

(e) What is the best split (between a_1 and a_2) according to the classification error rate?

Answer:

For attribute a_1: error rate $= 2/9$.

For attribute a_2: error rate $= 4/9$.

Therefore, according to error rate, a_1 produces the best split.

(f) What is the best split (between a_1 and a_2) according to the Gini index?

Answer:

For attribute a_1, the gini index is

$$\frac{4}{9}\left[1-(3/4)^2-(1/4)^2\right]+\frac{5}{9}\left[1-(1/5)^2-(4/5)^2\right]=0.3444.$$

For attribute a_2, the gini index is

$$\frac{5}{9}\left[1-(2/5)^2-(3/5)^2\right]+\frac{4}{9}\left[1-(2/4)^2-(2/4)^2\right]=0.4889.$$

Since the gini index for a_1 is smaller, it produces the better split.

5. Consider the following data set for a binary class problem.

A	B	Class Label
T	F	+
T	T	+
T	T	+
T	F	−
T	T	+
F	F	−
F	F	−
F	F	−
T	T	−
T	F	−

(a) Calculate the information gain when splitting on A and B. Which attribute would the decision tree induction algorithm choose?

Answer:

The contingency tables after splitting on attributes A and B are:

	$A = T$	$A = F$
+	4	0
−	3	3

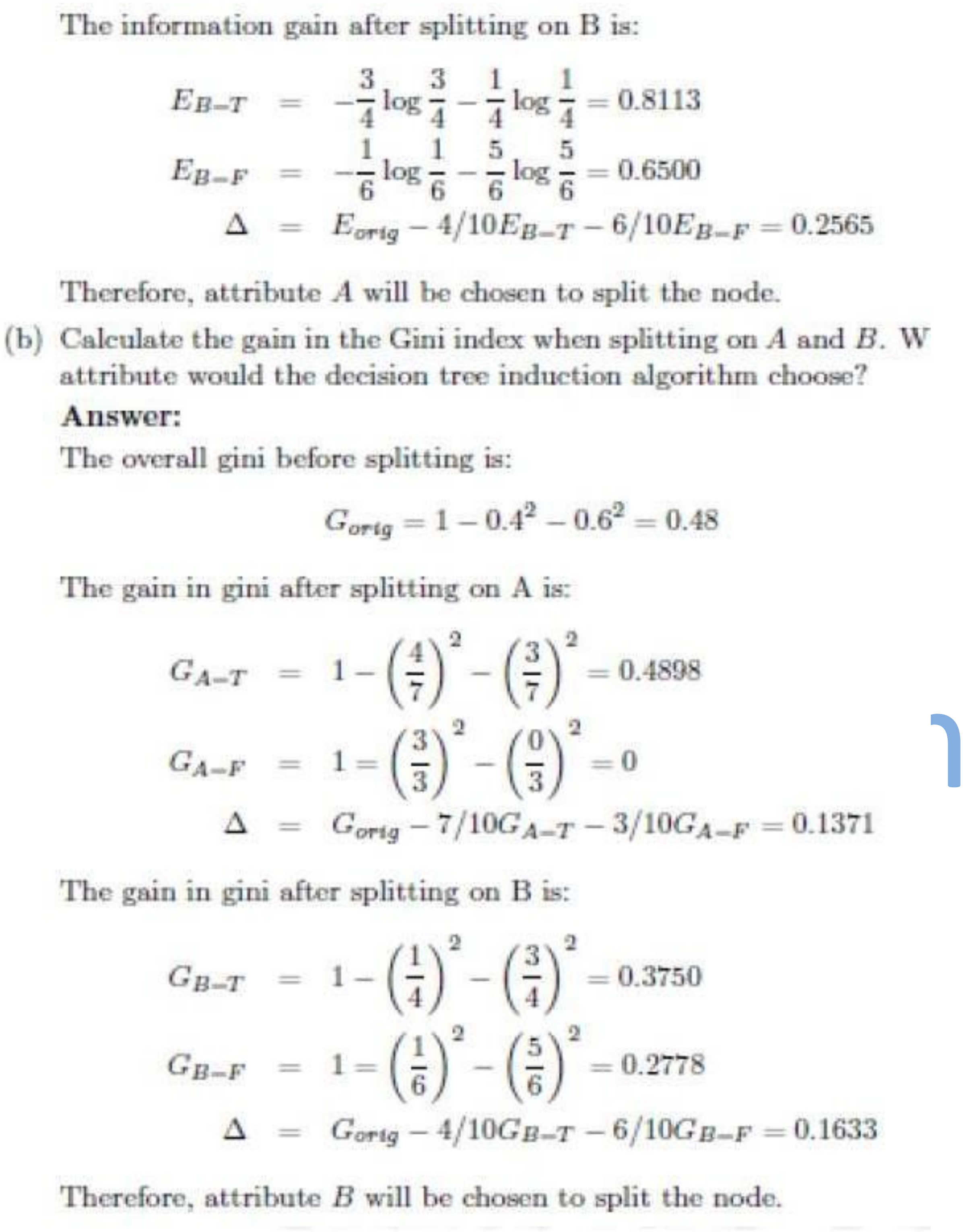
	$B = T$	$B = F$
+	3	1
−	1	5

The overall entropy before splitting is:

$$E_{orig} = -0.4 \log 0.4 - 0.6 \log 0.6 = 0.9710$$

The information gain after splitting on A is:

$$\begin{aligned} E_{A=T} &= -\frac{4}{7}\log\frac{4}{7} - \frac{3}{7}\log\frac{3}{7} = 0.9852 \\ E_{A=F} &= -\frac{3}{3}\log\frac{3}{3} - \frac{0}{3}\log\frac{0}{3} = 0 \\ \Delta &= E_{orig} - 7/10 E_{A=T} - 3/10 E_{A=F} = 0.2813 \end{aligned}$$



The information gain after splitting on B is:

$$
\begin{aligned}
E_{B-T} &= -\frac{3}{4}\log\frac{3}{4} - \frac{1}{4}\log\frac{1}{4} = 0.8113 \\
E_{B-F} &= -\frac{1}{6}\log\frac{1}{6} - \frac{5}{6}\log\frac{5}{6} = 0.6500 \\
\Delta &= E_{orig} - 4/10E_{B-T} - 6/10E_{B-F} = 0.2565
\end{aligned}
$$

Therefore, attribute A will be chosen to split the node.

(b) Calculate the gain in the Gini index when splitting on A and B. W attribute would the decision tree induction algorithm choose?

Answer:

The overall gini before splitting is:

$$G_{orig} = 1 - 0.4^2 - 0.6^2 = 0.48$$

The gain in gini after splitting on A is:

$$
\begin{aligned}
G_{A-T} &= 1 - \left(\frac{4}{7}\right)^2 - \left(\frac{3}{7}\right)^2 = 0.4898 \\
G_{A-F} &= 1 = \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0 \\
\Delta &= G_{orig} - 7/10G_{A-T} - 3/10G_{A-F} = 0.1371
\end{aligned}
$$

The gain in gini after splitting on B is:

$$
\begin{aligned}
G_{B-T} &= 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2 = 0.3750 \\
G_{B-F} &= 1 = \left(\frac{1}{6}\right)^2 - \left(\frac{5}{6}\right)^2 = 0.2778 \\
\Delta &= G_{orig} - 4/10G_{B-T} - 6/10G_{B-F} = 0.1633
\end{aligned}
$$

Therefore, attribute B will be chosen to split the node.

Model Over fitting:

The errors committed by a classification model are generally divided into two types: training errors and generalization errors.

Training error, is the number of misclassification errors committed on training records,
Generalization error is the expected error of the model on test records.
A good model must have low training error as well as low generalization error.

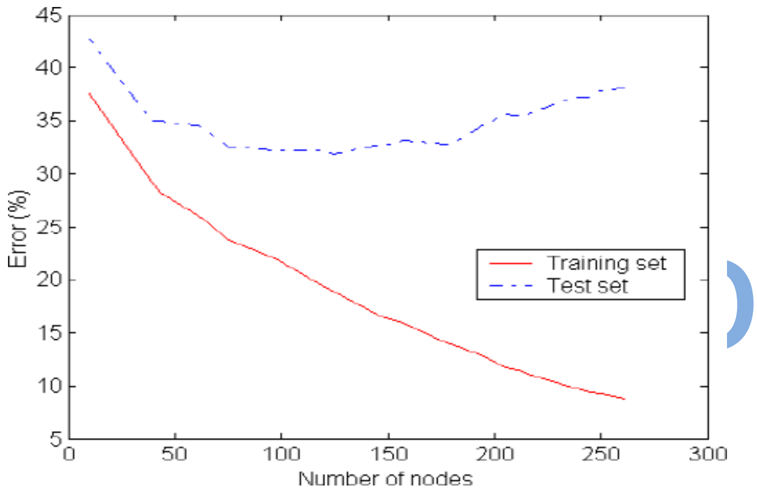
Underfitting : The training and test error rates of the model are large when the size of the tree is very small. This situation is known as model underfitting.
Underfitting occurs because the model has yet to learn the true structure ofthe data. As a result, it performs poorly on both the training and the test sets.

Overfitting.: As the number of nodes in the decision tree increases, the tree will have fewer training and test error . However, once the tree becomes too large, its test error rate begins to increase even though its training error rate continues to decrease. This phenomenon is known as model over fitting.

Reasons for over fitting:

- Presence of Noise
- Lack of Representative Samples

Figure shows the training and test error rates of the decision tree.



Estimating Generalization Errors:

Generalization errors: error on testing (∑e‟(t)) Methods for estimating generalization errors:

1) Optimistic approach: e'(t) = e(t)

2) Pessimistic approach:

○ For each leaf node: e‟(t) = (e(t)+0.5)

○ Total errors: e'(T) = e(T) + N □ 0.5 (N: number of leaf nodes)

- Ex: For a tree with 30 leaf nodes and 10 errors on training
- (out of 1000 instances): Training error = 10/1000 = 1%
- Generalization error = (10 + 30□0.5)/1000 = 2.5%

Reduced error pruning (REP):
Uses validation data set to estimate generalization error

How to Address Over fitting:

Pre-Pruning (Early Stopping Rule) ○

- Stop the algorithm before it becomes a fully-grown tree
- Typical stopping conditions for a node:

○ Stop if all instances belong to the same class

○ Stop if all the attribute values are the same

○

- More restrictive conditions:

○ Stop if number of instances is less than some user-specified threshold

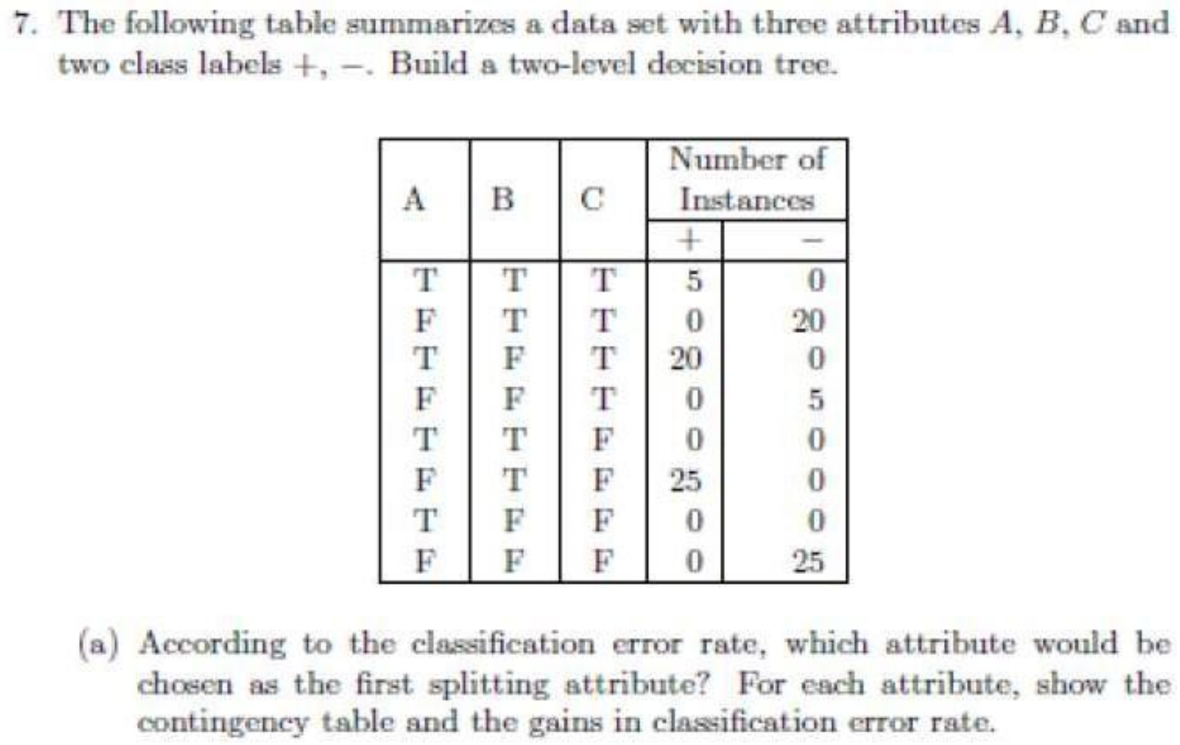
○ Stop if class distribution of instances are independent of the available features (e.g., using x 2 test)

○ Stop if expanding the current node does not improve impurity measures (e.g., Gini or information gain)

2 **Post-pruning**

- Grow decision tree to its entirety
- Trim the nodes of the decision tree in a bottom-up fashion
- If generalization error improves after trimming, replace sub-tree by a leaf node.
- Class label of leaf node is determined from majority class of instances in the sub-tree

Exercises:



7. The following table summarizes a data set with three attributes A, B, C and two class labels $+$, $-$. Build a two-level decision tree.

A	B	C	Number of Instances	
			+	−
T	T	T	5	0
F	T	T	0	20
T	F	T	20	0
F	F	T	0	5
T	T	F	0	0
F	T	F	25	0
T	F	F	0	0
F	F	F	0	25

(a) According to the classification error rate, which attribute would be chosen as the first splitting attribute? For each attribute, show the contingency table and the gains in classification error rate.

$$E_{orig} = \frac{25}{75}$$

After splitting on attribute B, the gain in error rate is:

	$B = T$	$B = F$
+	25	0
−	20	30

$$E_{B-T} = \frac{20}{45}$$
$$E_{B-F} = 0$$
$$\Delta_B = E_{orig} - \frac{45}{75}E_{B-T} - \frac{20}{75}E_{B-F} = \frac{5}{75}$$

After splitting on attribute C, the gain in error rate is:

	$C = T$	$C = F$
+	0	25
−	25	25

$$E_{C-T} = \frac{0}{25}$$
$$E_{C-F} = \frac{25}{50}$$
$$\Delta_C = E_{orig} - \frac{25}{75}E_{C-T} - \frac{50}{75}E_{C-F} = 0$$

The split will be made on attribute B.

(c) How many instances are misclassified by the resulting decision tree?

Answer:

20 instances are misclassified. (The error rate is $\frac{20}{100}$.)

(d) Repeat parts (a), (b), and (c) using C as the splitting attribute.

