Chapter 2
Synthesizing Conditional Patterns in a Database

Though frequent itemsets and association rules express interesting association among items of frequently occurring itemsets in a database, there may exist other types of interesting associations among the items. A critical analysis of frequent itemsets would provide more insight about a database. In this paper, we introduce the notion of conditional pattern in a database. Conditional patterns are interesting and useful for solving many problems. We propose an algorithm for mining conditional patterns in a database. Experiments are conducted on three real datasets. The results of the experiments show that conditional patterns store significant nuggets of knowledge about a database.

2.1 Introduction

Association analysis of items (Agrawal et al. 1993; Antonie and Zaïane 2004), and selecting right interestingness measures (Hilderman and Hamilton 1999; Tan et al. 2002) are two significant tasks at the heart of many data mining problems. An association analysis is generally associated with interesting patterns in a database, and the interestingness of a pattern is expressed by using some measures. A pattern would become interesting if the values of interestingness measures satisfy some conditions. Positive association rules (Agrawal et al. 1993) and negative association rules (Antonie and Zaïane 2004) are examples of two patterns that are synthesized from the itemset patterns in a database. Positive association rules are expressed by a forward implication $X \rightarrow Y$, where X and Y are itemsets in the database. X and Y are called the antecedent and consequent of the association rule respectively. The meaning attached to this type of association rules is that if all the items in X are purchased by a customer then it is likely that all the items in Y are purchased by the same customer at the same time. On the other hand, negative association rules are expressed by one of the following three forward implications: $X \rightarrow \neg Y$, $\neg X \rightarrow Y$, and $\neg X \rightarrow \neg Y$, where X and Y are itemsets in the given database. Let us consider the negative association rules of the form $X \rightarrow \neg Y$. The meaning attached to the negative association rules of the form $X \rightarrow \neg Y$ is that if all the items in X are

A. Adhikari and J. Adhikari, *Advances in Knowledge Discovery in Databases*,
Intelligent Systems Reference Library 79, DOI 10.1007/978-3-319-13212-9_2

purchased by a customer then it is unlikely that all the items in Y are purchased by the same customer at the same time. Though association rules express interesting association among items in frequent itemsets, they might not be sufficient for all kinds of association analysis of items in a given database.

The importance of an itemset could be judged by its support (Agrawal et al. 1993). *Support* (*supp*) of an itemset X in database D is the fraction of transactions in D containing X. Itemset X is *frequent* in D if $supp(X, D) \geq \alpha$, where α is user defined *minimum support level*. Itemset $X = \{x_1, x_2, \ldots, x_m\}$ corresponds to Boolean expression $x_1 \wedge x_2 \wedge \cdots \wedge x_m$. Thus, if the itemset $\{x_1, x_2, \ldots, x_m\}$ contains in a transaction then the Boolean expression $x_1 \wedge x_2 \wedge \cdots \wedge x_m$ is true for that transaction. On the other hand, if the itemset $\{x_1, x_2, \ldots, x_m\}$ does not contain in a transaction then the Boolean expression $x_1 \wedge x_2 \wedge \cdots \wedge x_m$ is false for that transaction. In general, let E be a Boolean expression on items in D. Then, $supp(E, D)$ is the fraction of transactions in D that satisfy E.

Frequent itemset mining has received significant attention in KDD community. Several implementations of mining frequent itemsets (FIMI 2004) have been reported. Frequent itemsets are important patterns in a database, since they determine the major characteristics of a database. Wu et al. (2005) have proposed a solution of inverse frequent itemset mining. Authors argued that one could efficiently generate a synthetic market basket database from the frequent itemsets and their supports. Let X and Y be two itemsets in database D. The characteristics of database D are revealed more by the pair $(X, supp(X, D))$ than that of $(Y, supp(Y, D))$, if $supp(X, D) > supp(Y, D)$. Thus, it is important to study frequent itemsets more than infrequent itemsets. Negative association rules are generated from infrequent itemsets. Thus, their applications in different problem domains are limited. The goal of this chapter is to study some kind of association among items which is not immediately available from frequent itemsets and association rules.

If X is frequent in D then every non-null subset of X is also frequent in D. Consider the following example.

Example 2.1 Let $D = \{\{a, b\}, \{a, b, c, d\}, \{a, b, c, h\}, \{a, b, g\}, \{a, b, h\}, \{a, c\}, \{a, c, d\}, \{b\}, \{b, c, d, h\}, \{b, d, g\}\}$. The frequent itemsets in D at minimum support level 0.2 are given as follows: $\{a\}(0.7)$, $\{b\}(0.8)$, $\{c\}(0.5)$, $\{d\}(0.4)$, $\{g\}(0.2)$, $\{h\}(0.3)$, $\{a, b\}(0.5)$, $\{a, c\}(0.4)$, $\{a, d\}(0.2)$, $\{a, h\}(0.2)$, $\{b, c\}(0.3)$, $\{b, d\}(0.3)$, $\{b, g\}(0.2)$, $\{b, h\}(0.3)$, $\{c, d\}(0.3)$, $\{c, h\}(0.2)$, $\{a, b, c\}(0.2)$, $\{a, b, h\}(0.2)$, $\{a, c, d\}(0.2)$, $\{b, c, d\}(0.2)$, $\{b, c, h\}(0.2)$. $X(\eta)$ denotes frequent itemset X with support η. Suppose we wish to study association among items in $\{a, b, c\}$. A frequent itemset mining algorithm could mine the following details about items in $\{a, b, c\}$.

Table 2.1 provides the information on how frequently a non-null subset of $\{a, b, c\}$ occurs in D. Such information might not be sufficient for all types of queries and analyses of items in $\{a, b, c\}$.

A positive association rule finds positive association between two disjoint non-null itemsets. Positive association rules in D are synthesized from frequent itemsets

Table 2.1 Frequent itemset $\{a, b, c\}$ and its non-null subsets at $\alpha = 0.2$

Itemset	$\{a\}$	$\{b\}$	$\{c\}$	$\{a, b\}$	$\{a, c\}$	$\{b, c\}$	$\{a, b, c\}$
Support	0.7	0.8	0.5	0.5	0.4	0.3	0.2

Table 2.2 Association rules generated from $\{a, b, c\}$ at $\alpha = 0.2$ and $\beta = 0.5$

Association rule	Support	Confidence
$\{a, c\} \rightarrow \{b\}$	0.2	0.66667
$\{b, c\} \rightarrow \{a\}$	0.2	0.66667

in D. A positive association rule r: $X \rightarrow Y$ in D is characterized by its support and confidence measures (Agrawal et al. 1993). *Support* of association rule r: $X \rightarrow Y$ in D is the fraction of transactions in D containing both X and Y. *Confidence* (*conf*) of association rule r in D is the fraction of transactions in D containing Y among the transactions containing X. An association rule r in D is interesting if $supp(r, D) \geq \alpha$, and $conf(r, D) \geq \beta$, where β is the *minimum confidence level*. The parameters α and β are user-defined inputs to an association rule mining algorithm. We synthesize association rules from $\{a, b, c\}$ of Example 2.1 as follows (Example 2.2).

Example 2.2 We continue here the discussion of Example 2.1. The interesting association rules generated from $\{a, b, c\}$ are given in Table 2.2.

The chapter is organized as follows. In Sect. 2.6, we introduce conditional pattern in a database. We discuss properties of conditional patterns in Sect. 2.3. In Sect. 2.4, we propose an algorithm for extracting conditional patterns in a database. The results of the experiments are given in Sect. 2.5. Also, we present an application of conditional patterns in this section. We discuss related work in Sect. 2.6.

2.2 Conditional Pattern

With reference to Examples 2.1 and 2.2, the study of items in $\{a, b, c\}$ might be incomplete if we know only the supports and the association rules with respect to non-null subsets of $\{a, b, c\}$. Thus, the information provided in Tables 2.1 and 2.2 might not be sufficient for all types of queries and analyses related to items in $\{a, b, c\}$. In fact, there are some queries related to items in $\{a, b, c\}$ whose answers are not immediately available from Tables 2.1 and 2.2. A few examples of such queries are given below.

