CHAPTER 20

Data Warehousing and Mining

This chapter covers data warehousing and data mining. Considering the growing importance of all the topics covered in this chapter, some of the sections of the chapter can be assigned as supplementary reading material, even in an introductory course. The material in the chapter is also suitable for laying the groundwork for an advanced course, or for professionals to keep in touch with recent developments.

Exercises

20.6 Draw a diagram that shows how the *classroom* relation of our university example as shown in Appendix A would be stored under a column-oriented storage structure.

Answer: The relation would be stored in three files, one per attribute, as shown below. We assume that the row number can be inferred implicitly from position, by using fixed size space for each attribute. Otherwise, the row number would also have to be stored explicitly.

building
Packard
Painter
Taylor
Watson
Watson

room_number
101
514
3128
100
120

capacity
500
10
70
30
50

20.7 Explain why the nested-loops join algorithm (see Section 12.5.1) would work poorly on database stored in a column-oriented manner. Describe an alternative algorithm that would work better and explain why your solution is better.

Answer: If the nested-loops join algorithm is used as is, it would require tuples for each of the relations to be assembled before they are joined. Assembling tuples can be expensive in a column store, since each attribute may come from a separate area of the disk; the overhead of assembly would be particularly wasteful if many tuples do not satisfy the join condition and would be discarded. In such a situation it would be better to first find which tuples match by accessing only the join columns of the relations. Sort-merge join, hash join, or indexed nested loops join can be used for this task. After the join is performed, only tuples that get output by the join need to be assembled; assembly can be done by sorting the join result on the record identifier of one of the relations and accessing the corresponding attributes, then resorting on record identifiers of the other relation to access its attributes.

20.8 Construct a decision-tree classifier with binary splits at each node, using tuples in relation $r(A, B, C)$ shown below as training data; attribute C denotes the class. Show the final tree, and with each node show the best split for each attribute along with its information gain value.

(1, 2, a), (2, 1, a), (2, 5, b), (3, 3, b), (3, 6, b),
(4, 5, b), (5, 5, c), (6, 3, b), (6, 7, c)

Answer: Figure 20.1 shows one possible decision tree for the data. Using the Gini purity metric, the purity of the initial data set is

$$1 - \sum_{i-1}^{k} p_i^2 = 1 - ((\frac{2}{9})^2 + (\frac{5}{9})^2 + (\frac{2}{9})^2) = 0.595259$$

The first branch splits on $B <= 2$, giving a purity score of $1 - 1^2 = 0$ for those attributes with $B <= 2$ (all are classified as a), and a purity score of

$$1 - ((\frac{2}{7})^2 + (\frac{5}{7})^2) = 0.40816$$

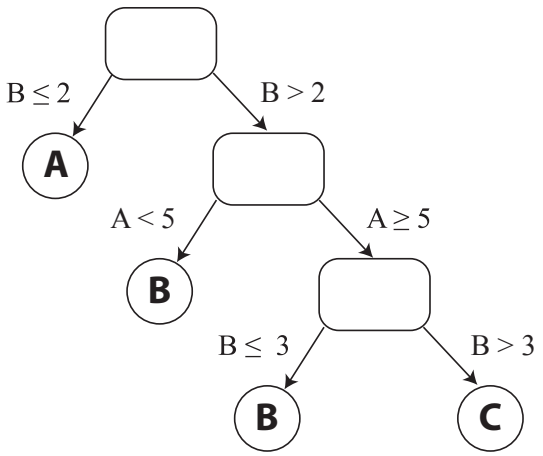

Figure 20.1 Decision tree for data on relation $r(A, B, C)$

for the remaining items. The weighted purity of the entire set is

$$\frac{2}{9} * 0 + \frac{7}{9} * 0.40816 = 0.31746$$

The information gain from this split is $0.595259 - 0.31746 = 0.27513$. Next, we split on $A < 5$. The 4 data items with $A < 5$ all have class b, and thus have purity 0. The remaining 3 items have purity

$$1 - ((\frac{1}{3})^2 + (\frac{2}{3})^2) = 0.44444$$

The weighted purity of these sets is

$$\frac{4}{7} * 0 + \frac{3}{7} * 0.44444 = 0.19048$$

The information gain from the second split is $0.40816 - 0.19048 = 0.21769$. Finally, we split on $B <= 3$. One data item satisfies this predicate and has class b. The other two items both have class c. The purity of these two sets is 0. The information gain from the final split is $0.21769 - 0 = 0.21769$.

