Data Mining
Association Rules

Mr. SHIVAM PANDEY

N.I.T.,WEST BENGAL

Association Rules:-

- Let A = $\{I_1, I_2, \ldots \ldots, I_m\}$ be a set of items. Let T, the transaction database be a set of transactions, where each transaction t is a subset of items. Thus t is a subset of A.
- Definition:- **Support:** A transaction t is said to *support* an item I_l, if I_l is present it t. t is said to support a subset of items $X \subseteq A$, if t supports each item I in X. An item set $X \subseteq A$ has a support s in T, denoted by $S(X)_T$, if s% of transactions in T support X.

- Let us consider the set of transactions in a bookshop as shown in the next slide
- We shall look at a set of only 6 transactions of purchases of books. In the first transaction, purchases are made of books on Compiler Construction (CC), Databases (D), Theory of Computations (TC), Computer Graphics (CG), and Artificial Neural Networks (ANN); We shall denote these subjects by CC, D, TC, CG and ANN, respectively. Thus we describe transactions as follows:

Fig. 1

1. t_1 = {ANN, CC, TC, CG}
2. t_2 = {CC, D, CG}
3. t_3 = {ANN, CC, TC, CG}
4. t_4 = {ANN, CC, D, CG}
5. t_5 = {ANN, CC, D, TC, CG}
6. t_6 = {CC, D, TC}

So A = {ANN, CC, D, TC, CG} and
T = {t_1, t_2, t_3, t_4, t_5, t_6}

- We can see that t_2 supports the items CC, D and CG. The item D is supported by 4 out of 6 transactions in T. Thus the *support* of D is 66.6%.
- Association Rule: For a given transaction database T, an association rule is an expression of the form $X \Rightarrow Y$, where X and Y are subsets of A and $X \Rightarrow Y$ holds the **confidence** τ, if τ% of transaction in T that support X also support Y. The rule $X \Rightarrow Y$ has **support** σ in the transaction set T if σ% of transactions in T support X U Y

- Consider the example of the bookshop. Assume that $\sigma = 50\%$ and $\tau = 60\%$. Clearly ANN $\Rightarrow$ CC holds. The confidence of this rule is, 100% because all the transactions that support ANN also support CC. On the other hand, CC $\Rightarrow$ ANN also holds, but its confidence is 66%

Problem Decomposition

- Find all sets of items, whose support is greater than the user specified minimum support, σ. Such item sets are called frequent item sets.
- Use the frequent item sets to generate the desired rules. The general idea is that if, say ABCD and AB are frequent item sets, then we can determine if the rules AB $\Rightarrow$ CD holds by checking the inequality shown in the next slide:

- $$\frac{\mathbf{s(\{A,B,C,D\})}}{\mathbf{s(\{A,B\})}} \geq \tau$$

- Where s(X) is the support of X in T.