- Find the support that a transaction contains item a but not items b and c, with respect to $\{a, b, c\}$.
- Find the support that a transaction contains items a and b but not item c, with respect to $\{a, b, c\}$.

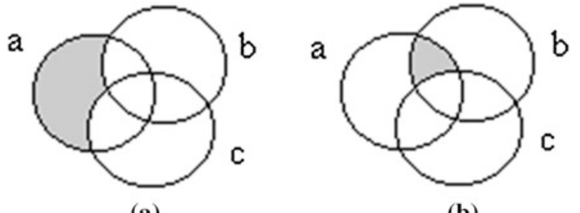
The above queries correspond to a specific type of pattern in a database. Some of these of patterns could have significant supports, since $\{a, b, c\}$ is a frequent itemset. In general, if we wish to study the association among the items in Y with negation of items in $X - Y$, then such analysis is not immediately available from frequent itemsets and positive association rules, for itemsets X and Y in a database such that $Y \subseteq X$. Such association analyses could be interesting, since the corresponding Boolean expressions could have high supports. Therefore, we need to mine such patterns for effective analyses of items in frequent itemsets.

Let $\langle Y, X \rangle$ be a pattern that a transaction in a database contains all the items of Y, but not items of $X - Y$, for itemsets X and Y in the database such that $Y \subseteq X$. Let $supp\langle Y, X, D\rangle$ be the support that a transaction in database D contains all the items of Y, but not items of $X - Y$, for itemsets X and Y in D such that $Y \subseteq X$. A pattern of type $\langle Y, X \rangle$ is called a *conditional pattern* (Adhikari and Rao 2008). A conditional pattern $\langle Y, X \rangle$ has two components: *pattern itemset* (Y) and *reference itemset* (X). Thus, a conditional pattern $\langle Y, X \rangle$ is associated with two values: $supp\langle Y, X, D\rangle$ and $supp(X, D)$. $supp\langle Y, X, D\rangle$ and $supp(X, D)$ are called *conditional support* (*csupp*) and *reference support* (*rsupp*) of conditional pattern $\langle Y, X \rangle$ in D, respectively. The conditional support and reference support of conditional pattern $\langle Y, X \rangle$ in D are denoted by $csupp\langle Y, X, D\rangle$ and $rsupp\langle Y, X, D\rangle$, respectively. In other words, $supp\langle Y, X, D\rangle$ and $supp(X, D)$ are denoted by $csupp\langle Y, X, D\rangle$ and $rsupp\langle Y, X, D\rangle$, respectively. A conditional pattern $\langle Y, X \rangle$ in D is *interesting* if $csupp\langle Y, X, D\rangle \geq \delta$ and $rsupp\langle Y, X, D\rangle \geq \alpha$, where δ is the *minimum conditional support level*. The parameters α and δ are user defined inputs to a conditional pattern mining algorithm.

Figure 2.1 give more insight about above two queries.

The shaded region in Fig. 2.1a is a set of transactions in D such that each transaction contains item a but not items b and c, with respect to $\{a, b, c\}$. The shaded region in Fig. 2.1b is a set of transactions in D such that each transaction contains items a and b but not item c, with respect to $\{a, b, c\}$. Thus, we get following formulas.

$$\begin{aligned} supp\langle\{a\},\{a,b,c\},D\rangle = {} & supp(\{a\},D) - supp(\{a,b\},D) \\ & - supp(\{a,c\},D) + supp(\{a,b,c\},D) \end{aligned} \tag{2.1}$$



(a) (b)

Fig. 2.1 *Shaded regions* in (**a**) and (**b**) correspond to conditional supports of $\langle\{a\}, \{a, b, c\}\rangle$ and $\langle\{a, b\}, \{a, b, c\}\rangle$ in D, respectively

Table 2.3 Conditional patterns with respect to {a, b, c} in D

Conditional pattern	csupp	Conditional pattern	csupp
$\langle\{a\}, X\rangle$	0	$\langle\{a, c\}, X\rangle$	0.2
$\langle\{b\}, X\rangle$	0.2	$\langle\{b, c\}, X\rangle$	0.1
$\langle\{c\}, X\rangle$	0	$\langle X, X\rangle$	1.0
$\langle\{a, b\}, X\rangle$	0.3		

Table 2.4 Non-trivial conditional patterns with respect to {*a*, *b*, *c*} at $\delta = 0.2$ and $\alpha = 0.2$

Conditional pattern	csupp	rsupp	Conditional pattern	csupp	rsupp
$\langle\{b\}, \{a, b, c\}\rangle$	0.2	0.2	$\langle\{a, c\}, \{a, b, c\}\rangle$	0.2	0.2
$\langle\{a, b\}, \{a, b, c\}\ \rangle$	0.3	0.2			

$$supp\langle\{a,b\},\{a,b,c\},D\rangle = supp(\{a,b\},D) - supp(\{a,b,c\},D) \tag{2.2}$$

A conditional pattern $\langle Y, X\rangle$ in a database is *trivial* if $Y = X$. A trivial conditional pattern is known when the corresponding frequent itemset gets extracted from the database. Thus, trivial conditional patterns get mined during mining of frequent itemsets. In the following example, we identify conditional patterns in *D* with respect to {*a*, *b*, *c*} of Example 2.1.

Example 2.3 Consider the frequent itemsets in *D* of Example 2.1. The conditional patterns with respect to $X = \{a, b, c\}$ in *D* are given in Table 2.3.

We define *size* of an itemset *X* as the number of items in *X*, denoted by $|X|$. Based on the sizes of pattern itemset and reference itemset, we could categorize conditional patterns in a database. The conditional patterns $\langle\{a\},X\rangle$, $\langle\{b\},X\rangle$ and $\langle\{c\},X\rangle$ belong to the same category. But, the conditional patterns $\langle\{a\},X\rangle$ and $\langle\{a,b\},X\rangle$ are of different categories. In general, two conditional patterns $\langle X, Y\rangle$ and $\langle P, Q\rangle$ in *D* are of the *same category*, if $|X| = |P|$ and $|Y| = |Q|$, for X, Y, P, Q are itemsets in *D*. All the conditional patterns mined with respect to a frequent itemset are not interesting. The interesting conditional patterns with respect to {*a*, *b*, *c*} in *D* are given in Table 2.4.

We observe that $csupp\langle Y,X,D\rangle \leq supp(Y,D)$, for $Y \subset X$. Nonetheless, $csupp\langle Y,X,D\rangle$ could be high, if *Y* is frequent in *D*. Thus, it is necessary to study such patterns in a database for effective analyses of items in frequent itemsets. The problem could be stated as follows.

We are given a database D of customer transactions. Extract interesting non-trivial conditional patterns from D.

2.3 Properties of Conditional Patterns

In this section, we present some interesting properties of conditional patterns in a database. Before presenting the properties, we introduce some notations. Let $X = \{x_1, x_2, \ldots, x_m\}$ and $Y = \{y_1, y_2, \ldots, y_p\}$. Then, $supp(X \cup Y, D)$ and $supp(X \cap Y, D)$ refer to $supp((x_1 \wedge x_2 \wedge \cdots \wedge x_m) \vee (y_1 \wedge y_2 \wedge \cdots \wedge y_p), D)$ and $supp((x_1 \wedge x_2 \wedge \cdots \wedge x_m) \wedge (y_1 \wedge y_2 \wedge \cdots \wedge y_p), D)$ respectively.

