ELSEVIER

Data & Knowledge Engineering 37 (2001) 85–115

DATA & KNOWLEDGE ENGINEERING

www.elsevier.com/locate/datak

Inter-transactional association rules for multi-dimensional contexts for prediction and their application to studying meteorological data

Ling Feng [a,*], Tharam Dillon [b], James Liu [c]

[a] *InfoLab, Department of Information Management, Tilburg University B 302, P.O. Box 90153, 5000 LE, Tilburg, The Netherlands*
[b] *Department of Computer Science and Computer Engineering, LaTrobe University, LaTrobe, Australia*
[c] *Department of Computing, Hong Kong Polytechnic University, Hong Kong, People's Republic of China*

Received 3 November 1999; received in revised form 31 January 2000; accepted 20 December 2000

Abstract

Inter-transactional association rules, first presented in our early work [H. Lu, J. Han, L. Feng, Stock movement prediction and *n*-dimensional inter-transaction association rules, in: Proceedings of the ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, Seattle, Washington, June 1998, pp. 12:1–12:7; H. Lu, L. Feng, J. Han, ACM Trans. Inf. Syst. 18 (4) (2000) 423–454], give a more general view of association relationships among items. Two kinds of algorithms, named Extended/Extended Hash-based Apriori (E/EH-Apriori) [Lu et al. (1998, 2000), loc. cit.] and First-Intra-Then-Inter (FITI) [K. H. Tung, H. Lu, J. Han, L. Feng, Breaking the barrier of transactions: Mining inter-transaction association rules, in: Proceedings ACM SIGKDD International Conference Knowledge Discovery and Data Mining, USA, August 1999, pp. 297–301], were presented for mining inter-transactional association rules from large data sets. A template-guided constraint-based inter-transactional association mining method was described in [L. Feng, H. Lu, J. Yu, J. Han, Mining inter-transaction association rules with templates, in: Proceedings ACM CIKM International Conference Information and Knowledge Management, USA, November 1999, pp. 225–233].

The current paper extends our previous work substantially in both theoretical and practical aspects. In the theoretical aspects, we improve the inter-transactional association rule framework by giving a more concise definition of inter-transactional association rules and related measurements, and investigate the closure property, theoretical foundations, multi-dimensional mining contexts, and performance issues in mining such extended association rules. We study the downward closure property problem within the inter-transactional association rule framework, and propose a solution for efficient mining of inter-transactional association rules. A set of examples, lemmas and theorems are provided to verify our discussions. We also present a hole-catered extended Apriori algorithm for mining inter-transactional association rules. Different from our previous work, here, we take data holes that possibly exist in the mining contexts into consideration. We also address some important technical issues, including *correctness*, *termination* and *computational complexity*, in this paper. In practice, we study the applicability of inter-transactional association rules to weather prediction, using multi-station meteorological data obtained from the Hong Kong Observatory headquarters. We report our experimental results as well as the experiences gained during the weather study. In

[*] Corresponding author.
E-mail addresses: ling@kub.nl (L. Feng), tharam@cs.latrobe.edu.au (T. Dillon), csnkliu@comp.polyu.edu.hk (J. Liu).

PII: S0169-023X(01)00003-9

particular, the deficiency of the current support/confidence-based association mining framework and its further extension in providing multi-dimensional predictive capabilities are addressed.

These extensions significantly augment the theory and practicality of the more general inter-transactional association rules. It is our hope that the work reported here could stimulate further interest not only in the applications of association rule techniques to non-transactional real-world data under multi-dimensional contexts, but also in the relevant theoretical and performance issues of association rule techniques.

Keywords: Intra-transactional/inter-transactional association rules; Multi-dimensional mining context; Downward closure property; Data holes; Weather prediction

1. Introduction

Meteorologists and weather forecasters base weather prediction mainly on numerical and statistical models apart from their subjective assessment of the dynamic atmospheric system. The classical approaches attempt to model the fluid and thermal dynamic systems for grid-point time series prediction based on boundary meteorological data. The simulation conducted often requires intensive computations, involving complex differential equations and computational algorithms. Moreover, the accuracy is bound by certain *inherited* constraints, such as the adoption of incomplete boundary conditions, model assumptions, and numerical instabilities, etc. [22].

There have been some studies which seek to make use of a few machine learning methods to derive forecasting rules from observational meteorological data provided by Bureau of Meteorology, Atmospheric Division of CSIRO in Australia, and the Wuhan Central Weather Services of China to address binary classification problems [8]. A constraint-based discovery algorithm was employed to detect forecasting rules from the domain data. Lately, Li et al. [20] extended Naive Bayesian classifiers to learn weather forecasting rules and for the prediction of multi-classification problems. For complex problems such as severe weather prediction, Lee et al. [19] presented a hybrid system to automate the satellite interpretation process and provided an objective analysis for tropical cyclones. However, all these methods fall short of deriving any explicit and direct association relationships by data mining for weather prediction. The extraction of association patterns from domain data was not the focus of these studies rather than the overall forecast of a certain specific meteorological phenomenon of interest.

Observing that many meteorological elements are correlated in nature, in this paper, we study meteorological data using association rule techniques, developed by the database and data mining communities, for weather prediction. Another different aspect from the previous machine learning and knowledge acquisition based approaches is that the latter only based their forecasts on data collected in a single station [19,20]. However, meteorological data are frequently gathered in a number of neighboring regional stations, and this has a bearing on the actual weather. In order to ensure that all the relevant data are utilized by the data mining techniques, it is important to make use of multi-station data. In this paper, therefore, we propose a method capable of doing association mining on multi-station atmospheric data.

1.1. Association rules for prediction

The problem of mining association rules was first introduced by Agrawal et al. [1]. The most often cited application of association rules is market basket analysis using transaction databases from supermarkets. The databases contain sales transaction records, each of which details the items bought by a customer in the transaction. Mining association rules is the process of discovering knowledge such as

R_1: 80% of customers who bought diapers also bought beer.

This can be expressed as "*diaper* ⇒ *beer* (25%, 80%)", where 80% is the *confidence level* of the rule and 25% is the *support level* of the rule, indicating how frequently the customers bought both diapers and beer. In general, an association rule takes the form of $X \Rightarrow Y(s,c)$, where X and Y are sets of items from the relevant databases, stating the presence of items in X in a transaction record will imply the presence of items in Y within the same transaction record, and s and c are *support* and *confidence*, respectively. The discovered knowledge can then be used in store floor planning, sales promotions, etc.

Mining association rules from large data sets has received considerable attention in recent years [1,2]. A large amount of work has been done in various directions, including efficient, Apriori-like mining methods [2,11,16,28,29,34,37,39], mining generalized, multi-level, or quantitative association rules [10,12,15,21,25,31,32], association rule mining query languages [24,38], constraint-based rule mining [5,13,26,35,38], incremental maintenance of discovered association rules [6], parallel and distributed mining [3,7,14], mining correlations and casual structures [4,33], cyclic and interesting association rule mining [27,30], etc.

Although association rules have demonstrated a strong potential value such as the improvement of market strategies in the retail industry, their emphasis is on *description* rather than *prediction*. As Mannila [23] says, "The association rules discovered from data have a simple meaning, but using them for prediction is not easy/interesting". For example, when we apply the above association rule concept to studying meteorological data, with each record listing various atmospheric observations including *wind direction*, *wind speed*, *temperature*, *relative humidity*, *rainfall*, and *mean sea level pressure* taken at a certain time in a certain area, we can find association rules like

R_2: If the humidity is medium wet, then there is no rain *in the same area at the same time*.

Although rule R_2 reflects some relationships among the meteorological elements, its role in weather prediction is inadequate, as users are often more concerned about the climate along a time dimension like

R_3: If the wind direction is east and the weather is warm, then it keeps warm *for the next 24 h*.

Unfortunately, the above knowledge cannot be discovered within the traditional association rule framework. In [17,18], we introduced a more general form of association rules, named *inter-transactional association rules*, which treats the traditional one as a special case. The traditional association rules are *intra-transactional* since they only capture associations among items within the same transactions, where the notion of the transaction could be the items bought by the same customer, the atmospheric events that happened at the same time, and so on. However, an inter-transactional association rule can represent not only the associations of items within transactions, but also the associations of items among different transactions along certain dimensions like the next 6 h, the next day, etc. Following the proposal of inter-transactional association rules, we presented two kinds of algorithms, named Extended/Extended Hash-based Apriori

(E/EH-Apriori) [17,18] and First-Intra-Then-Inter (FITI) [36], for mining inter-transactional association rules from large data sets.

1.2. Contributions of the paper

The current article makes substantial extensions to our previous work from both theoretical and practical aspects. In theoretical aspects, we improve the inter-transactional association rule framework by:

- Extending a series of rule-related concepts and terminologies, including *enhanced transactional database model*, *multi-dimensional mining contexts*, *extended items* (*transactions*), *normalized extended item* (*transaction*) *sets*, *database holes*, *possibly containing* and *containing relationships*. Based on these notions, a more concise definition of inter-transactional association rules and related measurements is given.
- Studying the property of the extended inter-transactional association rules. We examine the downward closure problem in mining such a more general association relationship, and provide a solution.
- Investigating the influence of data holes on mining multi-dimensional inter-transactional association rules.
- Presenting a hole-catered inter-transactional association mining method, together with its theoretical foundations and computational complexity.

In practice, we utilize meteorological data, obtained from multiple atmospheric stations in Hong Kong, to study the applicability of such an extended inter-transactional association rule technique to weather prediction. The data sets consist of meteorological records taken every 6 h each day from January 1993 to December 1997, covering six different geographical regions of Hong Kong. Unlike most data sets used in the previous association rule mining work, the data used in our experiments are non-transactional and possess two-dimensional properties (*time* and *space*) in nature. We report our experimental results as well as the experiences gained during the weather study. In particular, the deficiency of the current support/confidence-based association mining framework and its further extension in providing multi-dimensional predictive capabilities are addressed.

It is our hope that the work reported here could stimulate further interest not only in the practical applications of association rule techniques to non-transactional real-world data under multi-dimensional contexts, but also in the relevant theoretical and performance issues of association rule mining techniques.

1.3. Organization of the paper

The rest of the paper is organized as follows. In Section 2, we extend the scope of association rules from traditional intra-transactional associations to inter-transactional associations for prediction. In Section 3, we investigate the downward closure property problem with such general inter-transactional association rules. A hole-catered inter-transactional association mining method and its computational complexity are described in Section 4. The experimental results obtained from single-station and multi-station meteorological data sets are described in Section 5. In Section 6, we discuss how inter-transactional association rules can be extended to give multi-dimensional predictive capabilities. In addition, the inherent deficiency of association rules in weather prediction is also addressed. Section 7 concludes the paper with a brief discussion of future work.

2. Extending classical association rules for prediction

The fact that classical association rules only look at correlations of items within the same transaction limits predictive capabilities of association rules. When we extend the classical association concept from intra-transactional associations to inter-transactional associations, we can explore the correlative relations of items from different transactions along certain dimensions, giving an enlarged room for prediction. For instance, in a meteorological database whose records are organized by transaction time, inter-transactional association rules can represent associations of atmospheric phenomena along the *time* dimension.

In this section, we extend a series of concepts and terminologies, including enhanced transactional database model, multi-dimensional mining contexts, extended items (transactions), normalized extended item (transaction) sets, database holes, possibly containing and containing relationships, for inter-transactional association rule mining. Based on these notions, a formal definition of inter-transactional association rules and related measurements is given. Throughout the discussion, we assume that the following notation is used.

- A finite set of literals called items $\mathscr{I} = \{i_1, i_2, \ldots, i_n\}$.
- A finite set of transaction records $\mathscr{T} = \{t_1, t_2, \ldots, t_l\}$, where for $\forall t_i \in \mathscr{T}, t_i \subseteq \mathscr{I}$.
- A finite set of attributes called dimensional attributes $\mathscr{A} = \{a_1, a_2, \ldots, a_m\}$, whose domains are finite subsets of nonnegative integers.

2.1. Concepts and terminologies

2.1.1. An enhanced transactional database model

In classical association rule mining, records in a transactional database contain only items and are identified by their Transaction IDs (TIDs). Although transactions occur under certain *contexts* such as time, place, customers, etc., such contextual information has been ignored in classical association rule mining due to the fact that such rule mining was intra-transactional in nature. However, when we talk about inter-transactional associations across multiple transactions, the contexts of occurrence of transactions become important and must be taken into account.

Here, we enhance the traditional transactional database model by associating each transaction record with a number of attributes that describe the context within which the transaction happens. We call them *dimensional attributes*, because these attributes together constitute a multi-dimensional space and each transaction can be mapped to a certain point in this space. For a meteorological database where each transaction records observations of various meteorological elements taken at a certain time in a certain region, there are two dimensional attributes, namely, *time* and *region*. For a stock movement database, the dimensional attribute could be the *trading date*. Basically, dimensional attributes can be of any kind as long as they are meaningful to applications. Time, distance, temperature, latitude, etc., are typical dimensional attributes.

2.1.2. Multi-dimensional contexts

An m-dimensional mining context can be defined through m dimensional attributes $a_1, a_2, \ldots, a_m$, each of which represents a dimension. When $m = 1$, we have a single-dimensional mining context. Let $n_i = (n_i \cdot a_1, n_i \cdot a_2, \ldots, n_i \cdot a_m)$ and $n_j = (n_j \cdot a_1, n_j \cdot a_2, \ldots, n_j \cdot a_m)$ be two points in an m-dimensional space, whose values on the m dimensions are represented as $n_i \cdot a_1, n_i \cdot a_2, \ldots, n_i \cdot a_m$ and $n_j \cdot a_1, n_j \cdot a_2, \ldots, n_j \cdot a_m$, respectively. Two points n_i and n_j are equal if and only if for $\forall k\ (1 \leqslant k \leqslant m),\ n_i \cdot a_k = n_j \cdot a_k$.