Definitions

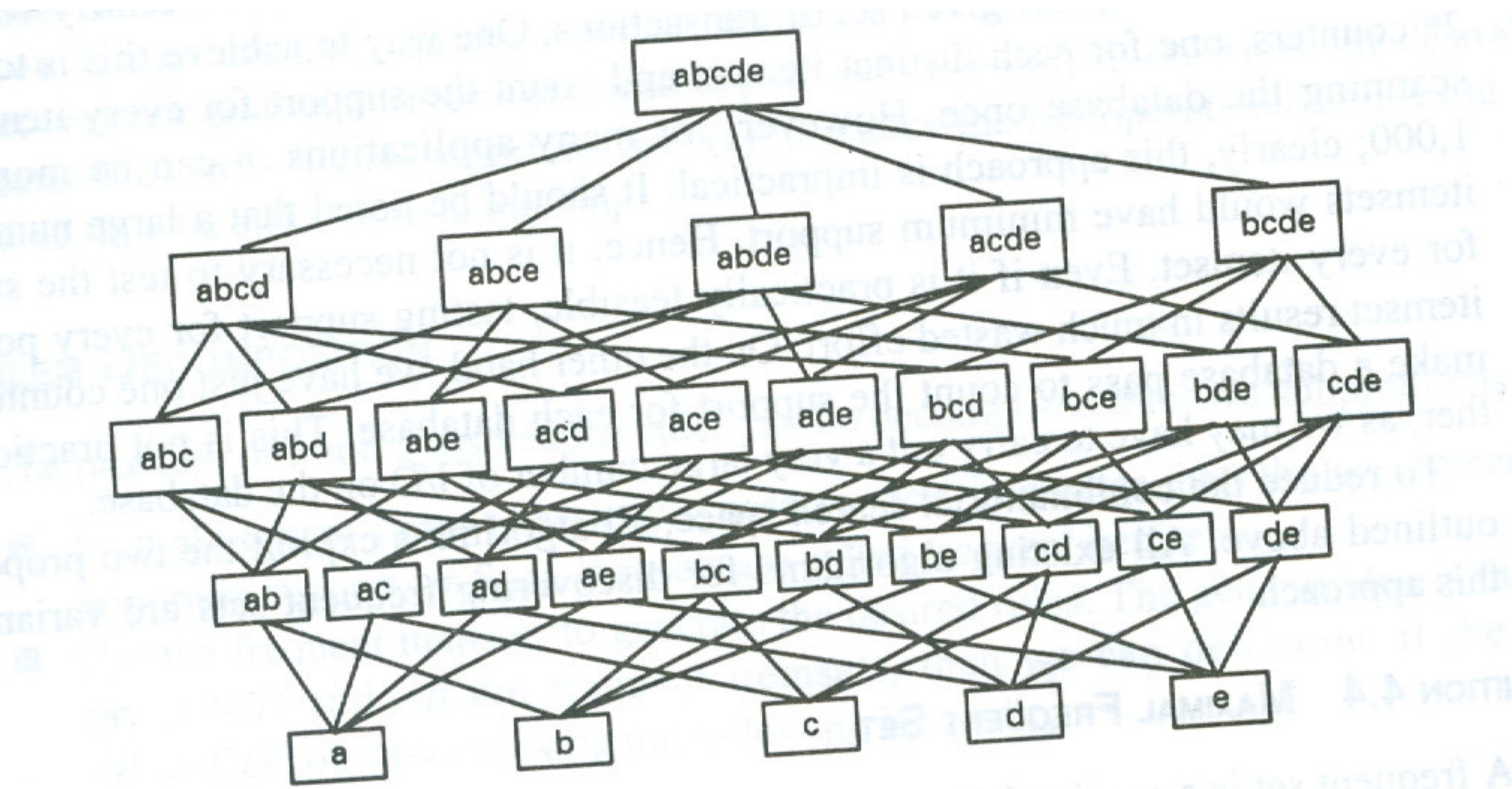
- Let T be the transaction database and σ be the user specified minimum support. An itemset $X \subseteq A$ is said to be a frequent itemset in T with respect to σ, if
- $s(X)_T \geq \sigma$
- In Example of Fig. 1, if we assume σ = 50%, then {ANN, CC, TC} is a frequent set as it is supported by at least 3 out of 6 transactions. We can see that any subset of this set is also a frequent set. On the other hand {ANN,CC,D} is not a frequent itemset and hence no set which properly contains this set is also a frequent set.

Definitions

- Maximal Frequent Set: A frequent set is a maximal frequent set if it is a frequent set and no superset of this is a frequent set.
- Border Set: An itemset is a border set if it is not a frequent set, but all its proper subsets are frequent sets.
- Note that if we know the set of all maximal frequent sets of a given T with respect to a σ, then we can find the set of all frequent sets without any extra scan of the database. Thus the set of all maximal frequent sets can act as a compact representation of the set of all frequent sets.

- We shall often refer to the lattice of subsets of A as shown in Fig. 2. For example, if A = {a,b,c,d,e}, then the lattice is given the following figure. In this lattice, the set of maximal frequent sets acts as a boundary between the set of all frequent sets and the set of all infrequent sets.

Fig. 2


abcde
abcd
abce
abde
acde
bcde
abc
abd
abe
acd
ace
ade
bcd
bce
bde
cde
ab
ac
ad
ae
bc
bd
be
cd
ce
de
a
b
c
d
e

- See Fig. 3 in the next slide
- A = {A1,A2,A3,A4,A5,A6,A7,A8,A9}. Assume **σ = 20%.** Since T contains 15 records, it means that an itemset that is supported by at least three transactions is a frequent set.