Lemma 2.1 *Let E be a Boolean expression that a transaction contains at least one item of itemset X in database D. Then,*

$$supp(E, D) = \sum_{\mathrm{Y} \subseteq \mathrm{X},\ \mathrm{Y} \neq \phi} supp\langle Y, X, D\rangle \tag{2.3}$$

Proof We re-state theorem of total probability (Feller 1968) in terms of supports as follows: For any m Boolean expressions $\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_\mathrm{m}$ in database D we have,

$$\begin{aligned} supp\left(\cup_{\mathrm{i}=1}^{\mathrm{m}} X_\mathrm{i}, D\right) = & \sum_{\mathrm{i}=1}^{\mathrm{m}} supp(X_\mathrm{i}, D) \\ & - \sum_{\mathrm{i}<\mathrm{j};\ \mathrm{i},\ \mathrm{j}=1}^{\mathrm{m}} supp\left(X_\mathrm{i} \cap X_\mathrm{j}, D\right) + \cdots + (-1)^{\mathrm{m}-1} supp\left(\cap_{\mathrm{i}=1}^{\mathrm{m}} X_\mathrm{i}, D\right). \end{aligned}$$

The events $\langle Y, X\rangle$ and $\langle Z, X\rangle$ are mutually exclusive, for $Y \neq Z, Y \subseteq X$ and $Z \subseteq X$. Thus, $supp(\langle Y, X\rangle \cap \langle Z, X\rangle, D) = 0$, for $Y \neq Z, Y \subseteq X$ and $Z \subseteq X$. □

Let $X = \{a, b, c\}$. With reference to Examples 2.1 and 2.3, $supp(a \vee b \vee c, D) = 1$ and $supp(a \vee b \vee c, D) = supp\langle\{a\}, X, D\rangle + supp\langle\{b\}, X, D\rangle + supp\langle\{c\}, X, D\rangle + supp\langle\{a, b\}, X, D\rangle + supp\langle\{a, c\}, X, D\rangle + supp\langle\{b, c\}, X, D\rangle + supp\langle X, X, D\rangle$. Thus, it validates Lemma 2.1.

Lemma 2.2 $supp(X, D) \leq \sum_{\mathrm{Y} \subseteq \mathrm{X},\ \mathrm{Y} \neq \phi} supp\langle Y, X, D\rangle$, *for any two itemsets X and Y in database D such that* $Y \subseteq X$.

Proof Let $X = \{x_1, x_2, \ldots, x_m\}$. Then, X corresponds to Boolean expression $x_1 \wedge x_2 \wedge \cdots \wedge x_m$ in D. Let E be a Boolean expression that a transaction contains at least one item of itemset X in D. Then, $supp(E, D) = \sum_{\mathrm{Y} \subseteq \mathrm{X},\ \mathrm{Y} \neq \phi} supp\langle Y, X, D\rangle$, (Lemma 2.1) = $supp\langle X, X, D\rangle + Q$, where $Q \geq 0$. Then, $supp(E, D) = supp(X, D) + Q$, since $supp(X, D) = supp\langle X, X, D\rangle$. The lemma follows. □

With reference to Examples 2.1 and 2.3, let $X = \{a, b, c\}$. $supp(X, D) = 0.2$. Now, $supp\langle\{a\}, X, D\rangle + supp\langle\{b\}, X, D\rangle + supp\langle\{c\}, X, D\rangle + supp\langle\{a, b\}, X,$

$D\rangle$ + $supp\langle\{a, c\}, X, D\rangle$ + $supp\langle\{b, c\}, X, D\rangle$ + $supp\langle X, X, D\rangle = 1.0 \geq 0.2$. Thus, it validates Lemma 2.2.

Lemma 2.3 *The conditional supports of* $\langle X, Y\rangle$ and $\langle X, Z\rangle$ *in a database may not be equal, for any three itemsets X, Y and Z in the database such that* $X \subseteq Y$ *and* $X \subseteq Z$.

Proof The itemsets $Y - X$ and $Z - X$ may not be the same. Thus, the lemma follows. □

A conditional pattern is so named due to Lemma 2.3. Using Example 2.1, we get $supp\langle\{a, b\}, \{a, b, h\}, D\rangle = 0.3$ and $supp\langle\{a, b\}, \{a, b, d\}, D\rangle = 0.4$. We observe that $supp\langle\{a, b\}, \{a, b, h\}, D\rangle \neq supp\langle\{a, b\}, \{a, b, d\}, D\rangle$.

Lemma 2.4 *There is no fixed ordered relationship between conditional supports of* $\langle Y, X\rangle$ *and* $\langle Z, X\rangle$ *in a database, for any three itemsets X, Y and Z in the database such that* $Z \subseteq Y \subseteq X$.

Proof Let X, Y and Z be three itemsets X, Y and Z in database D such that $supp\langle Y, X, D\rangle \leq supp\langle Z, X, D\rangle$, for some $Z \subseteq Y \subseteq X$. Also, there may exist another three itemsets P, Q and R in database D such that $supp\langle R, P, D\rangle \leq supp\langle Q, P, D\rangle$, for some $R \subseteq Q \subseteq P$. The proof is based on a counter example. With reference to database D of Example 2.1, let $X = \{a, b, c\}$, $Y = \{a, b\}$ and $Z = \{a\}$. Then, $supp\langle Z, X, D\rangle = supp\langle\{a\}, \{a, b, c\}, D\rangle = 0$, and $supp\langle Y, X, D\rangle = supp\langle\{a, b\}, \{a, b, c\}, D\rangle = 0.3$. Thus, $supp\langle Z, X, D\rangle \leq supp\langle Y, X, D\rangle$, for $Z \subseteq Y \subseteq X$. Let $A = \{b, d, h\}$, $B = \{b, d\}$ and $C = \{b\}$. Then, $supp\langle C, A, D\rangle = supp\langle\{b\}, \{b, d, h\}, D\rangle = 0.3$, and $supp\langle B, A, D\rangle = supp\langle\{b, d\}, \{b, d, h\}\rangle = 0.2$. Thus, $supp\langle B, A, D\rangle \leq supp\langle C, A\rangle$, for $C \subseteq B \subseteq A$. □

We could synthesize a set of frequent itemsets from a set of association rules. In particular, let r_1: $X \rightarrow Y$ and r_2: $X \rightarrow Z$ be two positive association rules in D, where X, Y and Z are three frequent itemsets in D. The set of frequent itemsets is synthesized from $\{r_1, r_2\}$ is $\{X, XY, XZ\}$. In a similar way, we could synthesize a set of frequent itemsets from a set of conditional patterns. In particular, let cp_1: $\langle x_1 \wedge x_2, x_1 \wedge x_2 \wedge x_3\rangle$, and cp_2: $\langle x_1 \wedge x_3, x_1 \wedge x_2 \wedge x_3\rangle$ are two conditional patterns in D, where x_i is an item in D, for $i = 1, 2, 3$. The set of frequent itemsets is synthesized from $\{cp_1, cp_2\}$ is $\{\{x_1, x_2\}, \{x_1, x_3\}, \{x_1, x_2, x_3\}\}$.

Example 2.4 Let us consider Table 2.2. The set of frequent itemsets synthesized from the set of positive association rules is given as follows: $\{\{a, c\}(0.4), \{b, c\}(0.3), \{a, b, c\}(0.2)\}$. Let us consider Table 2.4. The set of frequent itemsets synthesized from the set of conditional patterns is given as follows: $\{\{a, b\}(0.5), \{a, c\}(0.4), \{a, b, c\}(0.2)\}$.

From Example 2.4, we could conclude that the association rules and conditional patterns in a database may not represent the same information about a database. This because of the fact that the amount of information conveyed by association rules in a database is dependent on β, at a given α. Also, the information conveyed by the conditional patterns in a database is dependent on δ, at a given α. Thus, we have the following definition.

Definition 2.1 A set of association rules A and a set of conditional patterns C in a database convey the same information about a given database if the set of frequent itemsets synthesized from A is the same as the set of frequent itemsets synthesized from C.

Lemma 2.5 *The set association rules in a database at* $\beta = \alpha$ *and the set of conditional patterns in the database at* $\delta = 0$ *represent the same information about the database at a give* α.