A *relative distance* between n_i and n_j is defined as $\Delta\langle n_i, n_j\rangle = (n_j \cdot a_1 - n_i \cdot\ a_1, n_j \cdot a_2 - n_i \cdot a_2, \ldots, n_j \cdot a_m - n_i \cdot a_m)$. In the present paper, we also use the notation $\Delta_{(d_1,d_2,\ldots,d_m)}$, where $d_k = n_j \cdot a_k - n_i \cdot a_k\ (1 \leqslant k \leqslant m)$, to represent the relative distance between two points n_i and n_j in the m-dimensional space.

Besides the *absolute* representation $(n_i \cdot a_1, n_i \cdot a_2, \ldots, n_i \cdot a_m)$ for point n_i, we can also represent it by indicating its *relative distance* $\Delta\langle n_0, n_i\rangle$ from a certain *reference point* n_0, which is clear in the context of discourse. Let $\mathcal{N} = \{n_1, n_2, \ldots, n_u\}$ be a set of points in an m-dimensional space. The *largest reference point* of $\mathcal{N}$ is the point n_*, where for $\forall k\ (1 \leqslant k \leqslant m),\ n_* \cdot a_k = \min(n_1 \cdot a_k, n_2 \cdot a_k, \ldots, n_u \cdot a_k)$.

Example 2.1. Given two points in a two-dimensional space $n_1 = (0, 2)$ and $n_2 = (1, 1)$, the largest reference point of $\{n_1, n_2\}$ is $n_* = (0, 1)$, since $n_* \cdot a_1 = \min(n_1 \cdot a_1, n_2 \cdot a_1) = \min(0, 1) = 0$ and $n_* \cdot a_2 = \min(n_1 \cdot a_2, n_2 \cdot a_2) = \min(2, 1) = 1$.

Given a relative distance $\Delta\langle n_0, n_i\rangle$ from n_0, we can position the point n_i at $n_0 + \Delta\langle n_0, n_i\rangle$, i.e., $n_i = n_0 + \Delta\langle n_0, n_i\rangle$. Note that n_i, $\Delta\langle n_0, n_i\rangle$ and $\Delta_{(n_i\cdot a_1 - n_0\cdot a_1,\, n_i\cdot a_2 - n_0\cdot a_2,\, \ldots,\, n_i\cdot a_m - n_0\cdot a_m)}$ will be used interchangeably in the paper, since each of them refers to the same point n_i in the space.

2.1.3. Extended items (transactions)

The traditional concepts regarding *item* and *transaction* can be extended accordingly under an m-dimensional context. We call an item $i_k \in \mathcal{I}$ happening at the point $\Delta_{(d_1,d_2,\ldots,d_m)}$, i.e., at the point $(n_0 \cdot a_1 + d_1, n_0 \cdot a_2 + d_2, \ldots, n_0 \cdot a_m + d_m)$, an *extended item* and denote it as $\Delta_{(d_1,d_2,\ldots,d_m)}(i_k)$. In a similar fashion, we call a transaction $t_k \in \mathcal{T}$ happening at the point $\Delta_{(d_1,d_2,\ldots,d_m)}$ an *extended transaction* and denote it as $\Delta_{(d_1,d_2,\ldots,d_m)}(t_k)$. The set of all possible extended items, $\mathcal{I}_{\mathrm{E}}$, is defined as a set of $\Delta_{(d_1,d_2,\ldots,d_m)}(i_k)$ for any $i_k \in \mathcal{I}$ at all possible points $\Delta_{(d_1,d_2,\ldots,d_m)}$ in the m-dimensional space. $\mathcal{T}_{\mathrm{E}}$ is the set of all extended transactions, each of which contains a set of extended items, in the mining context.

2.1.4. Normalized extended item (transaction) sets

We call an extended itemset a *normalized extended itemset*, if all its extended items are positioned with respect to the largest reference point of the set. In other words, the extended items in the set have the minimal relative distance 0 for each dimension. Formally, let $I_{\mathrm{e}} = \{\Delta_{(d_{1,1},d_{1,2},\ldots,d_{1,m})}(i_1), \Delta_{(d_{2,1},d_{2,2},\ldots,d_{2,m})}(i_2), \ldots, \Delta_{(d_{k,1},d_{k,2},\ldots,d_{k,m})}(i_k)\}$ be an extended itemset. I_{e} is a normalized extended itemset, if and only if for $\forall j\ (1 \leqslant j \leqslant k) \forall i\ (1 \leqslant i \leqslant m)$, $\min(d_{j,i}) = 0$.

The normalization concept can be applied to an extended transaction set as well. We call an extended transaction set a *normalized extended transaction set*, if all its extended transactions are positioned with respect to the largest reference point of the set.

Any non-normalized extended item (transaction) set can be transformed into a normalized one through a *normalization process*, where the intention is to reposition all the involved extended items (transactions) based on the largest reference point of this set. We use $\mathscr{I}_{\mathrm{NE}}$ and $\mathscr{T}_{\mathrm{NE}}$ to denote the set of all possible normalized extended itemsets and normalized extended transaction sets, respectively.

In the following sections, we also call a set of (normalized) (extended) items simply a (normalized) (extended) *itemset*, and the number of items in a (normalized) (extended) itemset the *length* of this itemset. An itemset of length k is referred to as a k-itemset.

Example 2.2. Assume we have two extended itemsets $I_{\mathrm{e}} = \{\Delta_{(0,2)}(a), \Delta_{(1,0)}(b), \Delta_{(2,3)}(c)\}$ and $I'_{\mathrm{e}} = \{\Delta_{(1,2)}(a), \Delta_{(1,0)}(b), \Delta_{(2,3)}(c)\}$ in a two-dimensional space. I_{e} is a normalized extended itemset, since it has minimal value 0 for both dimensions, i.e., $\min(0,1,2) = 0$ and $\min(2,0,3) = 0$. But I'_{e} is not due to its non-zero minimal value for the first dimension.

Assume I'_{e} takes $n_0 = (n_0 \cdot a_1, n_0 \cdot a_2)$ as the reference point, from this we can locate the three points indicated by I'_{e} at $n_1 = (n_0 \cdot a_1 + 1, n_0 \cdot a_2 + 2), n_2 = (n_0 \cdot a_1 + 1, n_0 \cdot a_2 + 0)$, and $n_3 = (n_0 \cdot a_1 + 2, n_0 \cdot a_2 + 3)$, respectively. Following the definition, the largest reference point of I'_{e} should be $n_* = (n_0 \cdot a_1 + 1, n_0 \cdot a_2 + 0)$. Based on this new reference point n_*, we reposition the three extended items in I'_{e}, and finally obtain a normalized extended itemset $I''_{\mathrm{e}} = \{\Delta_{(0,2)}(a),\ \Delta_{(0,0)}(b),\ \Delta_{(1,3)}(c)\}$.

Property 2.1. *Any superset of a normalized extended item* (*transaction*) *set is also a normalized extend item* (*transaction*) *set.*

This property can be proven easily from the definition of normalized extended item (transaction) set.

2.1.5. Database holes in multi-dimensional contexts

Using m dimensional attributes, we can construct an m-dimensional context for the existence of transactions. Each transaction in the database can be mapped through a function to a certain point in this m-dimensional space. Transactions converging at the same point can be combined into one by performing the UNION operation on their itemsets. When a point p located at $(h_0, h_2, \ldots, h_m)$ receives no transactions, we say a *hole* occurs at p in this m-dimensional space. In general, there are three situations that can cause a hole at position p, namely, Case 1 and Case 2/(1), Case 2/(2) considered below:

Case 1 (p represents a *meaningless context*). The combination of dimensional values at $p = (h_0, h_1, \ldots, h_m)$ offers no way for transactions to occur. For example, suppose we have a database recording disease histories of all the patients in a mountainous region. The domains of its two dimensional attributes *age* and *elevation* have been discretized into $\{10, 20, 30, 40, 50, 60, 70, 80\}$ and $\{0, 500, 1000, 1500, 2000, 2500, 3000\}$, respectively. In this particular region, no patient older than 60 lives above the elevation 2500. Thus, the combinations of dimensional values at $(70, 2500)$, $(70, 3000)$, $(80, 2500)$, $(80, 3000)$ are meaningless contexts for database records, leaving four holes in this two-dimensional space.

Case 2 (p represents a *meaningful context*). Here, there are either (1) no transactions that occur at p. For instance, if all the young people of age 20 living at the elevation 1000 never go to the regional hospital, then the database has no disease records at (20, 1000). Under this circumstance, we can fill the hole with a transaction containing an empty item list; or (2) some transactions occur at p, but their contents are missing due to improper data entry and maintenance facilities. This kind of holes can be filled using missing data handling techniques.

As the holes caused by Case 2 can be filled by proper data pre-processing, we assume the databases to be mined contain only holes caused by Case 1. In the discussion that follows, holes mentioned imply the meaningless contexts for database transactions (i.e., Case 1).

Example 2.3. Fig. 1 shows a simple transactional database under a two-dimensional context. The domains of the two dimensional attributes X and Y have been discretized into 4 and 5 equal-sized intervals, respectively. There are three holes at $(1,1)$, $(1,3)$, and $(3,3)$. Therefore, the whole database has totally 17 transactions, whereas the itemsets of transaction t_{12} and transaction t_{16} are empty. Table 1 gives the transformed extended transactional database, where all the 17 transactions have been positioned with respect to a certain common reference point.

2.1.6. Two different relationships – possibly containing or containing

Due to the possible existence of data holes in the mining contexts, we introduce the notions of *possibly containing* and *containing* relationships between an extended transaction set and an extended item set. Let T_e be an extended transaction set in the database. We denote the normalization form of T_e by prefixing the foot label "e" with "n", i.e., T_{ne}. We define that T_e *possibly contains* a normalized extended itemset I_{ne}, if for $\forall\Delta_{(x_1,x_2,\ldots,x_m)}(i) \in I_{ne}$, there exists an extended transaction $\Delta_{(x_1,x_2,\ldots,x_m)}(t) \in T_{ne}$. We define that T_e *contains* I_{ne}, if for $\forall\Delta_{(x_1,x_2,\ldots,x_m)}(i) \in I_{ne}$, there exists an extended transaction $\Delta_{(x_1,x_2,\ldots,x_m)}(t) \in T_{ne}$ where $(i \in t)$. A *minimal* extended transaction set T_e that contains (or possibly contains) I_{ne} is the one, where there exists no other extended transaction set T_e' contains (or possibly contains) I_{ne}.

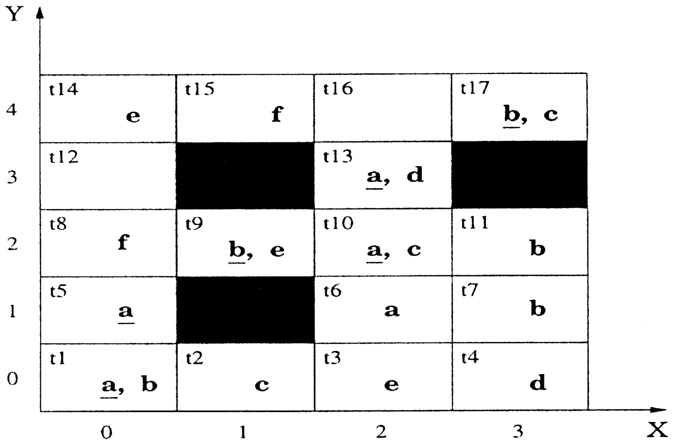

Fig. 1. A simple database under a two-dimensional context.

Table 1
A transformed two-dimensional extended transactional database example

Transaction	Extended transaction	Extended items
t_1	$\Delta_{(0,0)}(t_1)$	$\Delta_{(0,0)}(a), \Delta_{(0,0)}(b)$
t_2	$\Delta_{(1,0)}(t_2)$	$\Delta_{(1,0)}(c)$
t_3	$\Delta_{(2,0)}(t_3)$	$\Delta_{(2,0)}(e)$
t_4	$\Delta_{(3,0)}(t_4)$	$\Delta_{(3,0)}(d)$
t_5	$\Delta_{(0,1)}(t_5)$	$\Delta_{(0,1)}(a)$
t_6	$\Delta_{(2,1)}(t_6)$	$\Delta_{(2,1)}(a)$
t_7	$\Delta_{(3,1)}(t_7)$	$\Delta_{(3,1)}(b)$
t_8	$\Delta_{(0,2)}(t_8)$	$\Delta_{(0,2)}(f)$
t_9	$\Delta_{(1,2)}(t_9)$	$\Delta_{(1,2)}(b), \Delta_{1,2}(e)$
t_{10}	$\Delta_{(2,2)}(t_{10})$	$\Delta_{(2,2)}(a), \Delta_{(2,2)}(c)$
t_{11}	$\Delta_{(3,2)}(t_{11})$	$\Delta_{(3,2)}(b)$
t_{12}	$\Delta_{(0,3)}(t_{12})$	
t_{13}	$\Delta_{(2,3)}(t_{13})$	$\Delta_{(2,3)}(a), \Delta_{(2,3)}(d)$
t_{14}	$\Delta_{(0,4)}(t_{14})$	$\Delta_{(0,4)}(e)$
t_{15}	$\Delta_{(1,4)}(t_{15})$	$\Delta_{(1,4)}(f)$
t_{16}	$\Delta_{(2,4)}(t_{16})$	
t_{17}	$\Delta_{(3,4)}(t_{17})$	$\Delta_{(3,4)}(b), \Delta_{(3,4)}(c)$

Example 2.4. Let $I_{\text{ne}} = \{\Delta_{(0,0)}(a), \Delta_{(0,1)}(a), \Delta_{(1,2)}(b)\}$. According to the definition, a minimal extended transaction set that possibly contains I_{ne} is the one which after normalization has extended transactions (not holes) located at $\Delta_{(0,0)}$, $\Delta_{(0,1)}$ and $\Delta_{(1,2)}$. From the database shown in Fig. 1, we can find four minimal extended transaction sets that possibly contain I_{ne}. They are

$$T_{\text{e},1} = \{\Delta_{(0,0)}(t_1), \Delta_{(0,1)}(t_5), \Delta_{(1,2)}(t_9)\}, \quad T_{\text{e},2} = \{\Delta_{(2,0)}(t_3), \Delta_{(2,1)}(t_6), \Delta_{(3,2)}(t_{11})\},$$

$$T_{\text{e},3} = \{\Delta_{(0,2)}(t_8), \Delta_{(0,3)}(t_{12}), \Delta_{(1,4)}(t_{15})\}, \quad T_{\text{e},4} = \{\Delta_{(2,2)}(t_{10}), \Delta_{(2,3)}(t_{13}), \Delta_{(3,4)}(t_{17})\}.$$

After normalization, each of them accommodates transactions at $\Delta_{(0,0)}, \Delta_{(0,1)}$ and $\Delta_{(1,2)}$. Among these four extended transaction sets, only the first and the last extended transaction sets contain I_{ne}, since $(a \in t_1, a \in t_5, b \in t_9)$ and $(a \in t_{10}, a \in t_{13}, b \in t_{17})$.