Answer:

For the $C = T$ child node, the error rate before splitting is:
$E_{orig} = \frac{25}{50}$.
After splitting on attribute A, the gain in error rate is:

	$A = T$	$A = F$
+	25	0
−	0	25

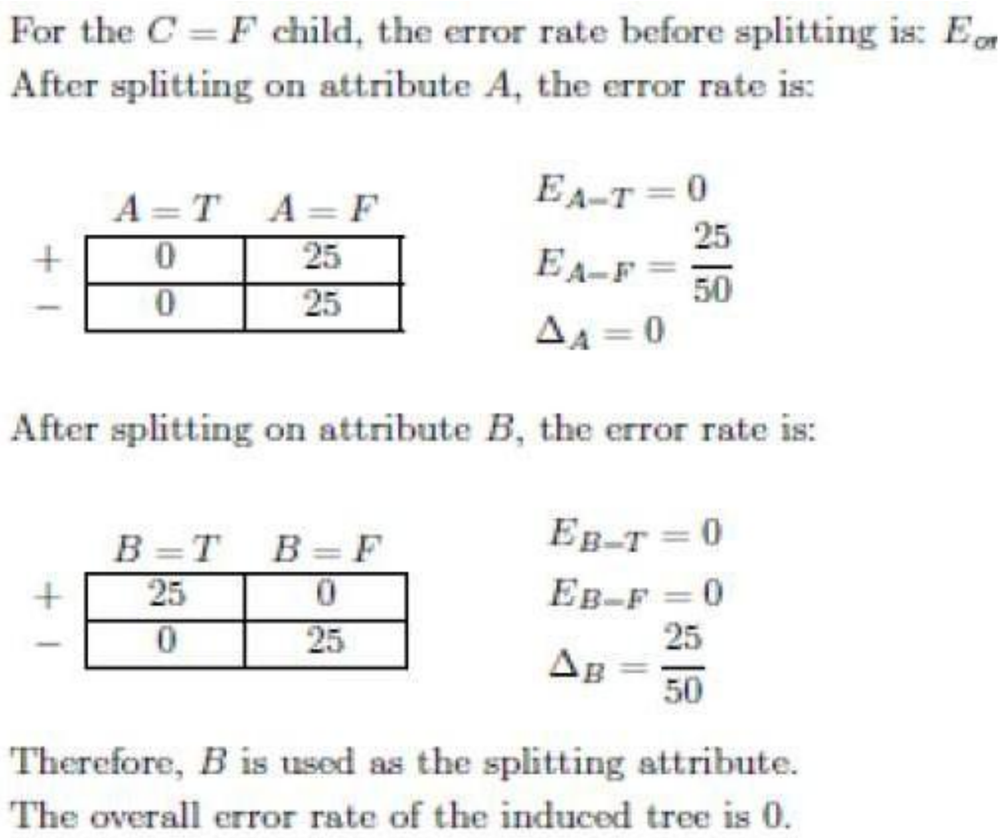
$$E_{A-T} = 0$$
$$E_{A-F} = 0$$
$$\Delta_A = \frac{25}{50}$$

After splitting on attribute B, the gain in error rate is:

	$B = T$	$B = F$
+	5	20
−	20	5

$$E_{B-T} = \frac{5}{25}$$
$$E_{B-F} = \frac{5}{25}$$
$$\Delta_B = \frac{15}{50}$$

Therefore, A is chosen as the splitting attribute.



For the $C = F$ child, the error rate before splitting is: E_{or}

After splitting on attribute A, the error rate is:

	$A = T$	$A = F$
+	0	25
−	0	25

$$E_{A-T} = 0$$

$$E_{A-F} = \frac{25}{50}$$

$$\Delta_A = 0$$

After splitting on attribute B, the error rate is:

	$B = T$	$B = F$
+	25	0
−	0	25

$$E_{B-T} = 0$$

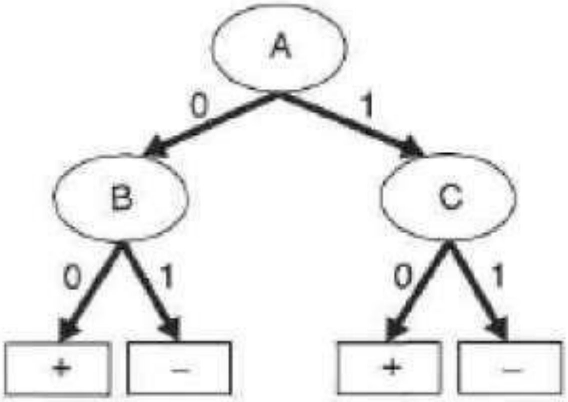
$$E_{B-F} = 0$$

$$\Delta_B = \frac{25}{50}$$

Therefore, B is used as the splitting attribute.

The overall error rate of the induced tree is 0.

8. Consider the decision tree shown in Figure 4.30.



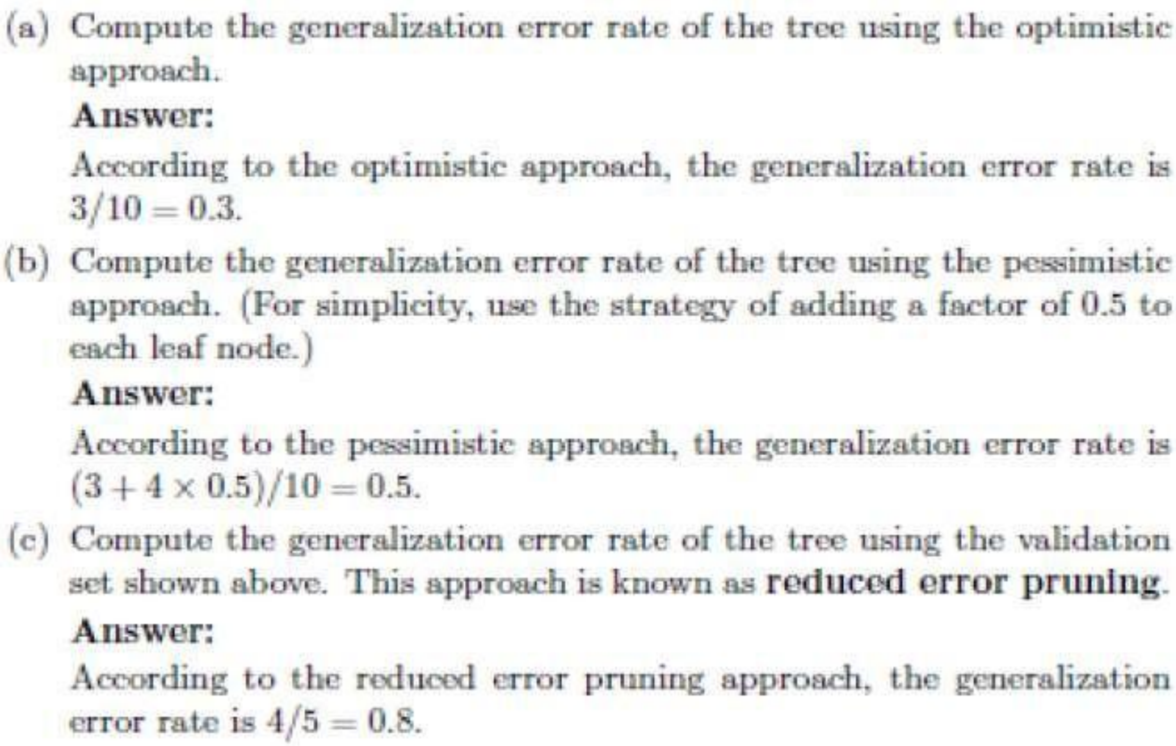
Training:

Instance	A	B	C	Class
1	0	0	0	+
2	0	0	1	+
3	0	1	0	+
4	0	1	1	−
5	1	0	0	+
6	1	0	0	+
7	1	1	0	−
8	1	0	1	+
9	1	1	0	−
10	1	1	0	−

Validation:

Instance	A	B	C	Class
11	0	0	0	+
12	0	1	1	+
13	1	1	0	+
14	1	0	1	−
15	1	0	0	+

Figure 4.30. Decision tree and data sets for Exercise 8.



(a) Compute the generalization error rate of the tree using the optimistic approach.

Answer:

According to the optimistic approach, the generalization error rate is $3/10 = 0.3$.

(b) Compute the generalization error rate of the tree using the pessimistic approach. (For simplicity, use the strategy of adding a factor of 0.5 to each leaf node.)

Answer:

According to the pessimistic approach, the generalization error rate is $(3 + 4 \times 0.5)/10 = 0.5$.

(c) Compute the generalization error rate of the tree using the validation set shown above. This approach is known as **reduced error pruning**.

Answer:

According to the reduced error pruning approach, the generalization error rate is $4/5 = 0.8$.

Answer:

The error rate for the data without partitioning on any attribute is

$$E_{orig} = 1 - \max(\frac{50}{100}, \frac{50}{100}) = \frac{50}{100}.$$

After splitting on attribute A, the gain in error rate is:

	$A = T$	$A = F$
+	25	25
−	0	50

$$E_{A-T} = 1 - \max(\frac{25}{25}, \frac{0}{25}) = \frac{0}{25} = 0$$

$$E_{A-F} = 1 - \max(\frac{25}{75}, \frac{50}{75}) = \frac{25}{75}$$

$$\Delta_A = E_{orig} - \frac{25}{100}E_{A-T} - \frac{75}{100}E_{A-F} = \frac{25}{100}$$

After splitting on attribute B, the gain in error rate is:

	$B = T$	$B = F$
+	30	20
−	20	30

$$E_{B-T} = \frac{20}{50}$$

$$E_{B-F} = \frac{20}{50}$$

$$\Delta_B = E_{orig} - \frac{50}{100}E_{B-T} - \frac{50}{100}E_{B-F} = \frac{10}{100}$$

After splitting on attribute C, the gain in error rate is:

	$C = T$	$C = F$
+	25	25
−	25	25

$$E_{C-T} = \frac{25}{50}$$

$$E_{C-F} = \frac{25}{50}$$

$$\Delta_C = E_{orig} - \frac{50}{100}E_{C-T} - \frac{50}{100}E_{C-F} = \frac{0}{100} = 0$$

The algorithm chooses attribute A because it has the highest gain.

(b) Repeat for the two children of the root node.

Answer:

Because the $A = T$ child node is pure, no further splitting is needed. For the $A = F$ child node, the distribution of training instances is:

B	C	Class label	
		+	−
T	T	0	20
F	T	0	5
T	F	25	0
F	F	0	25

The classification error of the $A = F$ child node is:

4.4 Rule-Based Classifier

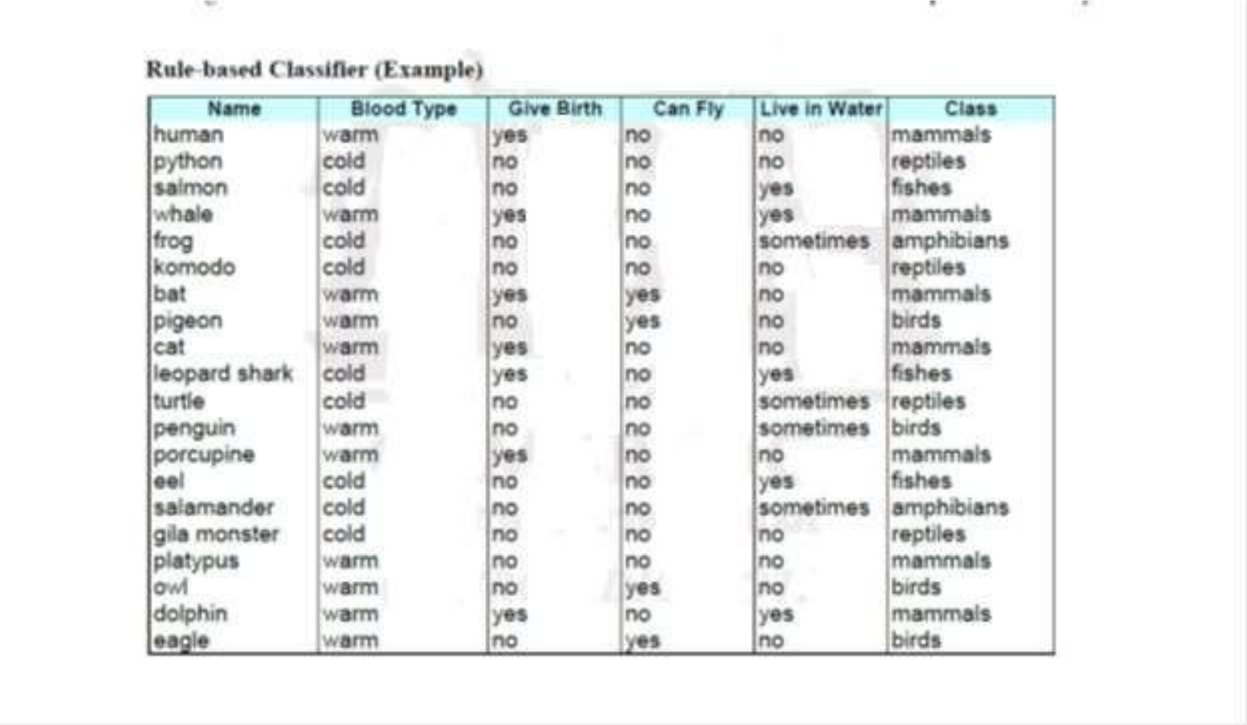
A rule-based classifier is a technique for classifying records using a collection of "if . . .then. . ." rules. The rules for the model are represented in a disjunctive normal form, . where R is known as the rule set and r;'s are the classification rules or disjuncts

Each classification rule can be expressed in the following way:

$$r_i : \quad (Condition_i) \longrightarrow y_i.$$

The left-hand side of the rule is called the rule antecedent or precondition.

The right-hand side of the rule is called the rule consequent, which contains the predicted class yi



Rule-based Classifier (Example)

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammals
python	cold	no	no	no	reptiles
salmon	cold	no	no	yes	fishes
whale	warm	yes	no	yes	mammals
frog	cold	no	no	sometimes	amphibians
komodo	cold	no	no	no	reptiles
bat	warm	yes	yes	no	mammals
pigeon	warm	no	yes	no	birds
cat	warm	yes	no	no	mammals
leopard shark	cold	yes	no	yes	fishes
turtle	cold	no	no	sometimes	reptiles
penguin	warm	no	no	sometimes	birds
porcupine	warm	yes	no	no	mammals
eel	cold	no	no	yes	fishes
salamander	cold	no	no	sometimes	amphibians
gila monster	cold	no	no	no	reptiles
platypus	warm	no	no	no	mammals
owl	warm	no	yes	no	birds
dolphin	warm	yes	no	yes	mammals
eagle	warm	no	yes	no	birds

R1: (Give Birth = no) $\wedge$ (Can Fly = yes) $\rightarrow$ Birds

R2: (Give Birth = no) $\wedge$ (Live in Water = yes) $\rightarrow$ Fishes

R3: (Give Birth = yes) $\wedge$ (Blood Type = warm) $\rightarrow$ Mammals

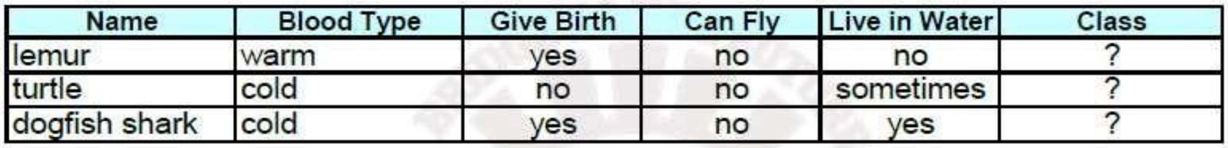
R4: (Give Birth = no) $\wedge$ (Can Fly = no) $\rightarrow$ Reptiles

R5: (Live in Water = sometimes) $\rightarrow$ Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
hawk	warm	no	yes	no	?
grizzly bear	warm	yes	no	no	?