Fig. 3

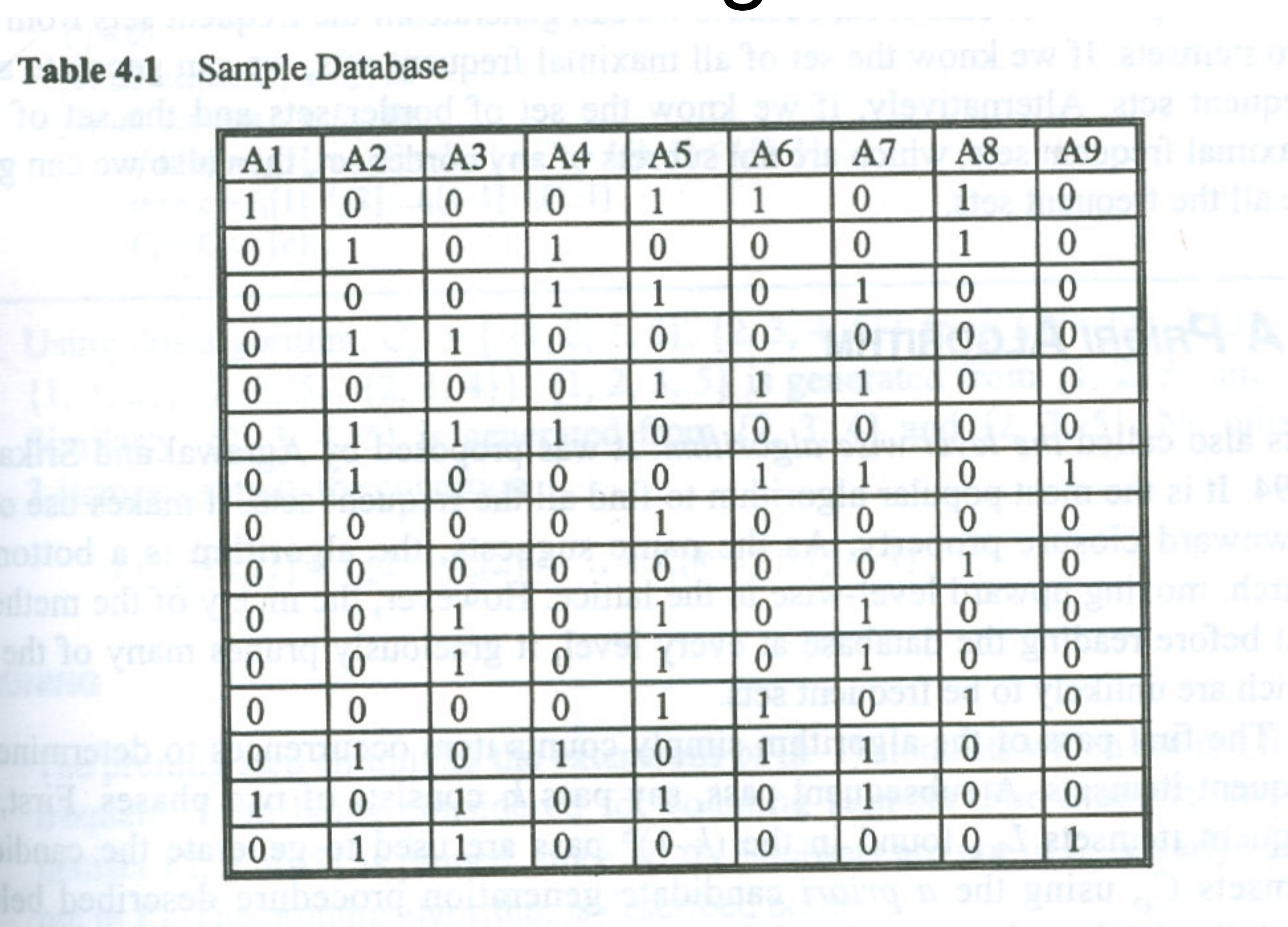


Table 4.1 Sample Database

A1	A2	A3	A4	A5	A6	A7	A8	A9
1	0	0	0	1	1	0	1	0
0	1	0	1	0	0	0	1	0
0	0	0	1	1	0	1	0	0
0	1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0
0	1	1	1	0	0	0	0	0
0	1	0	0	0	1	1	0	1
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	1	0
0	0	1	0	1	0	1	0	0
0	0	1	0	1	0	1	0	0
0	0	0	0	1	1	0	1	0
0	1	0	1	0	1	1	0	0
1	0	1	0	1	0	1	0	0
0	1	1	0	0	0	0	0	1