With the above notation, we are now in a position to formally define inter-transactional association rules and related measurements. Note that the definitions described in Section 2.2 are more concise and complete, compared to the previous ones appearing in our earlier work [9,17,18,36].

2.2. Inter-transactional association rules

Definition 2.1. *An inter-transactional association rule* under an m-dimensional context is an implication of the form $X \Rightarrow Y$, where

1. $X \subset \mathscr{I}_{\text{NE}}$ and $Y \subset \mathscr{I}_{\text{E}}$;
2. The extended items in X and Y are positioned with respect to the same reference point;
3. For $\forall\Delta_{(x_1,x_2,\ldots,x_m)}(i_x) \in X,\ \forall\Delta_{(y_1,y_2,\ldots,y_m)}(i_y) \in Y,\ x_j \leqslant y_j\ (1 \leqslant j \leqslant m)$;
4. $X \cap Y = \emptyset$.

Different from classical intra-transactional association rules, the inter-transactional association rules capture the occurrence contexts of associated items. The first clause of the definition requires the precedent and antecedent of an inter-transactional association rule to be a normalized extended itemset and an extended itemset, respectively. This is a more concise specification for the inter-transactional association rule format compared to the one in [9,17,18,36]. The second clause of the definition ensures items in X and Y are comparable in terms of their contextual positions. For prediction, in this study, each of the consequent items in Y takes place in a context later than any of its precedent items in X, as stated by the third clause.

Based on Definition 2.1, a rule that predicts a "warm weather" like "if the wind direction is east and the weather is currently warm, then it keeps warm for the next 12 h," can be expressed by a one-dimensional inter-transactional association rule "$\Delta_{(0)}(\textit{east wind direction})$, $\Delta_{(0)}(\textit{warm}) \Rightarrow \Delta_{(2)}(\textit{warm})$". Here, every interval represents 6 h.

Similar to intra-transactional association rules, we use *support* and *confidence* as two major measurements under the inter-transactional association mining framework. Traditionally, the support of a rule $X \Rightarrow Y$ is the fraction of transactions that contain $X \cup Y$ over the whole transactions, and the confidence of the rule is the fraction of transactions containing X that also contain Y. However, to measure inter-transactional association rules detected from a multi-dimensional space, the contextual information of transactions cannot be ignored, especially when such a mining space contains possibly meaningless contexts. Here, we extend our previous measurement definition [9,17,18,36] by exploring the two different relationships (i.e., *possibly containing* and *containing*) between an extended transaction set and an extended itemset. Such an extended definition is more general and can be applied no matter whether the mining context contains data holes or not.

Definition 2.2. Given a normalized extended itemset X and an extended itemset Y, let $|T_{xy}|$ be the total number of minimal extended transaction sets that contain $X \cup Y$, and $|T^{\mathrm{p}}_{xy}|$ be the total number of minimal extended transaction sets that possibly contain $X \cup Y$, and $|T_{x,xy^{\mathrm{p}}}|$ be the total number of minimal extended transaction sets in the database that contain X and meanwhile possibly contain $X \cup Y$. The *support* and *confidence* of an inter-transactional association rule $X \Rightarrow Y$ is defined as: $\mathrm{support}(X \Rightarrow Y) = |T_{xy}|/|T^{\mathrm{p}}_{xy}|$ and $\textit{confidence}(X \Rightarrow Y) = |T_{xy}|/|T_{x,xy^{\mathrm{p}}}|$.

Example 2.5. Suppose we have an inter-transactional association rule "$\Delta_{(0,0)}(a)$, $\Delta_{(0,1)}(a) \Rightarrow \Delta_{(1,2)}(b)$" detected from the database in Fig. 1, with $X = \{\Delta_{(0,0)}(a), \Delta_{(0,1)}(a)\}$ and $Y = \{\Delta_{(1,2)}(b)\}$. From Example 2.4, we know that the total number of minimal extended transaction sets that possibly contain and contain $X \cup Y$ is 4 and 2, respectively. Thus, $\textit{support} = |T_{xy}|/|T^{\mathrm{p}}_{xy}| = 2/4 = 50\%$. Among the four minimal extended transaction sets that possibly contain $X \cup Y$, only two extended transaction sets contain X. Thus, $\textit{confidence} = |T_{xy}|/|T_{x,xy^{\mathrm{p}}}| = 2/2 = 100\%$.

Given a user-specified minimum support (called *minsup*) and minimum confidence (called *minconf*), the task for mining inter-transactional association rules is to discover from a multi-dimensional database a complete set of inter-transactional association rules with $\textit{support} \geqslant \textit{minsup}$ and $\textit{confidence} \geqslant \textit{minconf}$, respectively.

3. The downward closure problem in mining inter-transactional association rules

The traditional intra-transactional association rule mining is decomposed into two steps:

(1) All itemsets that have support above the user-specified minimum support are first detected. These itemsets are called the *large* itemsets. All others are said to be *small.*

(2) From each large itemset, all the intra-transactional association rules that have minimum confidence are generated as follows: for a large itemset X and any $Y \subset X$, if $sup(X)/sup(X-Y) \geqslant minconf$, then the rule $X - Y \Rightarrow Y$ is a valid rule.

As the overall mining performance is dominated by the first step and the second step is relatively straightforward, most previous work has focused on deriving efficient algorithms for finding large itemsets. To avoid testing support for every possible itemset which is of exponential complexity in computation, almost all existing algorithms exploit the following *downward closure property* – "any subset of a large itemset is also large". For example, if a transaction contains itemset $\{a,b,c,d\}$, then it also contains any subset of $\{a,b,c,d\}$, e.g., $\{a\}, \{a,b\}, \{a,b,c\}$, etc. Conversely, any superset of a small itemset is also small. Therefore, if at some stage it is found that itemset $\{a,d,e\}$ is small, then none of its supersets, e.g., $\{a,d,e,f\}, \{a,d,e,f,g\}$, etc., need to be tested for minimum support. Based on this property, the discovery of large itemsets can be performed in a levelwise manner. First, supports for all itemsets of length 1 (i.e., 1-itemsets) are tested by scanning the entire database. The itemsets that are found to be small are discarded. Then, a set of 2-itemsets called *candidate* itemsets are generated by combining the large 1-itemsets are discarded. Similarly, candidate 3-itemsets are formed by combining large 2-itemsets, and their supports are tested by scanning the entire database and the small 2-itemsets, and their supports are counted with the small 3-itemsets discarded. This process is repeated until no more large itemsets are found.

3.1. Invalid downward closure property for inter-transactional association rules

Note that keeping the nice monotonicity of support level is the base of a large set of efficient intra-transactional association rule mining algorithms. Unfortunately, such a nice property does not hold under the more general inter-transactional association mining framework, especially when the mining context under investigation contains various data holes. To illustrate, let us look at the following one-dimensional database.

Example 3.1. The database in Fig. 2 has three holes which are meaningless contexts for database transactions. Let $X = \{\Delta_{(0)}(a), \Delta_{(1)}(b)\}$ be a normalized extended itemset. There are five minimal extended transaction sets that possibly contain X, which are $\{\Delta_{(0)}(t_1), \Delta_{(1)}(t_2)\}$, $\{\Delta_{(3)}(t_3), \Delta_{(4)}(t_4)\}$, $\{\Delta_{(4)}(t_4), \Delta_{(5)}(t_5)\}$, $\{\Delta_{(9)}(t_7), \Delta_{(10)}(t_8)\}$, $\{\Delta_{(10)}(t_8), \Delta_{(11)}(t_9)\}$. After normalization, each of them has transactions located at $\Delta_{(0)}$ and $\Delta_{(1)}$. We use a sliding window to capture each of these transaction sets. Among the five sliding windows $W_1, \ldots, W_5$, only W_1 and W_2 contain X. Hence, the support of X is $sup(X) = |T_X|/|T_X^{\mathrm{p}}| = 2/5 = 40\%$.

Similarly, let X' be a superset of X, where $X' = \{\Delta_{(0)}(a), \Delta_{(1)}(b), \{\Delta_{(3)}(c)\}$. Due to the enlarged scope of contextual requirement and database holes, only two minimal extended transaction sets possibly contain X', which are $\{\Delta_{(0)}(t_1), \Delta_{(1)}(t_2), \Delta_{(3)}(t_3)\}$ and $\{\Delta_{(4)}(t_4), \Delta_{(5)}(t_5), \Delta_{(7)}(t_6)\}$, encapsulated by two sliding windows W_1' and W_2', respectively. Note that the third position (indicated by a dotted line) within window W_1' and W_2', is not considered, since any transaction happening there

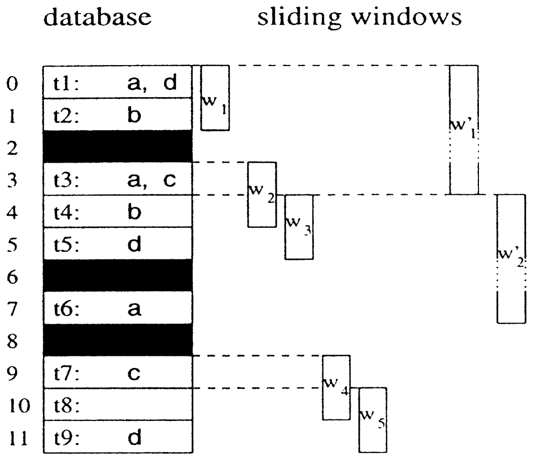

Fig. 2. A simple example database under a one-dimensional context.

(i.e., $\Delta_{(2)}$ and $\Delta_{(6)}$) does not belong to the considered extended transaction sets. Of the two windows W_1' and W_2', only W_1' contains X', thus, the support X' is $sup(X') = |T_{X'}|/|T_{X'}^{\mathrm{p}}| = 1/2 = 50\%$. If the minimum support level is set as $minsup = 45\%$, we have $sup(X') > minsup$, but $sup(X) < minsup$. This violates the downward closure property “any subset of a large itemset is also large”.

From Example 3.1, we can find that the invalidation of the downward closure property results from the smaller denominator in the computation of $sup(X') = |T_{X'}|/|T_{X'}^{\mathrm{p}}| = 1/2 = 50\%$ than that in the computation of $sup(X) = |T_X|/|T_X^{\mathrm{p}}| = 2/5 = 40\%$, i.e., the number of minimal extended transaction sets that possibly contain X' ($|T_{X'}^{\mathrm{p}}| = 2$) is smaller than the number of minimal extended transaction sets that possibly contain X ($|T_X^{\mathrm{p}}| = 5$).

In general we have the following lemmas regarding the concepts of *possibly containing* and *containing* relationships.

Lemma 3.1. *Given a normalized extended itemset X, for any of its superset $X' \supset X$, the number of minimal extended transaction sets in an m-dimensional database that* possibly contain *X is not less than the number of minimal extended transaction sets that* possibly contain *X', i.e., $|T_X^{\mathrm{p}}| \geqslant |T_{X'}^{\mathrm{p}}|$.*

Proof. Without loss of generality, assume that

$$X = \{\Delta_{(d_{1,1},\ldots,d_{1,m})}(i_1),\ \Delta_{(d_{2,1},\ldots,d_{2,m})}(i_2),\ \ldots,\ \Delta_{(d_{k,1},\ldots,d_{k,m})}(i_k)\},$$

$$X' = \{\Delta_{(d_{1,1},\ldots,d_{1,m})}(i_1),\ \Delta_{(d_{2,1},\ldots,d_{2,m})}(i_2),\ldots,\ \Delta_{(d_{k,1},\ldots,d_{k,m})}(i_k),\ \Delta_{(d_{k+1,1},\ldots,d_{k+1,m})}(i_k+1)\}.$$

There are two possible cases.

Case 1: X and X' have the same contextual point set, i.e.,

$$\exists\Delta_{(d_{j,1},\ldots,d_{j,m})}\ (1 \leqslant j \leqslant k),\quad \Delta_{(d_{k+1,1},\ldots,d_{k+1,m})} = \Delta_{(d_{j,1},\ldots,d_{j,m})}.$$

(1) For any minimal extended transaction set τ that possibly contains X, according to the "possibly containing" definition in Section 2, τ is also a minimal extended transaction set that possibly contains X', i.e., $\forall \tau \in T_X^{\mathrm{p}}$ implies $\tau \in T_{X'}^{\mathrm{p}}$.

(2) On the other hand, for any minimal extended transaction set $\bar{\tau}$ that does not possibly contain X, neither does it possibly contain X', i.e., $\forall \bar{\tau} \notin T_X^{\mathrm{p}}$ implies $\bar{\tau} \notin T_{X'}^{\mathrm{p}}$.

Based on (1) and (2), $|T_X^{\mathrm{p}}| = |T_{X'}^{\mathrm{p}}|$.

Case 2: X and X' have different contextual point sets, i.e.,

$$\forall \Delta_{(d_{j,1},\ldots,d_{j,m})}\ (1 \leq j \leq k), \quad \Delta_{(d_{k+1,1},\ldots,d_{k+1,m})} \neq \Delta_{(d_{j,1},\ldots,d_{j,m})}.$$

(1) Let τ be any minimal extended transaction set that possibly contains X. Assume $(r_1, \ldots, r_m)$ is the largest reference point of τ. Consider the contextual position $p = (r_1 + d_{k+1,1}, \ldots, r_m + d_{k+1,m})$ for the argumented extended item $\Delta_{(d_{k+1,1},\ldots,d_{k+1,m})}(i_{k+1})$ in X'.

