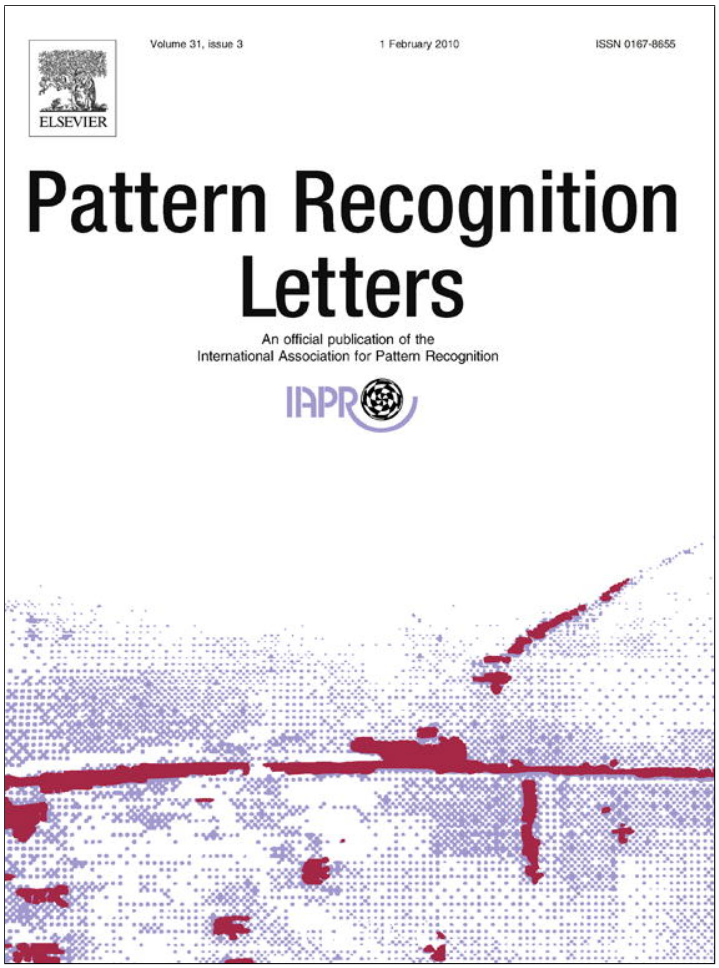

Pattern Recognition Letters 31 (2010) 179–187

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Measuring influence of an item in a database over time

Jhimli Adhikari [a,*], P.R. Rao [b]

[a] Department of Computer Science, Narayan Zantye College, Bicholim, Goa 403 529, India
[b] Department of Computer Science and Technology, Goa University, Goa 403 206, India

ARTICLE INFO

Article history:
Received 15 July 2008
Received in revised form 6 October 2009
Available online 21 October 2009

Communicated by W. Pedrycz

Keywords:
Confidence
Overall association
Overall influence
Synthesis of influence

ABSTRACT

Influence of items on some other items might not be the same as the association between these sets of items. Many tasks of data analysis are based on expressing influence of items on other items. In this paper, we introduce the notion of an overall influence of a set of items on another set of items. We also propose an extension to the notion of overall association between two items in a database. Using the notion of overall influence, we have designed two algorithms for influence analysis involving specific items in a database. As the number of databases increases on a yearly basis, we have adopted incremental approach in these algorithms. Experimental results are reported for both synthetic and real-world databases.

1. Introduction

Every time a customer interacts with business, we have an opportunity to gain strategic knowledge. Transactional data contains a wealth of information about customers and their purchase patterns. In fact, these data could be one of the most valuable assets, when used wisely. This has been recognized a long time ago by many large organizations such as supermarkets, insurance companies, healthcare organizations, telecommunications, and banks. These organizations have spent significant resources for collecting and analyzing transactional data. Many applications are based on inherent knowledge present in a database (Gary and Petersen, 2000; Wu et al., 2005; Adhikari et al., 2009). Such applications could be dealt with mining databases (Han et al., 2000; Agrawal and Srikant, 1994; Savasere et al., 1995). As a database changes over time, the inherent knowledge also changes. Therefore in the competitive market, knowledge-based decisions are more appropriate. Data mining algorithms are effective tools to support making such decisions. Data mining algorithms often extracts different patterns from a database. Some examples of patterns in a database are frequent item sets (Agrawal et al., 1993), association rules (Agrawal et al., 1993), negative association rules (Wu et al., 2004), Boolean expressions induced by itemset (Adhikari and Rao, 2007b) and conditional patterns (Adhikari and Rao, 2008a). Nevertheless, there are some applications for which association-based analysis might be inappropriate. For example, an organization might deal with a large number of items with its customers. The company might be interested in knowing how the purchase of a particular item affects the purchase of some other item. In this paper, we study such influences based on transactional time-stamped database.

Many companies transact a large number of products (items) with their customers. It might be required to perform data analyses involving different items. Such analyses might originate from different applications. One such analysis is identifying stable items (Adhikari et al., 2009) in databases over time. It could be useful in devising strategies for a company. Little work has been reported on data analyses over time. In this paper, we present another application involving different items in a database over time.

Consider a company that collects a huge amount of transactional data on a yearly basis. Let DT_i be the database corresponding to the ith year, $i = 1, 2, \ldots, k$. Each of these databases corresponds to a specific period of time. Thus, one could call these time databases. Each time database is mined using a traditional data mining technique (Adhikari and Rao, 2007a). In this application, we will deal with itemsets in a database. An itemset is a set of items in the database. Let I be the set of all items in the time databases. Each itemset X in a database D is associated with a statistical measure, called *support* (Agrawal et al., 1993), denoted by $supp(X, D)$. The support of an itemset is defined as the fraction of transactions containing the itemset.

Solutions to many problems are based on the study of relationships among variables. We will see later that the study of influence of a set of variables on another set of variables might not be the same as the association between these two sets of variables. Association analysis among variables has been studied well (Agrawal et al., 1993; Adhikari and Rao, 2007a, 2008b,c; Brin et al., 1997; Shapiro, 1991). In the context of studying association among

* Corresponding author. Fax: +91 0832 2361377.
E-mail addresses: jhimli_adhikari@yahoo.co.in (J. Adhikari), pralhaad@rediffmail.com (P.R. Rao).

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.10.010

variables using association rules one could conclude that the confidence of the association rule gives positive influence of antecedent on the consequent of the association rule. Such positive influences might not be sufficient for many data analyses.

Consider an established company possessing data over 50 consecutive years. Generally, the sales of a product vary from one season to another season. Also, a season re-appears on a yearly basis. Thus, we divide the entire database into a sequence of yearly databases. In this context, a yearly database could be considered as a time database. In this study, we estimate the influence of item x on y, for $x, y \in I$, where I is the set of all items in database D. In Section 3, we define the concept of influence of an itemset on another itemset.

An itemset could be viewed as a basic type of pattern in a database. Different types of pattern in a database could be derived from itemset patterns. For example, frequent itemset, association rule, negative association rule, Boolean expression induced by itemset and conditional pattern are examples of derived patterns in a database. Few applications have been reported on analysis of patterns over time. In this paper, we wish to study the influence of an item on a specific item/a set of specific items in a database.