The rule R1 covers a hawk => Bird

The rule R3 covers the grizzly bear => Mammal



Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

A lemur triggers rule R3, so it is classified as a mammal A turtle triggers both R4 and R5
A dogfish shark triggers none of the rules

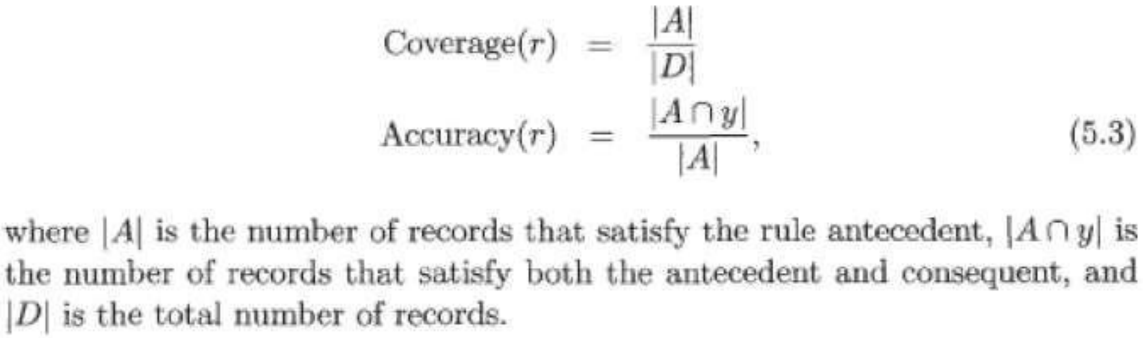
Rule Coverage and Accuracy

Coverage of a rule:

– Fraction of records that satisfy the antecedent of a rule

Accuracy of a rule:

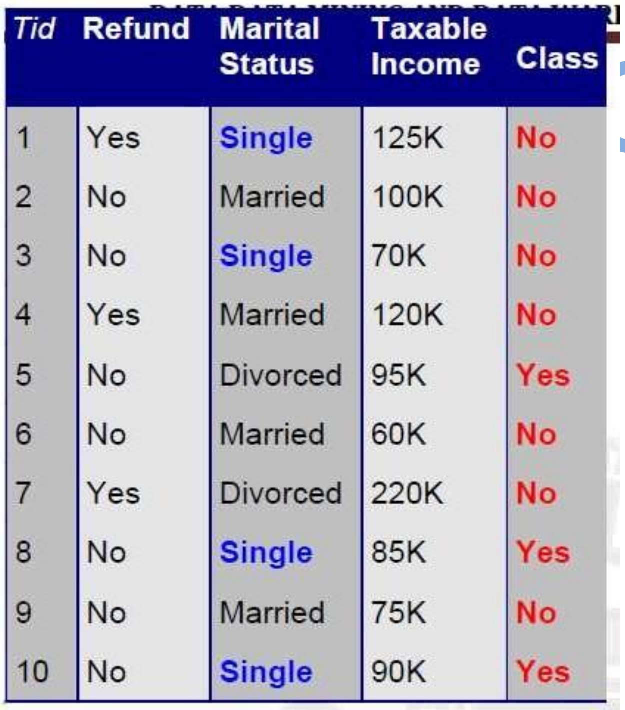
– Fraction of records that satisfy both the antecedent and consequent of a rule



$$\text{Coverage}(r) = \frac{|A|}{|D|}$$

$$\text{Accuracy}(r) = \frac{|A \cap y|}{|A|}, \qquad (5.3)$$

where $|A|$ is the number of records that satisfy the rule antecedent, $|A \cap y|$ is the number of records that satisfy both the antecedent and consequent, and $|D|$ is the total number of records.



Tid	Refund	Marital Status	Taxable Income	Class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Mutually Exclusive Rules The rules in a rule set .R are mutually exclusive if no two rules in .R are triggered by the same record. This property ensures that every record is covered by at most one rule in R. **Exhaustive Rules** A rule set -R has exhaustive coverage if there is a rule for each combination of attribute values. This property ensures that every record is covered by at least one rule in –R **Ordered Rules** In this approach, the rules in a rule set are ordered in decreasing order of their priority, which can be defined in many ways (e.g., based on accuracy, coverage, total description length, or the order in which the rules are generated). An ordered rule set is also known as a decision list. When a test record is presented, it is classified by the highest-ranked rule that covers the record. This avoids the problem of having conflicting classes predicted by multiple classification rules.

Rule-Ordering Schemes

Rule-based ordering

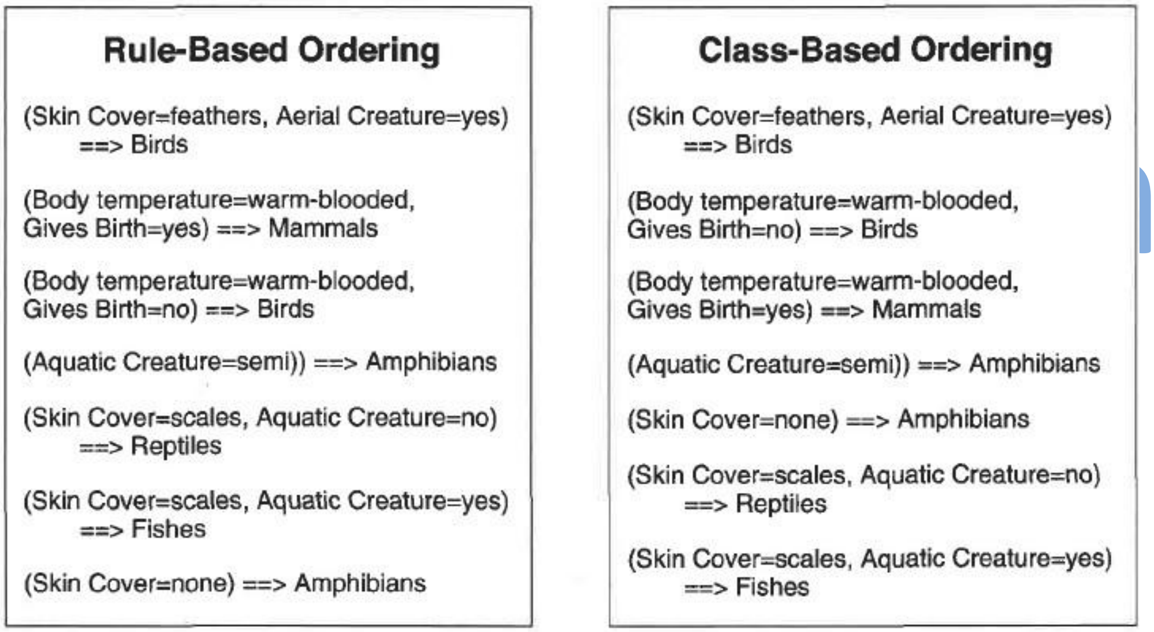
Individual rules are ranked based on their quality

- ✦ This approach orders the individual rules by some rule quality measure.
- ✦ This ordering scheme ensures that every test record is classified by the "best" rule covering it.

Class-based ordering

Rules that belong to the same class appear together

In this approach, rules that belong to the same class appear together in the rule set R. The rules are then collectively sorted on the basis of their class information.



Rule-Based Ordering

(Skin Cover=feathers, Aerial Creature=yes) ==> Birds

(Body temperature=warm-blooded, Gives Birth=yes) ==> Mammals

(Body temperature=warm-blooded, Gives Birth=no) ==> Birds

(Aquatic Creature=semi)) ==> Amphibians

(Skin Cover=scales, Aquatic Creature=no) ==> Reptiles

(Skin Cover=scales, Aquatic Creature=yes) ==> Fishes

(Skin Cover=none) ==> Amphibians

Class-Based Ordering

(Skin Cover=feathers, Aerial Creature=yes) ==> Birds

(Body temperature=warm-blooded, Gives Birth=no) ==> Birds

(Body temperature=warm-blooded, Gives Birth=yes) ==> Mammals

(Aquatic Creature=semi)) ==> Amphibians

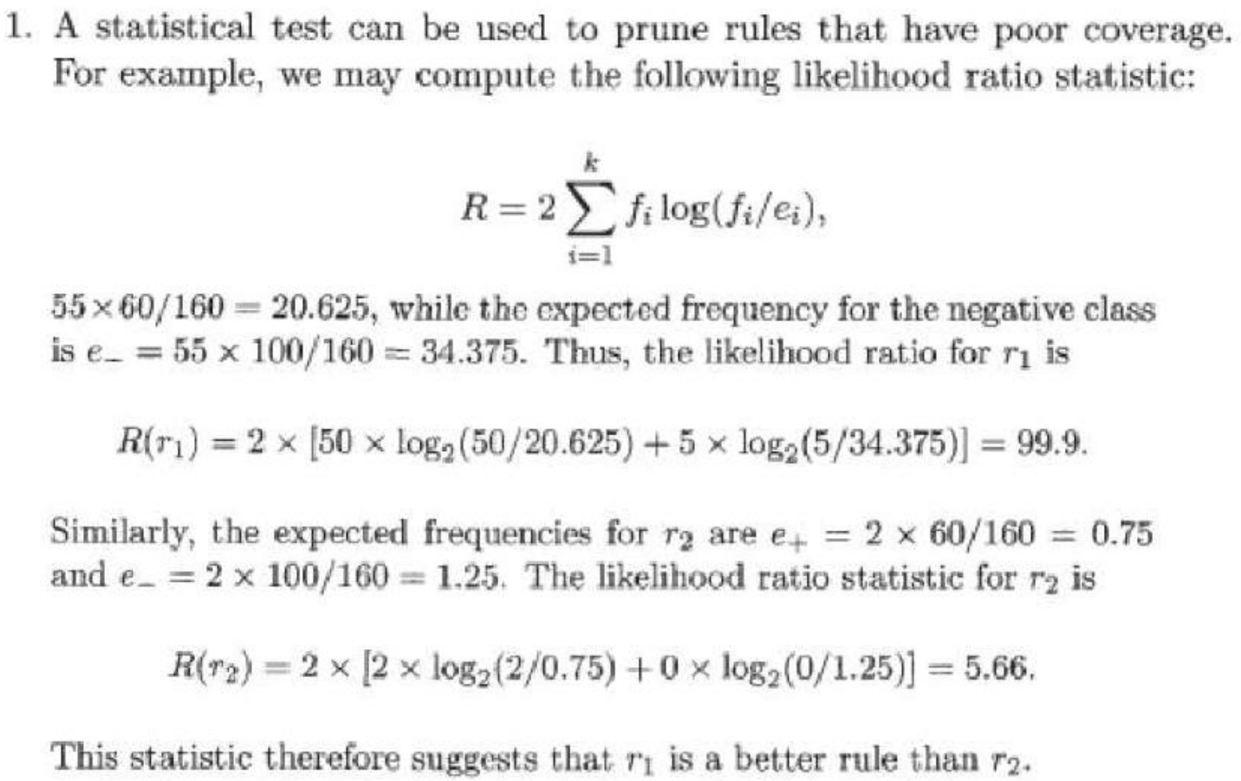
(Skin Cover=none) ==> Amphibians

(Skin Cover=scales, Aquatic Creature=no) ==> Reptiles

(Skin Cover=scales, Aquatic Creature=yes) ==> Fishes

Figure 5.1. Comparison between rule-based and class-based ordering schemes.

Rule Evaluation:



1. A statistical test can be used to prune rules that have poor coverage. For example, we may compute the following likelihood ratio statistic:

$$R = 2\sum_{i=1}^{k} f_i \log(f_i/e_i),$$

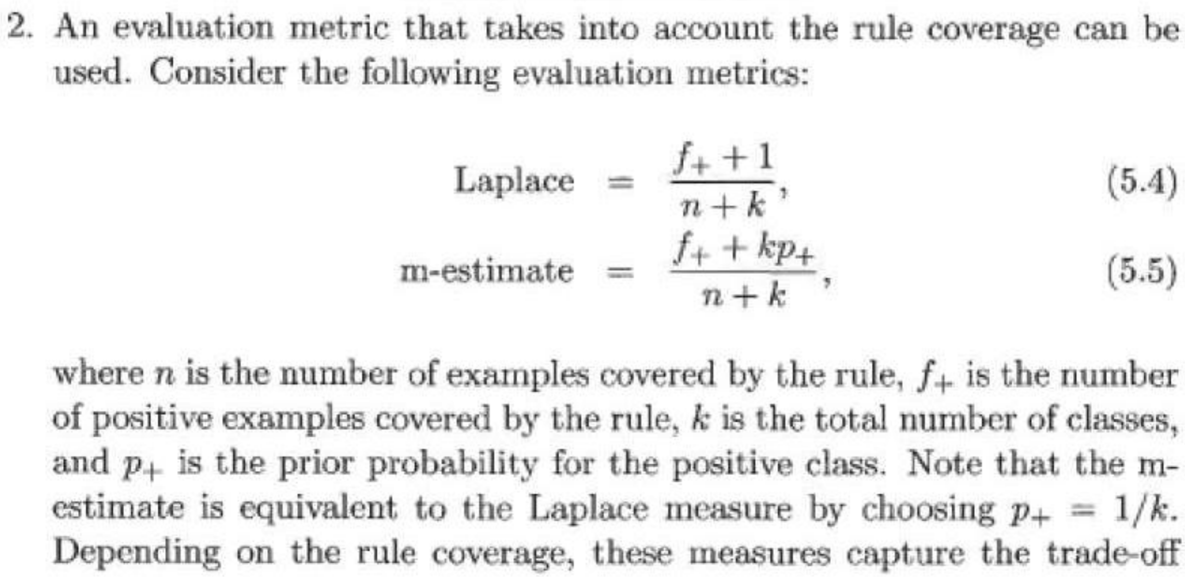
$55 \times 60/160 = 20.625$, while the expected frequency for the negative class is $e_- = 55 \times 100/160 = 34.375$. Thus, the likelihood ratio for r_1 is

$$R(r_1) = 2 \times [50 \times \log_2(50/20.625) + 5 \times \log_2(5/34.375)] = 99.9.$$

Similarly, the expected frequencies for r_2 are $e_+ = 2 \times 60/160 = 0.75$ and $e_- = 2 \times 100/160 = 1.25$. The likelihood ratio statistic for r_2 is

$$R(r_2) = 2 \times [2 \times \log_2(2/0.75) + 0 \times \log_2(0/1.25)] = 5.66.$$

This statistic therefore suggests that r_1 is a better rule than r_2.