(a) If p is a meaningful context with one transaction t_{k+1}, with the same largest reference point $(r_1, \ldots, r_m)$, then $\tau' = \tau \cup \{\Delta_{(r_1+d_{k+1,1},\ldots,r_m+d_{k+1,m})}(t_{k+1})\}$ is a minimal extended transaction set that possibly contains X', i.e., $\forall \tau \in T_X^{\mathrm{p}}$ implies $\tau' \in T_{X'}^{\mathrm{p}}$.

(b) If p is a meaningless context (hole) with no transaction, with the same largest reference point $(r_1, \ldots, r_m)$, we cannot find a minimal extended transaction set that possibly contains X'.

(c) If p is out of the contextual scope of database transactions, consequently no transactions exist at p, with the same largest reference point $(r_1, \ldots, r_m)$, we cannot find a minimal extended transaction set that possibly contains X', either.

As a result, from the same reference point of τ, we cannot always find a minimal extended transaction set that possibly contains X'.

(2) On the other hand, for any minimal extended transaction set $\bar{\tau}$ that does not possibly contain X, neither does $\bar{\tau}'$ possibly contain X'.

Based on (1) and (2), $|T_X^{\mathrm{p}}| \geqslant |T_{X'}^{\mathrm{p}}|$. Summarizing Cases 1 and 2, we have $|T_X^{\mathrm{p}}| \geqslant |T_{X'}^{\mathrm{p}}|$. The lemma gets proven. □

Lemma 3.2. *Given a normalized extended itemset X, for any of its superset $X' \supset X$, the number of minimal extended transaction sets in an m-dimensional database that* contain *X is not less than the number of minimal extended transaction sets that* contain *X', i.e., $|T_X| \geqslant |T_{X'}|$.*

Proof. (1) For any minimal extended transaction set τ' that contains X', since $X \subset X'$, then τ' also contains X. In other words, with the same largest reference point as τ', there also exists a minimal extended transaction set τ that contains X.

(2) However, for any minimal extended transaction set τ that contains X, there may not exist a minimal extended transaction set τ' with the same largest reference point of τ that contains X'. This is either because of the lack of a corresponding extended transaction set that possibly contains X', or the argumented item in X' is not included in the corresponding transaction. Therefore, $|T_X| \geqslant |T_{X'}|$. □

Based on Lemma 3.1 and 3.2, we have

$$sup(X) = \frac{|T_X|}{|T_X^{\mathrm{p}}|} \geqslant \frac{|T_{X'}|}{|T_X^{\mathrm{p}}|} = \frac{sup(X') * |T_{X'}^{\mathrm{p}}|}{|T_X^{\mathrm{p}}|} = \frac{|T_{X'}^{\mathrm{p}}|}{|T_X^{\mathrm{p}}|} * sup(X') \geqslant \frac{|T_{X'}^{\mathrm{p}}|}{|T_X^{\mathrm{p}}|} * minsup. \tag{3.1}$$

It indicates that in order for $sup(X')$ to satisfy the *minsup* requirement, the support of its subset may not necessarily achieve *minsup*, since $|T^{\mathrm{p}}_{X'}| \leqslant |T^{\mathrm{p}}_{X}|$ and $|T^{\mathrm{p}}_{X'}|/|T^{\mathrm{p}}_{X}| * minsup \leqslant minsup$.

Theorem 3.1. *Given two normalized extended itemsets X and X′ where* $X \subset X'$, $sup(X') \geqslant minsup$, *if and only if* $sup(X) \geqslant |T^{\mathrm{p}}_{X'}|/|T^{\mathrm{p}}_{X}| * minsup$.

The theorem is a straightforward derivation from Eq. (3.1).

Example 3.2. In Example 3.1, $|T^{\mathrm{p}}_{X}| = 5$ and $|T^{\mathrm{p}}_{X'}| = 2$. To ensure $sup(X') \geqslant minsup = 45\%$, $sup(X) \geqslant (2/5) * 45\% = 18\%$. Because $sup(X) = 40\% > 18\%$, we must incorporate X to generate candidate X' rather than dispose it as what the traditional mining algorithms do with the small itemsets in further candidate generation.

3.2. A proposal for solution of the downward closure problem

It is clear about the cause of the downward closure problem with the inter-transactional association framework. In practice, to determine whether a normalized extended candidate itemset X' shall be generated and counted, we need to check all the subsets of X'. Any of them dissatisfying the support requirement specified by Theorem 3.1 exempts X' from the further support-counting processes. However, it would be very time-consuming if for each subset X of X', the validity of $sup(X) \geqslant |T^{\mathrm{p}}_{X'}|/|T^{\mathrm{p}}_{X}| * minsup$ has to be tested, especially when the database considered is huge with a large amount of items and item combinations. Moreover, computing $|T^{\mathrm{p}}_{X'}|$ and $|T^{\mathrm{p}}_{X}|$ which vary with individual itemset also demands computational effort. In order to achieve efficient inter-transactional association mining performance, we adopt a compromise strategy using a lower-bound $|T^{\mathrm{p}}_{X'}|/|T^{\mathrm{p}}_{X}|$ rather than target a concise value based on the following equation:

$$sup(X) \geqslant \frac{|T^{\mathrm{p}}_{X'}|}{|T^{\mathrm{p}}_{X}|} * minsup \geqslant \alpha * minsup \qquad (0 < \alpha \leqslant 1).$$

That is, as long as any subset of X' has a support not less than $\alpha * minsup$, the potentially large candidate itemset X' shall be formed and counted during the next round of database scan. In this way, the computational overhead incurred for testing each subset of X' is lower.

Here, to derive α while keeping the precision loss low enough, we need to investigate how database holes affect the calculation of $|T^{\mathrm{p}}_{X'}|$ and $|T^{\mathrm{p}}_{X}|$. For ease of understanding, we start our discussion with the one-dimensional context, and then extend it to m-dimensions.

3.2.1. One-dimensional contexts

Given a normalized extended k-itemset $X = \{\Delta_{(d_1)}(i_1), \Delta_{(d_2)}(i_2), \ldots, \Delta_{(d_k)}(i_k)\}$, we define a window w of length l_X as shown in Fig. 2, where $l_X = \max(d_1, d_2, \ldots, d_k) - \min(d_1, \ldots d_k) + 1$. Window w slides along the dimension, capturing each time a potential minimal extended transaction set that possibly contains X. The function $TransExist(w, d_i)$ returns *true* if there is a transaction at the d_ith position of w; and *false* otherwise. We say a current window w catches an extended transaction set $\{\Delta_{(d_1)}(i_1), \Delta_{(d_2)}(i_2), \ldots, \Delta_{(d_k)}(i_k)\}$ if at any of its d_ith position where $d_i \in \{d_1, d_2, \ldots, d_k\}$, there exists a transaction rather than a hole (i.e., $TransExist(w, d_i) = $ true). For instance, in Fig. 2, window w'_1 and window w'_2 each capture a minimal extended transaction set that possibly contains $\{\Delta_{(0)}(a), \Delta_{(1)}(b), \Delta_{(3)}(c)\}$, since $\mathit{TransExist}\ (w'_1, 0) = \mathit{TransExist}(w'_1, 1) = \mathit{TransExist}(w'_1, 3) = \text{true}$, and $\mathit{TransExist}(w'_2, 0) = \mathit{TransExist}(w'_2, 1) = \mathit{TransExist}(w'_2, 3) = \text{true}$.

Given a one-dimensional database *db*, with respect to a certain reference point, assume the closest and the farthest locations where its transactions locate are $\Delta_{(0)}$ and $\Delta_{(U_{db})}$, respectively. In between $\Delta_{(0)}$ and $\Delta_{(U_{db})}$, there may be some hole positions. To count minimal extended transaction sets that possibly contain *X*, we need to slide the window constructed for *X* from the starting position $\Delta_{(0)}$ till $\Delta_{(U_{db}-l_x+1)}$. After $\Delta_{(U_{db}-l_x+1)}$, the window will outstrip the contextual scope of the database transactions and is not applicable any more. Therefore, we call $[0, U_{db} - l_X + 1]$ the *X-applicable sliding range*. All windows starting from within this range constitute a window set $\mathscr{W}_X$, referred to as *X-applicable window set* whose cardinality $|\mathscr{W}_X|$ is

$$|\mathscr{W}_X| = U_{db} - l_X + 1 - 0 + 1 = U_{db} - l_X + 2. \tag{3.2}$$

$l_X = \max(d_1, \ldots, d_k) + 1$ gives maximal contextual span of the items in *X*. Given the fact that in real-world applications, users are usually interested in associations happening within a certain range, such as gas stations and fast-food outlets within 50 miles, stock indexes rising within a week, etc., we introduce a *maxspan* threshold to limit the maximal value of d_i, where $d_i \in \{d_1, d_2, \ldots, d_k\}$, in the itemsets of interest. Therefore, we have

$$U_{db} - \mathit{maxspan} + 1 \leqslant |\mathscr{W}_X| = U_{db} - \max(d_1, \ldots, d_k) + 1 \leqslant U_{db} + 1. \tag{3.3}$$

Let $\mathscr{W}^-_{(X,d_i)}$ be a subset of $\mathscr{W}_X$, where $(d_i \in \{d_1, d_2, \ldots, d_k\}) \wedge (\forall w \in \mathscr{W}^-_{(X,d_i)}, \mathit{TransExists}(w, d_i) = \text{false})$. In other words, each window in $\mathscr{W}^-_{(X,d_i)}$ has a hole at its d_ith position. If the database has N_{hole} number of holes, then $|\mathscr{W}^-_{(X,d_i)}| \leqslant N_{\text{hole}}$, since N_{hole} holes produce at most N_{hole} number of windows having a hole at the d_ith position. In case some holes fall into the first window starting from $\Delta_{(0)}$, and their positions are lower than $\Delta_{(d_i)}$, these holes will not generate a hole window for $\mathscr{W}^-_{(X,d_i)}$.

The union

$$\mathscr{W}^-_X = \mathscr{W}^-_{(X,d_1)} \cup \mathscr{W}^-_{(X,d_2)} \cup \cdots \cup \mathscr{W}^-_{(X,d_k)} \tag{3.4}$$

returns a subset of *X*-applicable windows, such that any window in $\mathscr{W}^-_X$ contains one or several holes, i.e., $\forall w \in \mathscr{W}^-_{(X,d_i)}\ \exists d_i \in \{d_1, d_2, \ldots d_k\}, \mathit{TransExists}(w, d_i) = \text{false}$.

With the notion of *X*-applicable windows, the total number of minimal extended transaction sets that possibly contain X, $|T^{\text{p}}_X|$, is then equal to the difference between $|\mathscr{W}_X|$ and $|\mathscr{W}^-_X|$, i.e.,

$$|T^{\text{p}}_X| = |\mathscr{W}_X| - |\mathscr{W}^-_X|. \tag{3.5}$$

Example 3.3. The upper bound U_{db} of the context for the one-dimensional database in Fig. 3 is 9. Given a normalized extended itemset $X = \{\Delta_{(0)}(a), \Delta_{(1)}(b), \Delta_{(3)}(c)\}$, a window of length $l_X = \max(0, 1, 3) + 1 = 4$ can slide from within the *X*-applicable sliding range $[0, U_{db} - l_X + 1] = [0, 6]$, forming an *X*-applicable window set $\mathscr{W}_X = \{w_1, w_2, w_3, w_4, w_5, w_6, w_7\}$. $|\mathscr{W}_X| = U_{db} - l_X + 2 = 9 - 4 + 2 = 7$. Among the seven windows, w_3 and w_7 each have a hole at the 0th position, w_2 and w_6 each have a hole at the 1st position, and w_4 and w_6 each have a hole at the 3rd position. Hence, $\mathscr{W}^-_{(X,0)} = \{w_3, w_7\}, \mathscr{W}^-_{(X,1)} = \{w_2, w_6\}, \mathscr{W}^-_{(X,3)} = \{w_4, w_6\}$. $\mathscr{W}^-_X = \mathscr{W}^-_{(X,0)} \cup \mathscr{W}^-_{(X,1)} \cup \mathscr{W}^-_{(X,3)} = \{w_2, w_3, w_4, w_6, w_7\}$. Consequently, we have $\mathscr{W}_X - \mathscr{W}^-_X = \{w_1, w_5\}$. The formula $|T^{\text{p}}_X| = |\mathscr{W}_X| - |\mathscr{W}^-_X| = 7 - 5 = 2$ tells us that only two minimal extended transaction sets, $\{\Delta_{(0)}(t_1), \Delta_{(1)}(t_2), \Delta_{(3)}(t_3)\}$ and $\{\Delta_{(4)}(t_4), \Delta_{(5)}(t_5), \Delta_{(7)}(t_6)\}$, possibly contain *X*.

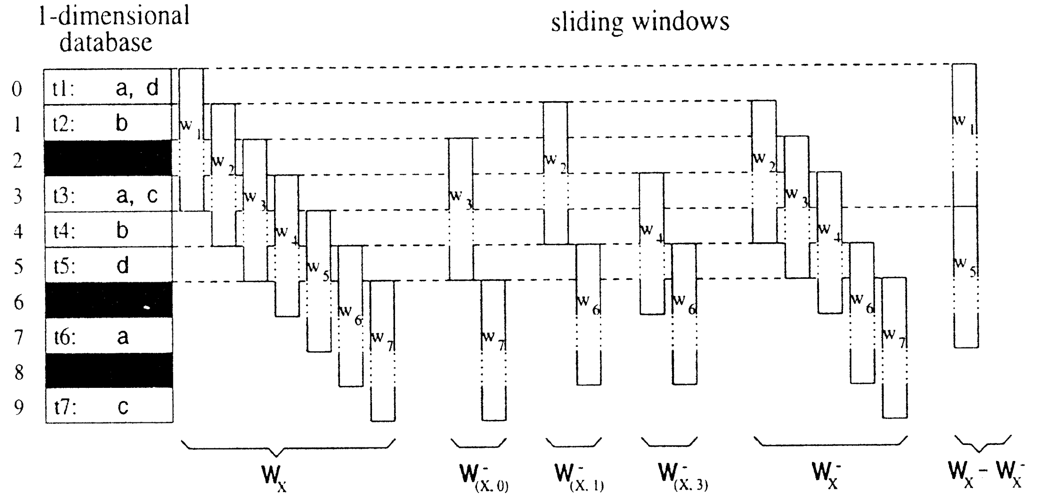

Fig. 3. Examples of applicable sliding windows for itemset *X*.