Most of the association analyses are based on a positive association between variables. Such positive association gives rise to positive influence of variables on other variables. Most of the real databases are large and sparse. In such cases an association analysis using positive influence might not be appropriate, if the overall influence of former variable on latter variable becomes negative. Thus, the concept of overall influence needs to be introduced.

The rest of the paper is organized as follows: in Section 2, we extend the notion of overall association between two items in a database. In Section 3, we introduce the notion of overall influence of an itemset on another itemset in a database. We study various properties of proposed measures. Also, we introduce the notion of overall influence of an item on a set of specific items in a database. In addition, we discuss the motivation of the proposed problem in this section. We state our problem in Section 4. We discuss work related to proposed problem in Section 5. In Section 6, we design an algorithm to measure the overall influence of an item on another item (incrementally). In addition, we design another algorithm of overall influence of an item on a set of specific items (incrementally). Experimental results are provided in Section 7. We conclude the paper in Section 8.

2. Association between two itemsets

Adhikari and Rao (2007a) have proposed a measure denoted by *OA*, for computing an overall association between two items in a market basket data. Using positive association *PA* between two items (Adhikari and Rao, 2007a), one could extend positive association between two itemsets in a database as follows:

$$PA(X,Y,D) = \frac{\#\text{transaction containing both } X \text{ and } Y, D}{\#\text{transaction containing at least one of } X \text{ and } Y, D},$$

where X and Y are itemsets in database D and "#P, D" is the number of transactions in D that satisfy the predicate P.

Similarly, negative association *NA* between two items (Adhikari and Rao, 2007a) could be extended as follows:

$$NA(X,Y,D) = \frac{\#\text{transaction containing exactly one of } X \text{ and } Y, D}{\#\text{transaction containing at least one of } X \text{ and } Y, D},$$

where X and Y are itemsets in database D.

Using *PA* and *NA*, *OA* between two itemsets X and Y in database D could be defined as follows:

$$OA(X,Y,D) = PA(X,Y,D) - NA(X,Y,D). \quad (1)$$

If $OA(X, Y, D)$ is positive, negative or zero then all the items in X together and all the items in Y together are positively, negatively or independently associated in D, respectively. We illustrate different types of association in the following example.

Example 1. Let database D_1 contain the following transactions: $\{a,d,e\}, \{a,b,c,d,g\}, \{a,b,e,g\}, \{b,c,g\}, \{d,e,g\}, \{b,e,f\}, \{c,d,e,f\}, \{a,b,c,d,f,g\}$, and $\{a,b,c,d,e\}$. We find here overall association between itemsets X, and Y, for some X, Y in D_1. In Table 1, supports of some itemsets are given below.

Here $PA(\{a, b\}, \{c, d\}, D_1) = 3/5$ and $NA(\{a, b\}, \{c, d\}, D_1) = 2/5$. Therefore, $OA(\{a, b\}, \{c, d\}, D_1) = 1/5$. In Table 2, overall associations are given.

In Table 2, we observe that the *OA* value between $\{a, b\}$ and $\{c, d\}$ as well as $\{a, c\}$ and $\{b, d\}$ are positive. But, the *OA* value between $\{c\}$ and $\{d, e\}$ is negative.

3. Concept of influence

Let X and Y be two itemsets in database D. We wish to find influence of X on Y in D. In the above section, we have proposed overall association between two itemsets. The influence of X on Y seems to be different from overall association between X and Y.

Let $X = \{x_1, x_2, \ldots, x_p\}$ and $Y = \{y_1, y_2, \ldots, y_q\}$ be two itemsets in database D. The influence of X on Y could be judged by the following events: (i) whether a customer purchases all the items of Y when they purchase all the items of X and (ii) whether a customer purchases all the items of Y when they do not purchase all the items of X. Such behaviors could be modeled using supports of $X \cap Y$ and $\neg X \cap Y$. The expression $supp(X \cap Y, D)/supp(X, D)$ measures the strength of positive association of X on Y. The expression $supp(\neg X \cap Y, D)/supp(\neg X, D)$ measures the strength of negative association of X on Y. Thus, the expressions $supp(X \cap Y, D)/supp(X, D)$ and $supp(\neg X \cap Y, D)/supp(\neg X, D)$ could be important in measuring overall influence of X on Y.

3.1. Influence of an itemset on another itemset

Let X and Y be the two itemsets in database D. The interestingness of an association rule $r_1: X \to Y$ could be expressed by its support and confidence (*conf*) measures (Agrawal et al., 1993). These measures are defined as follows. $supp(r_1, D) = supp(X \cap Y, D)$, and $conf(r_1, D) = supp(X \cap Y, D/supp(X, D)$. The measure $conf(r_1, D)$ could be interpreted as the fraction of transactions containing itemset Y among the transactions containing X in D. In other words, $conf(r_1, D)$ could be viewed as the *positive influence* (*PI*) of X on Y. Let us consider the negative association rule $r_2: \neg X \to Y$. Confidence of r_2 in D could be viewed as fractions of transactions containing Y among the transactions containing $\neg X$. In other words, confidence of r_2 in D could be viewed as *negative influence* (*NI*) of X on Y. In similar to overall association defined in (1), one could define *overall influence* (*OI*) of X on Y in a database as follows:

Table 1
Supports of itemsets in D_1.

Itemset($\{X\}$)	$\{a, b\}$	$\{c, d\}$	$\{a, c\}$	$\{b, d\}$	$\{d, e\}$	$\{e, g\}$
$supp(\{X\}, D_1)$	4/9	4/9	3/9	3/9	4/9	2/9

Table 2
Overall association between two itemsets in D_1.

Itemset($\{X, Y\}$)	$\{\{a, b\}, \{c, d\}\}$	$\{\{a, c\}, \{b, d\}\}$	$\{\{c\}, \{d, e\}\}$
$OA(X, Y, D_1)$	1/5	1	−3/7

Definition 1. Let X and Y be two itemsets in database D such that $X \cap Y = \phi$. Then overall influence of X on Y in D is defined as follows:

$$OI(X,Y,D) = supp(X \cap Y, D)/supp(X,D) - supp(\neg X \cap Y, D)/supp(\neg X, D). \quad (2)$$

$OI(X, Y, D)$ represents the difference of the influence on Y when X is present in a transaction and the influence on Y when X is not present in the transaction. Let γ be user-defined level of interestingness. Then $OI(X, Y, D)$ is *interesting* if $OI(X, Y, D) \geqslant \gamma$.

If $OI(X, Y, D) > 0$ then the itemset X has positive influence on itemset Y in D. In other words, all the items in X together help promoting itemset Y in D. If $OI(X, Y, D) < 0$ then X has negative influence on Y in D. In other words, all the items in X in D together do not help promoting together all the items in Y. If $OI(X, Y, D) = 0$ then X has no influence on Y in D. In Example 2, we illustrate the concept of overall influence.