Proof Let S be a set of frequent itemsets in database D. Also, let CLOSURE (S) = $\{s$: ($s \in S$), or ($s \neq \phi$ and $s \subseteq p \in S$)$\}$. Let SFIS(D, i) be the set of frequent itemsets of size i, for $i = 1, 2, \ldots$. The set of frequent itemsets synthesized from association rules in D at $\beta = \alpha$ is equal to CLOSURE $(\cup_{\mathrm{i} \geq 2} SFIS(D, i))$. Also, the set of frequent itemsets synthesized from the conditional patterns in D at $\delta = 0$ is equal to CLOSURE $(\cup_{\mathrm{i} \geq 2} SFIS(D, i))$. □

With reference to Example 2.1, the frequent itemsets in D at $\alpha = 0.4$ are given as follows: $\{a, b\}$(0.5), $\{a, c\}$(0.4). The association rules in D at $\beta = 0.4$ are given in Table 2.5.

The set frequent itemsets synthesized from the above association rules is equal to $\{\{a\}$(0.7), $\{b\}$(0.8), $\{c\}$(0.5), $\{a, b\}$(0.5), $\{a, c\}$(0.4)$\}$. The conditional patterns in D at $\delta = 0.4$ are given in Table 2.6.

The set of frequent itemsets synthesized from the above conditional patterns is equal to $\{\{a\}$(0.7), $\{b\}$(0.8), $\{c\}$(0.5), $\{a, b\}$(0.5), $\{a, c\}$(0.4)$\}$. Thus, the set of frequent itemsets synthesized from the above association rules and the set of

Table 2.5 Association rules in D at $\alpha = 0.4$ and $\beta = 0.4$

Association rule (r)	supp(r, D)	conf(r, D)
$\{a\} \rightarrow \{b\}$	0.5	0.71
$\{b\} \rightarrow \{a\}$	0.5	0.63
$\{a\} \rightarrow \{c\}$	0.4	0.57
$\{c\} \rightarrow \{a\}$	0.4	0.80

Table 2.6 Conditional patterns in D at $\alpha = 0.4$ and $\delta = 0$

Conditional pattern	csupp	rsupp	Conditional pattern	csupp	rsupp
$\langle\{a\}, \{a, b\}\rangle$	0.2	0.5	$\langle\{a\}, \{a, c\}\rangle$	0.3	0.4
$\langle\{b\}, \{a, b\}\rangle$	0.3	0.5	$\langle\{c\}, \{a, c\}\rangle$	0.1	0.4

frequent itemsets synthesized from the above conditional patterns are the same at $\beta = \alpha$ and $\delta = 0$. Thus, it validates Lemma 2.5.

Lemma 2.6 *Let the conditional pattern* $\langle Y, X\rangle$ *in database D is interesting at conditional support level* δ *and support level* α. *Then itemset Y is frequent at level* $\alpha + \delta$.

Proof $supp\langle Y, X, D\rangle \geq \delta$ and $supp(X, D) \geq \alpha$, since $\langle Y, X\rangle$ is interesting in D at conditional support level δ and support level α. The patterns X and $\langle Y, X\rangle$ in D can not occur in a transaction simultaneously. $supp(X, D) \geq \alpha$ implies $supp(Y, D) \geq \alpha$, since $Y \subseteq X$. Also, $supp\langle Y, X, D\rangle \geq \delta$ and thus, $supp(Y, D) \geq (\alpha + \delta)$. □

With reference to Example 2.3, $\langle\{b\}, \{a, b, c\}\rangle$ is interesting conditional pattern in D at δ = 0.2 and α = 0.2. With reference to Example 2.1, supp({b}, D) = 0.8 ≥ 0.2 + 0.2 = 0.4. Thus, it validates Lemma 2.6.

Lemma 2.7 Let $X_1, X_2, \ldots, X_m$ *be itemsets in database D such* $X_i \subseteq X_{i+1}$, *for* $i = 1, 2, \ldots, m - 1$. *Then*, $supp\langle Y, X_i, D\rangle \geq supp\langle Y, X_{i+1}, D\rangle$, *for* $Y \subseteq X_i$ *at every* $i = 1, 2, \ldots, m - 1$.

Proof Let $Y = \{a_1, a_2, \ldots, a_p\}$. Let $Z = X_{k+1} - X_k$, for $i = k$. Also let, $X_k = \{b_1, b_2, \ldots, b_q\}$, and $Z = \{c_1, c_2, \ldots, c_r\}$. Consider the following two Boolean expressions: $E_1 = a_1 \wedge a_2 \wedge \cdots \wedge a_p \wedge \neg b_1 \wedge \neg b_2 \wedge \cdots \wedge \neg b_q$ and $E_2 = a_1 \wedge a_2 \wedge \cdots \wedge a_p \wedge \neg b_1 \wedge \neg\, b_2 \wedge \cdots \wedge \neg b_q \wedge \neg c_1 \wedge \neg c_2 \wedge \cdots \wedge \neg c_r$. The Boolean expressions E_1 and E_2 correspond to conditional patterns $\langle Y, X_k\rangle$ and $\langle Y, X_{k+1}\rangle$, respectively. The expression E_2 is more restrictive than the expression E_1. Thus, $supp(E_1, D) \geq supp(E_2, D)$. □

With reference to database D of Example 2.1, let $Y = b$, $X_1 = \{a, b\}$ and $X_2 = \{a, b, c\}$. We have $supp\langle Y, X_1, D\rangle = 0.3$ and $supp\langle Y, X_2, D\rangle = 0.2$. We observe that $supp\langle Y, X_1, D\rangle \geq supp\langle Y, X_2, D\rangle$.

2.4 Mining Conditional Patterns

For mining conditional patterns in a database, we need to find their conditional supports. We calculate $supp\langle Y, X, D\rangle$ in terms of supports of relevant frequent itemsets, for $Y \subseteq X$. Let $X = Y \cup Z$, where $Z = \{a_1, a_2, \ldots, a_p\}$. The following theorem is useful for synthesizing conditional supports using relevant frequent itemsets in D.

Lemma 2.8 *Let* X, Y and Z *are itemsets in database D such that* $X = Y \cup Z$, *where* $Z = \{a_1, a_2, \ldots, a_p\}$. *Then,*

$$\begin{aligned} supp\langle Y, X, D\rangle = {} & supp(Y, D) - \sum_{\mathrm{i=1}}^{\mathrm{p}} supp(Y \cap \{a_{\mathrm{i}}\}, D) + \sum_{\mathrm{i<j;\ i,j=1}}^{\mathrm{p}} supp(Y \cap \{a_{\mathrm{i}}, a_{\mathrm{j}}\}, D) \\ & - \sum_{\mathrm{i<j<k;\ i,j,k=1}}^{\mathrm{p}} supp(Y \cap \{a_{\mathrm{i}}, a_{\mathrm{j}}, a_{\mathrm{k}}\}, D) + \cdots + (-1)^{\mathrm{p}} \\ & \times supp(Y \cap \{a_1, a_2, \ldots, a_{\mathrm{p}}\}, D) \end{aligned} \tag{2.4}$$

Proof We shall prove the result using method of induction on p. For $p = 1$, $X = Y \cap \{a_1\}$. Then, $supp\langle Y, X, D\rangle = supp(Y, D) - supp(Y \cap \{a_1\}, D)$. Thus, the result is true for $p = 1$. Let us assume that the result is true for $p = m$. □

We shall prove that the result is true for $p = m + 1$. Let $Z = \{a_1, a_2, \ldots, a_{\mathrm{m+1}}\}$. Due to the addition of item $a_{\mathrm{m+1}}$, many supports are required to be added to or, subtracted from the expression of $supp\langle Y, X, D\rangle$, for $p = m$. For example, $supp(Y \cap \{a_{\mathrm{m+1}}\}, D)$ is required to be subtracted, $supp(Y \cap \{a_{\mathrm{i}}, a_{\mathrm{m+1}}\}, D)$ is required to be added, for $1 \le i \le m$, and so on. Finally, the term $(-1)^{\mathrm{m+1}} \times supp(Y \cap \{a_1, a_2, \ldots, a_{\mathrm{m+1}}\}, D)$ is required to be added. Thus, the expression of $supp\langle Y, X, D\rangle$ at $p = m + 1$, is given as follows.

$$\begin{aligned} supp\langle Y, X, D\rangle = {} & supp(Y, D) - \sum_{\mathrm{i=1}}^{\mathrm{m+1}} supp(Y \cap \{a_{\mathrm{i}}\}, D) + \sum_{\mathrm{i<j;\ i,j=1}}^{\mathrm{m+1}} supp(Y \cap \{a_{\mathrm{i}}, a_{\mathrm{j}}\}, D) \\ & - \sum_{\mathrm{i<j<k;\ i,j,k=1}}^{\mathrm{m+1}} supp(Y \cap \{a_{\mathrm{i}}, a_{\mathrm{j}}, a_{\mathrm{k}}\}, D) + \cdots + (-1)^{\mathrm{m+1}} \\ & \times supp(Y \cap \{a_1, a_2, \ldots, a_{\mathrm{m+1}}\}, D). \end{aligned}$$

Formulas (2.1) and (2.2) validate above theorem. We shall use this formula in the proposed algorithm to compute conditional support of a conditional pattern.