Based on Eqs. (3.3)–(3.5), we have

$$\begin{aligned}|T_X^{\mathrm{p}}| &= |\mathcal{W}_X| - |\mathcal{W}_X^-| = |\mathcal{W}_X| - \left|\mathcal{W}^-_{(X,d_1)} \cup \mathcal{W}^-_{(X,d_2)} \cup \cdots \cup \mathcal{W}^-_{(X,d_k)}\right| \\ &\geqslant (U_{db} - maxspan + 1) - \left(\left|\mathcal{W}^-_{(X,d_1)}\right| + \left|\mathcal{W}^-_{(X,d_2)}\right| + \cdots + \left|\mathcal{W}^-_{(X,d_k)}\right|\right) \\ &\geqslant (U_{db} - maxspan + 1) - k * N_{\mathrm{hole}}.\end{aligned}$$

Similarly, for a superset of X,X', which has one more item, $|T_{X'}^{\mathrm{p}}| \geqslant (U_{db} - maxspan + 1) - (k+1) * N_{\mathrm{hole}}$. Since $|\mathcal{W}_X| \leqslant U_{db} + 1$ in Eq. (3.3),

$$sup(X) = \frac{|T_{X'}^{\mathrm{p}}|}{|T_X^{\mathrm{p}}|} * minsup \geqslant \frac{(U_{db} - maxspan + 1) - (k+1) * N_{\mathrm{hole}}}{U_{db} + 1} * minsup.$$

Based on the above derivation, we can come up with a revised theorem.

Theorem 3.2. *Given two normalized extended itemsets X and X′ (X ⊂ X′) in a one-dimensional context, sup(X′) ⩾ minsup, if and only if sup(X) ⩾ α ∗ minsup, where*

$$\alpha = \frac{(U_{db} - maxspan + 1) - (k+1) * N_{\mathrm{hole}}}{U_{db} + 1}.$$

Theorem 3.2 leads to the solution of the downward closure problem in mining inter-transactional association rules under one-dimensional contexts. That is, *any subset of a large extended itemset* (*i.e., the one with support not less than* minsup) *has support equal to or greater than* $\alpha * minsup$, where $\alpha = (U_{db} - maxspan + 1 - (k+1) * N_{\mathrm{hole}})/(U_{db} + 1)$. Candidate $(k+1)$-itemsets can thus be generated from k-itemsets whose supports above $\alpha * minsup$ instead of *minsup*. Details of the mining approach are given in Section 4.

3.2.2. m-Dimensional contexts

The above discussion for one-dimensional contexts can be extended to m-dimensional contexts. Assume the contextual scope of an m-dimensional database is $[\Delta_{(0,\ldots,0)}, \Delta_{(U_{db_1},\ldots,U_{db_m})}]$ with respect to a certain reference point. Given a normalized extended k-itemset $X = \{\Delta_{(d_{1,1},\ldots,d_{1,m})}(i_1), \Delta_{(d_{2,1},\ldots,d_{2,m})}(i_2), \ldots, \Delta_{(d_{k,1},\ldots,d_{k,m})}(i_k),\}$, an *m-dimensional cube* of length $(l_{X,1}, l_{X,2}, \ldots l_{X,m})$ along the m dimensions can be defined, where

$$\begin{aligned}
l_{X,1} &= \max(d_{1,1}, d_{2,1}, \ldots, d_{k,1}) - \min(d_{1,1}, d_{2,1}, \ldots, d_{k,1}) + 1 = \max(d_{1,1}, d_{2,1}, \ldots, d_{k,1}) + 1;\\
l_{X,2} &= \max(d_{1,2}, d_{2,2}, \ldots, d_{k,2}) - \min(d_{1,2}, d_{2,2}, \ldots, d_{k,2}) + 1 = \max(d_{1,2}, d_{2,2}, \ldots, d_{k,1}) + 1;\\
&\vdots\\
l_{X,m} &= \max(d_{1,m}, d_{2,m}, \ldots, d_{k,m}) - \min(d_{1,m}, d_{2,m}, \ldots, d_{k,m}) + 1 = \max(d_{1,m}, d_{2,m}, \ldots, d_{k,1}) + 1.
\end{aligned}$$

From the application viewpoint, we introduce a series of parameters $maxspan_1$, $maxspan_2, \ldots, maxspan_m$ to limit the maximal values of $d_{j,1}, d_{j,2}, \ldots, d_{j,m}$ $(1 \leqslant j \leqslant k)$ in the k-itemset X. That is,

$$\begin{aligned}
l_{X,1} &= \max(d_{1,1}, d_{2,1}, \ldots, d_{k,1}) + 1 \leqslant maxspan_1 + 1;\\
l_{X,2} &= \max(d_{1,2}, d_{2,2}, \ldots, d_{k,2}) + 1 \leqslant maxspan_2 + 1;\\
&\vdots\\
l_{X,m} &= \max(d_{1,m}, d_{2,m}, \ldots, d_{k,m}) + 1 \leqslant maxspan_m + 1.
\end{aligned}$$

To detect all the minimal extended transaction sets that possibly contain X, the defined cube is slid starting from within the X-applicable sliding range $[(0,\ldots,0), (U_{db_1} - l_{X,1} + 1, \ldots, U_{db_m} - l_{X,m} + 1)]$, forming an *X-applicable cube set* $\mathscr{W}_X$ with the cardinality

$$|\mathscr{W}_X| = (U_{db_1} - l_{X,1} + 2) * (U_{db_2} - l_{X,2} + 2) * \cdots * (U_{db_m} - l_{X,m} + 2) = \prod_{s=1}^{m} (U_{db_s} - l_{X,s} + 2), \tag{3.6}$$

$$\prod_{s=1} (U_{db_s} - maxspan_s + 1) \leqslant |\mathscr{W}_X| = \prod_{s=1}^{m} (U_{db_s} - \max(d_{1,s}, d_{2,s}, \ldots, d_{k,s}) + 1) \leqslant \prod_{s=1}^{m} (U_{db_s} + 1). \tag{3.7}$$

Similarly, let $\mathscr{W}^-_{(X,d_{i,1},d_{i,2},\ldots,d_{i,m})}$ be a subset of $\mathscr{W}_X$, where $(1 \leqslant i \leqslant k)$ and any cube in $\mathscr{W}^-_{(X,d_{i,1},d_{i,2},\ldots,d_{i,m})}$ has a hole at the $(d_{i,1}, d_{i,2}, \ldots, d_{i,m})$-th position. Apparently, $|\mathscr{W}^-_{(X,d_{i,1},d_{i,2},\ldots,d_{i,m})}| \leqslant N_{\text{hole}}$.

Let

$$\mathscr{W}^-_X = \mathscr{W}^-_{(X,d_{1,1},d_{1,2},\cdots,d_{1,m})} \cup \cdots \cup W^-_{(X,d_{k,1},d_{k,2},\ldots,d_{k,m})}. \tag{3.8}$$

Then

$$
\begin{aligned}
|T_X^{\mathrm{p}}| &= |\mathscr{W}_X| - |\mathscr{W}_X^-| = |\mathscr{W}_X| - \left|\mathscr{W}^-_{(X,d_{1,1},d_{1,2},\ldots,d_{1,m})} \cup \cdots \cup W^-_{(X,d_{k,1},d_{k,2},\ldots,d_{k,m})}\right| \\
&\geqslant \prod_{s=1}^{m}(U_{db_s} - maxspan_s + 1) - \left(\left|\mathscr{W}^-_{(X,d_{1,1},d_{1,2},\ldots,d_{1,m})}\right| + \cdots + \left|\mathscr{W}^-_{(X,d_{k,1},d_{k,2},\ldots,d_{k,m})}\right|\right) \\
&\geqslant \prod_{s=1}^{m}(U_{db_s} - maxspan_s + 1) - k * N_{\text{hole}}.
\end{aligned}
$$

For a superset of X, X', which has one more item, $|T_{X'}^{\mathrm{p}}| \geqslant \prod_{s=1}^{m}(U_{db_s} - \ maxspan_s + 1) - (k+1) * N_{\text{hole}}$. Also because $|\mathscr{W}_X| \leqslant \prod_{s=1}^{m}(U_{db_s} + 1)$ in Eq. (3.7), we can derive the following theorem for m-dimensional contexts.

Theorem 3.3. *Given two normalized extended itemsets X and X' $(X \subset X')$ in an m-dimensional context, $sup(X') \geqslant \ minsup$, if and only if $sup(X) \geqslant \alpha * minsup$, where*

$$
\alpha = \frac{\prod_{s=1}^{m}(U_{db_s} - maxspan_s + 1) - (k+1) * N_{\text{hole}}}{\prod_{s=1}^{m}(U_{db_s} + 1)}.
$$

Accordingly, we can derive the downward closure problem solution for mining inter-transactional association rules under m-dimensional contexts. That is, *any subset of a large itemset has support equal to or greater than $\alpha * minsup$*, where $\alpha = \prod_{s=1}^{m}(U_{db_s} - maxspan_s \ +1 - (k+1) * N_{\text{hole}}) / \prod_{s=1}^{m}(U_{db_s} + 1)$.

4. Mining inter-transactional association rules under one-dimensional contexts

Since inter-transactional association rules enrich the expressive power of traditional association rules by incorporating contextual semantics, the discovery of inter-transactional association rules demands more effort than mining traditional intra-transactional association rules. In this section, we investigate the feasibility of mining inter-transactional association rules in the presence of database holes under one-dimensional contexts. A hole-catered mining method is presented by extension of the classical Apriori algorithm [1,2]. The computational complexity incurred is also analyzed. Like traditional association rule mining, the discovery of inter-transactional association rules proceeds in two steps:

1. Find all extended itemsets with supports greater than or equal to a user-specified *minsup* threshold. We call these itemsets *large* extended itemsets. [1]
2. From the large extended itemsets that were discovered at step 1, derive inter-transactional association rules with confidence greater than or equal to a user-specified *minconf* threshold.

In the following, we address each of the two subproblems in detail.

4.1. Discovery of large extended itemsets

Algorithm 4.1 outlines a hole-catered extended Apriori method for finding all large extended itemsets.

[1] For simplicity of explanation, in the following, we use *itemset* and *extended itemset* interchangeably.

Let C_k represent the set of candidate k-itemsets, L_k represent the set of large k-itemsets with support not less than *minsup*, and L'_k represent the set of large k-itemsets with support not less than $actsup = \alpha * minsup$. The algorithm makes multiple passes over the database. Each pass consists of two phases. First, the set of all $(k-1)$-itemsets L'_{k-1}, found in the $(k-1)$th pass, is used to generate the candidate set C_k. The candidate generation procedure ensures that C_k is a superset of L'_k and L_k. The algorithm then scans the database. From every extended transaction located at a certain point in the one-dimensional space, it examines the existence of minimal extended transaction sets that possibly contain candidates and determines which candidates in C_k are contained and increments their counts. At the end of the pass, C_k is examined to check which of the candidates are actually large, yielding L_k. The algorithm terminates when L'_k becomes empty, as no candidate itemsets can be further generated.

4.1.1. Pass 1

Let $\mathscr{I} = \{i_1, i_2, \ldots, i_n\}$ be a set of items in the database. The candidate set C_1 of extended 1-itemsets is generated by attaching each item in $\mathscr{I}$ with all its possible occurring contexts within the range $[0, maxspan]$ (line 2). That is,

$$\begin{aligned} C_1 = \{ & \{\Delta_{(0)}(i_1)\}, \{\Delta_{(1)}(i_1)\}, \cdots, \{\Delta_{(maxspan)}(i_1)\}, \{\Delta_{(0)}(i_2)\}, \{\Delta_{(1)}(i_2)\}, \cdots, \{\Delta_{(maxspan)}(i_2)\}, \ldots, \\ & \{\Delta_{(0)}(i_n)\}, \{\Delta_{(1)}(i_n)\}, \cdots, \{\Delta_{(maxspan)}(i_n)\}\}. \end{aligned}$$

If compared with traditional association rule mining, where $C_1 = \{i_1, i_2, \ldots, i_n\}$ and $|C_1| = |\mathscr{I}|$, it looks like that the database has $(maxspan + 1)$ times the number of items, resulting in $|C_1| = |\mathscr{I}| * (maxspan + 1)$ candidate 1-itemsets to be counted.

After identifying candidates, the algorithm makes the first pass over the database to count minimal extended transaction sets that *contain* each candidate 1-itemset (line 5–6). Starting from each point $\Delta_{(r)}$ in the mining context, it checks whether a minimal extended transaction set contains a candidate $\{\Delta_{(d)}(i)\}$. If there is a transaction t located at $\Delta_{(r+d)}$ and containing item i, the count of $\{\Delta_{(d)}(i)\}$ increases by one.

In addition to counting candidate 1-itemsets as the traditional Apriori algorithm does during pass 1, the first database scan also has the responsibility of counting minimal extended transaction

Algorithm 4.1. A hole-catered extended Apriori algorithm for mining inter-transactional association rules.

Input: a one-dimensional database *DB* containing a set of items $\mathscr{I}$; a contextual range *maxspan* of interest to particular applications.

Output: a set of large itemsets L discovered from *DB*.