Example 2. We continue our discussion of Example 1. We have $PI(\{a, b\}, \{c, d\}, D_1) = 3/4$, $NI(\{a, b\}, \{c, d\}, D_1) = 1/5$, and $OI(\{a, b\}, \{c, d\}, D_1) = 11/20$. We observe that $PI(\{a, b\}, \{c, d\}, D_1)$ is more than $PA(\{a, b\}, \{c, d\}, D_1)$. Also, $NA(\{a, b\}, \{c, d\}, D_1)$ is more than $NI(\{a, b\}, \{c, d\}, D_1)$. So, $OI(\{a, b\}, \{c, d\}, D_1)$ is more than $OA(\{a, b\}, \{c, d\}, D_1)$. In similar to overall association, overall influence could be negative also. Let $X = \{c\}$ and $Y = \{d, e\}$. $PI(X, Y, D_1) = 2/5$, $NI(X, Y, D_1) = 1/2$, and $OI(X, Y, D_1) = -1/10$. Thus, overall influence between two itemsets could be negative as well as positive.

In most of the cases, the overall influence between two itemsets in a large database is negative. In real databases, it might be possible that the overall influence between the two itemsets is positive. In Example 3, we consider some special cases to illustrate the measure of overall influence.

Example 3. Let database D_2 contains following transactions: $\{a, b, e\}, \{a, e, g\}, \{b, e, g\}, \{a, b, d, e, g\}, \{b, d, e, g\}$ and $\{c, e, g\}$. We compute overall influence of an itemset X on another itemset Y under various cases.

Case 1: $supp(X, D_2) > supp(Y, D_2)$Let $X = \{e, g\}$, $Y = \{a, b\}$. $supp(X, D_2) = 5/6$, $supp(Y, D_2) = 2/6$ and $supp(X \cap Y, D_2) = 1/6$. We get$OI(X, Y, D_2) = -0.8$.

Case 2: $supp(X, D_2) < supp\ (Y, D_2)$Let $X = \{a, b\}$, $Y = \{e, g\}$. $supp(X, D_2) = 2/6$, $supp(Y, D_2) = 5/6$ and $supp(X \cap Y, D_2) = 1/6$. We get$OI(X, Y, D_2) = -0.5$.

Though the values of overall influence are negative for the above cases, the influence might turn positive for some databases. Let us consider another database $D_3 = \{\{a, b, c, d, g\}, \{b, c, g\}, \{c, d, g\}, \{a, b, c, d, e\}, \{b, c, e, g\}, \{a, b, c, d, e, g\}\}$

Case 1: $supp(X, D_3) > supp(Y, D_3)$Let $X = \{c, d\}$, $Y = \{a, b\}$. $supp(X, D_3) = 4/6$, $supp(Y, D_3) = 3/6$ and $supp(X \cap Y, D_3) = 3/6$. We get $OI(X, Y, D_3) = 0.5$.

Case 2: $supp(X, D_3) < supp(Y, D_3)$. Let $X = \{a, b\}$, $Y = \{c, d\}$. $supp(X, D_3) = 3/6$, $supp(Y, D_3) = 4/6$ and $supp(X \cap Y, D_3) = 3/6$. We get$OI(X, Y, D_3) = 0.667$.

3.2. Properties of influence measures

For the purpose of computing influence of an itemset on another itemset, one needs to express OI in terms of supports of relevant itemsets. From (2), we get OI as follows:

$$OI(X,Y,D) = supp(X \cap Y, D)/supp(X,D) - (supp(Y,D) - supp(X \cap Y, D))/(1 - supp(X,D)).$$

Finally, we get OI as follows:

$$OI(X,Y,D) = \frac{supp(X \cap Y, D) - supp(X,D) \times supp(Y,D)}{supp(X,D)[1 - supp(X,D)]}, \text{ if } supp(X,D) \neq 1 \text{ and } supp(Y,D) \neq 1$$
$$OI(X,Y,D) = 0, \text{ otherwise} \quad (3)$$

From the above formula one could observe that if support of itemset X in D is 1 then influence of other itemsets on X will be zero. On the other hand, if $supp(Y, D) = 1$ then $supp(X \cap Y, D) = supp(X, D)$ and $supp(X, D) \times supp(Y, D) = supp(X, D)$. Therefore, the numerator of formula (3) will result in zero and overall influence becomes zero. In the following lemma, we mention some properties of PI and NI. $OI(X, X, D) = 1$ at $X = Y$. Thus, $OI(X, X, D)$ at $X = Y$ could be termed as *trivial influence*.

Lemma 1. *For itemsets X, Y in D, the following properties are satisfied: (i) $0 \leqslant PI(X, Y, D) \leqslant 1$, (ii) $0 \leqslant NI(X, Y, D) \leqslant 1$, and (iii) $-1 \leqslant OI(X, Y, D) \leqslant 1$.*

Lemma 2. *$OI(X, YD) = \frac{supp(Y)|Corr(X,Y,D)-1|}{1-supp(X)}$, where $Corr(X, Y, D)$ is the correlation coefficient between itemsets X and Y in database D. If $Corr(X, Y, D) = 1$ then X and Y are independent in database D. In other words, if $OI(X, Y, D) = 0$ then X and Y are independent in D. If $Corr(X, Y, D) < 1$ then X and Y are negatively correlated in database D. In other words, if $OI(X, Y, D) < 0$ then X and Y are negatively correlated. If $Corr(X, Y, D) > 1$ then X and Y are positively correlated in database D. If $OI(X, Y, D) > 0$ then X and Y are positively correlated.*

3.3. Influence of an item on a set of specific items

Let $I = \{i_1, i_2, \ldots, i_m\}$ be the set of items in database D. Also, let $SI = \{s_1, s_2, \ldots, s_p\}$ be the set of specific items in database D. We would like to analyze the overall influence of each item on SI. The influence of an item on SI could be computed based on $OI(i_j, s_k, D)$, for $j = 1,2,\ldots,m$ and $k = 1,2,\ldots,p$. We say that the influence of i_j on s_k is interesting if $OI(i_j, s_k, D) \geqslant \gamma$, for $j = 1,2,\ldots,m$ and $k = 1,2,\ldots,p$. The value of γ depends on the level of data analysis to be performed. If the data analysis is performed in-depth then the value of γ is expected to be low. Also, the value of γ is dependent on the data to be analyzed. Normally, when the data is sparse the user needs to provide a low value of γ. On the other hand, γ could be given a reasonably high value for analyzing dense data. The procedure of determining influence of an item on a set of specific items could be explained using the following steps.

(i) Generate influence matrix (IM) of order $p \times m$ using $OI(i_j, s_k, D)$, for $j = 1,2,\ldots,m$ and $k = 1,2,\ldots,p$. (ii) An influence is counted when it is interesting. (iii) For each item, count the number of interesting influences on each of the specific items. (iv) The items in database D are sorted based on primary key as the number of interesting influences on the specific items, and secondary key as the support of an item. We explain steps (i)–(iv) using Example 4.

Example 4. Consider the database D_1 given in Example 1. Let $I = \{a, b, c, d, e, f, g\}$ and $SI = \{a, c, d\}$. The supports of items in D_1 are given in Table 3.