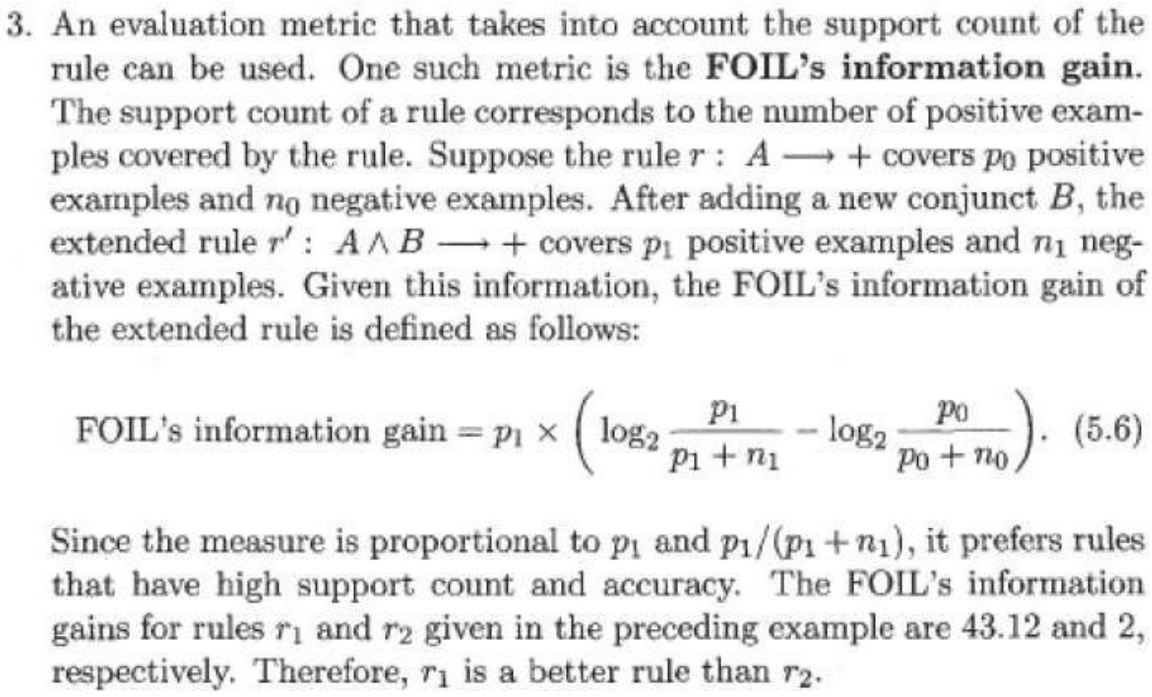


2. An evaluation metric that takes into account the rule coverage can be used. Consider the following evaluation metrics:

$$\text{Laplace} = \frac{f_+ + 1}{n + k}, \qquad (5.4)$$

$$\text{m-estimate} = \frac{f_+ + kp_+}{n + k}, \qquad (5.5)$$

where n is the number of examples covered by the rule, f_+ is the number of positive examples covered by the rule, k is the total number of classes, and p_+ is the prior probability for the positive class. Note that the m-estimate is equivalent to the Laplace measure by choosing $p_+ = 1/k$. Depending on the rule coverage, these measures capture the trade-off



3. An evaluation metric that takes into account the support count of the rule can be used. One such metric is the **FOIL's information gain.** The support count of a rule corresponds to the number of positive examples covered by the rule. Suppose the rule $r: A \longrightarrow +$ covers p_0 positive examples and n_0 negative examples. After adding a new conjunct B, the extended rule $r': A \wedge B \longrightarrow +$ covers p_1 positive examples and n_1 negative examples. Given this information, the FOIL's information gain of the extended rule is defined as follows:

$$\text{FOIL's information gain} = p_1 \times \left(\log_2 \frac{p_1}{p_1 + n_1} - \log_2 \frac{p_0}{p_0 + n_0} \right). \quad (5.6)$$

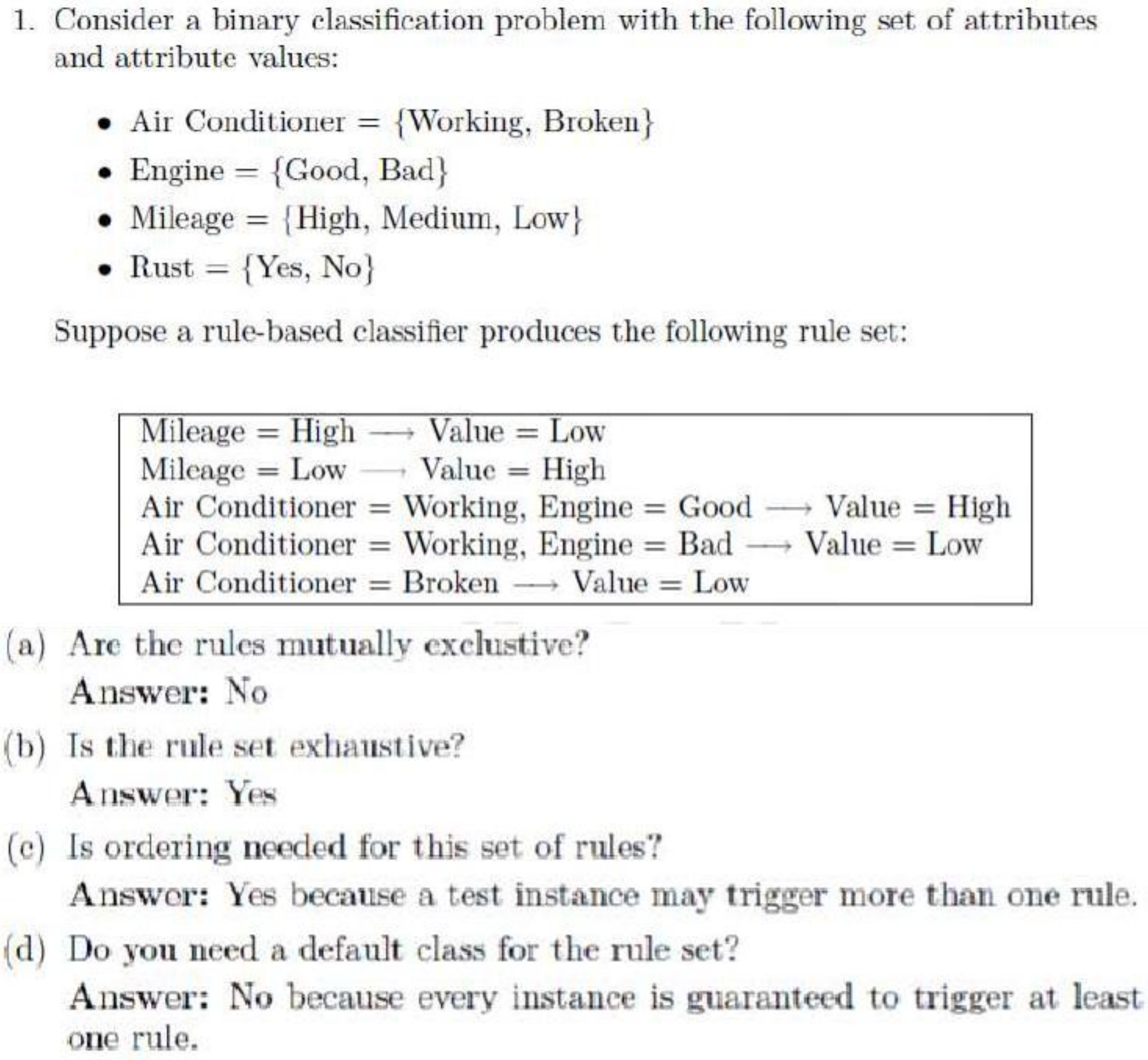
Since the measure is proportional to p_1 and $p_1/(p_1 + n_1)$, it prefers rules that have high support count and accuracy. The FOIL's information gains for rules r_1 and r_2 given in the preceding example are 43.12 and 2, respectively. Therefore, r_1 is a better rule than r_2.

Characteristics of Rule-Based Classifiers:

A rule-based classifier has the following characteristics:

- The expressiveness of a rule set is almost equivalent to that of a decision tree because a decision tree can be represented by a set of mutually exclusive and exhaustive rules. Both rule-based and decision tree classifiers create rectilinear partitions of the attribute space and assign a class to each partition. Nevertheless, if the rule-based classifier allows multiple rules to be triggered for a given record, then a more complex decision boundary can be constructed.

- Rule-based classifiers are generally used to produce descriptive models that are easier to interpret, but gives comparable performance to the decision tree classifier.



1. Consider a binary classification problem with the following set of attributes and attribute values:

- Air Conditioner = {Working, Broken}
- Engine = {Good, Bad}
- Mileage = {High, Medium, Low}
- Rust = {Yes, No}

Suppose a rule-based classifier produces the following rule set:

Mileage = High ⟶ Value = Low
Mileage = Low ⟶ Value = High
Air Conditioner = Working, Engine = Good ⟶ Value = High
Air Conditioner = Working, Engine = Bad ⟶ Value = Low
Air Conditioner = Broken ⟶ Value = Low

(a) Are the rules mutually exclustive?

Answer: No

(b) Is the rule set exhaustive?

Answer: Yes

(c) Is ordering needed for this set of rules?

Answer: Yes because a test instance may trigger more than one rule.

(d) Do you need a default class for the rule set?

Answer: No because every instance is guaranteed to trigger at least one rule.

4. Consider a training set that contains 100 positive examples and 400 negative examples. For each of the following candidate rules,

R_1: $A \longrightarrow +$ (covers 4 positive and 1 negative examples),
R_2: $B \longrightarrow +$ (covers 30 positive and 10 negative examples),
R_3: $C \longrightarrow +$ (covers 100 positive and 90 negative examples),

determine which is the best and worst candidate rule according to:

(a) Rule accuracy.

Answer:

The accuracies of the rules are 80% (for R_1), 75% (for R_2), and 52.6% (for R_3), respectively. Therefore R_1 is the best candidate and R_3 is the worst candidate according to rule accuracy.

(b) FOIL's information gain.

Answer:

Assume the initial rule is $\emptyset \longrightarrow +$. This rule covers $p_0 = 100$ positive examples and $n_0 = 400$ negative examples.

The rule R_1 covers $p_1 = 4$ positive examples and $n_1 = 1$ negative example. Therefore, the FOIL's information gain for this rule is

$$4 \times \left(\log_2 \frac{4}{5} - \log_2 \frac{100}{500} \right) = 8.$$

The rule R_2 covers $p_1 = 30$ positive examples and $n_1 = 10$ negative example. Therefore, the FOIL's information gain for this rule is

$$30 \times \left(\log_2 \frac{30}{40} - \log_2 \frac{100}{500} \right) = 57.2.$$

The rule R_3 covers $p_1 = 100$ positive examples and $n_1 = 90$ negative example. Therefore, the FOIL's information gain for this rule is

$$100 \times \left(\log_2 \frac{100}{190} - \log_2 \frac{100}{500} \right) = 139.6.$$

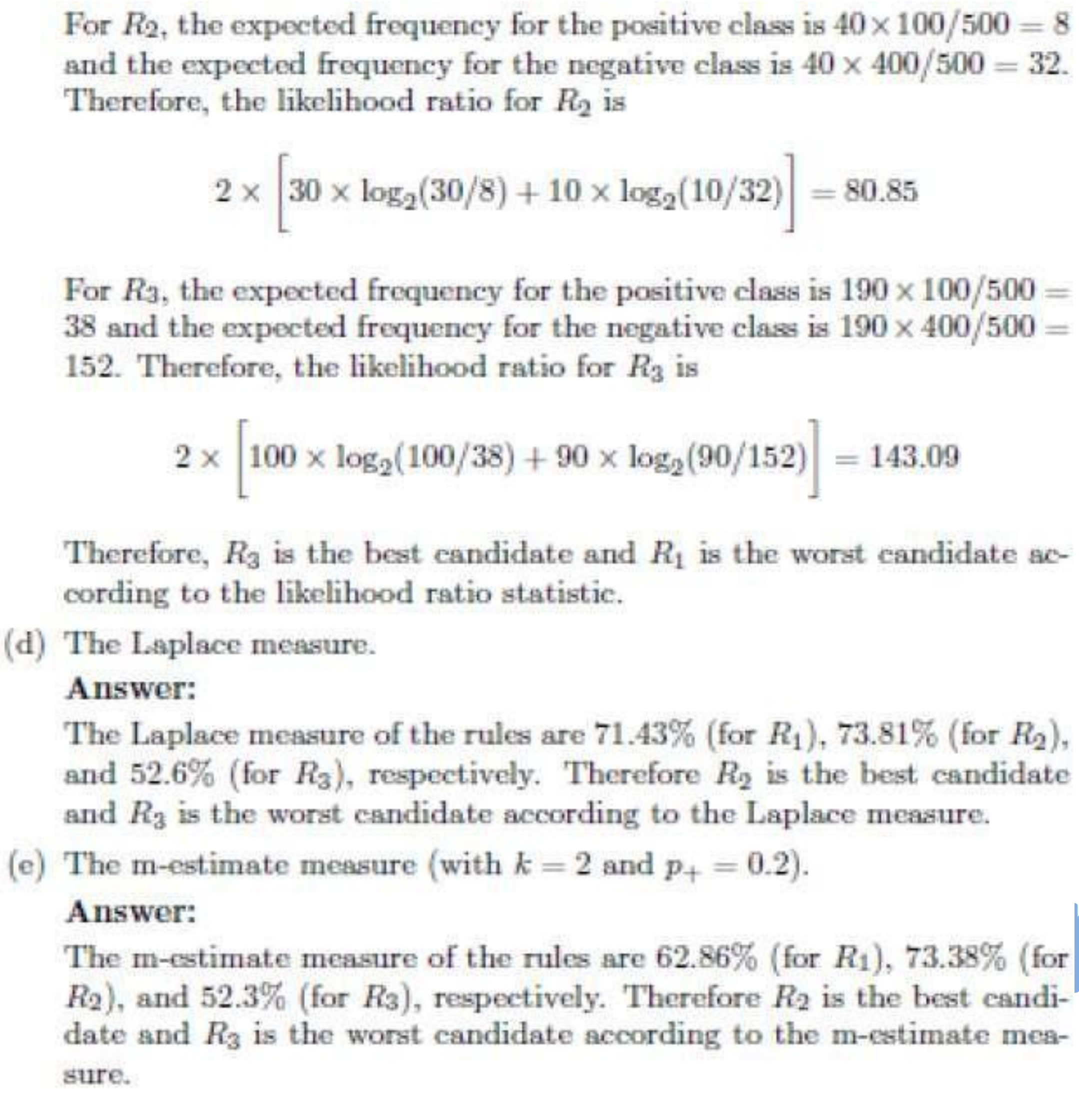
Therefore, R_3 is the best candidate and R_1 is the worst candidate according to FOIL's information gain.

(c) The likelihood ratio statistic.