Let $actsup = \alpha * minsup$, where $\alpha = (U_{db} - maxspan + 1 - (k+1) * N_{\text{hole}})/(U_{db} + 1)$.

```
k = 1
1       $L_1 = L'_1 = \emptyset$;
2       $C_1 = \{\{\Delta_{(d)}(i)\} \mid (i \in \mathscr{I}) \wedge (0 \leqslant d \leqslant maxspan)\}$;
3       foreach contextual point $\Delta_{(r)} \leqslant U_{db}$ do
4         Count-Possible-Contain-TranSet ($\Delta_{(r)}$);
5         foreach candidate 1-itemset $X = \{\Delta_{(d)}(i)\} \in C_1$ do
```

```
6              **if** $(\Delta_{(r+d)}(t) \in DB) \wedge (i \in t)$ **then** $X$.count++;
7          **endfor**

8          **foreach** candidate $X = \{\Delta_{(d)}(i)\} \in C_1$ **do** // obtain $L_1'$, $L_1$
9              $X$.pcount = Get-Possible-Contain-TranSet-Num($\{d\}$);
10             **if** ($X$.count/$X$.pcount $\geqslant$ *actsup*) **then**
11                 $L_1' \leftarrow X$;
12                 **if** ($X$.count/$X$.pcount $\geqslant$ *minsup*) **then** $L_1 \leftarrow X$;
13             **endif**
14         **endfor**

**k > 1**
15         **for** $(k = 2; L_{k-1}' \neq \phi; k++)$ **do**
16             $C_k =$ E-Apriori-Gen($L_{k-1}'$);
17             **foreach** contextual point $\Delta_{(r)} \leqslant U_{db}$ **do**
18                 **foreach** candidate $k$-itemset $X = \{\Delta_{(d_1)}(i_1), \ldots, \Delta_{(d_k)}(i_k)\} \in C_k$ **do**
19                     **if** $(\Delta_{(r+d_j)}(t_j) \in DB) \wedge (i_j \in t_j)(1 \leqslant j \leqslant k)$ **then** $X$. count++;
20             $L_k = L_k' = \emptyset$;
21             **foreach** candidate $X = \{\Delta_{(d_1)}(i_1), \ldots, \Delta_{(d_k)}(i_k)\} \in C_k$ **do** // obtain $L_k'$, $L_k$
22                 $X$.pcount = Get-Possible-Contain-TranSet-Num ($\{d_1, \ldots, d_k\}$);
23                 **if** ($X$.count/$X$.pcount $\geqslant$ *actsup*) **then**
24                     $L_k' \leftarrow X$;
25                     **if** ($X$.count/$X$.pcount $\geqslant$ *minsup*) **then** $L_k \leftarrow X$;
26                 **endif**
27             **endfor**
28         **endfor**
29         $L = U_{i=1}^{k-1} L_i$.
```

sets with various contextual combinations within $[0, maxspan]$ (line 4). This is required in computing supports of itemsets afterwards. For example, to get the support of itemset $X = \{\Delta_{(0)}(i_1), \Delta_{(0)}(i_2), \Delta_{(2)}(i_3)\}$, we need to know how many minimal extended transaction sets in the database have the relative contextual combination $\{\Delta_{(0)}, \Delta_{(2)}\}$. They constitute the set of minimal extended transaction sets that possibly contain X. Here, all possible combination of *different* contexts should be considered. The total number is

$$\binom{maxspan+1}{1} + \binom{maxspan+1}{2} + \cdots + \binom{maxspan+1}{maxspan+1} = 2^{(maxspan+1)} - 1$$

Example 4.1. When $maxspan = 3$, we have $2^{(3+1)} - 1 = 15$ combinatorial contexts as follows:

$$\begin{aligned}
&\text{size} = 1: \quad \{\Delta_{(0)}\}, \{\Delta_{(1)}\}, \{\Delta_{(2)}\}, \{\Delta_{(3)}\},\\
&\text{size} = 2: \quad \{\Delta_{(0)}, \Delta_{(1)}\}, \{\Delta_{(0)}, \Delta_{(2)}\}, \{\Delta_{(0)}, \Delta_{(3)}\}, \{\Delta_{(1)}, \Delta_{(2)}\}, \{\Delta_{(1)}, \Delta_{(3)}\}, \{\Delta_{(2)}, \Delta_{(3)}\},\\
&\text{size} = 3: \quad \{\Delta_{(0)}, \Delta_{(1)}, \Delta_{(2)}\}, \{\Delta_{(0)}, \Delta_{(1)}, \Delta_{(3)}\}, \{\Delta_{(0)}, \Delta_{(2)}, \Delta_{(3)}\}, \{\Delta_{(1)}, \Delta_{(2)}, \Delta_{(3)}\},\\
&\text{size} = 4: \quad \{\Delta_{(0)}, \Delta_{(1)}, \Delta_{(2)}, \Delta_{(3)}\}.
\end{aligned}$$

With $\Delta_{(r)}$ as the reference point, function *Count-Possible-Contain-TranSet* $(\Delta_{(r)})$ checks a list of consecutive positions $\Delta_{(r)}, \Delta_{(r+1)}, \ldots, \Delta_{(r+maxspan)}$, and adds one to the count values of those context combinations consisting of all transaction-filled positions. Referring to Example 4.1, suppose from a contextual point $\Delta_{(r)}$, three transactions locate at $\Delta_{(r)}, \Delta_{(r+1)}, \Delta_{(r+3)}$ but not $\Delta_{(r+2)}$, the counts of the following context combinations will be increased by one:

$$\{\Delta_{(0)}\}, \{\Delta_{(1)}\}, \{\Delta_{(3)}\}, \{\Delta_{(0)}, \Delta_{(1)}\}, \{\Delta_{(0)}, \Delta_{(3)}\}, \{\Delta_{(1)}, \Delta_{(3)}\}, \{\Delta_{(0)}, \Delta_{(1)}, \Delta_{(3)}\}.$$

Note that the final count value of a context combination obtained by checking all contextual points is in fact equal to the total number of minimal extended transaction sets that possibly contain itemsets of the same context combination. For instance, the total count 100 of $\{\Delta_{(0)}, \Delta_{(2)}\}$ indicates that there are 100 minimal extended transaction sets possibly containing 2-itemsets of the form $\{\Delta_{(0)}(x_1), \Delta_{(2)}(x_2)\}$.

Correspondingly, the function *Get-Possible-Contain-TranSet-Num*(*S*) takes a context combination in a set representation S as input and returns its count value (line 9). The support of an itemset X can thus be calculated by dividing *X.pcount* into *X.count*, where *X.pcount* is equal to the count of *X*'s context combination.

The first scan of the database will deliver the large set L_1 and L'_1 (line 10–13).

4.1.1.1. Computational complexity ($k=1$). The computation at pass 1 includes (1) generating candidate 1-itemsets, (2) counting context combinations, and (3) counting candidate 1-itemsets.

The time complexity of (1) is $\mathrm{O}((maxspan + 1) * |\mathscr{I}|)$, and the complexity of (2) and (3) is linear in the number of contextual points, i.e., the size of the mining space U_{db}. In the worst case, from each contextual point, all the following positions within *maxspan* including itself accommodate transactions, making the counts of all the $2^{(maxspan+1)} - 1$ context combinations increased by one. Thus, counting context combinations can be done in time $O((2^{(maxspan+1)} - 1) * U_{db})$. Similarly, to count candidate 1-itemsets, from each point, at most $(maxspan + 1)$ consecutive transactions are scanned, leading to total search time $\mathrm{O}((maxspan + 1) * l_{\mathrm{tran}} * U_{db})$, where l_{tran} is the average length of existing transactions in the database. Let $a = maxspan + 1$, the total computational complexity at Pass 1 is

$$\begin{aligned} Time_1 &= T_{\mathrm{generate}-C_1} + T_{\mathrm{count-context}} + T_{\mathrm{scanDB-count}-C_1} \\ &= \mathrm{O}(a * |\mathscr{I}|) + \mathrm{O}((2^a - 1) * U_{db}) + \mathrm{O}(a * l_{\mathrm{tran}} * U_{db}). \end{aligned}$$

4.1.2. Pass $k > 1$

Given L'_{k-1}, the candidate generation function *E-Apriori-Gen* (L'_{k-1}) returns a superset of L'_k and L_k (line 16). The procedure has two parts. In the *join* phase, two extended $(k-1)$-itemsets $X, X' \in L'_{k-1}$, which have all but one extended item in common, are joined to derive a candidate k-itemset. Let $X = \{\Delta_{(d_1)}(x_1), \ldots, \Delta_{(d_{k-2})}(x_{k-2}), \Delta_{(d_{k-1})}(x_{k-1})\}$ and $X' = \{\Delta_{(d_1)}(x_1), \ldots, \Delta_{d_{(k-2)}}(x_{k-2}), \Delta_{(d_k)}(x_k)\}$[2], where $\forall i \ (1 \leqslant i \leqslant k)(x_i \in \mathscr{I}) \wedge (0 \leqslant d_i \leqslant U_{db})$. We can generate a candidate k-itemset $X'' = \{\Delta_{(di)}(x_1), \ldots, \Delta_{(d_{k-2})}(x_{k-2}), \Delta_{(d_{k-1})}, (x_{k-1})\Delta_{(dx)}(x_k)\}$. All k-itemsets obtained in the join phase comprise a set

$$C_k^{\mathrm{join}} = \left\{ X \cup X' \,|\, (X, X' \in L'_{k-1}) \wedge (\text{The first } (k-2) \text{ extended items in } X \text{ and } X' \text{ are the same}) \right\}.$$

Next, in the *prune* phase, all those extended k-itemset(s) in C_k^{join} which have some $(k-1)$-subset(s) not in L'_{k-1} are discarded, leading to the candidate set C_k

$$C_k = \left\{X'' \,|\, (X'' \in C_k^{\text{join}}) \wedge (\forall Y \subset X''(|Y| = k-1) \rightarrow (Y \in L'_{k-1}))\right\}.$$

The candidate k-itemsets generated are counted by making one pass over the database (lines 17–19), from which L_k and L'_k are derived (lines 20–27). This candidate generation and counting process terminates after some iteration when $L'_{k-1} = \emptyset$ (line 15). Here, the total number of database passes is bound by the value of $|\mathscr{I}| * (maxspan + 1)$, which is the maximal length of extended itemsets in the database with all items in $\mathscr{I}$ at all possible contextual points in the range $[0, maxspan]$, i.e., $\left\{\Delta_{(0)}(i_1), \ldots, \Delta_{(0)}(i_n), \Delta_{(1)}(i_1), \ldots, \Delta_{(1)}(i_n), \ldots, \Delta_{(maxspan)}(i_1), \ldots, \Delta_{maxspan}(i_n)\right\}$

Example 4.2. Let $L'_3 = \{\{\Delta_{(0)}(a), \Delta_{(0)}(b), \Delta_{(0)}(c)\}, \quad \{\Delta_{(0)}(a), \Delta_{(0)}(b), \Delta_{(1)}(d)\}, \quad \{\Delta_{(0)}(a), \Delta_{(0)}(c), \Delta_{(1)}(d)\}, \quad \{\Delta_{(0)}(a), \Delta_{(0)}(c), \Delta_{(2)}(d)\}, \quad \{\Delta_{(0)}(b), \Delta_{(0)}(c), \Delta_{(1)}(d)\},\}$. After the join step, $C_4^{\text{join}} = \{\{\Delta_{(0)}(a), \Delta_{(0)}(b), \Delta_{(0)}(c), \Delta_{(1)}(d)\}, \quad \{\Delta_{(0)}(a), \Delta_{(0)}(c), \Delta_{(1)}(d), \Delta_{(2)}(d)\}\}$. The prune step will delete from C_4^{join} the itemset $\{\Delta_{(0)}(a), \Delta_{(0)}(c), \Delta_{(1)}(d), \Delta_{(2)}(d)\}$ because its subset $\{\Delta_{(0)}(a), \Delta_{(1)}(d), \Delta_{(2)}(d)\}$ is not in L'_3. We will then be left with only $\{\Delta_{(0)}(a), \Delta_{(0)}(b), \Delta_{(0)}(c), \Delta_{(1)}(d)\}$ in C_4.

4.1.2.1. Computational complexity ($k > 1$). We analyze the complexity at pass $k > 1$ in terms of $|L'_{k-1}|$, $|C_k^{\text{join}}|$ and U_{db}. The generation of C_k^{join} from L'_{k-1} is performed in time $O(|L'_{k-1}|^{\,2})$. For each k-itemset in C_k^{join}, totally $\binom{k}{k-1} - 2 (= k-2)$ $(k-1)$-subsets are checked for their existence in L'_{k-1}, requiring $O((k-2) * |C_k^{\text{join}}| * |L'_{k-1}|)$ time. Further,

$$\max(|C_k^{\text{join}}|) = \binom{|L'_{k-1}|}{2} = |L'_{k-1}| * (|L'_{k-1}| - 1)/2.$$

Similarly, the time spent at each pass in scanning the database and counting candidates can be estimated as $O(a * l_{\text{tran}} * U_{db})$ Thereby, the total time involved at pass $k > 1$ is

$$\begin{aligned} Time_k &= T_{\text{join}-L'_{k-1}} + T_{\text{generate}-C_k} + T_{\text{scanDB}-\text{count}-C_k} \\ &= O(|L'_{k-1}|^2) + O((k-2) * |L'_{k-1}|^2 * (|L'_{k-1}| - 1)/2) + O(a * l_{\text{tran}} * U_{db}) \end{aligned}$$

4.1.3. Correctness

The key to correctness of the above algorithm lies in the following lemma.

Lemma 4.1. *Any large k-itemset in L_k is included in C_k, i.e., $L_k \subseteq C_k$.*

Proof. We prove the lemma by induction.

(1) *When* $k = 1, L_1 \subseteq C_1$, *since* C_1 *includes all possible* 1-*itemsets which could be potentially large.*

(2) *Assume the lemma holds when* $k = s$.