In this case, the influence matrix is of order 3×7 as given below.

$$IM = \begin{array}{c|ccccccc} item & a & b & c & d & e & f & g \\ a & 1 & 0.333 & 0.100 & 0.333 & -0.167 & -0.333 & 0.350 \\ c & 0.100 & 0.333 & 1 & 0.333 & -0.667 & 0.167 & 0.350 \\ d & 0.300 & -0.500 & 0.300 & 1 & 0 & 0 & -0.150 \end{array}$$

Let γ be 0.2. Also let $x(\eta)$ denote η number of interesting influences of item x on different specific items. The numbers of interesting

Table 3
Supports of each items in D_1.

Items (x)	a	b	c	d	e	f	g
$supp(\{x\}, D_1)$	5/9	6/9	5/9	6/9	6/9	3/9	5/9

Table 4
A 2×2 contingency table for variables x and y.

	Y	$\neg Y$	Total
X	f_{11}	f_{10}	f_{1+}
$\neg X$	f_{01}	f_{00}	f_{0+}
Total	f_{+1}	f_{+0}	N

influences of different items in D_1 are given as follows. $a(2)$, $b(2)$, $c(2)$, $d(3)$, $e(0)$, $f(0)$, $g(2)$. The items being sorted using step (iv) are given as follows: $d(3)$, $b(2)$, $a(2)$, $c(2)$, $g(2)$, $e(0)$, $f(0)$. Given the set of specific items $\{a, c, d\}$, one could conclude that the item d has the maximum and the item f has a minimum influence on the specific items.

3.4. Motivation

The concept of influence might not be new in the literature of data mining. For example, $conf(X \rightarrow Y, D)$ refers to positive influence of X on Y. In other words, it implies how likely a customer purchases the items of Y when he has already purchased all the items of X. In addition, the concept of negative influence existed in the literature on data mining. $conf(\neg X \rightarrow Y, D)$ refers to the amount of negative influence of items of X in purchasing the items of Y. In other words, it implies how likely a customer purchases the items of Y when the customer has not purchased all the items of X. In many data analyses it might be required to consider the overall influence of a set of items on another set of items. Our work introduces the notion of overall influence that could be useful in dealing with many real life problems. In the following paragraph, we justify that an existing measure might not be appropriate to study the overall influence of an itemset on another itemset.

The analysis of relationships among variables is a fundamental task being at heart of many data mining problems. For example, metrics such as support, confidence, lift, correlation, and collective strength have been used extensively to evaluate the interestingness of association patterns. These metrics are defined in terms of the frequency counts tabulated in a 2×2 contingency table as shown in Table 4. To illustrate this, let us consider 10 example contingency tables, E_1 to E_{10}, given in Table 5. Tan et al. (2003) presented an overview of 21 interestingness measures proposed in the statistics, machine learning, and data mining literature.

In the following discussion, we shall observe why these measures fail to compute overall influence of an itemset on another itemset. In Examples 2 and 3, we have observed that the overall influence of an itemset on another itemset could be positive as well as negative. Thus, overall influence of an itemset on another itemset in a database lies in $[-1, 1]$. In a large database, where items are sparsely distributed over the transactions might result in negative overall influence of an itemset on another itemset. Based on these observations, one could consider the following five out of 21 interestingness measures since overall influence of an itemset on another itemset lies in $[-1, 1]$. These measures are presented in Table 6.

Table 6
Relevant interestingness measures for association patterns.

Symbol	Measure	Formula
ϕ	ϕ-coefficient	$\frac{P(\{x\}\cup\{y\}) - P(\{x\}) \times P(\{y\})}{\sqrt{P(\{x\}) \times P(\{y\}) \times (1-P(\{x\})) \times (1-P(\{y\}))}}$
Q	Yule's Q	$\frac{P(\{x\}\cup\{y\}) \times P(\neg(\{x\}\cap\{y\})) - P(\{x\}\cup\{y\}) \times P(\neg\{x\}\cup\{y\})}{P(\{x\}\cup\{y\}) \times P(\neg(\{x\}\cap\{y\})) + P(\{x\}\cup\{y\}) \times P(\neg\{x\}\cup\{y\})}$
Y	Yule's Y	$\frac{\sqrt{P(\{x\}\cup\{y\}) \times P(\neg(\{x\}\cap\{y\}))} - \sqrt{P(\{x\}\cup\{y\}) \times P(\neg\{x\}\cup\{y\})}}{\sqrt{P(\{x\}\cup\{y\}) \times P(\neg(\{x\}\cap\{y\}))} + \sqrt{P(\{x\}\cup\{y\}) \times P(\neg\{x\}\cup\{y\})}}$
κ	Cohen's	$\frac{P(\{x\}\cup\{y\}) + P(\neg\{x\}\cup\{y\}) - P(\{x\}) \times P(\{y\}) - P(\neg\{x\}) \times P(\neg\{y\})}{1 - P(\{x\}) \times P(\{y\}) - P(\neg\{x\}) \times P(\neg\{y\})}$
F	Certainty factor	$\max\left(\frac{P(\{y\}\|\{x\}) - P(\{y\})}{1 - P(\{y\})}, \frac{P(\{x\}\|\{y\}) - P(\{x\})}{1 - P(\{x\})}\right)$

Based on each formula present in Table 6, the above contingency tables have been ranked as shown in Table 7. A contingency table that gives the maximum value is ranked as number 1 based on the interestingness measure. For example, contingency tables $E1$ and $E2$ give maximum and the second maximum values based on ϕ.

Also, we rank the contingency tables based on the concept of overall influence explained in Example 1. In Table 8, we present the ranking of contingency tables when using the overall influence given by (3).

None of the five measures ranks contingency tables like the ranks given in Table 7. Thus, none of the above five measures serves as a measure of overall influence between two itemsets.

4. Problem statement

Let D be a database of customer transactions grown over a period of time. In this paper, we are interested in making an influence

Table 7
Ranking of contingency tables using above interestingness measures.

Example	ϕ	Q	Y	κ	F
$E1$	1	3	3	1	4
$E2$	2	1	1	2	1
$E3$	3	4	4	3	6
$E4$	4	2	2	5	2
$E5$	5	8	8	4	9
$E6$	6	7	7	7	7
$E7$	7	9	9	6	8
$E8$	8	10	10	8	10
$E9$	9	5	5	9	3
$E10$	10	6	6	10	5

Table 5
Examples of contingency tables.

Example	f_{11}	f_{10}	f_{01}	f_{00}
$E1$	8123	83	424	1370
$E2$	8330	2	622	1046
$E3$	9481	94	127	298
$E4$	3954	3080	5	2961
$E5$	2886	1363	1320	4431
$E6$	1500	2000	500	6000
$E7$	4000	2000	1000	3000
$E8$	4000	2000	2000	2000
$E9$	1720	7121	5	1154
$E10$	61	2483	4	7452

Table 8
Ranking of contingency tables using overall influence.

Example	Overall influence	Rank
$E1$	0.754	1
$E2$	0.627	3
$E3$	0.691	2
$E4$	0.560	4
$E5$	0.450	5
$E6$	0.352	7
$E7$	0.417	6
$E8$	0.167	9
$E9$	0.190	8
$E10$	0.023	10

analysis of a set of specific items. We will see how each of the specific items gets influenced by different items in the database. One could view the entire database as a sequence of time-based (temporal) databases. For instance, such databases may concern consecutive years. To provide an incremental solution to this problem, one might need to mine only the current time database and combine the mining result with the previous mining results. Thus, one needs to mine only the current database for the purpose of making an analysis based on entire database. As a result one can obtain cost-effective and faster analysis based on the entire database. Since the database grows over time, an incremental solution to influence analysis of specific items becomes natural and desirable.