Answer:

For R_1, the expected frequency for the positive class is $5 \times 100/500 = 1$ and the expected frequency for the negative class is $5 \times 400/500 = 4$. Therefore, the likelihood ratio for R_1 is

$$2 \times \left[4 \times \log_2(4/1) + 1 \times \log_2(1/4) \right] = 12.$$



For R_2, the expected frequency for the positive class is $40 \times 100/500 = 8$ and the expected frequency for the negative class is $40 \times 400/500 = 32$. Therefore, the likelihood ratio for R_2 is

$$2 \times \left[30 \times \log_2(30/8) + 10 \times \log_2(10/32)\right] = 80.85$$

For R_3, the expected frequency for the positive class is $190 \times 100/500 = 38$ and the expected frequency for the negative class is $190 \times 400/500 = 152$. Therefore, the likelihood ratio for R_3 is

$$2 \times \left[100 \times \log_2(100/38) + 90 \times \log_2(90/152)\right] = 143.09$$

Therefore, R_3 is the best candidate and R_1 is the worst candidate according to the likelihood ratio statistic.

(d) The Laplace measure.

Answer:

The Laplace measure of the rules are 71.43% (for R_1), 73.81% (for R_2), and 52.6% (for R_3), respectively. Therefore R_2 is the best candidate and R_3 is the worst candidate according to the Laplace measure.

(e) The m-estimate measure (with $k = 2$ and $p_+ = 0.2$).

Answer:

The m-estimate measure of the rules are 62.86% (for R_1), 73.38% (for R_2), and 52.3% (for R_3), respectively. Therefore R_2 is the best candidate and R_3 is the worst candidate according to the m-estimate measure.

5. Figure 5.1 illustrates the coverage of the classification rules $R1$, $R2$, and $R3$. Determine which is the best and worst rule according to:

   (a) The likelihood ratio statistic.

   Answer:

   There are 29 positive examples and 21 negative examples in the data set. $R1$ covers 12 positive examples and 3 negative examples. The expected frequency for the positive class is $15 \times 29/50 = 8.7$ and the expected frequency for the negative class is $15 \times 21/50 = 6.3$. Therefore, the likelihood ratio for $R1$ is

$$2 \times \left[12 \times \log_2(12/8.7) + 3 \times \log_2(3/6.3) \right] = 4.71.$$

   $R2$ covers 7 positive examples and 3 negative examples. The expected frequency for the positive class is $10 \times 29/50 = 5.8$ and the expected

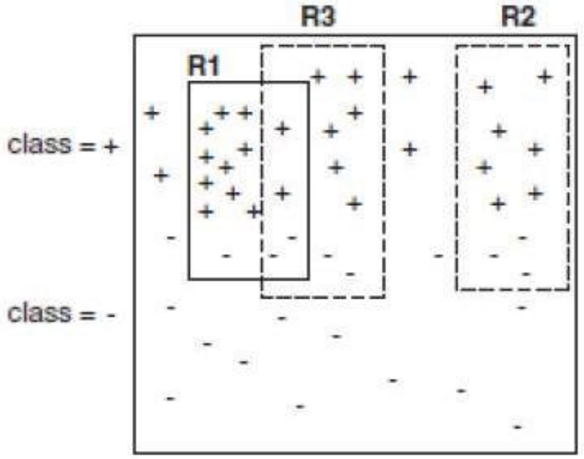


Figure 5.1. Elimination of training records by the sequential covering algorithm. $R1$, $R2$, and $R3$ represent regions covered by three different rules.

frequency for the negative class is $10 \times 21/50 = 4.2$. Therefore, the likelihood ratio for $R2$ is

$$2 \times \left[7 \times \log_2(7/5.8) + 3 \times \log_2(3/4.2)\right] = 0.89.$$

$R3$ covers 8 positive examples and 4 negative examples. The expected frequency for the positive class is $12 \times 29/50 = 6.96$ and the expected frequency for the negative class is $12 \times 21/50 = 5.04$. Therefore, the likelihood ratio for $R3$ is

$$2 \times \left[8 \times \log_2(8/6.96) + 4 \times \log_2(4/5.04)\right] = 0.5472.$$

$R1$ is the best rule and $R3$ is the worst rule according to the likelihood ratio statistic.

(b) The Laplace measure.

Answer:

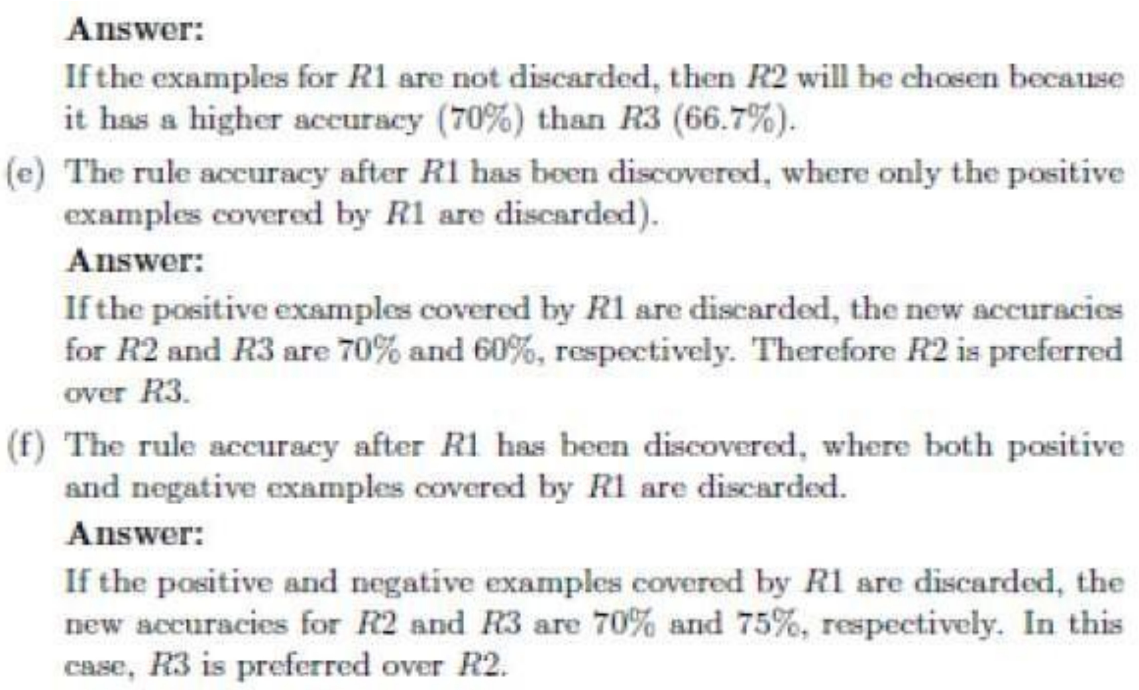
The Laplace measure for the rules are 76.47% (for $R1$), 66.67% (for $R2$), and 64.29% (for $R3$), respectively. Therefore $R1$ is the best rule and $R3$ is the worst rule according to the Laplace measure.

(c) The m-estimate measure (with $k = 2$ and $p_+ = 0.58$).

Answer:

The m-estimate measure for the rules are 77.41% (for $R1$), 68.0% (for $R2$), and 65.43% (for $R3$), respectively. Therefore $R1$ is the best rule and $R3$ is the worst rule according to the m-estimate measure.

(d) The rule accuracy after $R1$ has been discovered, where none of the examples covered by $R1$ are discarded).



Answer:

If the examples for $R1$ are not discarded, then $R2$ will be chosen because it has a higher accuracy (70%) than $R3$ (66.7%).

(e) The rule accuracy after $R1$ has been discovered, where only the positive examples covered by $R1$ are discarded).

Answer:

If the positive examples covered by $R1$ are discarded, the new accuracies for $R2$ and $R3$ are 70% and 60%, respectively. Therefore $R2$ is preferred over $R3$.

(f) The rule accuracy after $R1$ has been discovered, where both positive and negative examples covered by $R1$ are discarded.

Answer:

If the positive and negative examples covered by $R1$ are discarded, the new accuracies for $R2$ and $R3$ are 70% and 75%, respectively. In this case, $R3$ is preferred over $R2$.

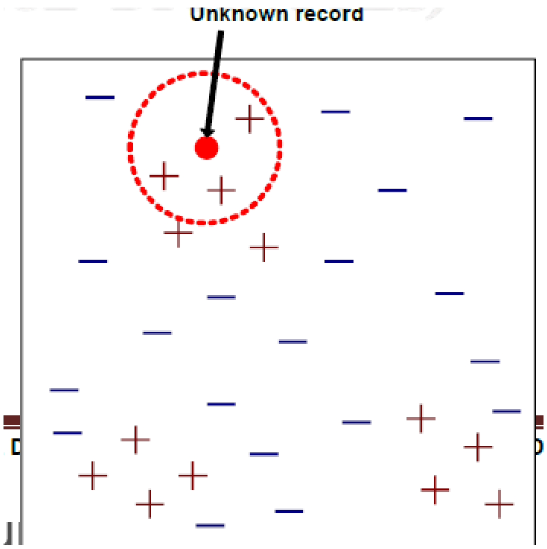
4.5 Nearest-Neighbor Classifiers

Requires three things

- The set of stored records
- Distance Metric to compute distance between records
- The value of *k*, the number of nearest neighbors to retrieve

To classify an unknown record:

- Compute distance to other training records
- Identify *k* nearest neighbors
- Use class labels of nearest neighbors to determine the class label of unknown record
- (e.g., by taking majority vote)



K-nearest neighbors of a record x are data points that have the k smallest distance to x Compute distance between two points:

– Euclidean distance

Determine the class from nearest neighbor list

– take the majority vote of class labels among the k-nearest neighbors Choosing the value of k:
– If k is too small, sensitive to noise points
– If k is too large, neighborhood may include points from other classes

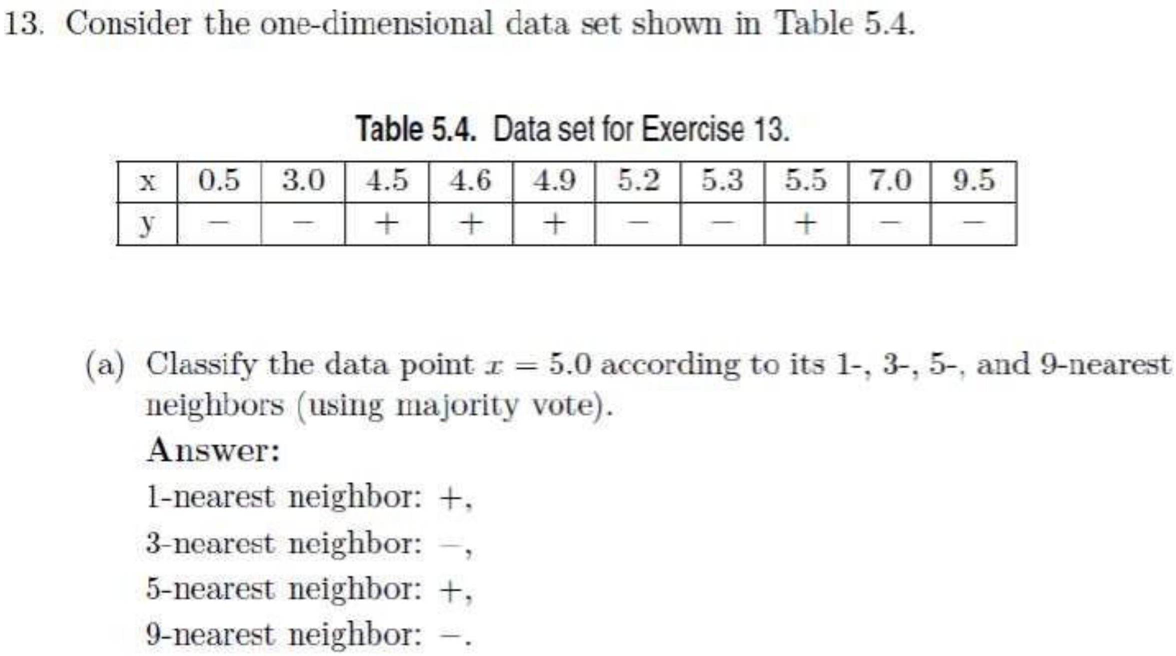
Characteristics of Nearest-Neighbor Classifiers:

○ Nearest-neighbor classification is part of a more general technique known as instance-based learning, which uses specific training instances to make predictions without having to maintain an abstraction (or model) derived from data. Instance-based learning algorithms require a proximity measure to determine the similarity or distance between instances and a classification function that returns the predicted class of a test instance based on its proximity to other instances.

○ Lazy learners such as nearest-neighbor classifiers do not require model building. However, classifying a test example can be quite expensive because we need to compute the proximity values individually between the test and training examples.

○ Nearest-neighbor classifiers can produce arbitrarily shaped decision boundaries. Such boundaries provide a more flexible model representation compared to decision tree and rule-based classifiers that are often constrained to rectilinear decision boundaries.

○ Nearest-neighbor classifiers can produce wrong predictions unless the appropriate proximity measure and data preprocessing steps are taken.



13. Consider the one-dimensional data set shown in Table 5.4.

Table 5.4. Data set for Exercise 13.

x	0.5	3.0	4.5	4.6	4.9	5.2	5.3	5.5	7.0	9.5
y	−	−	+	+	+	−	−	+	−	−

(a) Classify the data point $x = 5.0$ according to its 1-, 3-, 5-, and 9-nearest neighbors (using majority vote).

Answer:

1-nearest neighbor: +,

3-nearest neighbor: −,

5-nearest neighbor: +,

9-nearest neighbor: −.

4.6 Bayesian Classifiers

Bayes' Theorem:

Bayes‟ theorem is a way to figure out conditional probability. Conditional probability is the probability of an event happening, given that it has some relationship to one or more other events

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

Bayes' Theorem Problems Example #1

In a particular pain clinic, 10% of patients are prescribed narcotic pain killers. Overall, five percent of the clinic‟s patients are addicted to narcotics (including pain killers and illegal substances). Out of all the people prescribed pain pills, 8% are addicts. *If a patient is an addict, what is the probability that they will be prescribed pain pills?*

Step 1: **Figure out what your event "A" is from the question.** That information is in the italicized part of this particular question. The event that happens first (A) is being prescribed pain pills. That‟s given as 10%.

Step 2: **Figure out what your event "B" is from the question.** That information is also in the italicized part of this particular question. Event B is being an addict. That‟s given as 5%.

Step 3: **Figure out what the probability of event B (Step 2) given event A (Step 1)**. In other words, find what (B|A) is. We want to know "Given that people are prescribed pain pills, what‟s the probability they are an addict?" That is given in the question as 8%, or .8.

Step 4: **Insert your answers from Steps 1, 2 and 3 into the formula and solve.**

P(A|B) = P(B|A) * P(A) / P(B) = (0.08 * 0.1)/0.05 = 0.16

The probability of an addict being prescribed pain pills is 0.16 (16%).