Fig. 4

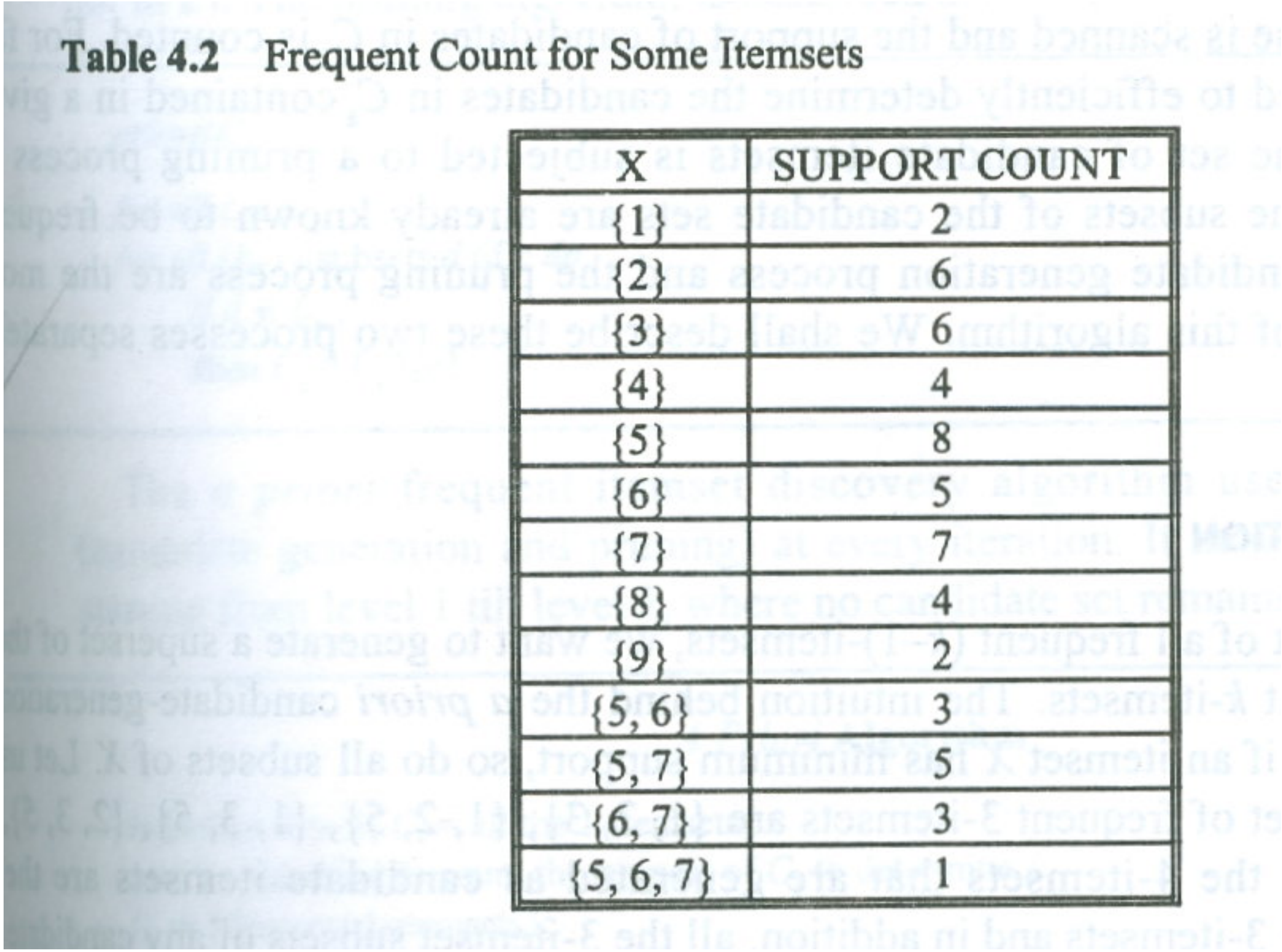


Table 4.2 Frequent Count for Some Itemsets

X	SUPPORT COUNT
{1}	2
{2}	6
{3}	6
{4}	4
{5}	8
{6}	5
{7}	7
{8}	4
{9}	2
{5, 6}	3
{5, 7}	5
{6, 7}	3
{5, 6, 7}	1

Applications of Data Mining – a close viewpoint

Several important data mining applications are concerned with pattern detection and recognition.