Each time database corresponds to the set of transactions made for a specific period of time. In this regard, the choice of time period corresponding to a database is an important issue. One could observe that the sales of items might vary over different seasons in a year. Instead of processing all the data together, we process data on a yearly basis. Then, the result of processing for the current year could be combined with that of previous years. Such incremental analysis might be appropriate since a season re-appears on a yearly basis. Otherwise, the processed result might be biased due to seasonal variations.

Our goal is to make an influence analysis of a set of items in a database. Let D_t be the database for the tth period of time, $t = 1, 2, \ldots, n$. For computing overall influence between two items in a database, one needs to mine supports of itemsets of sizes one and two. The *size* of an itemset refers to the number of items in the itemset. Let $D_{1,k}$ be the collection of databases D_1, $D_2, \ldots, D_k$. For computing $OI(x, y, D_{1,k+1})$, we assume that $OI(x, y, D_{1,k})$ is available to us for items x, y in $D_{1,k}$. In other words, for computing $OI(x, y, D_{1,k+1})$, we have $supp(x, D_{1,k})$, $supp(y, D_{1,k})$, and $supp(x \cap y, D_{1,k})$. Thus, our incremental procedure needs to compute $supp(x, D_{1,k+1})$, $supp(y, D_{1,k+1})$, and $supp(x \cap y, D_{1,k+1})$ using (i) $supp(x, D_{1,k})$, $supp(y, D_{1,k})$, and $supp(x \cap y, D_{1,k})$, (ii) $supp(x, D_{k+1})$, $supp(y, D_{k+1})$, and $supp(x \cap y, D_{k+1})$. In general, for an itemset X in $D_{1,k}$, $supp(X, D_{1,k+1})$ could be obtained incrementally as follows:

$$supp(X, D_{1,k+1}) = \frac{size(D_{k+1}) \times supp(X, D_{k+1}) + size(D_{1,k}) \times supp(X, D_{1,k})}{size(D_{k+1}) + size(D_{1,k})}. \quad (4)$$

The $size(D)$ refers to the number of transactions in database D.

5. Related work

For analyzing positive association between itemsets in a database, support–confidence framework was established by Agrawal et al. (1993). In Section 3.4, we have discussed why a confidence measure alone is not sufficient in determining an overall influence of an itemset on another itemset. Also, interestingness measures such as support, collective strength (Aggarwal and Yu, 1998) and Jaccard (Tan et al., 2003) are not relevant in this context, since they are single-argument measures.

The χ^2 test (Greenwood and Nikulin, 1996) only tells us whether two or more items are dependent. Such a test provides answers either "yes" or "no" to the question of whether the association is meaningful, and hence it might not be suitable for the specific requirement of our problem.

The interestingness measures such as lift (Tan et al., 2003), correlation (Tan et al., 2003), conviction (Brin et al., 1997), and odds-ratio (Tan et al., 2003) are semantically different from the measure of overall influence. Moreover, each of these measures lies in $[0, \infty)$.

Shapiro (1991) has proposed leverage measure in the context of mining strong rules in a database. It might not be suitable for the specific requirement of our problem.

6. Design of algorithms

Based on the discussion held in previous section, we design two algorithms for measuring influence of an item on another item and influence of an item on a set of specific items.

6.1. Designing algorithm for measuring overall influence of an item on another item

In this algorithm, we measure influence of an item on each of the items incrementally. We have expressed influence of an itemset on another itemset using supports of the relevant itemsets. Each itemset could be described by its *itemset* and *support*. We maintain arrays *IS1* and *IS2* for storing itemsets in D_1,k of size one and two, respectively. *Itemset* attribute of ith itemset in *IS1* could be accessed using the notation $IS1(i).itemset$. Similar notation is used to access *support* attribute of an itemset. Also, we maintain arrays $\Delta IS1$ and $\Delta IS2$ for storing itemsets in D_{k+1} of size one and two, respectively. We merge *IS1* and $\Delta IS1$ to obtain supports of itemsets of size one in $D_{1,k+1}$ and are stored in array *OIS1*. Similarly, we merge *IS2* and $\Delta IS2$ to obtain supports of itemsets of size two in $D_{1,k+1}$ and are stored in array *OIS2*. Using *OIS1* and *OIS2*, we compute an overall influence between items in $D_{1,k+1}$. The Overall influence between items is computed using formula (3) and stored in array *IOI*. The overall influence (oi) corresponding to jth pair of items is accessed by $IOI(j).oi$.

Algorithm 1. Find top q overall influences in the given database over time.

procedure *Top-q-OI*(q, *IS1*, *IS2*, Δ *IS1*, Δ*IS2*, *IOI*)
Inputs:
q: an integer representing the number of top influences
IS1: array of supports of itemsets of size one in $D_{1,k}$
IS2: array of supports of itemsets of size two in $D_{1,\ k}$
$\Delta IS1$: array of supports of itemsets of size one in D_{k+1}
$\Delta IS2$: array of supports of itemsets of size two in D_{k+1}
Outputs:
IOI: array of overall influences in $D_{1,k+1}$
01: sort array $\Delta IS1$ on *itemset* attribute in non-decreasing order;
02: sort array $\Delta IS2$ on *itemset* attribute in non-decreasing order;
03: call *Merge* (*IS1*, Δ*IS1*, *OIS1*);
04: call *Merge* (*IS2*, Δ*IS2*, *OIS2*);
05: **let** $j = 1$;
06: **for** $i = 1$ to $|OIS2|$ **do**
07: search $OIS2(i).item1$ in *OIS1*;
08: search $OIS2(i).item2$ in *OIS1*;
09: $IOI(j).oi = OI(OIS2(i).item1, OIS2(i).item2, D)$;
10: $IOI(j).item1 = OIS2(i).item1$; $IOI(j).item2 = OIS2(i).item2$;
11: increase j by 1;
12: $IOI(j).oi = OI(OIS2(i).item2, OIS2(i).item1, D)$;
13: $IOI(j).item1 = OIS2(i).item2$; $IOI\ (j).item2 = OIS2(i).item1$;
14: increase j by 1;
15: **end for**
16: sort array *IOI* in non-increasing order on oi attribute;
17: return first q influences;
18: **end procedure**;

Table 9
Database characteristics.

Database	*NT*	*ALT*	*AFI*	*NI*
mushroom (*M*)	8124	24.000	1624.800	120
ecoli (*E*)	336	7.000	25.835	91
random-68 (*R*)	3000	5.460	280.985	68
retail (*Rt*)	88,162	11.306	99.674	10,000

Table 10
Time database characteristics.