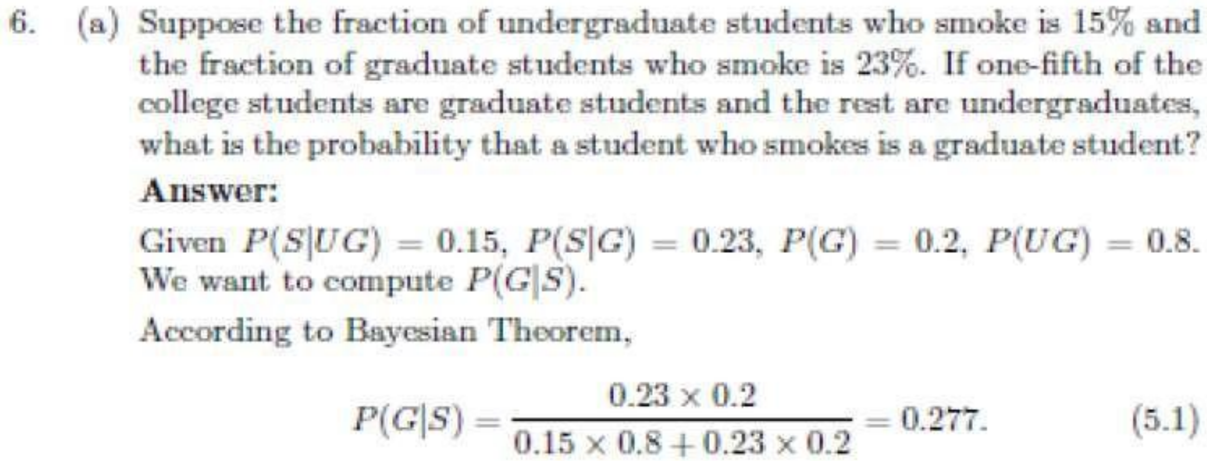
Bayes' Theorem Problems Example #2

Given:

– A doctor knows that meningitis causes stiff neck 50% of the time

- Prior probability of any patient having meningitis is 1/50,000
- Prior probability of any patient having stiff neck is 1/20

If a patient has stiff neck, what‟s the probability he/she has meningitis?



6. (a) Suppose the fraction of undergraduate students who smoke is 15% and the fraction of graduate students who smoke is 23%. If one-fifth of the college students are graduate students and the rest are undergraduates, what is the probability that a student who smokes is a graduate student?

Answer:

Given $P(S|UG) = 0.15$, $P(S|G) = 0.23$, $P(G) = 0.2$, $P(UG) = 0.8$. We want to compute $P(G|S)$.

According to Bayesian Theorem,

$$P(G|S) = \frac{0.23 \times 0.2}{0.15 \times 0.8 + 0.23 \times 0.2} = 0.277. \qquad (5.1)$$

Using the Bayes Theorem for Classification:

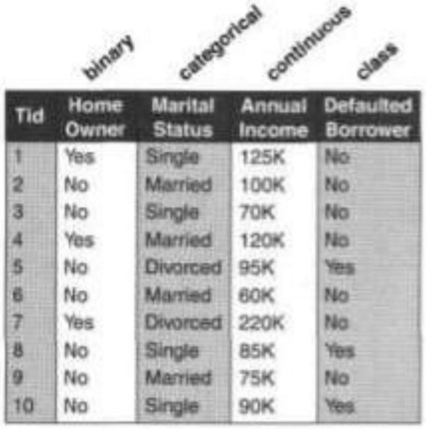
Let X denote the attribute set and Y denote the class variable. If the class variable has a non- deterministic relationship with the attributes, then we can treat X and Y as random variables and capture their relationship probabilistically using P(Y/X). This conditional probability is also known as the posterior probability for Y, as opposed to its prior probability, P(Y).

During the training phase, we need to learn the posterior probabilities P(Y/X) for every combination of X and Y based on information gathered from the training data.

By knowing these probabilities, a test record X' can be classified by finding the class Y’ that maximizes the posterior probability, P(Y'/X').

To illustrate this approach, consider the task of predicting whether a loan borrower will default on their payments.

Figure 5.9 shows a training set with the following attributes: House Owner, Marital Status, and Annual Income. Loan borrowers who defaulted on their payments are classified as Yes, while those who repaid their loans are classified as No

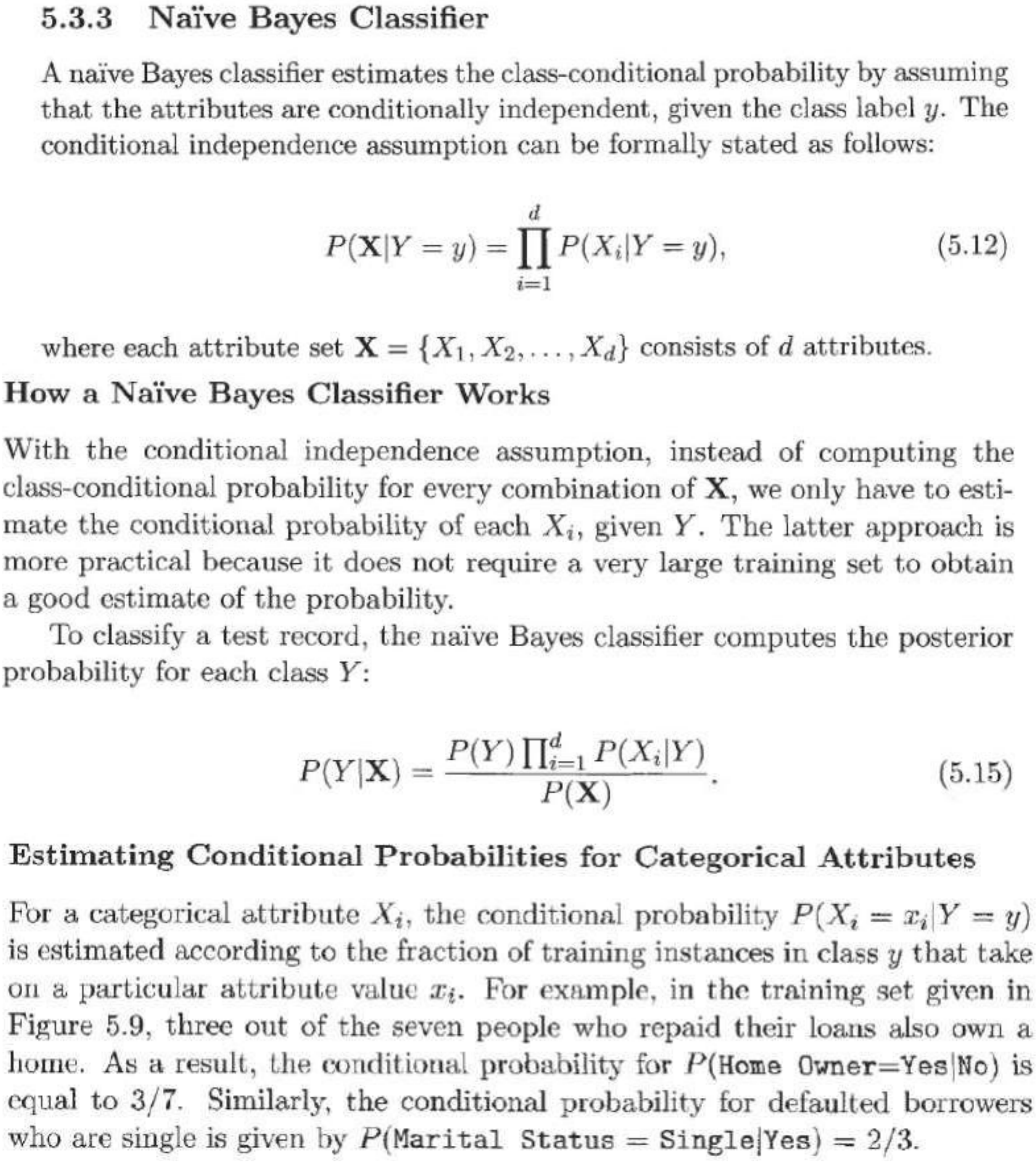


	binary	categorical	continuous	class
Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Figure 5.9. Training set for predicting the loan default problem.

Suppose we are given a test record with the following attribute set:

X :(Home Owner : No, Marital Status : Married, Annual Income : $120K).
To classify the record, we need to compute the posterior probabilities P(Yes/X) and P(No/X) based on information available in the training data.
If **P(Yes/X) > P(No/X),** then the record is classified as Yes; otherwise, it is classified as No



5.3.3 Naïve Bayes Classifier

A naïve Bayes classifier estimates the class-conditional probability by assuming that the attributes are conditionally independent, given the class label y. The conditional independence assumption can be formally stated as follows:

$$P(\mathbf{X}|Y=y)=\prod_{i=1}^{d}P(X_i|Y=y), \qquad (5.12)$$

where each attribute set $\mathbf{X}=\{X_1,X_2,\ldots,X_d\}$ consists of d attributes.

How a Naïve Bayes Classifier Works

With the conditional independence assumption, instead of computing the class-conditional probability for every combination of $\mathbf{X}$, we only have to estimate the conditional probability of each X_i, given Y. The latter approach is more practical because it does not require a very large training set to obtain a good estimate of the probability.

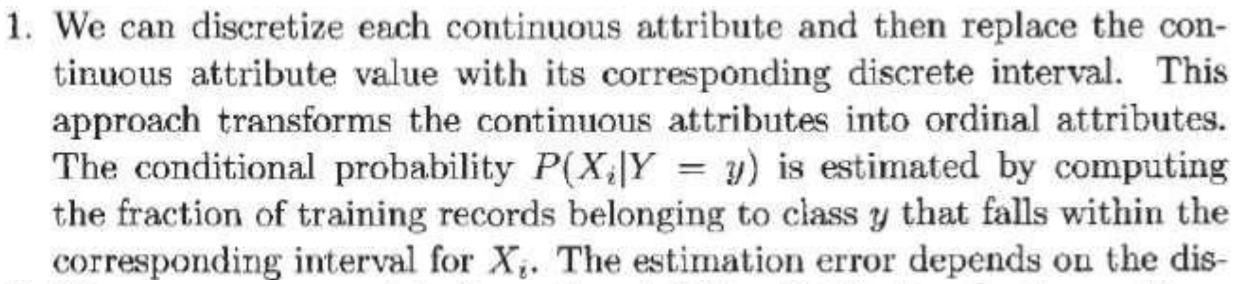
To classify a test record, the naïve Bayes classifier computes the posterior probability for each class Y:

$$P(Y|\mathbf{X})=\frac{P(Y)\prod_{i=1}^{d}P(X_i|Y)}{P(\mathbf{X})}. \qquad (5.15)$$

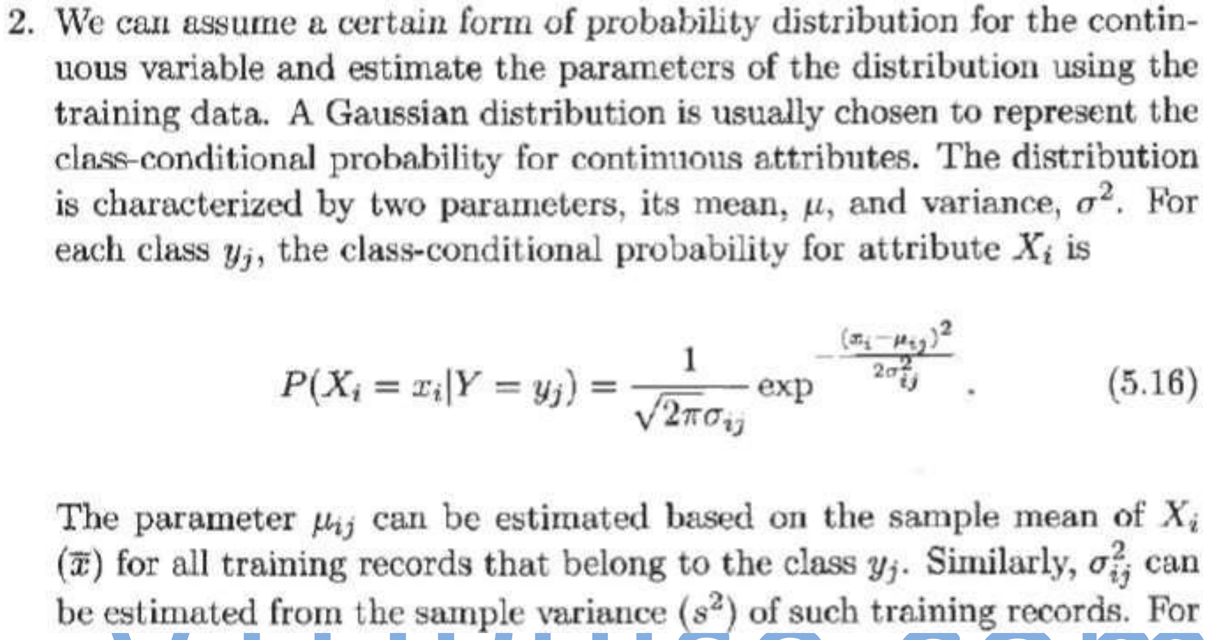
Estimating Conditional Probabilities for Categorical Attributes

For a categorical attribute X_i, the conditional probability $P(X_i=x_i|Y=y)$ is estimated according to the fraction of training instances in class y that take on a particular attribute value x_i. For example, in the training set given in Figure 5.9, three out of the seven people who repaid their loans also own a home. As a result, the conditional probability for $P(\texttt{Home Owner=Yes}|\texttt{No})$ is equal to 3/7. Similarly, the conditional probability for defaulted borrowers who are single is given by $P(\texttt{Marital Status = Single}|\texttt{Yes}) = 2/3$.