Let X be a large $(s+1)$-itemset in L_{s+1} $(X \in L_{s+1})$, where $X = \{\Delta_{(d_1)}(i_1), \ldots, \Delta_{(d_{s-1})}(i_{s-1}), \Delta_{(d_s)}(i_s), \Delta_{(d_{s+1})}(i_{s+1})\}$ without loss of generality. Let $Y \subset X, Y' \subset X$, where

$$Y = \{\Delta_{(d_1)}(i_1), \ldots, \Delta_{(d_{s-1})}(i_{s-1}), \Delta_{(d_s)}(i_s)\}, Y' = \{\Delta_{(d_1)}(i_1), \ldots, \Delta_{(d_{s-1})}(i_{s-1}), \Delta_{(d_{s+1})}(i_{s+1})\}.$$

[2] The extended items in an itemset are listed in an ascending order, where $\Delta_{(d_1)}(i_1) < \Delta_{(d_2)}(i_2)$ if and only if either $(d_1 < d_2)$ or $(d_1 = d_2 \wedge i_1 < i_2)$.

According to Theorem 3.2, any subset of X should be in L'_s. Thus, $Y \in L'_s$ and $Y' \in L'_s$. In the join phase, the algorithm joins Y and Y' to get $X \in C^{\text{join}}_{s+1}$. Thus, after the join phase, $L_{s+1} \subseteq C^{\text{join}}_{s+1}$. By similar reasoning, the prune step, where all itemsets whose s-subsets are not in L'_s are deleted from C^{join}_{s+1}, also does not discard X form C^{join}_{s+1}, leaving $X \in C_{s+1}$. Therefore, $L_{s+1} \subseteq C_{s+1}$.
Based on (1) and (2), the lemma is proven.

4.2. Generation of inter-transactional association rules

Using sets of large normalized extended itemsets, we can find the desired inter-transactional association rules according to the *minconf* measurement. The basic procedure for generating inter-transactional association rules is similar to that for generating classical association rules [2]: From a large normalized extended itemset X and any $Y \subset X$, if $sup(X)/sup(X - Y) \geqslant minconf$, then the rule $X - Y \Rightarrow Y$ is a desired inter-transactional association rule. For prediction purpose, all the items in the rule precedence $(X - Y)$ appear at lower contexts than the items in the rule consequence Y.

5. Experimental results

Two sets of experiments are conducted to examine the applicability of inter-transactional association rules to weather prediction.

5.1. Test with single-station meteorological data

We first run the algorithm against single-station meteorological data collected by the Hong Kong Observatory headquarters, which takes observations including *wind direction*, *wind speed*, *temperature*, *relative humidity*, *rainfall*, and *mean sea level pressure*, etc. every 6 h each day. In some records, certain atmospheric observations such as relative humidity, etc. are missing. Since the context constituted by the *time* dimension is valid for the whole set of meteorological records (transactions), we fill in these empty fields by averaging their nearby values. In this way, the data to be mined contain no missing fields, and also no database holes (i.e., meaningless contexts) exist in the mining space. Essentially, there is one dimension in this case, namely, time.

After pre-processing the data set, we apply the algorithm described in Section 4 to discover inter-transactional association rules from the 1995 meteorological records, and then examine their prediction accuracy using 1996 meteorological data from the same area in Hong Kong. Considering seasonal changes of weather, we extract records from May to October for our experiments, and there are therefore totally 736 records ($\text{total\_days} * \text{record\_num\_per\_day} = (31 + 30 + 31 + 31 + 30 + 31) * 4 = 736$) for each year. These raw data sets, containing continuous data, are further converted into appropriate formats with which the algorithms can work. Table 2 lists 32 items after the data transformation. Each record has six meteorological elements (items). The interval of every two consecutive records is 6 h. We set $\text{maxspan} = 11$ in order to detect the association rules for a 3-day horizon (i.e., $(11 + 1)/4 = 3$).

At *support* = 45% and *confidence* = 92%, from the 1995 meteorological data, we found only 1 classical association rule – “if the humidity is medium wet, then there is no rain at the same time” (which is quite obvious), but 580 inter-transactional association rules. Note that the number of inter-transactional association rules returned depends on the *maxspan* parameter setting. Table 3

Table 2
Items for single-station meteorological data

Meteorological element	ItemId	Value range	Meaning
Wind direction (°)	1	$(0, 45]$	North east
	2	$(45, 90]$	East
	3	$(90, 135]$	South east
	4	$(135, 180]$	South
	5	$(180, 225]$	South west
	6	$(225, 270]$	West
	7	$(270, 315]$	North west
	8	$(315, 360]$	North
Wind speed (km/h)	9	$(2, 12]$	Light
	10	$(12, 30]$	Moderate
	11	$(30, 40]$	Fresh
	12	$(40, 62]$	Strong
Rainfall (cm)	13	$[0, 0.05)$	No rain
	14	$[0.05, 0.1)$	Trace
	15	$[0.1, 4.9)$	Light
	16	$[4.9, 25.0)$	Moderate
	17	$[25.0, 100.0)$	Heavy
Relative humidity (%)	18	$[0, 50]$	Very dry
	19	$(50, 70]$	Dry
	20	$(70, 90]$	Medium wet
	21	$(90, 100]$	Wet
Temperature (°C)	22	$[0, 10]$	Very cold
	23	$(10, 16]$	Cold
	24	$(16, 22]$	Mild
	25	$(22, 28]$	Warm
	26	$(28, 50]$	Hot
Mean sea level pressure (hPa)	27	$(0, 9950]$	Very low
	28	$(9950, 10\,000]$	Low
	29	$(10\,000, 10\,050]$	Moderate
	30	$(10\,050, 10\,100]$	Slightly high
	31	$(10\,100, 10\,150]$	High
	32	$(10\,150, 20\,000]$	Very high

Table 3
Some significant inter-transactional association rules found from single-station meteorological data

Inter-transactional association rules	Sup.	Conf.	Pred. rate
$\Delta_{(0)}(2), \Delta_{(3)}(13) \Rightarrow \Delta_{(4)}(13)$ ("If there is an *east* wind direction and *no rain* in 18 h, then there will also be a *no rain* in 24 h")	46%	92%	90.0%
$\Delta_{(0)}(2), \Delta_{(0)}(25), \Delta_{(2)}(25) \Rightarrow \Delta_{(3)}(25)$ ("If it is currently *warm* with an *east* wind direction, and still *warm* 12 h later, then it will be continuously *warm* until 18 h later")	45%	95%	91.8%

lists some significant inter-transactional association rules found from the single-station meteorological data. We measure their predictive capabilities using the 1996 meteorological data recorded by the same station through *Prediction-Rate* $(X \Rightarrow Y) = sup(X \cup Y)/sup(X)$, which can achieve more than 90% prediction rate. From the test results, we find that, with inter-transactional association rules, more comprehensive and interesting knowledge can be discovered from the databases.

5.2. Test with multi-station meteorological data

Data used in the second experiment are from multiple atmospheric stations, located in the *west*, *north*, *south*, *east*, *middle*, and *middle-north* of Hong Kong, respectively. We combine the meteorological observations, except *rainfall*, taken by the above six stations at the same time into one record, plus their average rainfall to represent the overall rainfall behavior in Hong Kong. Missing data and discretization of continuous meteorological values are handled in the same way as in the first experiment, except for the discretization of *mean sea level pressure* element which follows a different conversion (as shown in Table 4) due to its larger value span for multi-station data than for single-station data.

The training records in this test range from January 1993 to June 1996, and the testing records range from July 1996 to December 1997, both covering all the seasons of a year. Thus, totally we have 5108 training records and 2192 testing records. Similarly, the interval of every two consecutive records is 6 h.

At *support* = 60% and *confidence* = 99%, we found 43 intra-transactional association rules and 1552 inter-transactional association rules. Compared to the ratio 1:580 obtained from the single-station meteorological data, the second experiment generates more inter-transactional association rules. This is because the multi-station meteorological data set contains more items than the former single-station data set by combining atmospheric elements from *six* different stations into each record. As a result, more candidate itemsets are generated and tested in the second experiment, and thus more large itemsets and association rules that satisfy the support and confidence requirements are returned. Table 5 shows some association rules found from the multi-station meteorological data. Note that most of the rules discovered from the multi-station data have higher prediction rate (more than 99%) than those discovered from the single-station data. This is expected since the number of training records is much larger than that in the first experiment.

Table 4
Discretization of mean sea level pressure for multi-station meteorological data

Meteorological element	Value range	Meaning
Mean sea level pressure (hPa)	(0, 9950]	Very low
	(9950, 14000]	Low
	(14000, 18000]	Moderate
	(18000, 22000]	Slightly high
	(22000, 26000]	High
	(26000, −)	Very high

Table 5
Some inter-transactional association rules detected from multi-station meteorological data

Extended association rules	Sup.	Conf.	Pred. rate
"If it is *hot* in the *south* and *no rain* at the moment, then the mean sea level pressure in the *middle* will be *low* within the next 24 h"	60%	99.7%	99.7%
"If it is *medium wet* in the *middle* and the mean sea level pressure in the *south* is *low* at the same time, then the *south* keeps the *low* mean sea level pressure for 12 h"	60%	99.4%	99.6%

6. Discussions

Along with interesting associations discovered, some inherent deficiencies in the current support/confidence-based association mining framework are also revealed during our weather study. In this section, we examine the problem with statistical measures in association rule detection. Several strategies are proposed to address this problem. Further extensions of association rules in providing multi-dimensional predictive capabilities are also discussed.

6.1. The problem with statistical measures in mining association rules

As described in Section 4, the discovery of association rules is divided into two phases based on two statistical measurements – support threshold *minsup* and confidence threshold *minconf*. In the first expensive phase, the database is searched for all large itemsets whose transaction supports are not less than *minsup*. After obtaining all large itemsets, the second phase generates from each large itemset association rules whose confidence is equal to or greater than *minconf*.

Under such a framework, frequently happening correlations which hold for numerous transactions can be easily and efficiently discovered from databases. However, for *infrequent* but *significant* cases which represent relatively small number of objects but have high confidence, the statistical measures are inadequate in eliciting association rules. For example, with a proper support threshold, we can detect large itemsets involving *no rain* within a reasonable time, since most of the time it has sunshine in Hong Kong, leaving plenty of *no rain* records in the database. In contrast, there are only a small number of records reporting *heavy rain* in the Hong Kong area. If we want to know which factors are related with *heavy rain*, one straightforward method is to set the support threshold low enough to be sure those combinations of items including *heavy rain* can pass the support threshold as large itemsets and enter the second rule-generation phase. However, a low support requirement will inevitably lead to too many candidate itemsets being counted and many unrelated itemsets being returned as larger itemsets, making the mining time much longer and intolerable. Another severe problem with such a low support threshold method is that there could be too many rules being generated - it could be much more than the number of database records.

In order to detect association relationship from infrequent representatives, we propose the following two strategies.

6.1.1. Strategy I

Instead of utilizing a *support threshold – minsup* to simply delimit large and small itemsets, we can make use of a *support range* with lower and upper bounds $[sup_l, sup_u]$ to filter out from large itemsets *weakly* and *strongly* supported itemsets further. Weakly supported itemsets are the itemsets whose supports fall into the range of $[sup_l, sup_u]$, and strongly supported itemsets have supports greater than sup_u. Note that although weakly supported itemsets represent infrequent objects, their transaction supports should not be too low (above the lower bound sup_l), since enough evidence is still a necessity to ensure the reliability of corresponding association patterns derived later. By restricting the support range of itemsets, we can eliminate those strongly supported itemsets from the resulting large itemsets, and return to users a small number of association rules which have high confidence.

The problem with this strategy is that the monotonic property of support range of itemsets cannot be guaranteed. That is, if an itemset has a support within $[sup_l, sup_u]$, its subset may not stay within the same support range. For example, given a support range requirement [20%, 40%], assume that $sup(\{a,b\}) = 80\%, sup(\{a,c\}) = 75\%$, and $sup(\{a,b,c\}) = 30\%$. Although itemset $\{a,b,c\}$ meets the support requirement [20%, 40%], its subsets $\{a,b\}$ and $\{a,c\}$ do not. Therefore, when we generate candidate 3-itemsets from two 2-itemsets, we cannot prune out $\{a,b\}$ and $\{a,c\}$ as they both together can derive $\{a,b,c\}$, which is a desirable input itemset to the final rule generation.

Since we cannot reduce the number of candidate itemsets due to the inapplicability of monotonicity, at each pass made over the database, the mining process needs to count lots of candidates, yielding a very poor mining performance. Note that keeping the monotonicity is the basis of a large number of efficient intra-transactional association rule mining algorithms presented before.

6.1.2. Strategy II

In order to cater for infrequent situations without degrading mining efficiency we propose a more focused mining strategy. The idea is to remove those unrelated and uninteresting items from data sets before starting the mining process. For instance, if a user wants to investigate the correlative relationship of *heavy rain*, we can eliminate its mutually exclusive items like *no rain* from all the data records where it appears. In this way, we can reduce not only the database size, but also the number and length of candidate itemsets since unrelated and frequently-occurring items will not participate in any itemset. Hence, the effort spent in scanning the database and counting candidates' supports can be greatly decreased, leading to a much improved mining performance. Also, by working around a few interesting items, the mining can be focused and the cost incurred is proportionate to what the user wants and gets.

6.2. Enhancing multi-dimensional predictive capabilities of association rules

The performance study in this paper focused on inter-transactional association mining along one dimension-time. Although the multi-station data being studied possess the property of two dimensions-time and space, the domain for the spatial dimension contains only six values (i.e., *west*, *north*, *south*, *east*, *middle* and *middle-north*), which is far less than that for the time dimension. Hence, for the sake of mining efficiency, we combine atmospheric records from different stations taken at the same time, and re-organize the original data set along one dimension. Nevertheless, when we have large domains for both dimensions, such a simple combination is not feasible, and we need to generalize the method described in this paper to mine inter-transactional associations along multiple dimensions.

Similarly, we can detect multi-dimensional inter-transactional association rules based on large itemsets. The candidate itemsets under multi-dimensional contexts can be generated in the same way as those under single-dimension. However, the candidate counting effort increases dramatically with the number of dimensions. For example, to compute the support of a normalized extended itemset $I_{\text{ne}} = \{\Delta_{(0,0)}(a), \Delta_{(0,1)}(a), \Delta_{(1,2)}(b)\}$ under a two-dimensional context as shown in Fig. 1 where $(U_{db_x} = 3)$ and $(U_{db_y} = 4)$, from each possible reference point (x, y) $(0 \leqslant x \leqslant U_{db_x}$ and $0 \leqslant y \leqslant U_{db_y})$ in the space, we need to check 3 surrounding positions $(x, y), (x, y+1)$ and $(x+1, y+2)$. If no hole exists, we then check whether the corresponding transactions contain I_{ne}; Otherwise, we stop and move to the next reference point, from which a new round of checking starts. In total, $(U_{db_x} + 1) * (U_{db_y} + 1) = 4 * 5 = 20$ reference points are probed during one scan of the database, leading to a huge search space as compared to the case of one-dimension.