D	NT	ALT	AFI	NI	D	NT	ALT	AFI	NI
M_0	812	24.000	295.273	66	M_5	812	24.000	221.454	88
M_1	812	24.000	286.588	68	M_6	812	24.000	216.533	90
M_2	812	24.000	249.846	78	M_7	812	24.000	191.059	102
M_3	812	24.000	282.435	69	M_8	812	24.000	229.271	85
M_4	812	24.000	259.840	75	M_9	816	24.000	227.721	86
E_0	33	7.000	4.620	50	E_5	33	7.000	3.915	59
E_1	33	7.000	5.133	45	E_6	33	7.000	3.500	66
E_2	33	7.000	5.500	42	E_7	33	7.000	3.915	59
E_3	33	7.000	4.813	48	E_8	33	7.000	3.397	68
E_4	33	7.000	3.397	68	E_9	39	7.000	4.550	60
R_0	300	5.590	28.676	68	R_5	300	5.140	26.676	68
R_1	300	5.417	28.000	68	R_6	300	5.510	28.353	68
R_2	300	5.360	27.647	68	R_7	300	5.497	28.338	68
R_3	300	5.543	28.456	68	R_8	300	5.537	28.471	68
R_4	300	5.533	28.382	68	R_9	300	5.477	28.235	68
Rt_0	9000	11.244	12.070	8384	Rt_5	9000	10.856	16.710	5847
Rt_1	9000	11.209	12.265	8225	Rt_6	9000	11.200	17.416	5788
Rt_2	9000	11.337	14.597	6990	Rt_7	9000	11.155	17.346	5788
Rt_3	9000	11.490	16.663	6206	Rt_8	9000	11.997	18.690	5777
Rt_4	9000	10.957	16.039	6148	Rt_9	7162	11.692	15.348	5456

Table 11
Top 10 overall influences in different databases.

M ($supp$ = 0.15)			E ($supp$ = 0.12)			R ($supp$ = 0.03)			Rt ($supp$ = 0.12)		
$\{x\}$	$\{y\}$	OI	$\{x\}$	$\{y\}$	OI	$\{x\}$	$\{y\}$	OI	$\{x\}$	$\{y\}$	OI
86	34	0.997	24	48	0.946	19	29	−0.017	41	39	0.200
34	86	0.992	89	50	0.913	29	19	−0.020	39	48	0.180
58	24	0.991	53	48	0.693	8	56	−0.023	48	39	0.175
67	76	0.986	63	50	0.665	56	8	−0.023	41	48	0.129
76	67	0.986	87	50	0.660	15	14	−0.031	39	41	0.114
24	58	0.963	56	50	0.621	14	15	−0.032	48	41	0.071
93	59	0.895	61	50	0.618	18	52	−0.035	48	7	−0.234
93	76	0.884	27	48	0.618	52	18	−0.036	39	7	−0.292
93	67	0.881	83	50	0.540	54	58	−0.044	48	2	−0.293
102	24	0.875	56	48	0.488	58	54	−0.047	48	1	−0.316

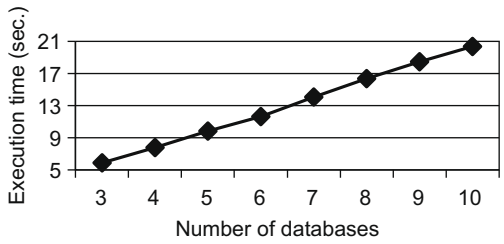

Fig. 1. Execution time versus number of databases at $supp$ = 0.2 (*mushroom*).

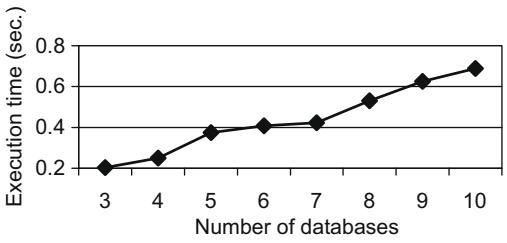

Fig. 2. Execution time versus number of databases at $supp$ = 0.12 (*ecoli*).

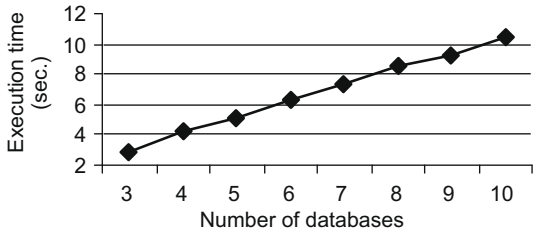

Fig. 3. Execution time versus number of databases at $supp$ = 0.03 (*random-68*).

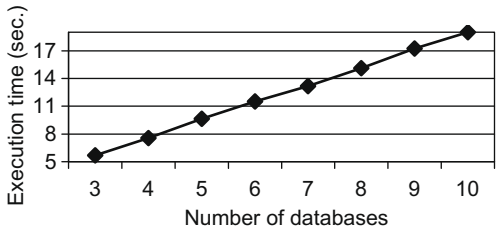

Fig. 4. Execution time versus number of databases at $supp$ = 0.2 (*retail*).

The procedure $Merge(A, B, C)$ merges sorted arrays A and B and generates output array C. In this context, sorting is based on support of an itemset. The time complexity of procedure *Merge* is $O(|A| + |B|)$ (Knuth, 1998). Now, *OIS1* contains the supports of items in $D_{1,k+1}$. Also, *OIS2* contains the supports of itemsets of size two in $D_{1,k+1}$. The information contained in *OIS1* and *OIS2* is used to compute overall influence of an item on another item in $D_{1,k+1}$. Using line 09 we have computed influence of a singleton itemset on

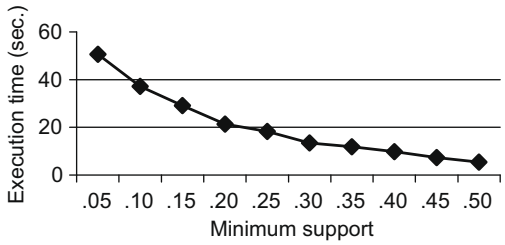

Fig. 5. Execution time versus minimum support (*mushroom*).

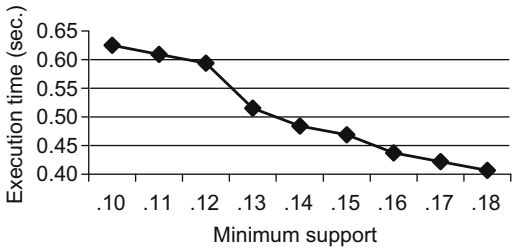

Fig. 6. Execution time versus minimum support (*ecoli*).

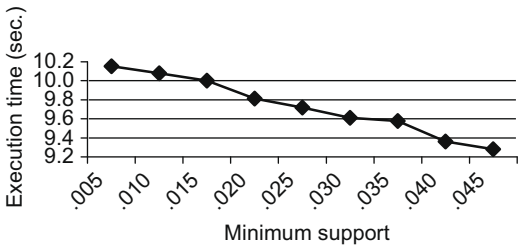

Fig. 7. Execution time versus minimum support (*random-68*).

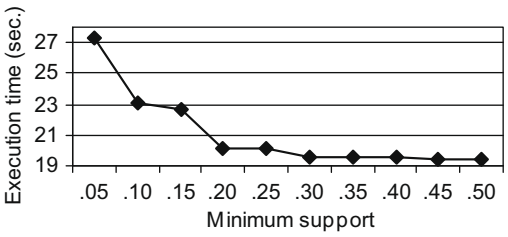

Fig. 8. Execution time versus minimum support (*retail*).