1. **Spotting fraudulent behavior** by detecting regions of space defining different types of transactions, where the data points significantly differ from the rest.
2. **Astronomy:** Another use is in astronomy, where detection of unusual stars or galaxies or nebulas or super galaxies may lead to the discovery of previously unknown phenomena and terrestrial body.

Applications of Data Mining contd..

3. **Another Task:** Finding combinations of items that occur frequently in transaction databases

Association Rules

An association rule is a simple probabilistic statement about the co-occurrence of certain events in a database.

An association rule takes the following form:

If A = 1 and B = 1 THEN C = 1 with probability p

- Where A, B, and C are binary variables and p = p(C =1 | A =1, B =1), i.e. the conditional probability that C = 1 given that A = 1 and B = 1.
- The conditional probability p is sometimes referred to as the ‘accuracy’ of ‘confidence’ of the rule.

Informal a priori Algorithm for Association Rule Learning

1. In general we moves from frequent sets of size (k-1) to frequent sets of size k
2. In the above we can prune any sets of size k that contain a subset of (k-1) items that themselves are not frequent at the (k-1) level.
3. From example, if we have only frequent sets {A=1, B=1} and {B=1, C=1}, we could combine them to get the candidate k =3 frequent set {A=1, B=1, C=1}. However, if the subset of items (A=1, C=1} was not frequent, then {A=1, B=1, C=1} could not be frequent, and could safely be **pruned.**

Algorithm contd.

- 4. Given the pruned list of candidate frequent sets of size k, the algorithm performs another linear scan of the database to determine which of these sets are in fact frequent
- 5. The confirmed frequent sets of size k (if any) are combined to generate all possible frequent sets containing (k+1) events, followed by pruning, and another scan of the database, and so on – until no more frequent sets can be generated

- In the worst case, all possible sets of events are frequent and the algorithm takes exponential time.
- However since practically the data are often very sparse , the cardinality of the largest frequent set is usually quiet small.

Finding Frequent Sets and Association Rules

- If the frequent sets are known, the finding association rules is simple.

  The association rule has the form:

  $X \Rightarrow Y$

  If the rule has frequency at least ‘s’ then the set ‘X’ must by definition have frequency at least ‘s’.

Formal *a priori* Algorithm for Association Rule

```
$i = 0$;
$c_i = \{ \{A\} \mid A$ is a variable$\}$;
while $c_i$ is not empty do
    database pass:
        for each set in $c_i$, test whether it is frequent;
    Let $L_i$ be the collection of frequent set from $c_i$;
    Candidate generation:
        let $C_{i+1}$ be those sets of size $i+1$
            Whose all subsets are frequent;
End.
```