Lemma 2.9 *The maximum number of non-trivial conditional patterns is equal to* $\sum_{X \in \mathrm{SFIS(D)};\ |X| \ge 2} 2^{|X|-2}$, *where SFIS*(*D*) *is the set of frequent itemsets in database D.*

Proof The number of subsets of X is equal to $2^{|X|-2}$ such that $Y \ne \phi$, for $Y \subset X$. Each such subset of X corresponds to a non-trivial conditional pattern with reference to X. Thus, the lemma follows. □

The interestingness of a conditional pattern is judged by its conditional support and reference support. By combining both the measures one could define many interestingness measures of a conditional pattern. An appealing measure of interestingness of a conditional pattern $\langle Y, X\rangle$ in database D could be $csupp\langle Y, X, D\rangle + rsupp\langle Y, X, D\rangle$.

2.4.1 Algorithm Design

For mining conditional patterns in a database, we make use of an existing frequent itemset mining algorithm (Agrawal and Srikant 1994; Han et al. 2000; Savasere et al. 1995). There are two approaches of mining conditional patterns in a database.

In the first approach, we could synthesize conditional patterns from current frequent itemset extracted during the mining process. As soon as a frequent itemset is found during the mining process, we could call an algorithm of finding conditional patterns that generates conditional patterns from the current frequent itemset. When a frequent itemset is extracted, then all the non-null subsets of the frequent itemest have already been extracted. Thus, we could synthesize all the conditional patterns from the current frequent itemset extracted from the database. In the second approach, we could synthesize conditional patterns from the frequent itemsets in the given database after mining of all frequent itemsets. Thus, all the frequent itemsets are processed at the end of mining task. These two approaches seem to be the same so far as the computational complexity is concerned. In this chapter, we have followed the second approach of synthesizing conditional patterns. During the process of mining frequent itemsets, the frequent itemsets of smaller size get extracted before the frequent itemsets of larger size. The frequent itemsets are stored in array *SFIS* and get sorted based on their size automatically. During the processing of current frequent itemset, all the non-null subsets are available before the current itemset in *SFIS*.

Before presenting proposed algorithm of synthesizing the conditional patterns, we first state how we have designed the synthesizing algorithm. The frequent itemsets of size one can not generate conditional patterns. Thus, the algorithm skips processing frequent itemsets of size one. There are $2^{|\mathrm{X}|-1}$ non-null subsets of an itemset X. Each non-null subset of X may correspond to an interesting conditional pattern, for $|X| \geq 2$. The subset X of X corresponds to a trivial conditional pattern. Thus, we need to process $2^{|\mathrm{X}|-2}$ subsets of X.

One could view a conditional pattern as an object having following attributes: *pattern*, *reference*, *csupp*, and *rsupp*. We use an array *CP* to store conditional patterns in a database. The y attribute of *i*th conditional pattern is accessed by notation $CP(i) \cdot \mathrm{y}$. Also, a frequent itemset could be viewed as an object described by a set of attributes. A frequent itemset could be described by the following attributes: itemset and supp. Let N be the number of frequent itemsets in the given database D. The variables i and j are used to index the frequent itemset being processed and the conditional pattern being synthesized, respectively. An algorithm for synthesizing interesting non-trivial conditional patterns is presented below.

Algorithm 2.1. Synthesize interesting non-trivial conditional patterns in a database.

```
procedure conditional-pattern-synthesis (N, SFIS)
Input:
N: number of frequent itemsets in the given database
SFIS: array of frequent itemsets in the given database
Output:
Interesting non-trivial conditional patterns
01: let $i = 1$;
02: let $j = 1$;
03: while ($|SFIS(i)| = 1$) do
04:     increase $i$ by 1;
05: end while
06: while ($i \le N$) do
07:    $CP(j).rsupp = SFIS(i).supp$; $CP(j).reference = SFIS(i).itemset$;
08:    let $sum = 0$;
09:    for $k = 1$ to $(2^{|SFIS(i).itemset| - 1})$ do
10:       let $tempItemset$ = $k$-th subset of $SFIS(i).itemset$;
11:       if ($SFIS(i).itemset = tempItemset$) then goto line 24; end if
12:       let $kk = 1$;
13:       while ($kk \le i$) do
14:          if ($SFIS(kk).itemset = tempItemset$) then
15:             $sum = sum + (-1)^{|SFIS(kk).itemset| - |tempItemset|} \times SFIS(kk).supp$;
16:             goto line 21;
17:          end if
18:          increase $kk$ by 1;
19:       end while
20:    end for
21:    if ($sum \ge \delta$) then
22:       $CP(i).csupp = sum$; $CP(i).pattern = tempItemset$;
23:       increase $j$ by 1;
24:    end if
25:    increase $i$ by 1;
26: end while
27: sort conditional patterns on ($csupp + rsupp$) in non-increasing order;
28: for $k = 1$ to $j$ do
29:    display $k$-th conditional pattern;
30: end for
31: end procedure
```

In this section, we explain and justify the statements of the above algorithm. The important parts of the algorithm are explained as follows: The frequent itemsets of size one generate trivial conditional patterns. Thus, we have skipped processing frequent itemsets of size one using lines 3–5. We synthesize conditional patterns using lines 6–26. There are $2^{|X|-1}$ non-null subsets for an itemset X. Each subset is considered using a *for*-loop in lines 9–20. The algorithm synthesizes conditional patterns with reference to a frequent itemset X, for $|X| \geq 2$. The algorithm bypasses processing itemset Y, if $Y = X$. When we synthesize conditional patterns with reference to a frequent itemset, we have already finished synthesizing its subsets. All the non-null subsets appear on or before the frequent itemset in *SFIS*. Thus, if a frequent itemset *X* located at position *i*, then we search for a subset of *X* from index 1 to *i* in *SFIS*, since *SFIS* is sorted non-decreasing order on length of an itemset. Thus, it justifies the condition of *while*-loop at line 13. Formula (2.4) expresses $supp\langle Y, X, D\rangle$ in terms of $supp(Y \cap Z, D)$, for all $Z \subseteq X - Y$. The co-efficient of $supp(Y \cap Z, D)$ is $(-1)^{|Z|}$ in the expression of $supp\langle Y, X, D\rangle$. Thus, $supp\langle Y, X, D\rangle = \sum_{Z \subseteq X-Y} (-1)^{|Z|} \times supp\langle Y, X, D\rangle$. This formula has been applied at line 15 to calculate $supp\langle Y, X, D\rangle$. A conditional pattern is interesting if the conditional support is greater than or equal to δ, provided the reference support of the itemset is greater than or equal to α. We need not check the reference support, since we deal with the frequent itemsets. In line 21, we check whether the currently synthesized conditional pattern is interesting. The details of a synthesized conditional pattern are stored using lines 7 and 22. At line 27, we sort all interesting conditional patterns in the given database. Finally, we display interesting conditional patterns using lines 28–30.