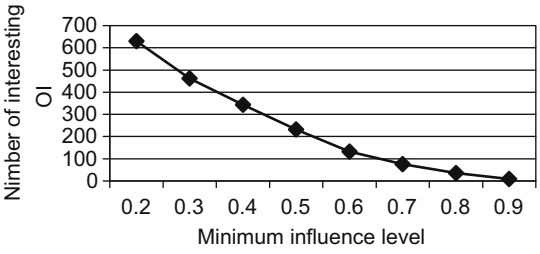

Fig. 9. Number of interesting *OI* values versus γ at *supp* = 0.2 (*mushroom*).

another singleton itemset. Suppose {6, 8} be a frequent 2-itemset in $D_{1,k+1}$ stored in fourth cell of *OIS2*. Then $OI(OIS2(4).item1, OIS2(4).item2, D_{1,k+1})$ refers to overall influence of {6} on {8} in $D_{1,k+1}$. In lines 6–15, we have computed and stored overall influences of a singleton itemset on another singleton itemset in $D_{1,k+1}$. In line 16, we have sorted overall influences in non-increasing order. Finally, we display first q overall influences.

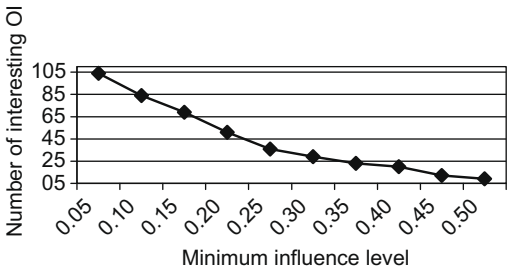

Fig. 10. Number of interesting *OI* values versus γ at *supp* = 0.12 (*ecoli*).

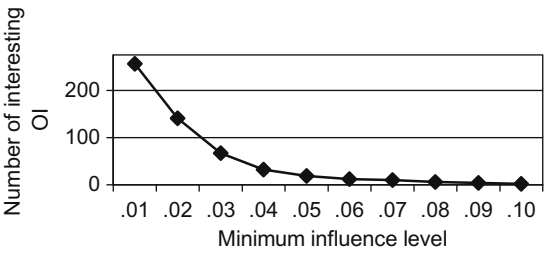

Fig. 11. Number of interesting *OI* values versus γ at *supp* = 0.015 (*random-68*).

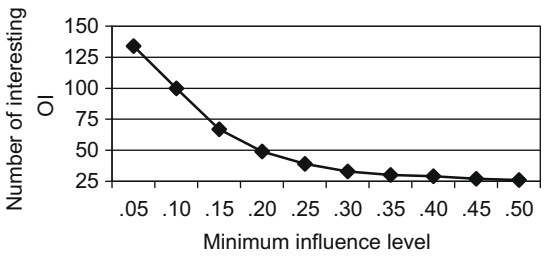

Fig. 12. Number of interesting *OI* values versus γ at*supp* = 0.02 (*retail*).

Let *IS1* and *IS2* contain M and N itemsets, respectively. Let $\Delta IS1$ and $\Delta IS2$ contain m and n elements, respectively. Lines 1 and 2 take $O(m \times log(m))$ and $O(n \times log(n))$ time, respectively. Also, lines 3 and 4 take $O(M + m)$ and $O(N + n)$ time, respectively. Each of the search statements in lines 7 and 8 take $O(log(M + m))$ time, since *OIS1* is sorted. The sort statement in line 16 takes time $O((N + n) \times log\ (N + n))$. The time complexity of lines 6–15 is $O((N + n) \times log(M + m))$. Thus, the time complexity of algorithm *Top-q-OI* is *maximum* $\{O(M + m), O((N + n) \times log(N + n)), O((N + n) \times log(M + m))\}$.

6.2. Designing algorithm for measuring overall influence of an item on each of the specific items

One could store specific items in an array. The proposed algorithm seems to be the same as Algorithm 1 except that every time it measures an overall influence of an item on a specific item.

6.3. Designing algorithm for identifying top influential items on a set of specific items

In Algorithm 2, we find influence of an item on a set of specific items in a database. We construct influence matrix (*IM*) from the arrays of specific items (*SI*) and overall influence between items

(*IOI*). The algorithm scans *IM* for each item to count the number of interesting influences which are stored in array called *count*. Finally, we sort *count* on descending order on primary key count value and secondary key support.

Algorithm 2. Find influence of an item on a set of specific items in the database over time.

procedure *Top-q-items*(q, *SI*, *IS1*, *IS2*, Δ*IS1*, Δ*IS2*, *OIS1*, *OIS2*, *IOI*)
Inputs:
q: an integer representing the number of top influences
SI: array of specific items
IS1, *IS2*, Δ*IS1*, Δ*IS2*, *OIS1*, *OIS2*, *IOI*: as specified in Algorithm 1
Outputs:
count: array of number of interesting influences
01: **for** $i = 1$ to $|SI|$ **do**
02: **for** $j = 1$ to $|IOI|$ **do**
03: **if**($SI(i) = IOI(j).item1$) **then**
04: $IM(i)(j) = IOI(j).oi$;
05: **end if**
06: **end for**
07: **end for**
08: **for** $j = 1$ to $|IOI|$ **do**
09: **let** $count(j) = 0$;
10: **for** $i = 1$ to $|SI|$ **do**
11: **if** $(IM(j)(i) \geqslant \gamma)$ **then**
12: increase $count(j)$ by 1;
13: **end if**
14: **end for**
15: **end for**
16: sort *count* on non-increasing order on primary key count value and secondary key support;
17: return first q items;
end procedure

Let array *SI* contains p items. Line 1 repeats for p times. Line 2 repeats $O(M + m)$ times. So, lines 1-7 take $O(p \times (M + m))$ time. Line 8 repeats $O(M + m)$ times. Line 10 repeats p times. Thus, line 8-15 take $O(p \times (M + m))$ time. Therefore, the time complexity of the above algorithm is $O(p \times (M + m))$, where $M > m$. Also, sorting statement at line 16 takes $O((M + m) \times log\,(M + m))$. The time complexity of algorithm *Top-q-items* is $maximum\{O(p \times (M + m)), O((M + m) \times log(M + m))\}$.

7. Experiments

We have carried out several experiments to study the effectiveness of the proposed analysis. All the experiments have been implemented on a 1.6 GHz Pentium IV with 256 MB of memory using visual C++ (version 6.0) software. We present the experimental results using three real-world databases and one synthetic database. The databases *mushroom*, *retail* (Frequent itemset mining dataset repository) and *ecoli* are real-world databases. Database *ecoli* is a subset of *ecoli database* (UCI ML repository) and it has been processed for the purpose of conducting experiments. *random-68* is a Synthetic database. The symbols used in different tables are explained as follows. Let *D*, *NT*, *ALT*, *AFI*, and *NI* denote database, the number of transactions, average length of a transaction, average frequency of an item, and number of items, respectively. The details of these databases are given in Table 9.