Sample Database

This database is used subsequently; also σ = 20 % is assumed

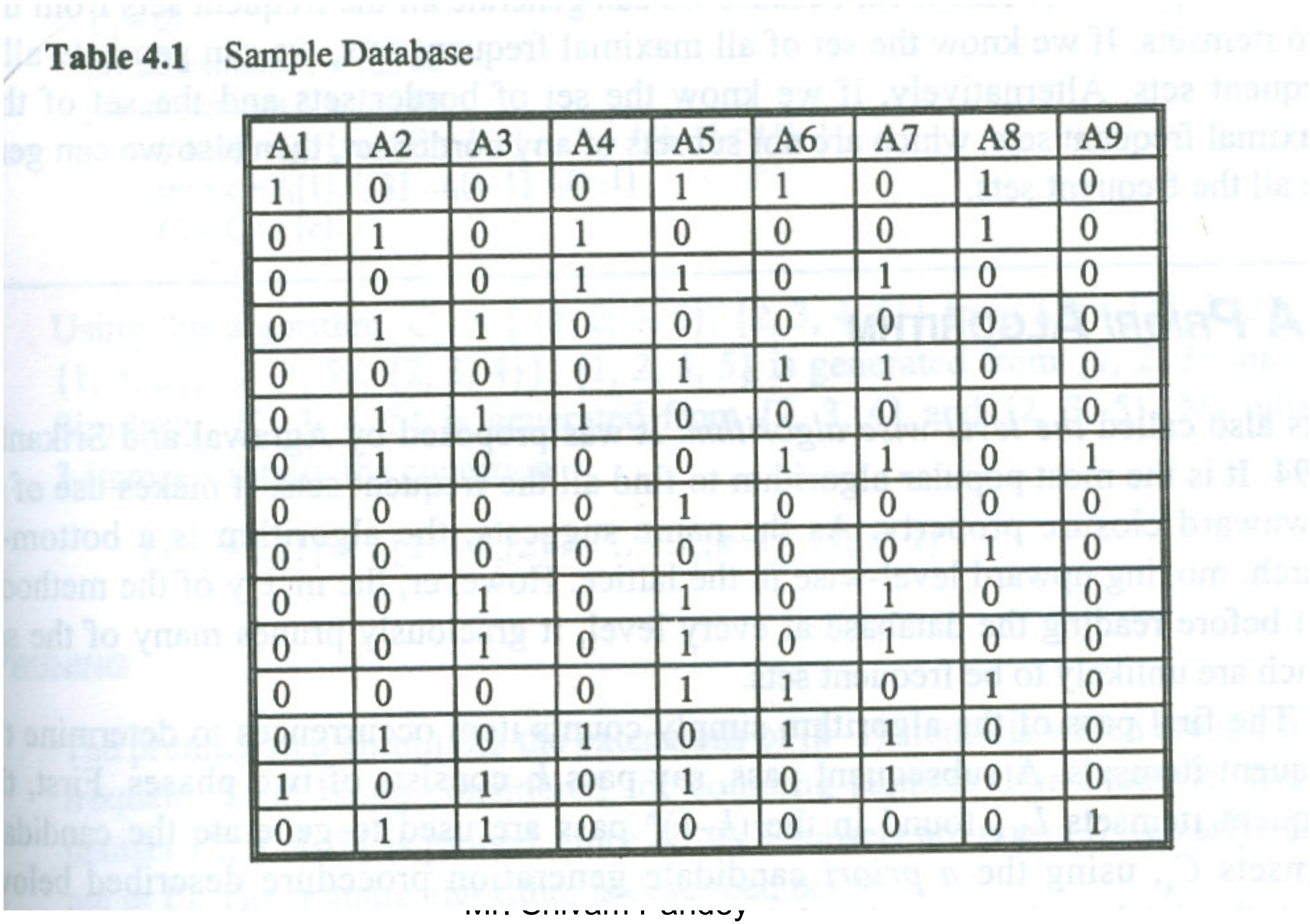


Table 4.1 Sample Database

A1	A2	A3	A4	A5	A6	A7	A8	A9
1	0	0	0	1	1	0	1	0
0	1	0	1	0	0	0	1	0
0	0	0	1	1	0	1	0	0
0	1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0
0	1	1	1	0	0	0	0	0
0	1	0	0	0	1	1	0	1
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	1	0
0	0	1	0	1	0	1	0	0
0	0	1	0	1	0	1	0	0
0	0	0	0	1	1	0	1	0
0	1	0	1	0	1	1	0	0
1	0	1	0	1	0	1	0	0
0	1	1	0	0	0	0	0	1

Frequency count of item sets of previous database:

Legend : Item set {i,j} indicates {Ai, Aj}

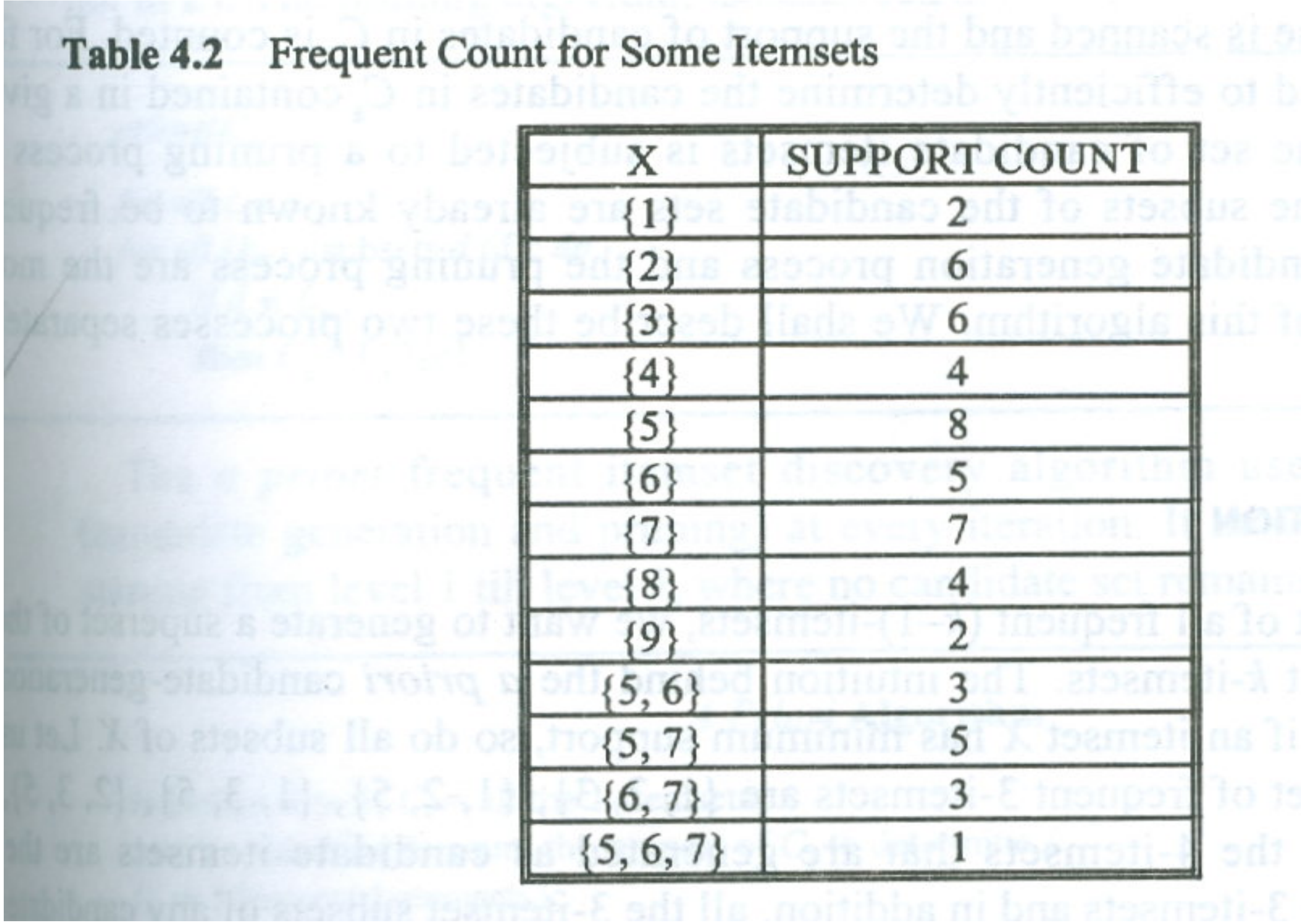
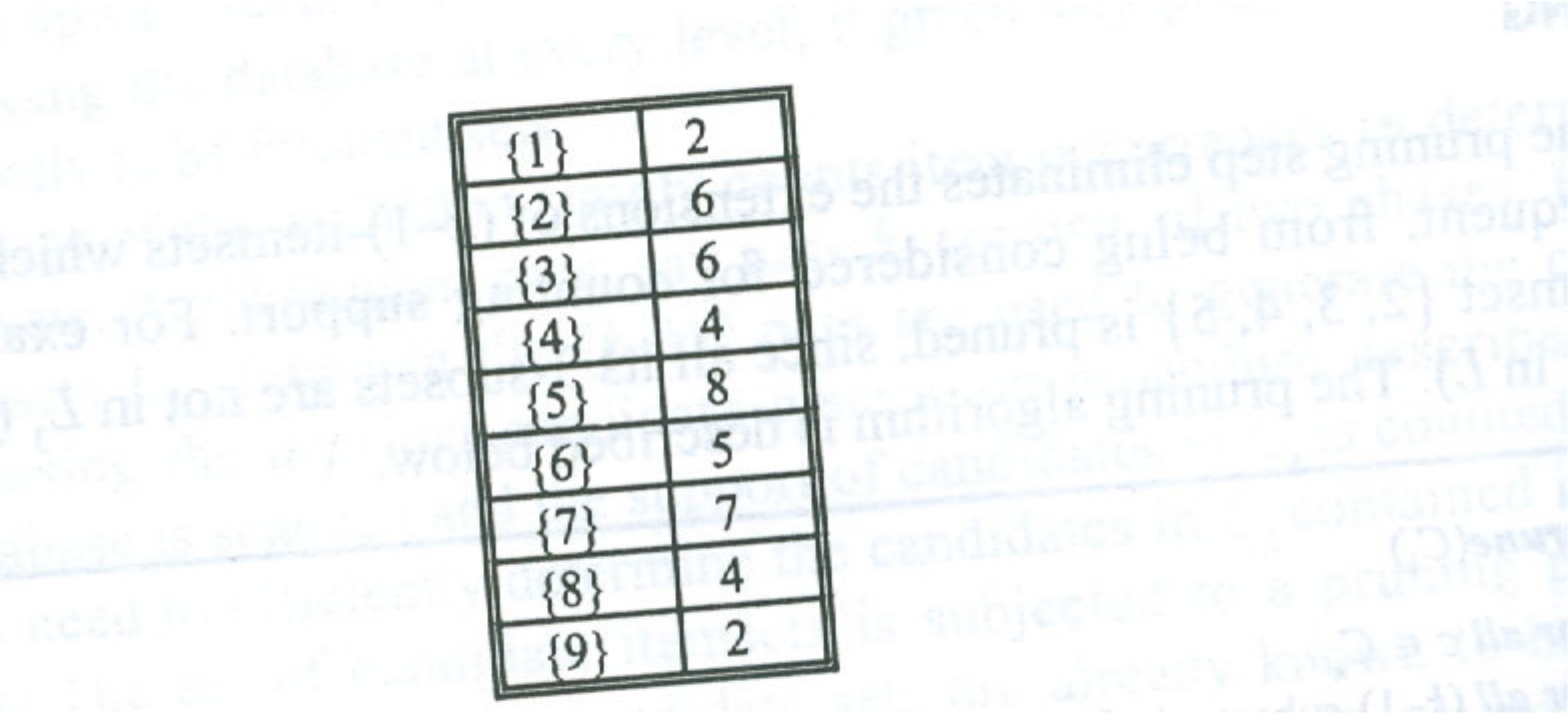