Estimating Conditional Probabilities for Continuous Attributes:
There are two ways to estimate the class-conditional probabilities for continuous Attributes in naive Bayes classifiers:



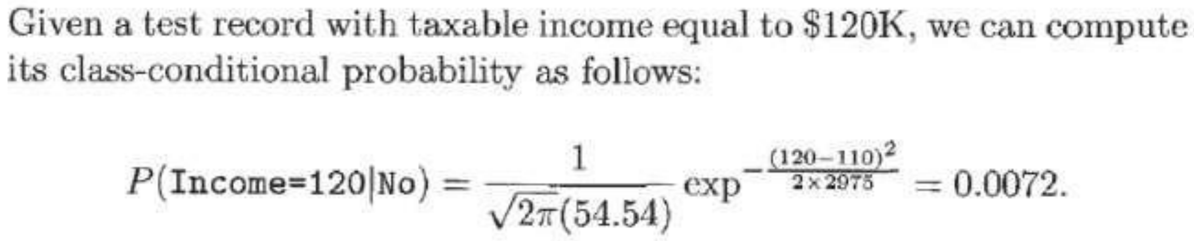
1. We can discretize each continuous attribute and then replace the continuous attribute value with its corresponding discrete interval. This approach transforms the continuous attributes into ordinal attributes. The conditional probability $P(X_i|Y = y)$ is estimated by computing the fraction of training records belonging to class y that falls within the corresponding interval for X_i. The estimation error depends on the dis-



2. We can assume a certain form of probability distribution for the continuous variable and estimate the parameters of the distribution using the training data. A Gaussian distribution is usually chosen to represent the class-conditional probability for continuous attributes. The distribution is characterized by two parameters, its mean, μ, and variance, σ^2. For each class y_j, the class-conditional probability for attribute X_i is

$$P(X_i = x_i|Y = y_j) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} \exp^{-\frac{(x_i-\mu_{ij})^2}{2\sigma_{ij}^2}}. \qquad (5.16)$$

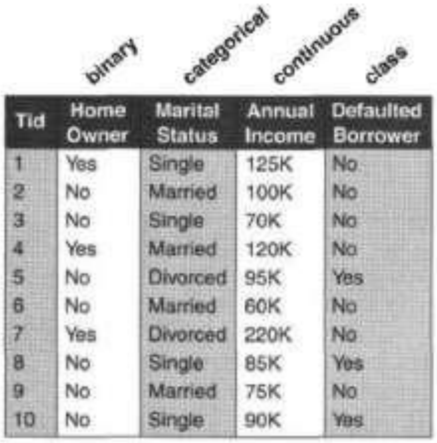
The parameter μ_{ij} can be estimated based on the sample mean of X_i ($\overline{x}$) for all training records that belong to the class y_j. Similarly, σ_{ij}^2 can be estimated from the sample variance (s^2) of such training records. For



Given a test record with taxable income equal to \$120K, we can compute its class-conditional probability as follows:

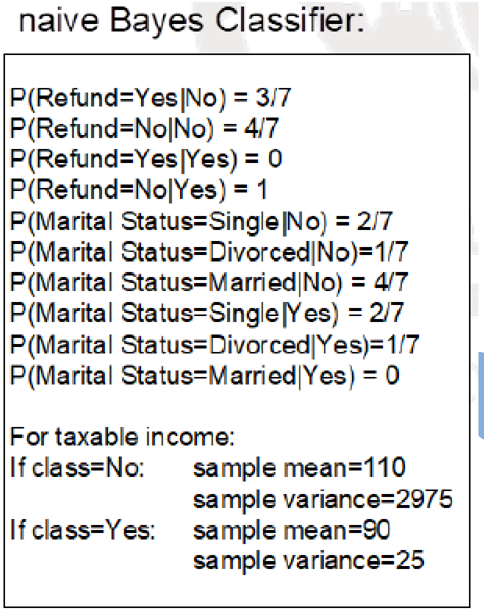
$$P(\texttt{Income=120}|\texttt{No}) = \frac{1}{\sqrt{2\pi}(54.54)} \exp^{-\frac{(120-110)^2}{2\times 2975}} = 0.0072.$$

Example of Naïve Bayes Classifier: Consider the training record


binary
categorical
continuous
class

Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

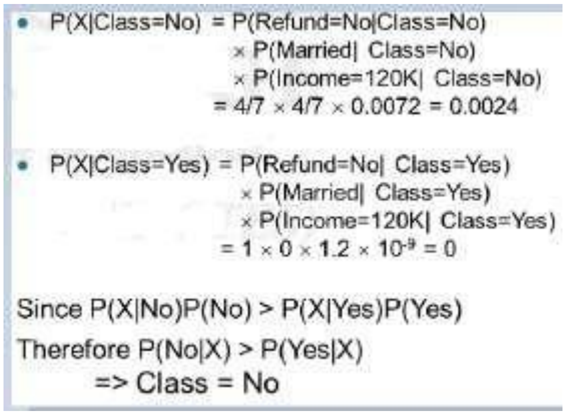
Figure 5.9. Training set for predicting the loan default problem.



naive Bayes Classifier:

P(Refund=Yes|No) = 3/7
P(Refund=No|No) = 4/7
P(Refund=Yes|Yes) = 0
P(Refund=No|Yes) = 1
P(Marital Status=Single|No) = 2/7
P(Marital Status=Divorced|No)=1/7
P(Marital Status=Married|No) = 4/7
P(Marital Status=Single|Yes) = 2/7
P(Marital Status=Divorced|Yes)=1/7
P(Marital Status=Married|Yes) = 0

For taxable income:
If class=No: sample mean=110
sample variance=2975
If class=Yes: sample mean=90
sample variance=25



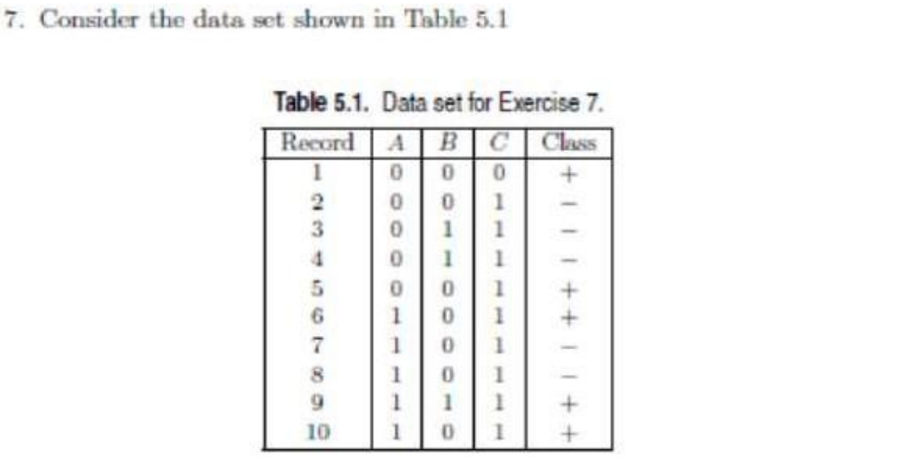
- P(X|Class=No) = P(Refund=No|Class=No) × P(Married| Class=No) × P(Income=120K| Class=No) = 4/7 × 4/7 × 0.0072 = 0.0024
- P(X|Class=Yes) = P(Refund=No| Class=Yes) × P(Married| Class=Yes) × P(Income=120K| Class=Yes) = 1 × 0 × 1.2 × 10^{-9} = 0

Since P(X|No)P(No) > P(X|Yes)P(Yes)
Therefore P(No|X) > P(Yes|X)
=> Class = No

M-estimate of Conditional Probability:

$$P(x_i|y_j) = \frac{n_c + mp}{n + m},$$

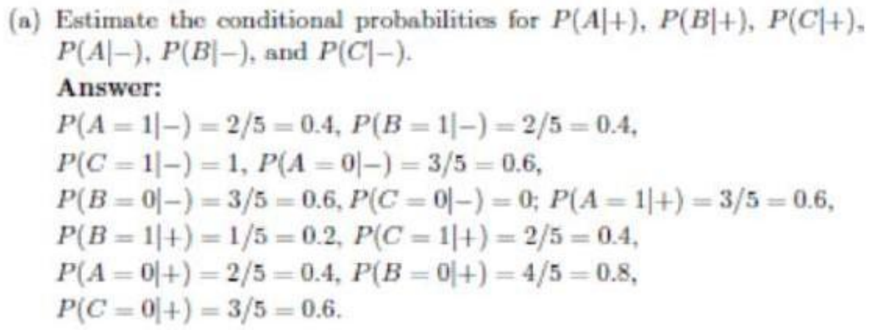
where n is the total number of instances from class Yj, nc is the number of training examples from class Yi that take on the value Xi, m is a parameter known as the equivalent sample size, and p is a user-specified parameter.



7. Consider the data set shown in Table 5.1

Table 5.1. Data set for Exercise 7.

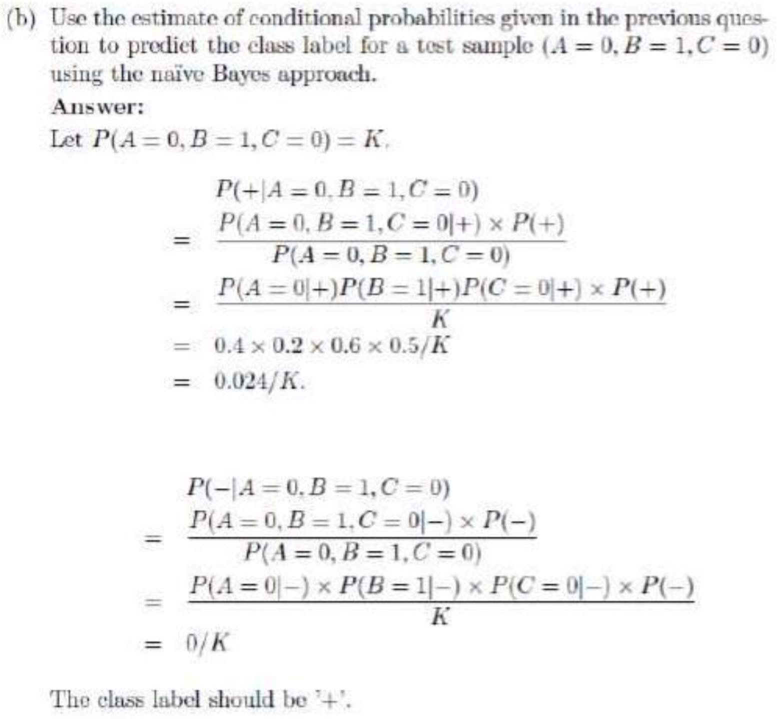
Record	A	B	C	Class
1	0	0	0	+
2	0	0	1	−
3	0	1	1	−
4	0	1	1	−
5	0	0	1	+
6	1	0	1	+
7	1	0	1	−
8	1	0	1	−
9	1	1	1	+
10	1	0	1	+



(a) Estimate the conditional probabilities for $P(A|+)$, $P(B|+)$, $P(C|+)$, $P(A|-)$, $P(B|-)$, and $P(C|-)$.

Answer:

$P(A = 1|-) = 2/5 = 0.4$, $P(B = 1|-) = 2/5 = 0.4$,
$P(C = 1|-) = 1$, $P(A = 0|-) = 3/5 = 0.6$,
$P(B = 0|-) = 3/5 = 0.6$, $P(C = 0|-) = 0$; $P(A = 1|+) = 3/5 = 0.6$,
$P(B = 1|+) = 1/5 = 0.2$, $P(C = 1|+) = 2/5 = 0.4$,
$P(A = 0|+) = 2/5 = 0.4$, $P(B = 0|+) = 4/5 = 0.8$,
$P(C = 0|+) = 3/5 = 0.6$.



(b) Use the estimate of conditional probabilities given in the previous question to predict the class label for a test sample $(A = 0, B = 1, C = 0)$ using the naïve Bayes approach.

Answer:

Let $P(A = 0, B = 1, C = 0) = K$.

$$\begin{aligned}
& P(+|A = 0, B = 1, C = 0) \\
= & \frac{P(A = 0, B = 1, C = 0|+) \times P(+)}{P(A = 0, B = 1, C = 0)} \\
= & \frac{P(A = 0|+)P(B = 1|+)P(C = 0|+) \times P(+)}{K} \\
= & 0.4 \times 0.2 \times 0.6 \times 0.5/K \\
= & 0.024/K.
\end{aligned}$$

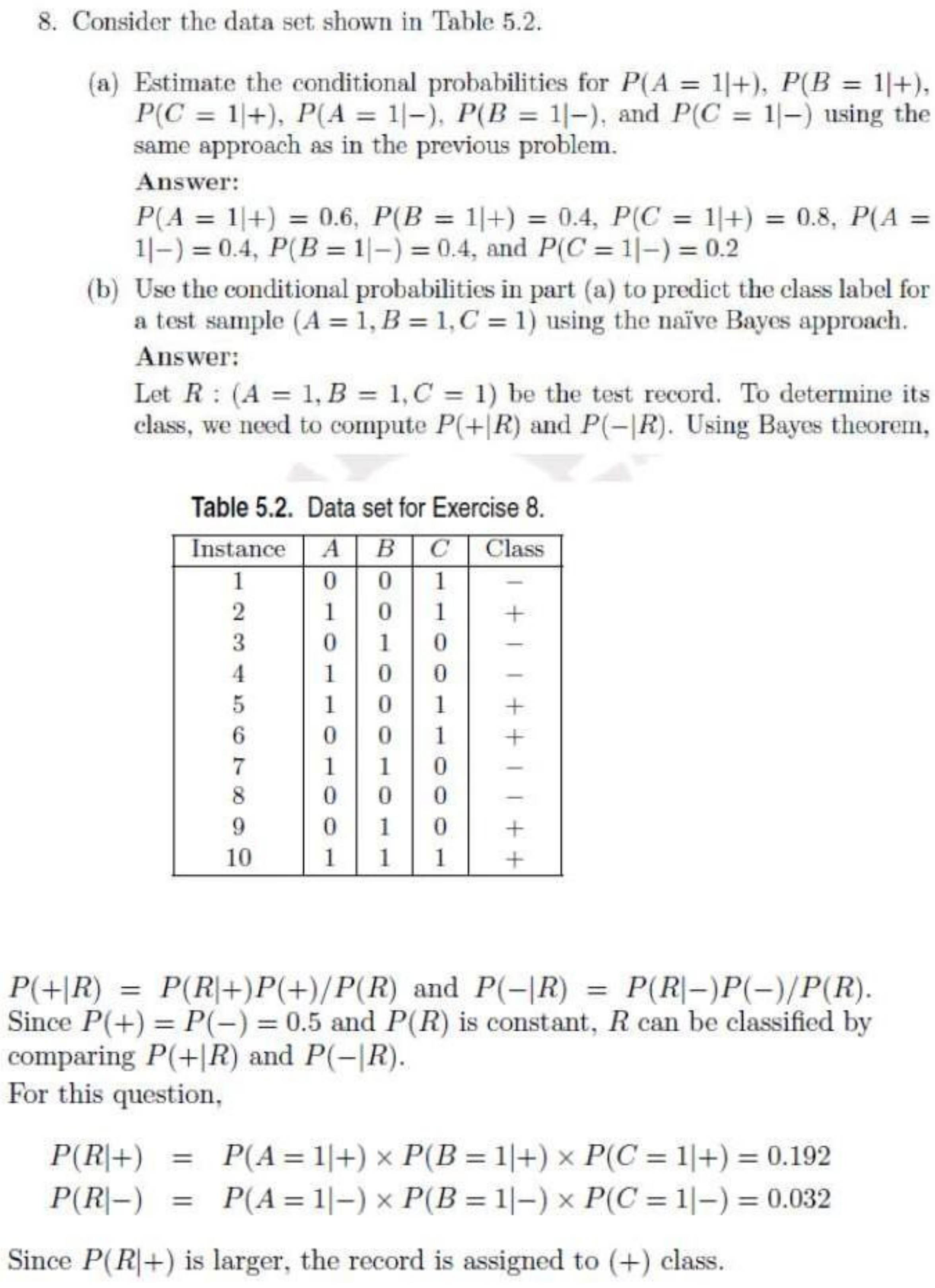
$$\begin{aligned}
& P(-|A = 0, B = 1, C = 0) \\
= & \frac{P(A = 0, B = 1, C = 0|-) \times P(-)}{P(A = 0, B = 1, C = 0)} \\
= & \frac{P(A = 0|-) \times P(B = 1|-) \times P(C = 0|-) \times P(-)}{K} \\
= & 0/K
\end{aligned}$$

The class label should be '+'.

(c) Estimate the conditional probabilities using the m-estimate approach, with $p = 1/2$ and $m = 4$.

Answer:

$P(A = 0|+) = (2 + 2)/(5 + 4) = 4/9$,
$P(A = 0|-) = (3 + 2)/(5 + 4) = 5/9$,
$P(B = 1|+) = (1 + 2)/(5 + 4) = 3/9$,
$P(B = 1|-) = (2 + 2)/(5 + 4) = 4/9$,
$P(C = 0|+) = (3 + 2)/(5 + 4) = 5/9$,
$P(C = 0|-) = (0 + 2)/(5 + 4) = 2/9$.