Lemma 2.10 *Algorithm conditional-pattern-synthesis executes in* $O(N^2 \times 2^p)$ *time, where N is the number of frequent itemsets in the database.*

Proof Lines 3–5 take $O(N)$ time. The *while*-loop at line 6 repeats maximum *N* times. Let the average size of the frequent itemsets of size greater than 1 be *p*. Thus, the *for*-loop at line 9 repeats 2^{p-1} times. The *while*-loop at line 13 repeats maximum *N* times. Thus, the time complexity of lines 6–26 is equal to $O(N^2 \times 2^p)$. The time complexity of line 27 is equal to $O(N \times 2^p \times \log(N \times 2^p))$, since the number of conditional patterns is equal to $O(N \times 2^p)$. The time complexity of lines 28–30 is equal to $O(N \times 2^p)$. Therefore, the time complexity of the algorithm is *maximum* $\{O(N^2 \times 2^p), O(N \times 2^p \times \log(N \times 2^p))\}$. □

2.5 Experiments

We have carried out several experiments to study the effectiveness of our approach. We present experimental results using three real databases. Database *retail* (Frequent itemset mining dataset repository 2004) is obtained from an anonymous Belgian retail supermarket store. Databases *BMS-Web-Wiew-1* and *BMS-Web-Wiew-2* can be found from KDD CUP 2000 (Frequent itemset mining dataset repository 2004). We present some characteristics of these databases in Table 2.7.

Let *NT*, *AFI*, *ALT*, and *NI* denote the number of transactions, the average frequency of an item, the average length of a transaction, and the number of items in the corresponding database respectively. Top five interesting conditional patterns of available categories are shown in Table 2.8. We have implemented apriori algorithm for the purpose of mining conditional patterns in the given databases. The

Table 2.7 Database characteristics

Database	*NT*	*ALT*	*AFI*	*NI*
retail	88,162	11.305755	99.673800	10,000
BMS-Web-Wiew-1	1,49,639	2.000000	155.711759	1,922
BMS-Web-Wiew-2	3,58,278	2.000000	7165.560000	100

Table 2.8 Top 5 conditional patterns of each category available in *retail* at $\alpha = 0.05$ and $\delta = 0.03$

Conditional pattern	*csupp*	*rsupp*
$\langle\{39\}, \{1, 39\}\rangle$	0.520451	0.066332
$\langle\{39\}, \{8, 39\}\rangle$	0.524421	0.062362
$\langle\{39\}, \{0, 39\}\rangle$	0.526871	0.059912
$\langle\{39\}, \{2, 39\}\rangle$	0.525612	0.061171
$\langle\{39\}, \{3, 39\}\rangle$	0.525714	0.061069
$\langle\{39\}, \{39, 41, 48\}\rangle$	0.210317	0.083551
$\langle\{39\}, \{32, 39, 48\}\rangle$	0.221603	0.061274
$\langle\{39\}, \{38, 39, 48\}\rangle$	0.208106	0.069213
$\langle\{48\}, \{39, 41, 48\}\rangle$	0.139482	0.083551
$\langle\{32\}, \{32, 39, 48\}\rangle$	0.049432	0.061274
$\langle\{39,48\}, \{32, 39, 48\}\rangle$	0.269277	0.061274
$\langle\{39, 48\}, \{39, 41, 48\}\rangle$	0.247000	0.083551
$\langle\{39, 48\}, \{38, 39, 48\}\rangle$	0.261337	0.069213
$\langle\{38, 39\}, \{38, 39, 48\}\rangle$	0.048127	0.069213
$\langle\{32, 39\}, \{32, 39, 48\}\rangle$	0.034629	0.061274

conditional patterns in a database are ranked based on the sum of conditional support and reference support.

In both *BMS-Web-Wiew-1* and *BMS-Web-Wiew-2*, only one category of conditional patterns is available, since the maximum length of a transaction in each of these two databases is 2.

We have also conducted experiments for finding time needed to mine conditional patterns in different databases. The execution time for finding conditional patterns in a database increases as the size, i.e., the number of transactions contained in a database increases. We observe this phenomenon in Figs. 2.2 and 2.3. We have also conducted experiments to find time needed to synthesize conditional patterns in a database. The time (only) for synthesizing conditional patterns in each of the above databases is equal to 0 ms at the respective values of α and δ shown in Tables 2.8, 2.9 and 2.10.

We have also conducted experiments for finding the number of conditional patterns in a database at a given α. The number of conditional patterns in a database decreases as α increases. We observe this phenomenon in Figs. 2.4 and 2.5.

We have also conducted experiments for finding execution time needed for mining conditional patterns in a database at a given α. The execution time needed

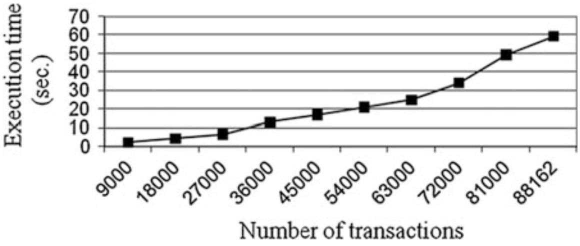


Fig. 2.2 Execution time versus the number of transactions in *retail*

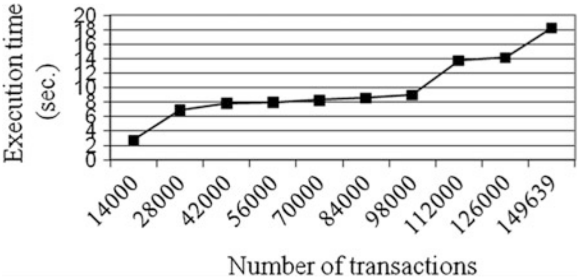


Fig. 2.3 Execution time versus the number of transactions in *BMS-Web-Wiew-1*

Table 2.9 Top 5 conditional patterns of each category available in *BMS-Web-Wiew-1* at $\alpha = 0.01$ and $\delta = 0.009$

Conditional pattern	*csupp*	*rsupp*
⟨{5}, {1, 5}⟩	0.235453	0.013740
⟨{5}, {3, 7}⟩	0.236135	0.013058
⟨{5}, {5, 7}⟩	0.235293	0.013900
⟨{5}, {5, 9}⟩	0.236335	0.012858
⟨{7}, {7, 9}⟩	0.203563	0.011568

Table 2.10 Top 5 conditional patterns of each category available in *BMS-Web-Wiew-2* at $\alpha = 0.009$ and $\delta = 0.007$

Conditional pattern	*csupp*	*rsupp*
⟨{7}, {1, 7}⟩	0.174072	0.022943
⟨{7}, {6, 7}⟩	0.185401	0.011614
⟨{7}, {7, 9}⟩	0.175810	0.021204
⟨{7}, {0, 7}⟩	0.185702	0.011312
⟨{7}, {2, 7}⟩	0.185747	0.011268

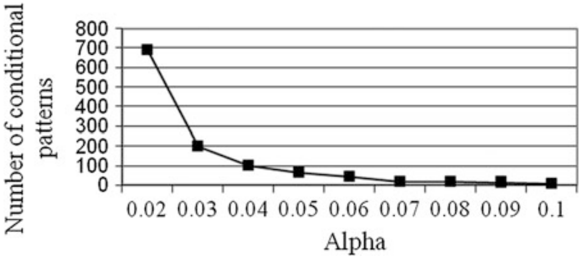
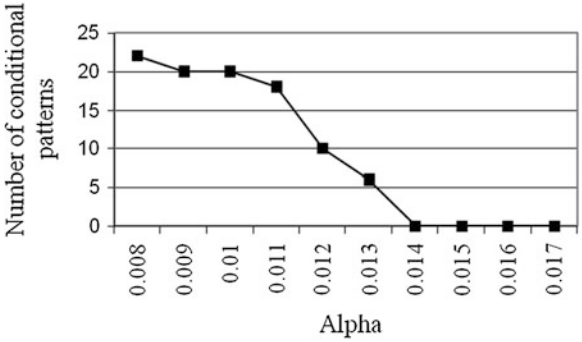


Fig. 2.4 Number of conditional patterns versus α for *retail*

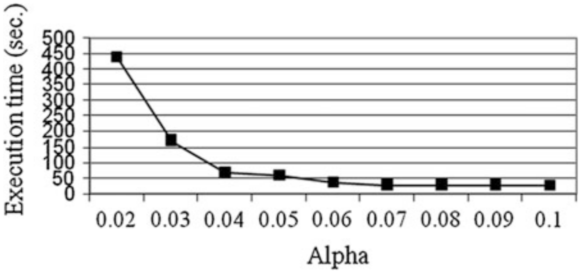
Fig. 2.5 Number of conditional patterns versus α for *BMS-Web-Wiew-1*



for mining conditional patterns in a database decreases as α increases. We observe this phenomenon in Figs. 2.6 and 2.7.