To reduce the mining cost, a constraint-based strategy can be enforced in mining multi-dimensional inter-transactional association rules. This is motivated by the fact that users usually have certain interesting contexts and objectives in mind. For example, a user may want to know how the rain today affects the weather for 1 day in the region located 2 unit distances away. In this case, the rule contexts which are worth investigation include $\Delta_{(0,0)}$ and $\Delta_{(4,2)}$, where its x-coordinate denoting time (1 unit interval represents 6 h) and y-coordinate denoting region. In general, we can use a boolean expression $\mathscr{C}^m(x_1, x_2, \ldots, x_m)$ to specify the conditions that interesting m-dimensional contexts shall satisfy. Without loss of generality, we can assume that $\mathscr{C}^m$ is in a disjunctive normal form $D_1 \vee D_2 \vee \cdots \vee D_s$, where each D_i is of the form $d_{i_1} \wedge d_{i_2} \wedge \cdots \wedge d_{i_t}$. Each element d_{i_j} can be either a condition on individual dimension (e.g, $x_1 \leqslant 4, x_2 = 0$), or a condition involving several dimensions (e.g., $x_1 = x_2, x_1 = x_2 + 2$). For instance, the contexts that interest the above weather investigator can be expressed as $\mathscr{C}^2(x_1, x_2) : (x_1 = 0 \wedge x_2 = 0) \vee (x_1 = 4 \wedge x_2 = 2)$. The expression $\mathscr{C}^m(x_1, x_2, \ldots, x_m) : (x_1 = x_2 = \cdots = x_m) \wedge (0 \leqslant x_1 \leqslant 4)$ requires all the dimensional values to change at the same pace while the maximal scope is limited to 4.

By focusing on interesting portions, we can reduce the database search space, hence improving the mining performance. More detailed discussion regarding template-guided constrained association rule mining can be found in our other paper [9].

7. Conclusion

In this paper, we extend the traditional association rule framework from intra-transactional associations to inter-transactional associations to enforce its predictive capability in prediction such as weather forecasting. A formal definition of inter-transactional association rules and related measurements is given. We investigate the property, theoretical foundations, multi-dimensional mining contexts and performance issues in mining such inter-transactional association rules, especially when the databases to be mined contain various holes in multi-dimensional contexts. The proposed hole-catered extended Apriori algorithm is applied to both single-station and multi-station meteorological data sets obtained from the Hong Kong Observatory headquarters. Our test results show that with the extended inter-transactional association rules, more comprehensive and interesting association relationships can be found. Further extensions of the method and the deficiencies of association rules for prediction are also discussed in the paper.

We are currently conducting research into the development of efficient mining algorithms under 2- and *m*-dimensional contexts. Parallel and distributed mining inter-transactional association rules are also an interesting area of work we plan to explore.

Acknowledgements

The work is supported by the Department of Computing, Hong Kong Polytechnic University, under the project titled “Data Mining Techniques with Applications to Business Problems”.

The authors would like to thank anonymous reviewers for many valuable comments and suggestions that have improved the quality of this paper.

References

[1] R. Agrawal, T. Imielinski, A. Swami, Mining association rules between sets of items in large databases, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington DC, USA, May 1993, pp. 207–216.

[2] R. Agrawal, R. Srikant, Fast algorithms for mining association rules, in: Proceedings of the 20th International Conference on Very Large Data Bases, Santiago, Chile, September 1994, pp. 478–499.

[3] R. Agrawal, J.C Shafer, Parallel mining of association rules: Design, implementation and experience, Technical Report IBM Research Report RJ 10004, 1996.

[4] S. Brin, R. Motwani, C. Silverstein, Beyond market basket: generalizing association rules to correlations, in: Proceedings of the ACM SIGMOD International Conference in Management of Data, Tucson, Arizona, USA, June 1997, pp. 265–276.

[5] E. Baralis, G. Psaila, Designing templates for mining association rules, J. Intell. Inf. Syst. 9 (1) (1997) 7–32.

[6] D. Cheung, J. Han, V. Ng, C.Y. Wong, Maintenence of discovered association rules in large databases: an incremental updating technique, in: Proceedings of the International Conference on Data Engineering, New Orleans, Louisiana, USA, February 1996, pp. 106–114.

[7] D.W. Cheung, V.T. Ng, A.W. Fu, Y.J. Fu, Efficient mining of association rules in distributed databases, IEEE Trans. Knowledge Data Eng. 8 (6) (1996) 911–922.

[8] H. Dai, Machine learning of weather forecasting rules from large meteorological data bases, Adv. Atmospheric Sci. 13 (4) (1996) 471–488.

[9] L. Feng, H. Lu, J. Yu, J. Han, Mining inter-transaction association rules with templates, in: Proceedings ACM CIKM International Conference Information and Knowledge Management, USA, November 1999, pp. 225–233.

[10] T. Fukuda, Y. Morimoto, S. Morishita, T. Tokuyama, Data mining using two-dimensional optimized association rules: Schema, algorithms, and visualization, in: Proceedings ACM SIGMOD International Conference Management of Data, Montreal, Canada, June 1996, pp. 13–23.

[11] M. Fang, N. Shivakumar, H. Garcia-Molina, R. Motwani, J.D. Ullman, Computing iceberg queries efficiently, in: Proceedings of the 24th International Conference on Very Large Data Bases, New York, USA, August 1998, pp. 299–310.

[12] J. Han, Y. Fu, Discovery of multiple-level association rules from large databases, in: Proceedings of the 21st International Conference on Very Large Data Bases, Zurich, Switzerland, September 1995, pp. 420–431.

[13] J. Han, Y. Fu, Meta-rule-guided mining of association rules in relational databases, in: Proceedings of the First International Workshop on Integration of Knowledge Discovery with Deductive and Object-Oriented Database, Singapore, December 1995, pp. 39–46.

[14] E.-H. Han, G. Karypis, V. Kumar, Scalable parallel data mining for association rules, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, Tucson, Arizona, USA, June 1997, pp. 277–288.

[15] M. Kamber, J. Han, J.Y. Chiang, Metarule-guided mining of multi-dimensional association rules using data cubes, in: Proceedings of the International Conference on Knowledge Discovery and Data Mining, California, USA, August 1997, pp. 207–210.

[16] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, A.I. Verkamo, Finding interesting rules from large sets of discovered association rules, in: Proceedings of the Third International Conference on Information and Knowledge Management, Gaithersburg, Maryland, November 1994, pp. 401–408.

[17] H. Lu, L. Feng, J. Han, Beyond intra-transactional association analysis: Mining multi-dimensional inter-transaction association rules, ACM Trans. Inf. Syst. 18 (4) (2000) 423–454.

[18] H. Lu, J. Han, L. Feng, Stock movement prediction and *n*-dimensional inter-transaction association rules, in: Proceedings of the ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, Seattle, Washington, Jume 1998, pp. 12:1–12:7.

[19] R. Lee, J. Liu, An automatic satellite interpretation of tropical cyclone patterns using elastic graph dynamic link model, Int. J. Pattern Recognition Artificial Intell. 13 (8) (1999) 1251–1270.

[20] B. Li, J. Liu, H. Dai, Forecasting from low quality data with applications in weather forecasting, Int. J. Comput. Inf. 22 (3) (1998) 351–358.

[21] B. Lent, A. Swami, J. Widom, Clustering association rules, in: Proceedings of the International Conference on Data Engineering, Birmingham, England, April 1997, pp. 220–231.

[22] J. Liu, L. Wong, A case study for Hong Kong weather forecasting, in: Proceedings of the International Conference on Neural Information Processing, Hong Kong, September 1996.

[23] H. Mannila, Data mining and hierarchical models, Finnish Statistical Society, November 1997.

[24] R. Meo, G. Psaila, S. Ceri, A new SQL-like operator for mining association rules, in: Proceedings of the 22nd International Conference on Very Large Data Bases, Mumbai, India, September 1996, pp. 122–133.

[25] R.J. Miller, Y. Yang, Association rules over interval data, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, Tucson, Arizona, USA, June 1997, pp. 452–461.

[26] R. Ng, L.V.S. Lakshmanan, J. Han, A. Pang, Exploratory mining and pruning optimizations of constrained association rules, in: Proc. of the ACM SIGMOD International Conference on Management of Data, Seattle, Washington, June 1998, pp. 13–24.

[27] B. Özden, A. Ramaswamy, A. Silberschatz, Cyclic association rules, in: Proceedings of the International Conference on Data Engineering, 1998.

[28] J.-S. Park, M.-S. Chen, P.S. Yu, An effective hash based algorithm for mining association rules, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, San Jose, CA, May 1995, pp. 175–186.

[29] J.-S. Park, M.-S. Chen, P.S. Yu, Mining association rules with adjustable accuracy, Technical Report IBM Research Report, 1995.

[30] S. Ramaswamy, S. Mahajan, A. Silberschatz, On the discovery of interesting patterns in association rules, in: Proceedings of the 24th International Conference on Very Large Data Bases, New York, USA, August 1998, pp. 368–379.

[31] R. Srikant, R. Agrawal, Mining generalized association rules, in: Proceedings of the 21st International Conference Very Large Data Bases, Zurich, Switzerland, September 1995, pp. 409–419.

[32] R. Srikant, R. Agrawal, Mining quantitative association rules in large relational tables, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, Montreal, Canada, June 1996, pp. 1–12.

[33] C. Silverstein, S. Brin, R. Motwani, J.D. Ullman, Scalable techniques for mining casual structures, in: Proceedings of the 24th International Conference on Very Large Data Bases, New York, USA, August 1998, pp. 594–605.

[34] A. Savasere, E. Omiecinski, S. Navathe, An efficient algorithm for mining association rules in large databases, in: Proceedings of the 21st International Conference on Very Large Data Bases, Zurich, Switzerland, September 1995, pp. 432–443.

[35] R. Srikant, Q. Vu, R. Agrawal, Mining association rules with item constraints, in: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, Newport Beach, California, August 1997, pp. 67–73.

[36] K.H. Tung, H. Lu, J. Han, L. Feng, Breaking the barrier of transactions: Mining inter-transaction association rules, in: Proceedings ACM SIGKDD International Conference Knowledge Discovery and Data Mining, USA, August 1999, pp. 297–301.

[37] H. Toivonen, Sampling large databases for association rules, in: Proceedings of the 22nd Conference on Very Large Data Bases, Mumbai, India, September 1996, pp. 134–145.

[38] D. Tsur, J.D. Ullman, S. Abitboul, C. Clifton, R. Motwani, S. Nestorov, Query flocks: a generalization of association-rule mining, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, Seattle, Washington, June 1998, pp. 1–12.

[39] M.J. Zaki, S. Parthasarathy, M. Ogihara, W. Li, New algorithms for fast discovery of association rules, in: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, Newport Beach, CA, USA, August 1997, pp. 283–286.

Ling Feng got the B.Sc. and Ph.D. degrees in Computer Science from Huazhong University of Science and Technology, China in 1990 and 1995 with distinct graduate honor. She was a postdoctoral fellow in the Department of Information Systems and Computer Science at National University of Singapore from 1995 to 1997, and a lecturer in the Department of Computing at Hong Kong Polytechnic University from 1997 to 1999. She is currently an assistant professor in the InfoLab in the Department of Information Management at Tilburg University in the Netherlands. Her research interests include data mining and its applications, data warehouses, digital libraries, distributed object-oriented database management systems, knowledge-based information systems, integration of database and Web technologies. She has been engaged in 12 different international and national projects as principal investigator, project leader, chief designer and developer, and published over 40 research papers in international and national conferences and journals. She is a member of ACM, IEEE and AIS.

T. Dillon is a Professor of Computing at Hong Kong Polytechnic University and the professor of Computer Science and Computer Engineering at La Trobe University in Melbourne, Australia, and Director of the Applied Computing Research Institute. His research interests include Data Mining Internet Computing, e-commerce, hybrid neuro-symbolic systems, neural nets, software engineering, database systems and computer networks. He has also worked with industry and commerce in developing systems in telecommunications, health care systems; e-commerce, logistics, power systems, banking and finance. He is editor-in-chief of the *International Journal of Computer Systems Science and Engineering* and the *International Journal of Engineering Intelligent Systems*, as well as co-editor of the *Journal of Electric Power and Energy Systems*. He is an Advisory Editor of the IEEE Transactions on Neural Networks in the USA. He is on the advisory editorial board of Applied Intelligence published by Kluwer in USA and Computer Communications published by Elsevier in the UK. He has published over 400 papers in international and national journals and conference and has written four books and edited five other books. He is fellow of the Institution of Electrical and Electronic Engineers (USA), fellow of the Institution of Engineers (Australia), fellow of the Australian Computer Society.

James Liu received the B.Sc. (Hons) and M.Phil. degrees in mathematics and computational modeling from Murdoch University, Australia, in 1982 and 1987, respectively. He received his Ph.D. degree in Artificial Intelligence from La Trobe University, Australia, in 1992. While working on his degree, he worked as a computer scientist at Defence Signal Directorate in Australia from 1988 till 1990. He joined the Aeronautical Research Laboratory (ARL) of Defence Science and Technology Organization in Australia as a research scientist in 1990. At ARL, he helped perform AI research in areas of human factors and mission enhancement.Dr. Liu was appointed as Assistant Professor in 1994 in the Department of Computing at Hong Kong Polytechnic University. He has published technical papers on subjects in expert system verification, forecasting systems, pattern recognition and biometrics technology application. His current interests include intelligent business computing, multilingual system development, weather simulation and forecasting, data mining and Web-based information systems. He is a member of IEEE and AAAI.