8. Consider the data set shown in Table 5.2.

(a) Estimate the conditional probabilities for $P(A = 1|+)$, $P(B = 1|+)$, $P(C = 1|+)$, $P(A = 1|-)$, $P(B = 1|-)$, and $P(C = 1|-)$ using the same approach as in the previous problem.

Answer:

$P(A = 1|+) = 0.6$, $P(B = 1|+) = 0.4$, $P(C = 1|+) = 0.8$, $P(A = 1|-) = 0.4$, $P(B = 1|-) = 0.4$, and $P(C = 1|-) = 0.2$

(b) Use the conditional probabilities in part (a) to predict the class label for a test sample $(A = 1, B = 1, C = 1)$ using the naïve Bayes approach.

Answer:

Let $R : (A = 1, B = 1, C = 1)$ be the test record. To determine its class, we need to compute $P(+|R)$ and $P(-|R)$. Using Bayes theorem,

Table 5.2. Data set for Exercise 8.

Instance	A	B	C	Class
1	0	0	1	−
2	1	0	1	+
3	0	1	0	−
4	1	0	0	−
5	1	0	1	+
6	0	0	1	+
7	1	1	0	−
8	0	0	0	−
9	0	1	0	+
10	1	1	1	+

$P(+|R) = P(R|+)P(+)/P(R)$ and $P(-|R) = P(R|-)P(-)/P(R)$. Since $P(+) = P(-) = 0.5$ and $P(R)$ is constant, R can be classified by comparing $P(+|R)$ and $P(-|R)$.

For this question,

$$\begin{aligned} P(R|+) &= P(A = 1|+) \times P(B = 1|+) \times P(C = 1|+) = 0.192 \\ P(R|-) &= P(A = 1|-) \times P(B = 1|-) \times P(C = 1|-) = 0.032 \end{aligned}$$

Since $P(R|+)$ is larger, the record is assigned to $(+)$ class.

Characteristics of Naive Bayes Classifiers:

Naive Bayes classifiers generally have the following characteristics:

- They are robust to isolated noise points because such points are averaged out when estimating conditional probabilities from data.

Naive Bayes classifiers can also handle missing values by ignoring the example during model building and classification.

- They are robust to irrelevant attributes.

- Correlated attributes can degrade the performance of naive Bayes classifiers because the conditional independence assumption no longer holds for such attributes.

Bayesian Belief Networks

Bayesian networks represent an advanced form of general Bayesian probability

A Bayesian network is a graphical model that encodes probabilistic relationships among variables of interest.

A Bayesian belief network (BBN), or simply, Bayesian network, provides a graphical representation of the probabilistic relationships among a set of random variables. There are two key elements of a Bayesian network:

1. A directed acyclic graph (dag) encoding the dependence relationships among a set of variables.
2. A probability table associating each node to its immediate parent nodes.

Consider three random variables,A, B, and C, in which A and B are independent variables and each has a direct influence on a third variable, C.

The relationships among the variables can be summarized into the directed acyclic graph shown in Figure 5.12(a).

Each node in the graph represents a variable, and each arc asserts the dependence relationship between the pair of variables. If there is a directed arc from X to Y, then X is the parent of Y and Y is the child of X. F\rrthermore, if there is a directed path in the network from X to Z, then X is an ancestor of Z, whlle Z is a descendant of X.

For example, in the diagram shown in Figure 5.12(b), A is a descendant of D and D is an ancestor of B. Both B and D arc also non-descendants of A.

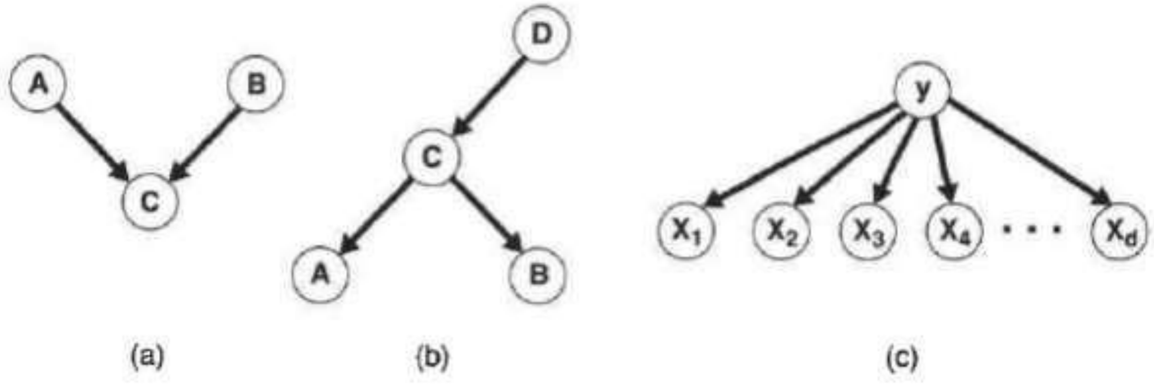


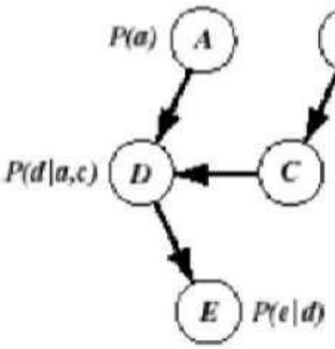
Figure 5.12. Representing probabilistic relationships using directed acyclic graphs.

In the diagram shown in Figure 5.12(b), A is conditionally independent of both B and D given C because the nodes for B and D are non-descendants of node A.

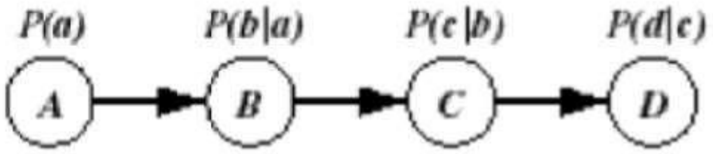
The conditional independence assumption made by a naive Bayes classifier can also be represented using a Bayesian network, as shown in Figure 5.12(c), where gr is the target class and {Xt,Xz,...,Xa} is the attribute set.

Besides the conditional independence conditions imposed by the network topology, each node is also associated with a probability table.

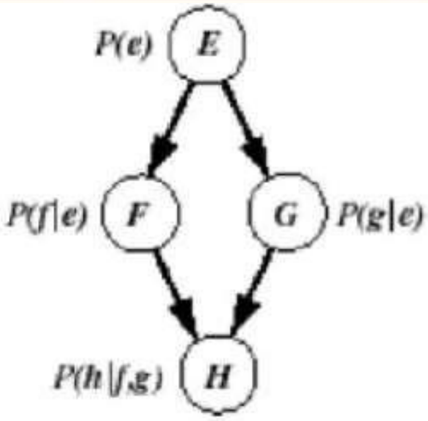
1. If a node X does not have any parents, then the table contains only the prior probability P(X).
2. If a node X has only one parent, Y, then the table contains the conditional probability P(XIY).
3. If a node X has multiple parents, {Y1,,Y2, . . . ,Yn}, then the table contains the conditional probability P(XlY1,Y2,. . ., Yn.).



P(a, b, c, d, e) =P(a)P(b)P(c|b)P(d|a,c)P(e|d)



P(a, b, c, d) = P(a)P(b|a)P(c|b)P(d|c)

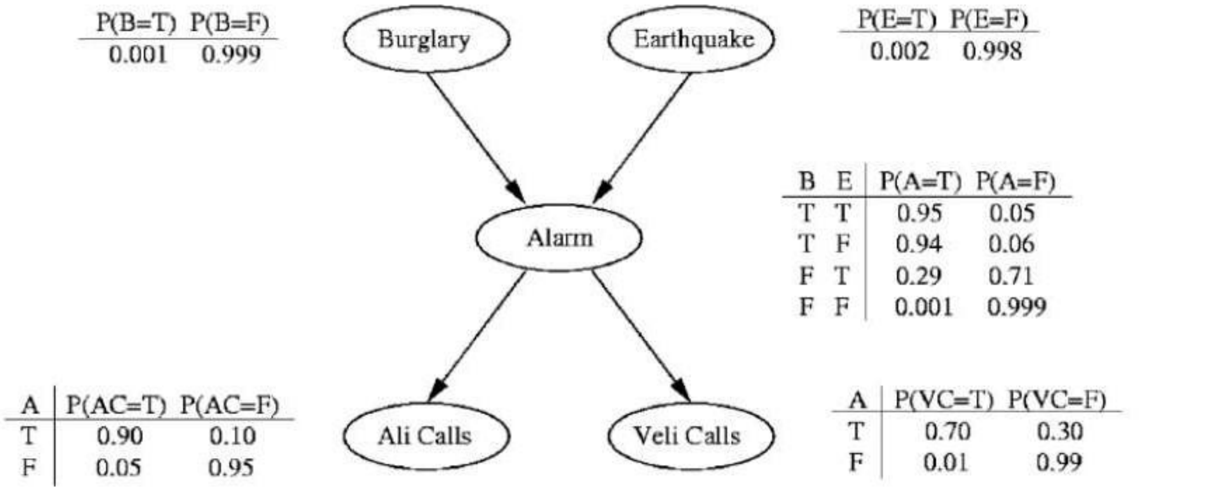


P(e, f, g, h) =P(e)P(f|e)P(g|e)P(h|f, g)

EXAMPLE:

You have a new burglar alarm installed at home.

- It is fairly reliable at detecting burglary, but also sometimes responds to minor earthquakes.
- You have two neighbors, Ali and Veli, who promised to call you at work when they hear the alarm.
- Ali always calls when he hears the alarm, but sometimes confuses telephone ringing with the alarm and calls too.
- Veli likes loud music and sometimes misses the alarm.
- Given the evidence of who has or has not called, we would like to estimate the probability of a burglary.



P(B=T)	P(B=F)
0.001	0.999

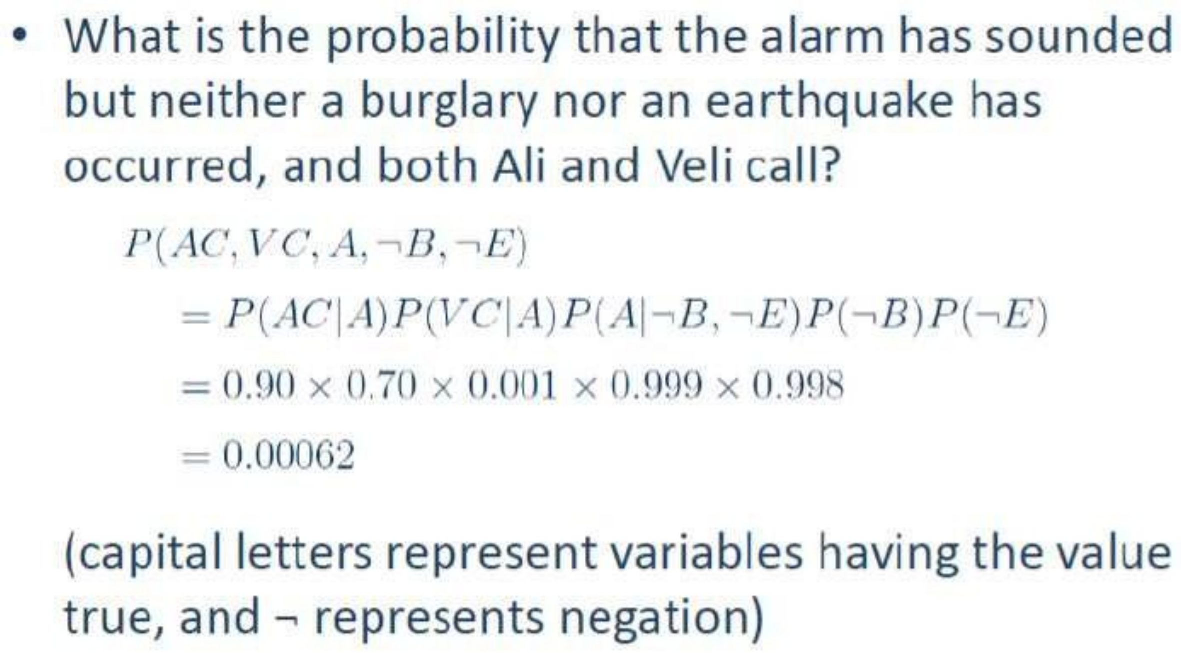
P(E=T)	P(E=F)
0.002	0.998

B	E	P(A=T)	P(A=F)
T	T	0.95	0.05
T	F	0.94	0.06
F	T	0.29	0.71
F	F	0.001	0.999

A	P(AC=T)	P(AC=F)
T	0.90	0.10
F	0.05	0.95

A	P(VC=T)	P(VC=F)
T	0.70	0.30
F	0.01	0.99

The Bayesian network for the burglar alarm example. Burglary (B) and earthquake (E) directly affect the probability of the alarm (A) going off, but whether or not Ali calls (AC) or Veli calls (VC) depends only on the alarm.
(Russell and Norvig, Artificial Intelligence: A Modern Approach, 1995)



- What is the probability that the alarm has sounded but neither a burglary nor an earthquake has occurred, and both Ali and Veli call?

$$
\begin{aligned}
&P(AC, VC, A, \neg B, \neg E)\\
&\quad = P(AC|A)P(VC|A)P(A|\neg B, \neg E)P(\neg B)P(\neg E)\\
&\quad = 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998\\
&\quad = 0.00062
\end{aligned}
$$

(capital letters represent variables having the value true, and ¬ represents negation)

Characteristics of BBN

Following are some of the general characteristics of the BBN method:

- BBN provides an approach for capturing the prior knowledge of a particular domain using a graphical model. The network can also be used to encode causal dependencies among variables.

- Constructing the network can be time consuming and requires a large amount of effort. However, once the structure of the network has been determined, adding a new variable is quite straightforward.

- Bayesian networks are well suited to dealing with incomplete data. Instances with missing attributes can be handled by summing or integrating the probabilities over all possible values of the attribute.

- Because the data is combined probabilistically with prior knowledge, the method is quite robust to model over fitting.

4.7 Important Questions:

1. What is classification. Explain the general approach for solving a classification problem with an example.
2. How decision trees are used for classification. Explain decision tree induction algorithm for classification.
3. Write Hunts algorithm and illustrate it‟s working.
4. Explain the Methods for Expressing Attribute Test Conditions.
5. Explain various measures for selecting the best split with an example.
6. Explain the importance of evaluation criterion for classification methods.
7. Explain the characteristics of decision tree Induction.
8. Explain Model Over fitting. What are the reasons for overfitting? How to address overfitting problems

9. Explain how to estimate generalization errors.
10. List characteristics of decision tree induction.
11. Give the difference between rule-based ordering and class-based ordering scheme.
12. Explain rule-based classifier and its characteristics.
13. Explain the characteristics of rule based classifier
14. How to improve accuracy of classification. Explain
15. Explain k-nearest neighbor classification algorithm.
16. Explain any characteristics of the nearest neighbor classifier.
17. What is Baye‟s theorem? Show how it is used for classification.
18. Explain with an example how naïve Baye „s algorithm used for classification.
20. Discuss the two common strategies for growing a classification rule.
21. Explain sequential covering algorithm for rule extraction.
22. Explain model building in Bayesian networks.