Also, we have conducted experiments to study the relationship between the size of a database and the number of conditional patterns in it. The experiments are conducted on databases *retail* and *BMS-Web-Wiew-1*. The results of the experiments are shown in Figs. 2.8 and 2.9. From the graphs in Figs. 2.8 and 2.9, we could conclude that there is no universal relationship between the size of a database and the number of conditional patterns in it.

Fig. 2.6 Execution time versus α for *retail*



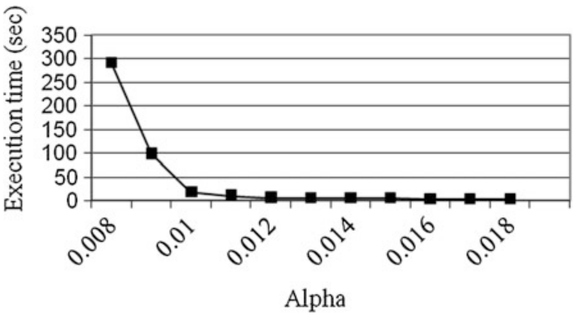


Fig. 2.7 Execution time versus α for *BMS-Web-Wiew-1*

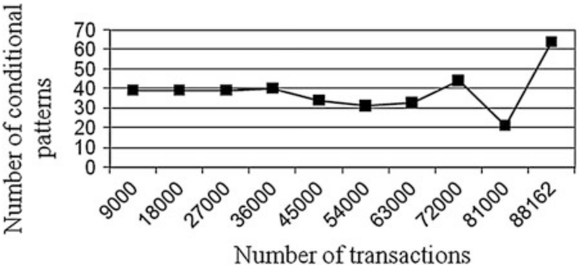


Fig. 2.8 Number of conditional patterns versus the number of transactions in *retail*

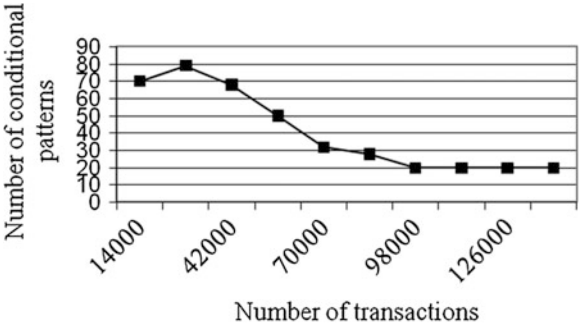


Fig. 2.9 Number of conditional patterns versus the number of transactions in *BMS-Web-Wiew-1*

Also, we have conducted experiments to study the relationship between the number of conditional patterns and conditional support. The experiments have been conducted on databases *retail* and *BMS-Web-Wiew-1*. The number of conditional patterns in a database decreases as δ increases. We observe this phenomenon in Figs. 2.10 and 2.11.

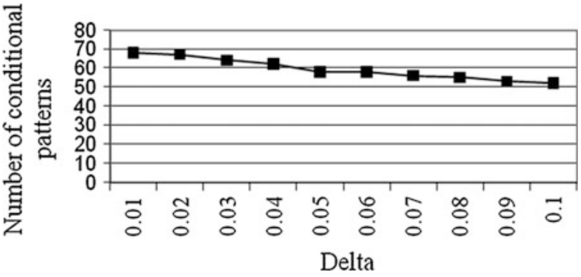


Fig. 2.10 Number of conditional patterns versus δ for *retail*

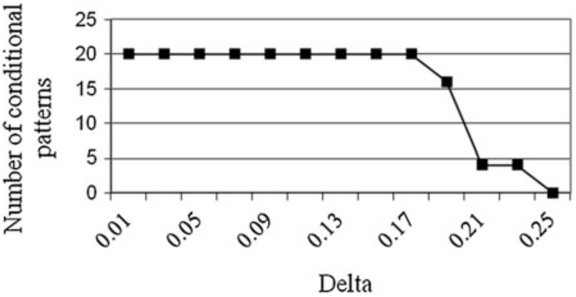


Fig. 2.11 Number of conditional patterns versus δ for *BMS-Web-Wiew-1*

2.5.1 An Application

Adhikari and Rao (2007) have proposed a technique for mining arbitrary Boolean expressions induced by frequent itemsets using conditional patterns in a database. The pattern itemset of a conditional pattern with respect to itemset X = {a_1, a_2, …, a_m} is of the form {b_1, b_2, …, b_m}, where $b_i = a_i$, or $\neg a_i$, for i = 1, 2, …, m. Let Ψ(X) be the set of all such pattern itemsets with respect to X. Then Ψ(X) could be called as the *generator* of Boolean expressions induced by X. Ψ(X) contains $2^m - 1$ pattern itemsets. A pattern itemset of the corresponding conditional pattern is also called a *minterm*, or *standard product*. Every Boolean expression of items of X could be constructed using pattern itemsets in Ψ(X). In particular, let X = {a, b, c}. Then, Ψ(X) = {{a, b, c}, {a, b, ¬c}, {a, ¬b, c}, {a, ¬b, ¬c}, {¬a, b, c}, {¬a, b, ¬c}, {¬a, ¬b, c}}. Boolean expression ¬b ∧ c could be expressed by the pattern itemsets as follows: (a ∧ ¬b ∧ c) ∨ (¬a ∧ ¬b ∧ c). Every Boolean expression could be expressed by pattern itemsets in the corresponding generator. A Boolean expression expressed as a sum of pattern itemsets is said to be in *canonical form*. Each pattern itemset corresponds to a set of transactions in D. In the following, we show how each pattern itemset with respect to {a, b, c} corresponds to a set of transactions in D.

The shaded region in Fig. 2.12a contains the set of transactions containing the items a, b and c with respect to {a, b, c}. Thus, it corresponds to the pattern itemset of ⟨{a, b, c}, {a, b, c}⟩. The shaded region in Fig. 2.12b contains the set of transactions containing the items a, and b, but not the item c, with respect to

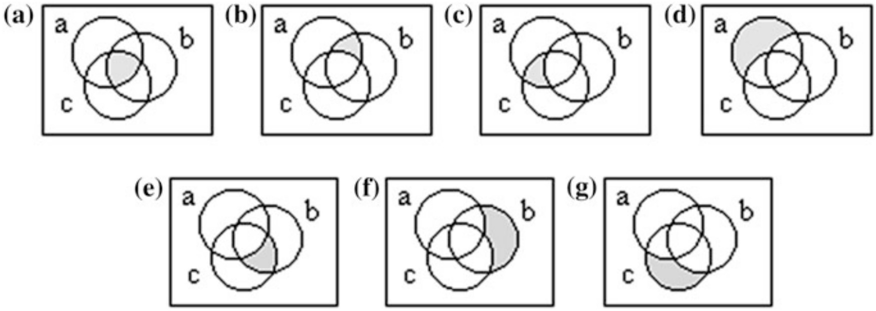


Fig. 2.12 Generator of {a, b, c}. **a** a ∧ b ∧ c, **b** a ∧ b ∧ ¬c, **c** a ∧ ¬b ∧ c, **d** a ∧ ¬b ∧ ¬c, **e** ¬a ∧ b ∧ c, **f** ¬a ∧ b ∧ ¬c, **g** ¬a ∧ ¬b ∧ c