Each database has been divided into 10 databases, called input databases, for the purpose of conducting experiments on multiple time databases. The input databases obtained from *mushroom*, *ecoli*, *random-68* and *retail* are named as M_i, E_i, R_i, and Rt_i, $i = 0, 1, \ldots, 9$. We present some characteristics of the input databases in Table 10. Top 10 overall influences in different databases are presented in Table 11.

We have studied the execution time with respect to the number of data sources. We observe in Figs. 1–4 that this time increases as the number of data sources gets higher.

The size of each input database generated from *mushroom* and *retail* are significantly larger than an input database generated from *ecoli*. As a result, we observe a steeper relationship in Figs. 1 and 4. The number of frequent itemsets decreases as the minimum support increases.

In Figs. 5–8 it is shown how the execution time decreases over the increase of the minimum support value.

By comparing Figs. 1–4, one notes that the steepness of a graph increases as the size of branch databases increase. Similar observation holds true for Figs. 5–8.

In Section 3.1 we have explained the concept of interesting overall influence. Given a threshold value of γ, we have counted the number of overall influences. In Figs. 9–12 we have shown how the number of interesting overall influence decreases over the increase of the minimum influence level.

Figs. 9–12 also provide another type of insight. As the size of a transaction increases, the number of interesting overall influences also increases, provided the number of transactions in a branch database and the level of overall influence remain constant. The average transaction length of mushroom branch databases is significantly higher than that of other branch databases. The mining algorithm generates a large number of interesting overall influences even at minimum influence level 0.2.

We have taken specific items in different databases in Table 12. Based on the requirement of association analysis one could choose specific items in time databases.

The influences of different items on a set of specific items in different databases are presented in Table 13. In the *mushroom*

Table 12
Specific items in different databases

M	*E*	*R*	*Rt*
SI = {1,2,3,6,9,10,11,13,16,23}	*SI* = {37,39,40,41,42,44,48,,49,50,51}	*SI* = {1,2,3,4,5,6,7,8,9,10}	*SI* = {0,1,2,3,4,5,6,7,8,9}

Table 13
Influences of different items on a set of specific items in different databases.

M (*supp* = 0.2)		*E* (*supp* = 0.12)		*R* (*supp* = 0.015)		*Rt* (*supp* = 0.03)	
γ	$x(\eta)$	γ	$x(\eta)$	γ	$x(\eta)$	γ	$x(\eta)$
0.3	86(3),34(3),36(3),39(2), 59(2),63(2),2(2),93 (2), 36(2),23(2),90(1),24(1)	0.07	48(5),37(2),50(1), 42(1),44(1),39(1), 40(1),49(1),41(1)	0.05	18(5),15(3),65(2),55(2), 61(2),7(1),54(1), 27(1),35(1),66(1),22(1)	0.05	413(8),310(2),0(1), 1(1),8(1),2(1),3(1), 5(1),9(1),4(1)

database, item 86 is the most influential item because 3 specific items are influenced by it. Item 48 in *ecoli* database exhibits a significant influence on the set of specific items. It shows that item 48 has high influence on 5 out of 10 specific items. In the same way one could conclude that item 18 in *random-68* is the most influential item with respect to the given set of specific items. 5 out of 10 specific items are influenced significantly by item 18. Item 413 influences 8 out of 10 specific items significantly in *retail* database. Therefore, it is the most influential item in *retail*.

8. Conclusions

The concept of positive influence might not be sufficient in many data analyses. One could perform an effective data analysis by using the measure of overall influence. Measuring influence over time becomes an important issue, since many companies possess data for a long period of time so that they could be exploited in an efficient manner. In this paper, we have designed two algorithms using the measure of overall influence. The first algorithm reports all the significant influences in a database. In the second algorithm we have sorted items based on their influences on a set of specific items. Such analyses might be interesting since the proposed measure of influence considers both positive and negative influence of an itemset on another itemset.

Acknowledgements

Authors would like to thank the anonymous reviewers for their constructive comments that helped to improve the quality of the paper significantly.

References

Adhikari, A., Rao, P.R., 2007a. Study of select items in multiple databases by grouping. In: Proceedings Internat. Conf. on Artificial Intelligence, pp. 1699–1718.

Adhikari, A., Rao, P.R., 2007b. A framework for mining arbitrary Boolean expressions induced by frequent itemsets. In: Proceedings Internat. Conf. on Artificial Intelligence, pp. 5–23.

Adhikari, A., Rao, P.R., 2008a. Efficient clustering of databases induced by local patterns. Decision Support Systems 44 (4), 925–943.

Adhikari, A., Rao, P.R., 2008b. Mining conditional patterns in a database. Pattern Recognition Lett. 29 (10), 1515–1523.

Adhikari, A., Rao, P.R., 2008c. Synthesizing heavy association rules in different real data sources. Pattern Recognition Lett. 29 (1), 59–71.

Adhikari, J., Rao, P.R., Adhikari, A., 2009. Clustering items in different data sources induced by stability. Internat. Arab J. Inform. Technol. 6 (4), 394–402.

Aggarwal, C., Yu, P., 1998. A new framework for itemset generation. In: Proc. PODS, pp. 18–24.

Agrawal, R., Imielinski, T., Swami, A., 1993. Mining association rules between sets of items in large databases. In: Proc. ACM SIGMOD Conf. on Management of Data, pp. 207–216.

Agrawal, R., Srikant, R., 1994. Fast algorithms for mining association rules. In: Proc. Internat. Conf. on Very Large Databases, pp. 487–499.

Brin, S., Motwani, R., Ullman, J.D., Tsur, S., 1997. Dynamic itemset counting and implication rules for market basket data. In: Proc. ACM SIGMOD Internat. Conf. on Management of Data, pp. 255–264.

Frequent itemset mining dataset repository, xxxx. <http://fimi.cs.helsinki.fi/data>.

Gary, J.R., Petersen, A., 2000. Analysis of cross category dependence in market basket selection. J. Retail. 76 (3), 367–392.

Greenwood, P.E., Nikulin, M.S., 1996. A Guide to Chi-Squared Testing, first ed. Wiley-Interscience.

Han, J., Pei, J., Yiwen, Y., 2000. Mining frequent patterns without candidate generation, In: Proc. ACM-SIGMOD Internat. Conf. on Management of Data, pp. 1–12.

Knuth, D.E., 1998. The Art of Computer Programming Sorting and Searching, vol. 3, second ed. Addison-Wesley Professional.

Savasere, A., Omiecinski, E., Navathe, S., 1995. An efficient algorithm for mining association rules in large databases. In: Proc. Internat. Conf. on Very Large Data Bases, pp. 432–443.

Shapiro, P., 1991. Discovery, analysis, and presentation of strong rules. Knowledge Discovery in Databases, pp. 229–248.

Tan, P.N., Kumar, V., Srivastava, J., 2003. Selecting the right interestingness measure for association patterns. In: Proc. SIGKDD Conference, pp. 32–41.

UCI ML repository, xxxx. <http://www.ics.uci.edu/~mlearn/MLSummary.html>.

Wu, X., Zhang, C., Zhang, S., 2004. Efficient mining of both positive and negative association rules. ACM Trans. Inform. Systems 22 (3), 381–405.

Wu, X., Zhang, C., Zhang, S., 2005. Database classification for multi-database mining. Inform. Systems 30 (1), 71–88.