Table 4.2 Frequent Count for Some Itemsets

X	SUPPORT COUNT
{1}	2
{2}	6
{3}	6
{4}	4
{5}	8
{6}	5
{7}	7
{8}	4
{9}	2
{5, 6}	3
{5, 7}	5
{6, 7}	3
{5, 6, 7}	1

The frequent 1- item sets and their support counts are:



{1}	2
{2}	6
{3}	6
{4}	4
{5}	8
{6}	5
{7}	7
{8}	4
{9}	2

Illustration of *a priori* algorithm from frequent 1 item set and corresponding support count (previous slide) [σ = 20 %]

- L1 := { {2} → 6, {3} → 6, {4} → 4, {5} → 8, {6} → 5, {7} → 7, {8} → 4 }
- k = 2
- In the candidate generation step, we get:
- C_2 = { {2,3}, {2,4}, {2,5}, {2,6}, {2,7}, {2,8}, {3,4}, {3,5}, {3,6}, {3,7}, {3,8}, {4,5}, {4,6}, {4,7}, {4,8}, {5,6}, {5,7}, {5,8}, {6,7}, {6,8}, {7,8}
- The Pruning step does not change C_2

- From the database, count the support of elements in C_2 :
- L_2 := { {2,3} → 3, {2,4} → 3, {3,5} → 3, {3, 7} → 3, {5,6} → 3, {5,7} → 5, {6, 7} → 3

- k := 3
- In the candidate generation step,
- Combining {2,3} and {2,4} , we get {2,3,4}
- Combining {3,5} and (3,7} , we get {3,5,7} and
- Combining {5,6} and {5,7}, we get {5,6,7}
- Thus C_3 := {{2,3,4}, {3,5,7}, {5,6,7}}

- The pruning step prunes {2,3,4} as one of its subset {3,4} is of size 1 (which is less than 2; which makes σ =20%).
- Thus the pruned C_3 is

- {{3,5,7}, {5,6,7}}

- Read the database to count the support of the item sets in C_3 [$\sigma = 20\%$]
- $L_3 := \{\{3,5,7\} \rightarrow 3\}$
- $k := 4$
- Since L_3 contains only one element, C_4 is empty and hence the algorithm stops, returning the set of frequent sets along with their respective support values. The set is:
- $L := L_1 \cup L_2 \cup L_3$

- **End of Association Rules**