{a, b, c}. Thus, it corresponds to the pattern itemset of ⟨{a, b}, {a, b, c}⟩. The shaded region in Fig. 2.12c contains the set of transactions containing the items a and c, but not the item b, with respect to {a, b, c}. Thus, it corresponds to the pattern itemset of ⟨{a, c}, {a, b, c}⟩. The shaded region in Fig. 2.12d contains the set of transactions containing the item a, but not the items b and c, with respect to {a, b, c}. Thus, it corresponds to the pattern itemset of ⟨{a}, {a, b, c}⟩. The shaded region in Fig. 2.12e contains the set of transactions containing the items b and c, but not the item a, with respect to {a, b, c}. Thus, it corresponds to the pattern itemset of ⟨{b, c}, {a, b, c}⟩. The shaded region in Fig. 2.12f contains the set of transactions containing the item b, but not the items a and c, with respect to {a, b, c}. Thus, it corresponds to the pattern itemset of ⟨{b}, {a, b, c}⟩. Finally, the shaded region in Fig. 2.12g contains the set of transactions containing the item c, but not the items a and b, with respect to {a, b, c}. Thus, it corresponds to the pattern itemset of ⟨{c}, {a, b, c}⟩.

Let X = {a, b, c}. Consider the Boolean expressions E_1({a, b, c}) = c ∨ (a ∧ ¬b) and E_2({a, b, c}) = ¬a ∧ ¬c given in Fig. 2.13.

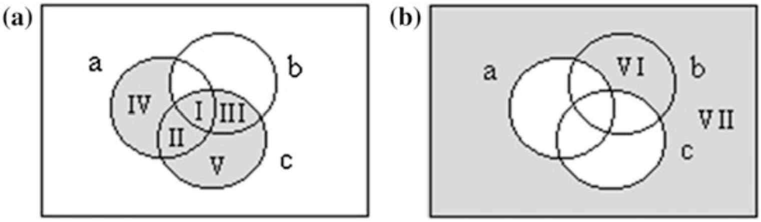


Fig. 2.13 Boolean expressions E_1({a, b, c}) = c ∨ (a ∧ ¬b) and E_2({a, b, c}) = ¬a ∧ ¬c represent the shaded areas of (**a**) and (**b**) respectively

The supports of above Boolean expressions could be computed as follows. supp (E_1, D) could be obtained by adding the supports of regions I, II, III, IV, and V. These regions are mutually exclusive. Each of these regions corresponds to a member of $\Psi(\{a, b, c\})$. Thus, supp(E_1, D) = supp(a ∧ b ∧ c, D) + supp(a ∧ ¬b ∧ c, D) + supp(¬a ∧ b ∧ c, D) + supp(a ∧ ¬b ∧ ¬c, D) + supp(¬a ∧ ¬b ∧ c, D). Also, supp(E_2, D) could be obtained by adding the supports of regions VI and VII. But, the region VII does not correspond to any member of $\Psi(\{a, b, c\})$. Now, supp(¬E_2, D) = supp(a ∧ b ∧ c, D) + supp(a ∧ ¬b ∧ c, D) + supp(¬a ∧ b ∧ c, D) + supp (a ∧ ¬b ∧ ¬c, D) + supp(¬a ∧ ¬b ∧ c, D) + supp(a ∧ b ∧ ¬c, D). Therefore, supp(E_2, D) = 1 − supp(¬E_2, D).

2.6 Related Work

Agrawal et al. (1993) introduce association rule and support-confidence framework and an algorithm to mine frequent itemsets. The algorithm is sometimes called AIS after the authors' initials. Since then, many algorithms have been reported to generate association rules in a database. Association rule mining finds interesting association between two itemsets in a database. Agrawal and Srikant (1994) introduce apriori algorithm that uses breadth-first search strategy to count the support of itemsets. The algorithm uses an improved candidate generation function, which exploits the downward closure property of support and makes it more efficient than AIS. Han et al. (2000) describe the data mining method FP-growth that uses an extended prefix-tree structure to store the databases in a compressed form. FP-growth adopts a divide-and-conquer approach to decompose both the mining tasks and databases. It uses a pattern fragment growth method to avoid the costly process of candidate generation and testing. Savasere et al. (1995) have introduced partition algorithm. The database is scanned only twice. For the first scan, the database is partitioned and in each partition support is counted. Then the counts are merged to generate potential frequent itemsets. In the second scan, the potential frequent itemsets are counted to find the actual frequent itemsets.

In the context of pattern synthesis, Viswanath et al. (2006) have proposed a novel pattern synthesis method called partition based pattern synthesis which can generate an artificial training set of exponential order when compared with that of the given original training set.

In the context of other applications of data mining, Hong and Weiss (2001) have examined a few successful application areas and their technical challenges to show how the demand for data mining of massive data warehouses has fuelled advances in automated predictive methods.

2.7 Conclusion and Future Work

Frequent itemsets could be considered as the basic ingredient of a database. Thus, we could analyze the characteristics of a database in more detail by mining various patterns with respect to frequent itemsets. This chapter introduces conditional patterns in a database and proposes an algorithm to mine them. Thus, we could reveal more characteristics of a database using conditional patterns. Also, we have observed that conditional patterns store significant nuggets of knowledge about a database that are not immediately available from frequent itemsets and association rules. In Sect. 2.5.1, we have presented an application of conditional patterns in a database. In future also, we shall search for more applications of conditional patterns in a database.

References

Adhikari A, Rao PR (2007) A framework for synthesizing arbitrary Boolean expressions induced by frequent itemsets. In: Proceedings of 3rd Indian international conference on artificial intelligence, pp 5–23

Adhikari A, Rao PR (2008) Mining conditional patterns in a database. Pattern Recogn Lett 29 (10):1515–1523

Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Proceedings of 20th very large databases (VLDB) conference, pp 487–499

Agrawal R, Imielinski T, Swami A (1993) Mining association rules between sets of items in large databases. In: Proceedings of ACM SIGMOD conference management of data, pp 207–216

Antonie M-L, Zaïane OR (2004) Mining positive and negative association rules: an approach for confined rules. In: Proceedings of PKDD, pp 27–38

Feller W (1968) An introduction to probability theory and its applications, vol 1, 3rd edn. Wiley, New York

FIMI (2004) http://fimi.cs.helsinki.fi/src/

Frequent itemset mining dataset repository (2004) http://fimi.cs.helsinki.fi/data

Han J, Pei J, Yiwen Y (2000) Mining frequent patterns without candidate generation. In: Proceedings of ACM SIGMOD conference management of data, pp 1–12

Hilderman RJ, Hamilton HJ (1999) Knowledge discovery and interestingness measures: a survey. In: Technical report CS-99-04, Department of Computer Science, University of Regina

Hong SJ, Weiss SM (2001) Advances in predictive models for data mining. Pattern Recogn Lett 22(1):55–61

Savasere A, Omiecinski E, Navathe S (1995) An efficient algorithm for mining association rules in large databases. In: Proceedings of the 21st international conference on very large data bases, pp 432–443

Tan P-N, Kumar V, Srivastava J (2002) Selecting the right interestingness measure for association patterns. In: Proceedings of SIGKDD conference, pp 32–41

Viswanath P, Murty MN, Bhatnagar S (2006) Partition based pattern synthesis technique with efficient algorithms for nearest neighbor classification. Pattern Recogn Lett 27(14):1714–1724

Wu X, Wu Y, Wang Y. Li Y (2005) Privacy-aware market basket data set generation: a feasible approach for inverse frequent set mining. In: Proceedings of SIAM international conference on data mining, pp 103–114

Springer

http://www.springer.com/978-3-319-13211-2

Advances in Knowledge Discovery in Databases
Adhikari, A.; Adhikari, J.
2015, XV, 370 p. 136 illus., Hardcover
ISBN: 978-3-319-13211-2