20.9 Suppose half of all the transactions in a clothes shop purchase jeans, and one third of all transactions in the shop purchase T-shirts. Suppose also that half of the transactions that purchase jeans also purchase T-shirts. Write down all the (nontrivial) association rules you can deduce from the above information, giving support and confidence of each rule.

Answer: The rules are as follows. The last rule can be deduced from the previous ones.

Rule	Support	Conf.
$\forall\, transactions\ T,\ \textbf{true} \Rightarrow buys(T, jeans)$	50%	50%
$\forall\, transactions\ T,\ \textbf{true} \Rightarrow buys(T, t\text{-}shirts)$	33%	33%
$\forall\, transactions\ T,\ buys(T, jeans) \Rightarrow buys(T, t\text{-}shirts)$	25%	50%
$\forall\, transactions\ T,\ buys(T, t\text{-}shirts) \Rightarrow buys(T, jeans)$	25%	75%

20.10 Consider the problem of finding large itemsets.

a. Describe how to find the support for a given collection of itemsets by using a single scan of the data. Assume that the itemsets and associated information, such as counts, will fit in memory.

b. Suppose an itemset has support less than j. Show that no superset of this itemset can have support greater than or equal to j.

Answer:

a. Let $\{S_1, S_2, \ldots, S_n\}$ be the collection of item-sets for which we want to find the support. Associate a counter $count(S_i)$ with each item-set S_i.
Initialize each counter to zero. Now examine the transactions one-by-one. Let $S(T)$ be the item-set for a transaction T. For each item-set S_i that is a subset of $S(T)$, increment the corresponding counter $count(S_i)$.
When all the transactions have been scanned, the values of $count(S_i)$ for each i will give the support for item-set S_i.

b. Let A be an item-set. Consider any item-set B which is a superset of A. Let τ_A and τ_B be the sets of transactions that purchase all items in A and all items in B, respectively. For example, suppose A is $\{a, b, c\}$, and B is $\{a, b, c, d\}$.
A transaction that purchases all items from B must also have purchased all items from A (since $A \subseteq B$). Thus, every transaction in τ_B is also in τ_A. This implies that the number of transactions in τ_B is at most the number of transactions in τ_A. In other words, the support for B is at most the support for A.
Thus, if any item-set has support less than j, all supersets of this item-set have support less than j.

20.11 Create a small example of a set of transactions showing that although many transactions contain two items, that is, the itemset containing the two items has a high support, purchase of one of the items may have a negative correlation with purchase of the other.

Answer: The following set of transactions involve fruit purchases:

Transaction_ID	Item
T-1	orange
T-1	banana
T-1	apple
T-2	orange
T-2	banana
T-3	orange
T-3	apple
T-4	orange
T-4	banana
T-4	grapes
T-5	banana
T-5	apple
T-6	banana
T-6	grapes

Consider the association rule

$$orange \Rightarrow banana$$

This rule is satisfied in 3 out of the 6 transactions, so the support value is 50 percent. However, the correlation between purchasing oranges and purchasing bananas in this data set is -0.32.

20.12 The organization of parts, chapters, sections, and subsections in a book is related to clustering. Explain why, and to what form of clustering.

Answer: The organization of a book's content is a form of **hierarchical clustering**. Contents within a single subsection are closely related, whereas different parts of a book cover a more diverse range of topics.

20.13 Suggest how predictive mining techniques can be used by a sports team, using your favorite sport as an example.

Answer: Given the large amount of statistics collected during and about sporting events, there are many ways a sports team can make use of predictive data mining:

- Some players may be more effective in certain situations or environments, so data mining can predict when each player should be used.
- Specific strategies may be more effective against certain teams or during certain situations in the game.
- Predictive mining can estimate the outcome of a match beforehand, information which could be useful to a team before entering a tournament.